

# 播放库

## SDK 编程指南

(for Android)

### V7.3.4.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
VERSION 7.3.4.X(2017.03) .....	2
3. 错误代码及说明 .....	4
3.1 基本错误码说明 .....	4
3.2 鱼眼错误码 .....	5
4. 函数调用顺序 .....	6
4.1 文件模式 .....	6
4.2 流模式 .....	7
4.3 同步回放模块 .....	8
4.4 鱼眼模块 .....	8
4.5 硬解码模块 .....	10
4.6 预录模块 .....	11
4.7 私有信息功能模块 .....	11
5. 函数说明 .....	13
5.1 系统操作及错误号获取 .....	13
5.1.1 获取播放库版本号 <i>getSdkVersion</i> .....	13
5.1.2 获取错误码 <i>getLastError</i> .....	13
5.1.3 获取播放端口 <i>getPort</i> .....	13
5.1.4 释放播放端口 <i>freePort</i> .....	13
5.2 文件操作 .....	15
5.2.1 打开数据文件 <i>openFile</i> .....	15
5.2.2 关闭数据文件 <i>closeFile</i> .....	15
5.3 流操作 .....	16
5.3.1 设置流播放模式 <i>setStreamOpenMode</i> .....	16
5.3.2 打开流 <i>openStream</i> .....	16
5.3.3 关闭流 <i>closeStream</i> .....	17
5.3.4 输入流数据 <i>inputData</i> .....	17
5.4 播放控制 .....	18
5.4.1 开启播放 <i>play</i> .....	18
5.4.2 开启播放 <i>playEx</i> .....	18
5.4.3 停止播放 <i>stop</i> .....	18
5.4.4 暂停数据播放 <i>pause</i> .....	18
5.4.5 快速播放 <i>fast</i> .....	19
5.4.6 慢速播放 <i>slow</i> .....	19

5.4.7 以独占方式播放音频 <code>playSound</code> .....	19
5.4.8 关闭音频 <code>stopSound</code> .....	20
5.4.9 以共享方式播放音频 <code>playSoundShare</code> .....	20
5.4.10 停止共享音频播放 <code>stopSoundShare</code> .....	20
5.4.11 算法调节音量 <code>adjustWaveAudio</code> .....	21
5.4.12 设置文件当前播放位置（百分比） <code>setPlayPos</code> .....	21
5.4.13 获取文件当前播放位置（百分比） <code>getPlayPos</code> .....	21
5.4.14 设置文件当前播放时间（毫秒） <code>setPlayedTimeEx</code> .....	22
5.4.15 获取文件当前播放时间（毫秒） <code>getPlayedTimeEx</code> .....	22
5.4.16 获取文件当前播放时间 <code>getPlayedTime</code> .....	22
5.4.17 设置文件当前播放帧号 <code>setCurrentFrameNum</code> .....	22
5.4.18 获取文件当前播放帧号 <code>getCurrentFrameNum</code> .....	23
5.4.19 设置音视频同步 <code>syncToAudio</code> .....	23
5.4.20 设置窗口透明度 <code>setWindowTransparency</code> .....	23
5.4.21 设置变分辨率回调函数 <code>setEcnTypeChgCB</code> .....	23
5.4.22 刷新当前显示窗口 <code>refreshPlay</code> .....	24
5.4.23 获取文件在线定位偏移量 <code>getPlayTimeOffset</code> .....	24
5.5 获取播放或解码信息 .....	26
5.5.1 设置垂直翻转 <code>verticalFlip</code> .....	26
5.5.2 获取全局时间 <code>getSystemTime</code> .....	26
5.5.3 获取文件总帧数 <code>getFileTotalFrames</code> .....	26
5.5.4 获取当前帧率 <code>getCurrentFrameRate</code> .....	27
5.5.5 获取已解码的视频帧数 <code>getPlayedFrames</code> .....	27
5.5.6 获取原始图像大小 <code>getPictureSize</code> .....	27
5.6 解码操作及控制 .....	28
5.6.1 设置视频帧解码类型 <code>setDecodeFrameType</code> .....	28
5.6.2 设置跳 I 帧解码 <code>setIframeDecInterval</code> .....	28
5.6.3 设置解码回调函数 <code>setDecodeCB</code> .....	28
5.6.4 设置全局时间解码回调函数 <code>setDecodeCBEx</code> .....	29
5.6.5 文件结束回调 <code>setFileEndCB</code> .....	31
5.6.6 解码密钥 <code>setSecretKey</code> .....	31
5.7 显示操作 .....	32
5.7.1 设置或增加显示区域 <code>setDisplayRegion</code> .....	32
5.7.2 设置或增加显示区域 <code>setDisplayRegionEx</code> .....	33
5.7.3 设置或销毁显示区域 <code>setVideoWindow</code> .....	34
5.7.4 设置或销毁显示区域 <code>setVideoWindowEx</code> .....	34
5.7.5 设置同步回放组 <code>setSycGroup</code> .....	35
5.8 缓冲区操作 .....	36
解码前源缓冲区 .....	36
5.8.1 获取源缓冲区剩余数据大小 <code>getSourceBufferRemain</code> .....	36
5.8.2 清空所有缓冲区的剩余数据 <code>resetSourceBuffer</code> .....	36
5.8.3 清空指定缓冲区的数据 <code>resetBuffer</code> .....	36
解码后播放缓冲区 .....	37
5.8.4 设置显示缓冲区缓冲帧数 <code>setDisplayBuf</code> .....	37

5.8.5 获取显示缓冲区缓冲帧数 <code>getDisplayBuf</code> .....	37
源、播放缓冲区.....	37
5.8.6 获取指定缓冲区的大小 <code>getBufferValue</code> .....	37
5.9 索引 .....	39
5.9.1 索引回调 <code>setFileRefCB</code> .....	39
5.10 抓图.....	40
5.10.1 显示回调 <code>setDisplayCB</code> .....	40
5.10.2 带全局时间显示回调 <code>setDisplayCBEx</code> .....	40
5.10.3 截取当前画面 BMP 图片 <code>getBMP</code> .....	41
5.10.4 私有信息 BMP 抓图 <code>getBmpEx</code> .....	42
5.10.5 截取当前画面 JPEG 图片 <code>getJPEG</code> .....	42
5.11 硬解码.....	44
5.11.1 设置硬解码优先 <code>setHardDecode</code> .....	44
5.11.2 设置最大硬解码路数 <code>setMaxHardDecodePort</code> .....	45
5.11.3 获取当前解码类型 <code>getDecoderType</code> .....	45
5.11.4 切换到软解 <code>switchToSoft</code> .....	45
5.11.5 切换到硬解 <code>switchToHard</code> .....	46
5.12 预录像功能.....	48
5.12.1 设置预录像数据回调 <code>setPreRecordCallBack</code> .....	48
5.12.2 设置预录像数据回调 <code>setPreRecordCallBackEx</code> .....	49
5.12.3 设置预录像开关 <code>setPreRecordFlag</code> .....	50
5.13 鱼眼功能.....	52
5.13.1 广角图像矫正 <code>setImageCorrection</code> .....	52
5.13.2 设置鱼眼矫正类型 <code>setFECDisplayEffect</code> .....	52
5.13.3 获取鱼眼矫正参数 <code>getFECDisplayParam</code> .....	53
5.13.4 设置鱼眼矫正参数 <code>setFECDisplayParam</code> .....	53
5.13.5 开启鱼眼功能 <code>enableFEC</code> .....	54
5.13.6 关闭鱼眼功能 <code>disableFEC</code> .....	54
5.13.7 获取鱼眼矫正处理子端口 <code>getPortFEC</code> .....	54
5.13.8 删除鱼眼矫正处理子端口 <code>delPortFEC</code> .....	56
5.13.9 设置鱼眼矫正参数 <code>setParamFEC</code> .....	56
5.13.10 获取鱼眼矫正参数 <code>getParamFEC</code> .....	58
5.13.11 设置显示窗口 <code>setWnd</code> .....	59
5.13.12 设置显示窗口 <code>setWndEx</code> .....	60
5.13.13 获取当前选择鱼眼子端口 <code>getCurrentPTZPort</code> .....	61
5.13.14 设置当前选择鱼眼子端口 <code>setCurrentPTZPort</code> .....	61
5.13.15 画矩形或者不规则框选择 <code>setPTZoutLineShowMode</code> .....	61
5.13.16 获取当前子端口号对应的当前 PTZ 坐标值 <code>setPTZ2Window</code> .....	62
5.13.17 获取当前子端口号对应的当前 PTZ 坐标值 <code>setFECDisplayPTZ</code> .....	63
5.13.18 设置鱼眼 3D 旋转（此设置只针对 3D 鱼眼起作用） <code>setFish3DRotate</code> .....	63
5.13.19 获取鱼眼抓图所需缓存大小 <code>getFishCapPicSize</code> .....	65
5.13.20 鱼眼抓图 <code>getFishCapPic</code> .....	66
5.14 私有数据.....	67
5.14.1 显示私有数据 <code>renderPrivateData</code> .....	67

5.14.2 显示私有数据 renderPrivateDataEx .....	67
5.14.5 私有数据回调 setAdditionalCallback .....	69
5.14.6 私有数据回调 setIVSDrawFunCallback .....	69
5.14.7 设置字体库路径 setOverlayPrilInfoFlag .....	70
5.15 倒放 .....	71
5.15.1 倒放 reversePlay .....	71
<b>6. 见问题排查 .....</b>	<b>72</b>
关于流模式接口 .....	72
Question 1 送入到播放库 openStream 接口报错 .....	72
流模式预览 .....	72
Question 2 实时流预览有卡顿 .....	72
Question 3 实时流预览不显示图像 .....	72
Question 4 实时流预览延时 .....	73
流模式播放 .....	73
Question 5 流模式下快放 .....	73
Question 6 实时流预览显示第一帧画面比较慢 .....	74
关于加密码流 .....	74
Question 7 加密码流无法正常播放 .....	74
关于抓图 .....	74
Question 8 如何将捕获的图片保存在内存中以供后续处理 .....	74
关于播放声音问题 .....	74
Question 9 音频播放卡顿，视频播放流畅 .....	74
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 .....	74
Question 11 播放文件，无法听到声音 .....	74
关于文件模式问题 .....	75
Question 12 实时流播放正常，文件播放有快放现象 .....	75
Question 13 文件打开失败 .....	75
Question 14 文件模式如何定位 .....	75
关于解码回调问题 .....	75
Question 15 如何获取多路数据流解码后的数据 .....	75
Question 16 实时流数据用解码回调有丢帧现象 .....	76
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 .....	76
Question 18 如何自己实现显示 .....	76
Question 19 解码回调和显示回调函数的区别 .....	76
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 .....	76
关于同步回放问题 .....	76
Question 21 如何实现多个文件的同步回放 .....	76
关于 ANDROID 版本的播放库问题 .....	77
Question 22 Android 版本的播放库不能正常预览 .....	77

# 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。

系统环境：

软件环境	Android 操作系统 4.0 及以上（硬解码要求 4.1.2 及以上）的手机和平板
硬件环境	完整支持 ARM32、ARM64 架构 CPU（x86 可以兼容，不保证稳定性）

## 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 7.3.4.X(2017.03)

- 新增接口：  
带全局时间显示回调: [setDisplayCBEx](#);  
带全局时间解码回调: [setDecodeCBEx](#);  
设置预录像回调(带全局时间): [setPreRecordCallBackEx](#);  
开启/关闭共享音频播放 [playSoundShare](#)、[stopSoundShare](#)  
设置鱼眼 3D 旋转 [setFish3DRotate](#);  
鱼眼抓图: [getFishCapPicSize](#)、[getFishCapPic](#);  
私有信息 BMP 抓图: [getBmpEx](#);  
倒放: [reversePlay](#);
- 修改功能：鱼眼新增 4 种 3D 效果，在 [getPortFEC](#) 里面设置：FEC\_CORRECT\_SEM（半球）、FEC\_CORRECT\_CYC（圆柱）、FEC\_CORRECT\_PLA（小行星）、FEC\_CORRECT\_CYC\_SPL（剪开圆柱）。

### Version 7.3.3.X(2016.10)

- 新增接口：  
动态调整音量大小: [adjustWaveAudio](#);  
硬解切软解 [switchToSoft](#);  
软解切硬解 [switchToHard](#)  
获取 MP4 封装流式播放数据长度偏移量 [getPlayTimeOffset](#);  
设置隔 I 帧解码 [setIframeDecInterval](#);  
刷新一帧数据 [refreshPlay](#);  
设置私有数据回调 [setIVSDrawFunCallback](#);
- 修改功能：fast,快放接口最大支持倍速调整为 128 倍速，由于性能限制无法到达 128 倍速，可配合 [setDecodeFrameType](#) 设置只解 I 帧，[setIframeDecInterval](#) 隔 I 帧解码使用

### Version 7.3.2.X(2016.0630)



- 鱼眼多画面 PTZ 矫正: [enableFEC](#)、[disableFEC](#)、[getPortFEC](#)、[delPortFEC](#)、[setParamFEC](#)、[getParamFEC](#)、[setWnd](#)、[setWndEx](#)、[getCurrentPTZPort](#)、[setCurrentPTZPort](#)、[setPTZoutLineShowMode](#)、[setPTZ2Window](#)。
- 私有数据显示(POS 信息、图片叠加不支持): [renderPrivateDataEx](#)、[renderPrivateData](#)。
- 音视频同步播放: [syncToAudio](#)。
- 增加窗口透明度设置: [setWindowTransparency](#)。

## 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: [setSyncGroup](#)。
- 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

### 3.1 基本错误码说明

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
PLAYM4_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_NOT_FIND	33	资源没找到
PLAYM4_NEED_LARGER_BUFFER	34	需要更大 BUFFER
PLAYM4_DEMUX_ERROR	35	Demux 错误
PLAYM4_FAIL_UNKNOWN	99	错误

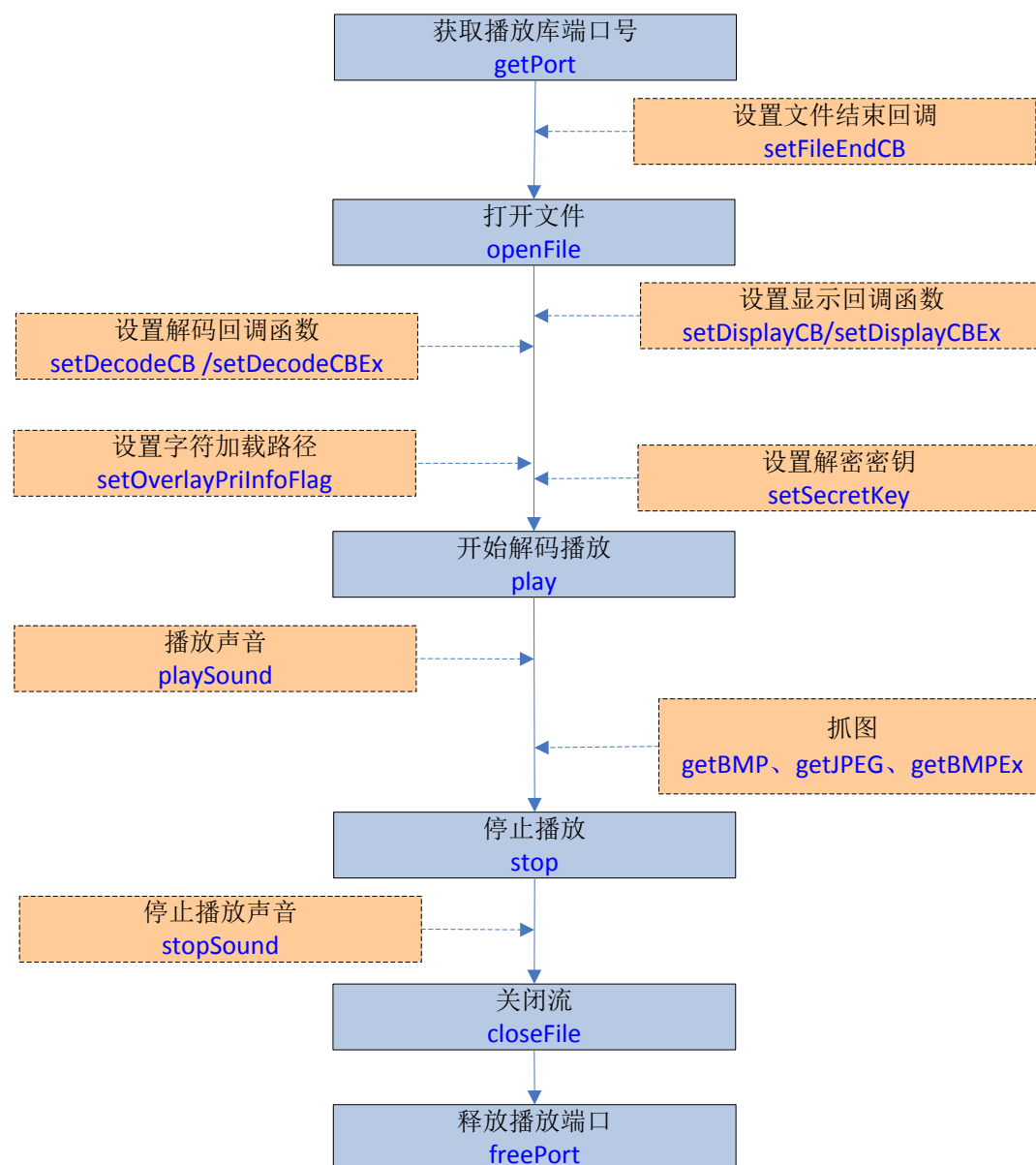
## 3.2 鱼眼错误码

	代码	说明
PLAYM4_FEC_ERR_ENABLEFAIL	100	鱼眼模块加载失败
PLAYM4_FEC_ERR_NOTENABLE	101	鱼眼模块没有加载;
PLAYM4_FEC_ERR_NOSUBPORT	102	子端口没有分配
PLAYM4_FEC_ERR_PARAMNOTINIT	103	没有初始化对应端口的参数
PLAYM4_FEC_ERR_SUBPORTOVER	104	子端口已经用完
PLAYM4_FEC_ERR_EFFECTNOTSUPPORT	105	该安装方式下这种效果不支持
PLAYM4_FEC_ERR_INVALIDWND	106	非法的窗口;
PLAYM4_FEC_ERR_PTZOVERFLOW	107	PTZ 位置越界;
PLAYM4_FEC_ERR_RADIUSINVALID	108	圆心参数非法;
PLAYM4_FEC_ERR_UPDATENOTSUPPORT	109	指定的安装方式和矫正效果, 该参数更新不支持;
PLAYM4_FEC_ERR_NOPLAYPORT	110	播放库端口没有启用;
PLAYM4_FEC_ERR_PARAMVALID	111	参数为空;
PLAYM4_FEC_ERR_INVALIDPORT	112	非法子窗口;
PLAYM4_FEC_ERR_PTZZOOMOVER	113	PTZ 矫正范围越界;
PLAYM4_FEC_ERR_OVERMAXPORT	114	矫正通道饱和, 最大支持的矫正通道为四个;
PLAYM4_FEC_ERR_ENABLED	115	该端口已经启用了鱼眼模块;
PLAYM4_FEC_ERR_D3DACCENOTENABLE	116	D3D 加速没有开启;
PLAYM4_FEC_ERR_PLACE_TYPE	117	安装方式不正确; 一个播放库port需要对应一种安装方式

[返回目录](#)

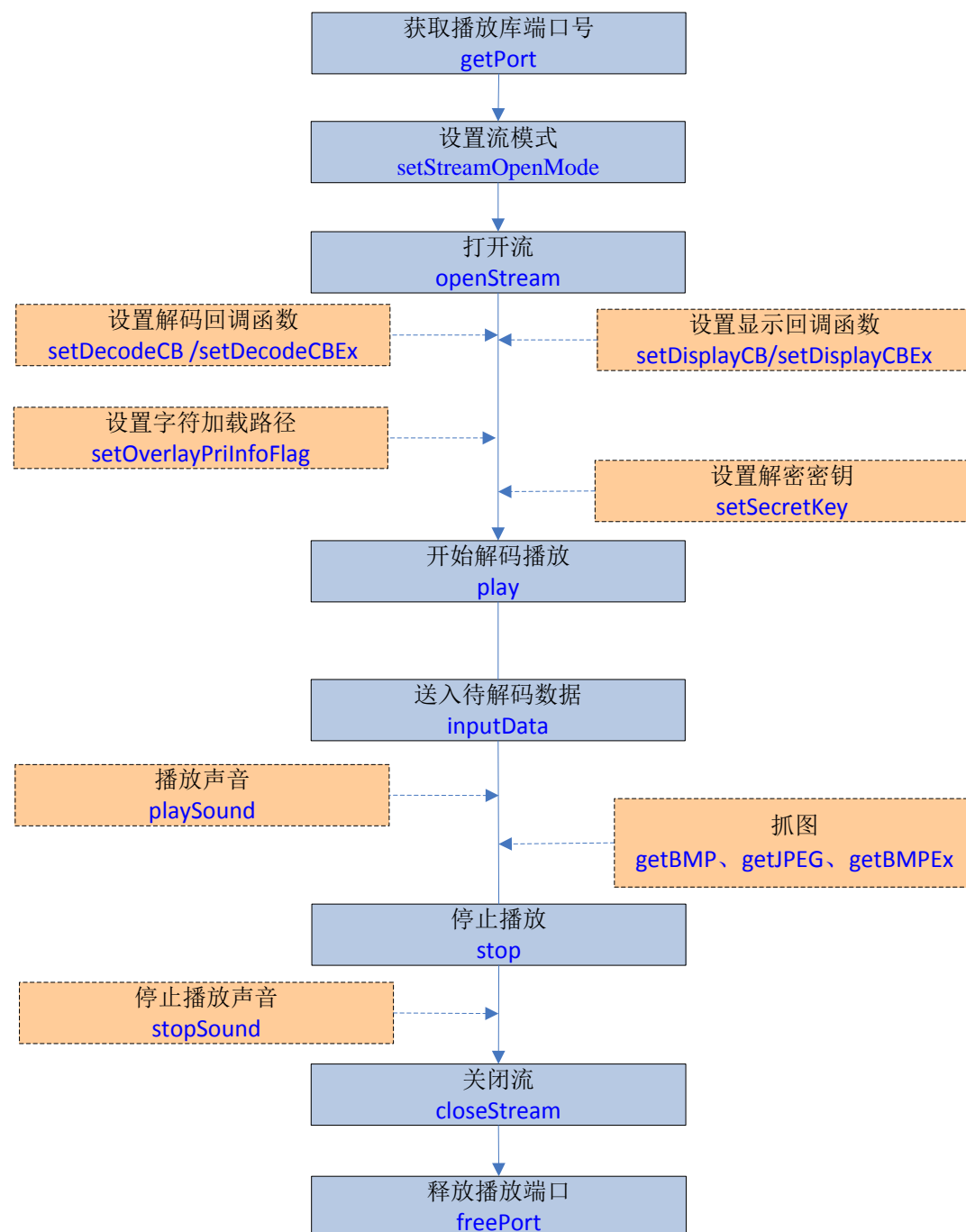
## 4. 函数调用顺序

### 4.1 文件模式



- 实现文件模块需要先获取播放库端口号，且需要确认文件是海康设备的录像文件。
- 文件模式下设置的 setDisplayCBEx 回调，需要在 stop 前调用 setDisplayCBEx(null)，不调用会导致 while 循环退不出。(建议所有设置的回调，在关闭播放库 stop 前面，均需要设置为 null)

## 4.2 流模式



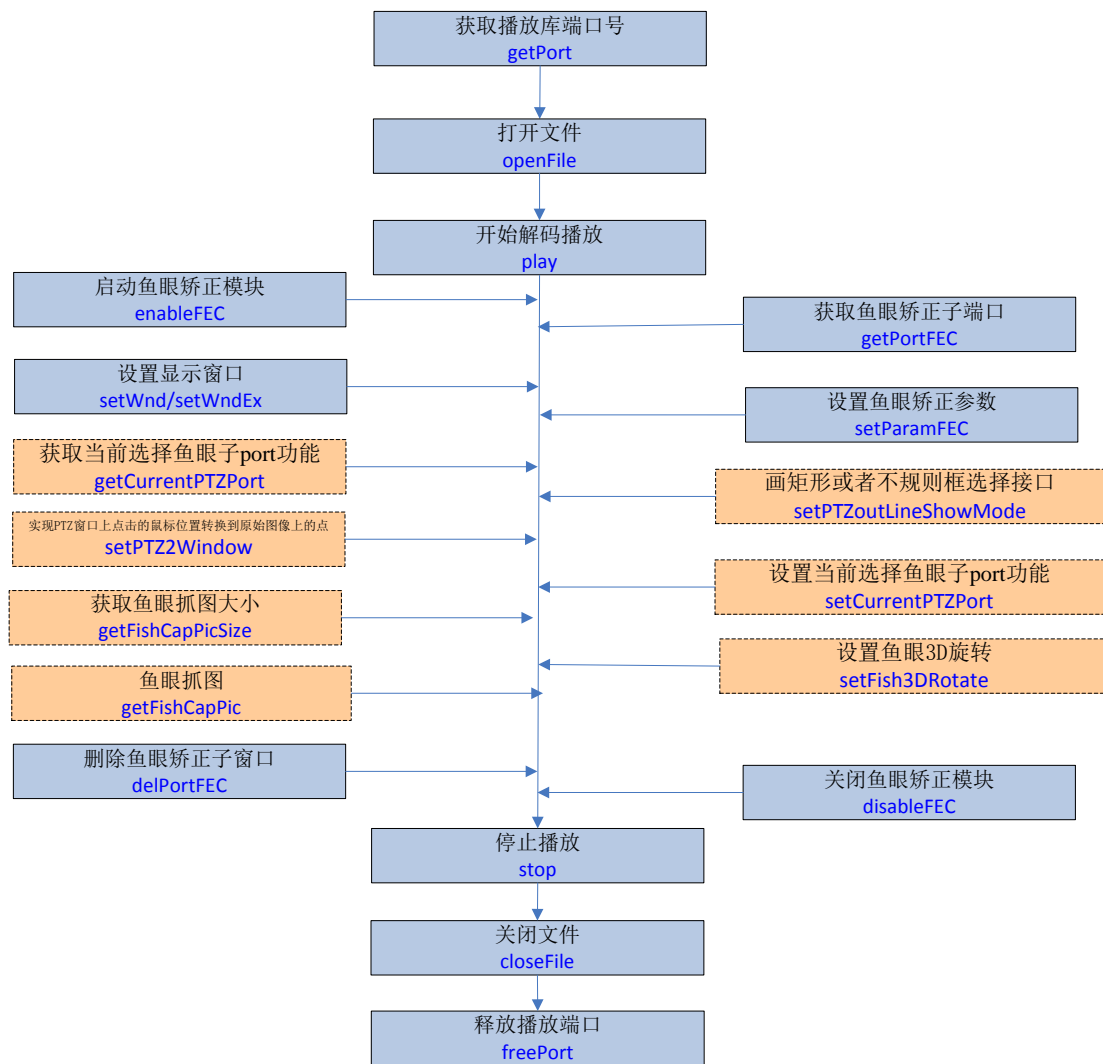
- 流模式模块需要先获取播放库通道号，然后设置流模式（相关接口 `getPort`、`setStreamOpenMode`），送入正确的海康媒体信息头打开数据流（相关接口 `openStream`），开始解码播放（相关接口 `play`），然后送入正确的海康数据（相关接口 `inputData`）。
- 实时流需要获取解码后的数据，则可以调用解码回调接口（相关接口 `setDecodeCB`）。
- 多路实时流若都需要获取解码后的数据，每个需要获取解码数据的通道都可以调用解码回调接口（`setDecodeCB`），在解码回调函数中根据 `nPort` 号判断是具体哪路数据。

## 4.3 同步回放模块



- 同步回放是指多个文件在有相同的时间段，然后多个窗口按照相同的时间戳播放。其中需要调用接口（SetSycGroup）。

## 4.4 鱼眼模块



鱼眼功能模块是对鱼眼摄像机图像的矫正，使用时需要开启预览模块、获取鱼眼矫正处理子端口、设置鱼眼矫正参数、设置显示窗口、设置鱼眼窗口的绘图回调。

- 开启和关闭鱼眼模块：开启鱼眼模块基本参数，调用的接口 enableFEC、disableFEC。
- 获取/删除鱼眼矫正处理子端口：调用的接口有 getPortEFC、delPortFEC。
- 设置/获取鱼眼矫正参数：调用的接口有 setParamFEC、getParamFEC。
- 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。建议使用 7.3.2 及其以后版本的鱼眼接口。
- 设置显示窗口：调用的接口有 setWnd、setWndEx。
- 鱼眼抓图接口：先调用 getFishCapPicSize 获取图片大小，然后调用 getFishCapPic 实现鱼眼抓图。
- 以上流程图是以文件模式为例画图，主要显示鱼眼模块接口如何调用，流模式也支持鱼眼模式功能。
- 鱼眼显示窗口销毁后无法再创建。

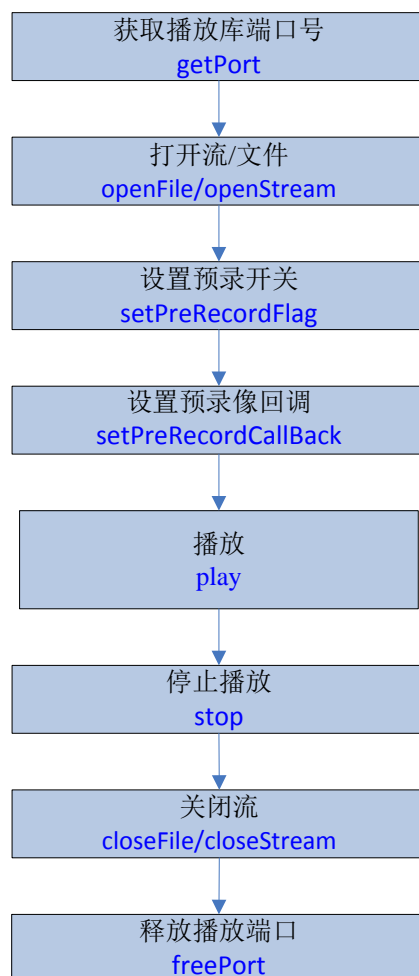
## 4.5 硬解码模块



- 硬解模块: 设定硬解码开关使得播放库优先采用 GPU 解码, 替换原先的视频解码模块。
- 开启硬解码优先调用接口: `setHardDecode`, 符合硬解码条件的码流会优先进行硬解码, 硬解码失败时会主动切换到软解。
- 硬解切软解 `switchtoSoft` 后需要调用 `play` (不调用 `play` 可能会停止), `switchtoSoft` 后不调用 `play` 条件下再调用 `switchToHard`, 会报顺序错误。
- 软解切硬解 `switchToHard` 主要是用来手动切换将软解切换为硬解, 如果已经是硬解, 则调用 `switchToHard` 返回错误码 2。调用 `switchToHard` 完后必须调用 `play` 才能切换到硬解, 只调用 `switchToHard` 会出现程序崩溃。(软解成功之后调用此接口) 软解切硬解时, 需要销毁解码器和重新等待 I 帧, 需要等到 2-3 秒左右时间画面播放。

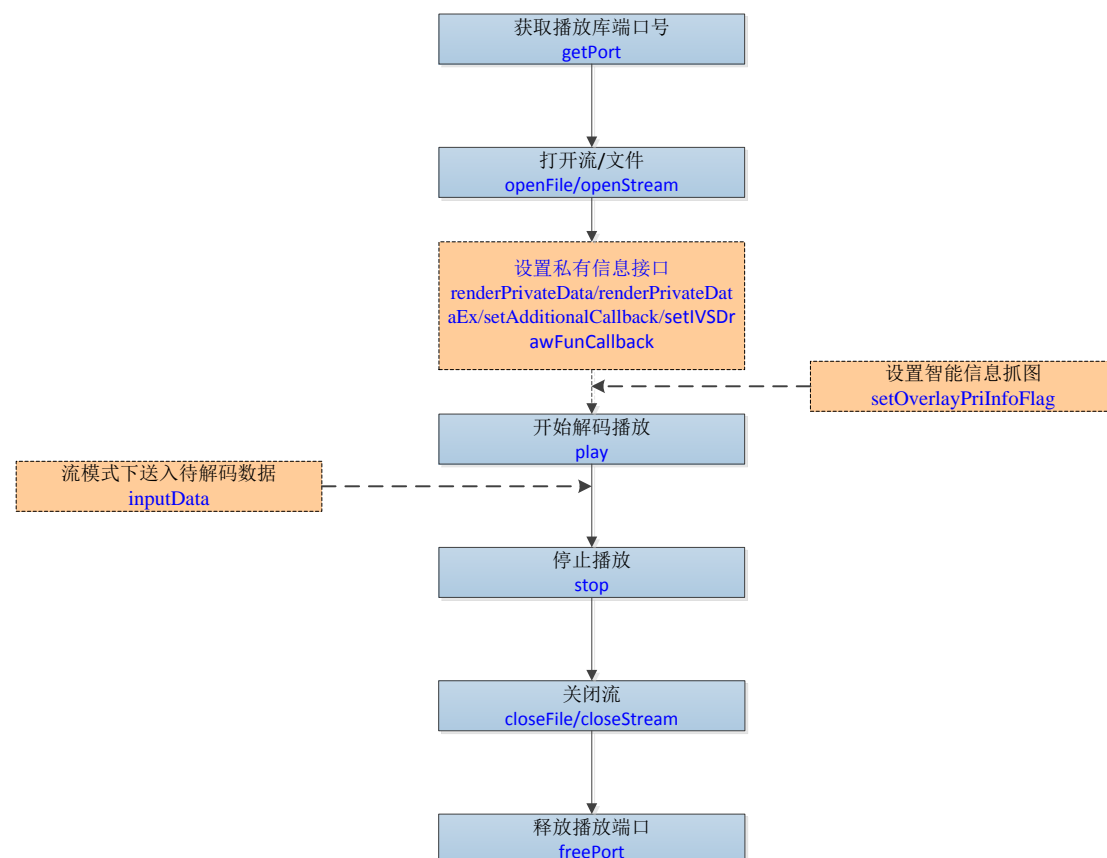


## 4.6 预录模块



- 预录功能模块是设备通过网络传输过来的码流通过回调函数给用户,方便用户保存为本地文件查看。
- 设置预录开关: 设置预录开关标志位,调用的接口 `setPreRecordFlag`;
- 设置预录数据回调函数: 调用的接口 `setPreRecordCallBack/setPreRecordCallBackEx`。

## 4.7 私有信息功能模块



- 私有信息模块：适用于带有智能信息的码流，若无智能信息的码流则调用此接口没有效果。
- 设置私有信息显示（大类型：如智能分析、移动侦测、POS 信息后叠加、图片叠加、热成像信息）：`renderPrivateData`；
- 设置私有信息显示（子类：
  - 当 `nIntelType` 为 `PLAYM4_RENDER_FIRE_DETCET` 时，子类型取值为 `PLAYM4_FIRE_ALARM`；
  - 当 `nIntelType` 为 `PLAYM4_RENDER_TEM` 时，子类型取值为 `PLAYM4_TEM_FLAG`）；`renderPrivateDataEx`。  
设置私有信息回调：
- 设置私有信息画图回调：`setIVSDrawFunCallback`。
- `renderPrivateData`、`renderPrivateDataEx`、`setAdditionalCallback`、`setIVSDrawFunCallback` 在 `play` 之前或者之后调用都可以；设置私有信息抓图：`setOverlayPriInfoFlag`。

# 5. 函数说明

## 5.1 系统操作及错误号获取

### 5.1.1 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](#)

### 5.1.2 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 <a href="#">错误码宏定义</a>	
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](#)

### 5.1.3 获取播放端口 **getPort**

函 数:	int getPort()	
参 数:		
返回值:	获取播放端口，返回-1 表示失败	
说 明:	获取未使用的通道号,通道号的上限为 32。	
注 意:		

[返回目录](#)

### 5.1.4 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](#)

## 5.2 文件操作

### 5.2.1 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:	在获取到端口资源后, 送入有效文件路径, 解析文件层封装和编码信息, 判断是否有关键帧, 计算总帧数和总时间。 其中海康播放库只支持海康编码设备的编码录像文件。第三方的标准文件暂不做支持。	
注 意:	<b>Android</b> 平台下建议文件大小不超过 2G, 大于等于 2G 文件, 经测试在有些手机上 openFile 会返回不支持(跟手机平台有关)。	

[返回目录](#)

### 5.2.2 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](#)

## 5.3 流操作

### 5.3.1 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:	<p>若码流为 RTP 封装的, 则需要注意正确送入 RTP 的文件头, 以及数据包大小和内容。</p> <p>设置流播放模式, 其中若设置为 STREAM_REALTIME 模式则表示尽量保证实时性, 防止数据阻塞; 且数据检查严格;</p> <p>设置为 STREAM_FILE 则表示按时间戳播放。</p>	
注 意:	STREAME_REALTIME 模式下 fast、slow 调用无效且不返回错误码。	

[返回目录](#)

### 5.3.2 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:	<p>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。</p> <p>流模式下进行定位控制和单帧操作, 需由外部控制数据。播放时候会检查帧号连续性, 如果帧号不连续会跳关键帧播放。</p> <p>支持类型在文件模式的基础上增加 RTP 封装码流的支持。</p>	
注 意:		

[返回目录](#)

### 5.3.3 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](#)

### 5.3.4 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。 未打开文件流, 或者文件流数据海康的播放库不能正常解码。	
注 意:		

[返回目录](#)

## 5.4 播放控制

### 5.4.1 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。在 play 之前需要 holder 创建成功。 如果已经播放, 重置当前播放速度为正常速度。	
注 意:	I 帧大于 2MB 时可能无法解析。	

[返回目录](#)

### 5.4.2 开启播放 **playEx**

函 数:	boolean playEx(int nPort, SurfaceTexture texture)	
参 数:	int nPort	播放通道号
	SurfaceTexture texture	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 texture 窗口调整, 要全屏显示, 只要把 texture 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:	I 帧大于 2MB 时可能无法解析。	

[返回目录](#)

### 5.4.3 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](#)

### 5.4.4 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 <code>true</code> ；失败返回 <code>false</code>
说明:	
注意:	

[返回目录](#)

### 5.4.5 快速播放 **fast**

函数:	<code>boolean fast(int nPort)</code>	
参数:	<code>int nPort</code>	播放端口
返回值:	成功返回 <code>true</code> ；失败返回 <code>false</code>	
说明:	<p>每次调用将使当前播放速度加快一倍。</p> <p><b>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。</b></p> <p>最大播放速度跟码流和手机 CPU 性能有很大关系，在 CPU 处于 90% 以上，或者一个核满的情况下，播放速度可能上不去，这属于硬件解码性能不足导致）。</p> <p>每次调用将使当前播放速度加快一倍，正常速度调用此接口最多调用 4 次；要恢复正常播放调用 <code>play()</code>，从当前位置开始正常播放；</p> <p>高清码流在高倍速播放时，由于受到解码和显示的限制，可能达不到所设置的速度。</p>	
注意:	<code>fast(nPort)</code> 快放接口最大支持倍速调整为 128 倍速，由于性能限制无法到达 128 倍速，可配合 <code>setDecodeFrameType</code> 设置只解 I 帧，和 <code>setIFrameDecInterval</code> 隔 I 帧解码使用。	

[返回目录](#)

### 5.4.6 慢速播放 **slow**

函数:	<code>boolean slow(int nPort)</code>	
参数:	<code>int nPort</code>	播放端口
返回值:	成功返回 <code>true</code> ；失败返回 <code>false</code>	
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:	降低到原播放速度的一般（最低 1/16 倍速），播放状态不变。	

[返回目录](#)

### 5.4.7 以独占方式播放音频 **playSound**

函数:	<code>boolean playSound(int nPort)</code>	
参数:	<code>int nPort</code>	播放端口
返回值:	成功返回 <code>true</code> ；失败返回 <code>false</code>	
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 <code>openStream</code> 的时候 40 字节文件头未申明音频编码类型，<code>playSound</code> 会报错。</p> <p>在码流不包含音频数据时调用该接口返回成功，不播放声音。</p>	

<b>注 意：</b>	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 纯视频文件，playSound / playSoundShare 返回成功，但是没有音频播放。兼容如 果头中没音频的话返回不成功会影响功能。
-------------	--

[返回目录](#)

### 5.4.8 关闭音频 **stopSound**

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

[返回目录](#)

### 5.4.9 以共享方式播放音频 **playSoundShare**

<b>函 数：</b>	boolean playSoundShare(int nPort)	
<b>参 数：</b>	int nPort	播放端口
<b>返回值：</b>	成功返回 true；失败返回 false	
<b>说 明：</b>	<p>该接口是以共享方式打开声音，而 playSound 和 stopSound 是以独占方式播放声音的。</p> <p><b>playSoundShare、stopSoundShare 需要配对调用；在播放声音过程中，建议 playSound、stopSound 配对混用。</b></p> <p>G 系列音频采样率按照封装层上的数据播放，如果解码文件或者码流的封装层上标识的音频采样率不正确，会导致声音播放异常。</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 openStream 的时候 40 字节文件头未申明音频编码类型，播放声音会报错。</p> <p>在码流不包含音频数据时调用该接口返回成功，不播放声音。</p> <p>纯视频文件，playSound / playSoundShare 返回成功，但是没有音频播放。兼容如果头中没音频的话返回不成功会影响功能。</p>	
<b>注 意：</b>		

[返回目录](#)

### 5.4.10 停止共享音频播放 **stopSoundShare**

<b>函 数：</b>	boolean stopSoundShare(int nPort)	
<b>参 数：</b>	int nPort	播放端口
<b>返回值：</b>	成功返回 true；失败返回 false	
<b>说 明：</b>	<p><b>playSoundShare、stopSoundShare 需要配对调用；在播放声音过程中，建议 playSound、stopSound 配对混用。</b></p> <p>G 系列音频采样率按照封装层上的数据播放，如果解码文件或者码流的封装层上标识的音频采样率不正确，会导致声音播放异常。</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。 如果 <code>openStream</code> 的时候 40 字节文件头未申明音频编码类型，播放声音会报错。 在码流不包含音频数据时调用该接口返回成功，不播放声音。
<b>注 意：</b>	

[返回目录](#)

### 5.4.11 算法调节音量 **adjustWaveAudio**

函 数：	boolean adjustWaveAudio(int nPort, int nValue)	
参 数：	int nPort	播放端口
	int nValue	调整参数，0 表示不调整，nValue 取值范围为(-100,100)。
返回值：	成功返回 true；失败返回 false	
说 明：	此接口主要用来在开启音频播放时动态调整音量大小，nValue 取值范围为(-100,100)。 音量从很高调节到很低时，需要一定的衰减时间来缓解音量的突变，反应出的现象会有短暂的延时生效。	
<b>注 意：</b>	如果是系统音量很小的话，调用 adjustWaveAudio 调用音量大小不是很明显。	

[返回目录](#)

### 5.4.12 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：	设置文件播放指针的相对位置（百分比）。 如果在建立文件索引的前提下使用,则为精确定位,否则即为粗略定位； <b>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。</b> 在进行精准定位时，由于需要解码一个 GOP 的数据，在解码性能不足的情况下，画面会停顿几秒再继续播放。	
<b>注 意：</b>		

[返回目录](#)

### 5.4.13 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	
<b>注 意：</b>		

[返回目录](#)

### 5.4.14 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，则为精确定位，否则即为粗略定位， <b>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</b> 。 在进行精准定位时，由于需要解码一个 GOP 的数据，在解码性能不足的情况下，画面会停顿几秒再继续播放。	

[返回目录](#)

### 5.4.15 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:	成功返回当前播放时间，以毫秒为单位；失败返回 0	
说 明:	根据时间获取文件播放位置	
注 意:		

[返回目录](#)

### 5.4.16 获取文件当前播放时间 **getPlayedTime**

函 数:	int getPlayedTime(int nPort)	
参 数:	int nPort	播放通道号
返回值:	成功返回当前播放时间，以秒为单位；失败返回 0	
说 明:	根据时间获取文件播放位置	
注 意:		

[返回目录](#)

### 5.4.17 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，则为精确定位，否则即为粗略定位， <b>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</b> 。	

在进行精准定位时，由于需要解码一个 GOP 的数据，在解码性能不足的情况下，画面会停顿几秒再继续播放。

[返回目录](#)

#### 5.4.18 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](#)

#### 5.4.19 设置音视频同步 **syncToAudio**

函 数:	boolean syncToAudio(int nPort, int nSyncToAudio)	
参 数:	int nPort	播放端口
	int nSyncToAudio	是否开启音视频同步，其中 0 为关闭，1 为开启。
返回值:	成功返回 true；失败返回 false	
说 明:	Synctoaudio 接口用于保存音视频同步开关值，跟接口调用顺序没有关系，播放库逻辑为调用此接口返回为 true	
注 意:		

[返回目录](#)

#### 5.4.20 设置窗口透明度 **setWindowTransparency**

函 数:	boolean setWindowTransparency(int nPort, float fTrans)	
参 数:	int nPort	播放端口
	float fTrans	透明度参数，取值范围 0~1.0，默认设定为 0.5。
返回值:	成功返回 true；失败返回 false	
说 明:	只能在软解情况下使用，设置的内部渲染层的透明，非窗口透明，需要移动窗口才能看出效果	
注 意:		

[返回目录](#)

#### 5.4.21 设置变分辨率回调函数 **setEcnTypeChgCB**

函 数:	boolean setEcnTypeChgCB(int nPort, PlayerEncTypeChgCB typeChgCB)	
参 数:	int nPort	播放端口
	PlayerEncTypeChgCB typeChgCB	变分辨率回调函数

	<p>PlayerEncTypeChgCB 回调函数</p> <pre>public void onEncTypeChg(int nPort);</pre> <p>回调函数参数说明</p> <table><tr><td>int</td><td>nPort</td><td>播放端口</td></tr></table>	int	nPort	播放端口	
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](#)

5.4.22 刷新当前显示窗口 **refreshPlay**

函 数:	boolean refreshPlay(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:	此接口主要用于刷新一帧数据，主要用于软解，硬解不支持。	
注 意:		

[返回目录](#)

5.4.23 获取文件在线定位偏移量 **getPlayTimeOffset**

函 数:	int getPlayTimeOffset(int nPort, int nTime)	
参 数:	int nPort	播放端口
	int nTime	定位的播放时间，单位是 ms。取值范围：[0,录像文件播放总长度]
返回值:	成功返回偏移量；失败返回-1	
说 明:	<p>此接口主要用于远程回放下 MP4 封装流式播放获取指定时间的 MP4 数据长度偏移量，用于 MP4 封装流式定位。<b>不支持 MP4 封装文件播放模式。</b></p> <p>MP4 封装流式播放，不支持以下两种类型：</p> <ul style="list-style-type: none"><li>➤ 不支持 MPEG4 编码的 MP4 封装码流</li><li>➤ 不支持后置索引 MP4 封装码流</li></ul> <p>适用于远程回放情况下对回放文件进行定位。</p> <p>适用于文件流方式播放文件（仅用于 MP4 封装流式定位使用，其他封装视频或文件播放不建议使用）。即本地文件以流模式打开后调用此接口进行定位，获取到对应的文件偏移量，调用文件操作接口对文件进行偏移即可实现流式定位；具体实现如下：</p> <ul style="list-style-type: none"><li>➤ 文件流方式定位后播放停止，等待送入相应偏移位置的数据（重新定位后的数据）后继续播放。</li><li>➤ 即在调用 getPlayTimeOffset 后，需停止 inputData 送数据，然后调用 resetSourceBuffer 接口清空播放库内部缓存后，再重新读取指定偏移量数据 inputData 到播放库。</li></ul>	

	<p>用户指定一个播放定位的时间，播放库根据这个时间返回需要偏移的数据长度，用户根据这个长度重新送数据到播放库播放。使用场景跟回放比较类似，使用海康取流库，现在一般使用在海康车载设备对接。</p> <pre>openStream play ... inputData getPlayTimeOffset 获取到定位需要偏移的长度后 resetSourceBuffer inputData中送入按照getPlayTimeOffset获取的偏移后的数据</pre>
注 意：	

[返回目录](#)

## 5.5 获取播放或解码信息

### 5.5.1 设置垂直翻转 **verticalFlip**

函 数:	boolean verticalFlip(int nPort, int nFlag)	
参 数:	int nPort int nFlag	播放端口 是否设置翻转，若为 1 则为翻转；为 0 则不翻转
返回值:	成功返回 true,失败返回 false	
说 明:	只对原始窗口起翻转效果，对多窗口如同时开启电子放大，三代鱼眼矫正不起作用；开启硬解码不支持图像翻转。	
注 意:	热成像和温度信息不处理翻转效果，显示信息会出现不翻转或翻转错误的情况。	

[返回目录](#)

### 5.5.2 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](#)

### 5.5.3 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](#)

### 5.5.4 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](#)

### 5.5.5 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](#)

### 5.5.6 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。 <b>在尚未解码出第一帧图像前，返回默认值 352x288。</b> 此函数获取的是实际码流分辨率大小，需要正常解码显示后使用。 未调用 play 时调用此接口返回值无实际参考意义。	
注 意:		

[返回目录](#)

## 5.6 解码操作及控制

### 5.6.1 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
	<code>#define DECODE_NORMAIL</code>	0 正常解码
	<code>#define DECODE_KEY_FRAME</code>	1 只解关键帧
	<code>#define DECODE_NONE</code>	2 不解视频帧
	<code>#define DECODE_VIDEO_HALF</code>	3 SVC 码流只解码 1/2
	<code>#define DECODE_VIDEO_QUARTER</code>	4 SVC 码流只解码 1/4
	<code>#define DECODE_VIDEO_EIGHTH</code>	5 SVC 码流只解码 1/8
	<code>#define DECODE_ALL_FRAME</code>	6 解码类型为 4 时表示 8 倍速高分辨率码流不自动抽帧, 默认全解
返回值:	成功返回 true; 失败返回 false	
说 明:	在解码前调用此接口。 解码类型为 4 时表示 8 倍速高分辨率码流不自动抽帧, 默认全解。 实时流时设置只解关键帧时, 音频可能会有卡顿的现象	

[返回目录](#)

### 5.6.2 设置跳 I 帧解码 **setIFrameDecInterval**

函 数:	boolean setIFrameDecInterval(int nPort, int nInterval)	
参 数:	LONG      nPort	播放通道号
	int        nInterval	设置跳 I 帧个数, 表示往后跳几个 I 帧
返回值:	成功返回 true; 失败返回 false	
说 明:	此接口主要用于设置给 I 帧解码, 调用前需要设置 setDecodeFrameType(1)只解关键帧, 否则返回错误码 2。 支持 svc 隔帧解码, 调用 setDecodeFrameType 接口进行设置。	

[返回目录](#)

### 5.6.3 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
	PlayerDecodeCB 回调函数 <code>public void onDecode(int nPort,</code>	

	<pre>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>播放器通道号</td></tr><tr><td>byte[]</td><td>data</td><td>解码后的音视频数据指针</td></tr><tr><td>int</td><td>nDataLen</td><td>解码后的音视频数据 pBuf 的长度</td></tr><tr><td>int</td><td>nWidth</td><td>视频图像宽/音频声道数</td></tr><tr><td>int</td><td>nHeight</td><td>视频图像高/音频采样率</td></tr><tr><td>int</td><td>nFrameTime</td><td>解码时间戳</td></tr><tr><td>int</td><td>nDataType</td><td>解码当前帧类型, 详见下文宏定义说明</td></tr><tr><td>int</td><td>Reserved</td><td>保留</td></tr></table> <p>宏定义说明</p> <table><tr><td>T_AUDIO16</td><td>= 101;</td><td>音频数据, 采样率 16khz, 单声道, 每个采样点 16 位表示。</td></tr><tr><td>T_AUDIO8</td><td>= 100;</td><td>音频数据, 采样率 8khz, 单声道, 每个采样点 16 位表示。</td></tr><tr><td>T_RGB32</td><td>= 7;</td><td>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。</td></tr><tr><td>T_UYVY</td><td>= 1;</td><td>视频数据, uyvy 格式。</td></tr><tr><td>T_YV12</td><td>= 3;</td><td>“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”, 第一个像素位于图像左上角。</td></tr><tr><td></td><td></td><td>视频数据, yv12 格式。排列顺序“Y0-Y1-.....”, “V0-V1....”, “U0-U1-.....”。</td></tr></table> <p>关于图像格式详细信息, 用户可以查阅相关资料。</p>	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	T_AUDIO16	= 101;	音频数据, 采样率 16khz, 单声道, 每个采样点 16 位表示。	T_AUDIO8	= 100;	音频数据, 采样率 8khz, 单声道, 每个采样点 16 位表示。	T_RGB32	= 7;	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。	T_UYVY	= 1;	视频数据, uyvy 格式。	T_YV12	= 3;	“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”, 第一个像素位于图像左上角。			视频数据, yv12 格式。排列顺序“Y0-Y1-.....”, “V0-V1....”, “U0-U1-.....”。	
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																																										
T_AUDIO16	= 101;	音频数据, 采样率 16khz, 单声道, 每个采样点 16 位表示。																																										
T_AUDIO8	= 100;	音频数据, 采样率 8khz, 单声道, 每个采样点 16 位表示。																																										
T_RGB32	= 7;	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。																																										
T_UYVY	= 1;	视频数据, uyvy 格式。																																										
T_YV12	= 3;	“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”, 第一个像素位于图像左上角。																																										
		视频数据, yv12 格式。排列顺序“Y0-Y1-.....”, “V0-V1....”, “U0-U1-.....”。																																										
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。</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](#)

5.6.4 设置全局时间解码回调函数 **setDecodeCBEx**

函 数:	boolean setDecodeCBEx(int nPort, PlayerCallBack.PlayerDecodeCBEx decodeCBEx)	
参 数:	int nPort	播放通道号
	PlayerDecodeCBEx	解码回调函数
	decodeCBEx	
	PlayerDecodeCBEx 回调函数	

	<pre>public void onDecodeEx(int nPort, byte[] data, int nDataLen, int nWidth, int nHeight, int nFrameTime, int nDataType, int Reserved,int nYear, int nMon, int nDay, int nHour, int nMin, int nSec, int nMs);</pre> <p>参数说明:</p> <table><tr><td>int</td><td>nPort</td><td>播放器通道号</td></tr><tr><td>byte[]</td><td>data</td><td>解码后的音视频数据指针</td></tr><tr><td>int</td><td>nDataLen</td><td>解码后的音视频数据 pBuf 的长度</td></tr><tr><td>int</td><td>nWidth</td><td>视频图像宽/音频声道数</td></tr><tr><td>int</td><td>nHeight</td><td>视频图像高/音频采样率</td></tr><tr><td>int</td><td>nFrameTime</td><td>解码时间戳</td></tr><tr><td>int</td><td>nDataType</td><td>解码当前帧类型，详见下文宏定义说明</td></tr><tr><td>int</td><td>Reserved</td><td>保留</td></tr><tr><td>int</td><td>nYear</td><td>年</td></tr><tr><td>int</td><td>nMon</td><td>月</td></tr><tr><td>int</td><td>nDay</td><td>日</td></tr><tr><td>int</td><td>nHour</td><td>时</td></tr><tr><td>int</td><td>nMin</td><td>分</td></tr><tr><td>int</td><td>nSec</td><td>秒</td></tr><tr><td>int</td><td>nMs</td><td>毫秒</td></tr></table> <p>宏定义说明</p> <table><tr><td>T_AUDIO16</td><td>= 101;</td><td>音频数据,采样率 16khz, 单声道, 每个采样点 16 位表示。</td></tr><tr><td>T_AUDIO8</td><td>= 100;</td><td>音频数据,采样率 8khz,单声道, 每个采样点 16 位表示。</td></tr><tr><td>T_RGB32</td><td>= 7;</td><td>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。</td></tr><tr><td>T_UYVY</td><td>= 1;</td><td>视频数据, uyvy 格式。</td></tr><tr><td>T_YV12</td><td>= 3;</td><td>“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”, 第一个像素位于图像左上角。</td></tr><tr><td></td><td></td><td>视频数据, yv12 格式。排列顺序“Y0-Y1-.....”, “V0-V1....”, “U0-U1-.....”。</td></tr></table> <p>关于图像格式详细信息，用户可以查阅相关资料。</p>	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	int	nYear	年	int	nMon	月	int	nDay	日	int	nHour	时	int	nMin	分	int	nSec	秒	int	nMs	毫秒	T_AUDIO16	= 101;	音频数据,采样率 16khz, 单声道, 每个采样点 16 位表示。	T_AUDIO8	= 100;	音频数据,采样率 8khz,单声道, 每个采样点 16 位表示。	T_RGB32	= 7;	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。	T_UYVY	= 1;	视频数据, uyvy 格式。	T_YV12	= 3;	“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”, 第一个像素位于图像左上角。			视频数据, yv12 格式。排列顺序“Y0-Y1-.....”, “V0-V1....”, “U0-U1-.....”。
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																																																														
int	nYear	年																																																														
int	nMon	月																																																														
int	nDay	日																																																														
int	nHour	时																																																														
int	nMin	分																																																														
int	nSec	秒																																																														
int	nMs	毫秒																																																														
T_AUDIO16	= 101;	音频数据,采样率 16khz, 单声道, 每个采样点 16 位表示。																																																														
T_AUDIO8	= 100;	音频数据,采样率 8khz,单声道, 每个采样点 16 位表示。																																																														
T_RGB32	= 7;	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。																																																														
T_UYVY	= 1;	视频数据, uyvy 格式。																																																														
T_YV12	= 3;	“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”, 第一个像素位于图像左上角。																																																														
		视频数据, yv12 格式。排列顺序“Y0-Y1-.....”, “V0-V1....”, “U0-U1-.....”。																																																														
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。</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>回调函数中获取的年、月、日、时、分、秒、毫秒信息是当前数据码流中解析出来的全局时间，是 DSP 编解码流时打入的。</p>																																																															

音视频播放正常，可以通过 `setVideowindow` 这个接口来选择显示或不显示。

[返回目录](#)

5.6.5 文件结束回调 **setFileEndCB**

函 数:	boolean setFileEndCB(int nPort, PlayerPlayEndCB playEndCB)	
参 数:	int nPort	播放通道号
	PlayerPlayEndCB playEndCB	文件结束回调接口对象
	回调函数 PlayerPlayEndCB public void onPlayEnd(int nPort);	
	参数说明	
	int nPort	播放库通道号
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。	
注 意:	在 <code>openFile</code> 之前调用才有效。且只适用于文件模式。	

[返回目录](#)

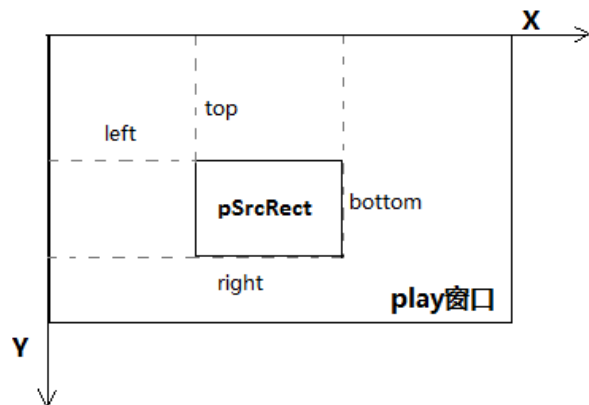
5.6.6 解码密钥 **setSecretKey**

函 数:	boolean setSecretKey(int nPort, int nKeyType, byte[] pSecretKey, int nKeyLen)	
参 数:	int nPort	播放通道号
	int lKeyType	密钥类型（取值为 0 不加密；取值为 1 标识 AES 加密）
	byte[] pSecretKey	密钥串首地址
	int lKeyLen	密钥长度，单位为 bit（1 字节为 8 bit）
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 <code>openSteam/openFile</code> 之前调用才有效。	
注 意:	一般只有海康私有封装和 PS 封装层有水印，其他没有水印。编码只支持标准 264 格式的水印。	

[返回目录](#)

5.7 显示操作

5.7.1 设置或增加显示区域 **setDisplayRegion**

函 数:	boolean setDisplayRegion(int nPort, int nRegionNum, MPRect pSrcRect, SurfaceHolder hDestWnd, int bEnable)	
参 数:	<div><div><div>int nPort</div><div>int nRegionNum</div><div>MPRect pSrcRect</div><div>SurfaceHolder hDestWnd</div><div>int bEnable</div></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>	<div>播放通道号</div> <div>显示区域索引，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><div>区域左上角点的X 轴坐标</div><div>区域右下角点的X 轴坐标</div><div>区域左上角点的Y 轴坐标</div><div>区域右下角点的Y 轴坐标</div><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	

	<p>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。</p> <p>➤ 当前版本为了跟三代鱼眼同时使用，限制 <b>nRegionNum</b> 取值为 0、1。其中 0 是对原始窗口进行电子放大，1 是对新开的窗口进行电子放大。</p> <p>➤ <b>setdisplayregion</b>或者<b>setdisplayregionex</b>实现电子放大功能时，若返回参数错误时，依然可以调用<b>siwtchtohard</b>切换到硬解码功能，即电子放大功能设定失败，不影响当前软解码切换到硬解码。</p>
--	--

[返回目录](#)

5.7.2 设置或增加显示区域 **setDisplayRegionEx**

函 数:	<b>boolean setDisplayRegionEx(int nPort, int nRegionNum, MPRect pSrcRect, SurfaceTexture texture, int bEnable)</b>	
参 数:	<div><div><div>int nPort</div><div>int nRegionNum</div><div>MPRect pSrcRect</div><div>SurfaceTexture texture</div><div>int bEnable</div></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>	<div><p>播放通道号</p><p>显示区域索引，0~(MAX_DISPLAY_WND-1)</p><p>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</p><p>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。</p><p>打开（设置）或关闭显示区域</p></div> <div><p>区域左上角点的X 轴坐标</p><p>区域右下角点的X 轴坐标</p><p>区域左上角点的Y 轴坐标</p><p>区域右下角点的Y 轴坐标</p></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:	<ul style="list-style-type: none"> <li>➤ 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。</li> <li>➤ 当前版本为了跟三代鱼眼同时使用, 限制 nRegionNum 取值为 0、1。其中 0 是对原始窗口进行电子放大, 1 是对新开的窗口进行电子放大。</li> <li>➤ Setdisplayregion 或者 setdisplayregionex 实现电子放大功能时, 若返回参数错误时, 依然可以调用 siwtchtohard 切换到硬解码功能, 即电子放大功能设定失败, 不影响当前软解码切换到硬解码。</li> </ul>

[返回目录](#)

### 5.7.3 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	端口索引
	SurfaceHolder holder	设置显示窗口句柄
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。 电子放大时调用该接口电子放大效果会消失	

[返回目录](#)

### 5.7.4 设置或销毁显示区域 **setVideoWindowEx**

函 数:	boolean setVideoWindowEx(int nPort, int nRegionNum, SurfaceTexture texture)	
参 数:	int nPort	播放库端口
	int nRegionNum	端口索引
	SurfaceTexture texture	设置显示窗口句柄
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 SurfaceTexture 拥有生命周期。 电子放大时调用该接口电子放大效果会消失	

[返回目录](#)



5.7.5 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:	<p>可以多次调用将指定的 nPort 加入到同步组中；</p> <p>setSycGroup 在 play 之前调用；</p> <p>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；</p> <p>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。</p>	
注 意:		

[返回目录](#)

## 5.8 缓冲区操作

### 解码前源缓冲区

#### 5.8.1 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](#)

#### 5.8.2 清空所有缓冲区的剩余数据 **resetSourceBuffer**

函 数:	boolean resetSourceBuffer(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:	清空所有缓冲区的剩余数据	
注 意:		

[返回目录](#)

#### 5.8.3 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 (暂不支持)
	BUF_VIDEO_RENDER	解码后视频缓冲区总节点个数,以帧为单位。
	BUF_AUDIO_RENDER	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。
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

### 5.8.4 设置显示缓冲区缓冲帧数 **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>重置回调标志位为有效状态。</p> <p>设置播放缓冲区（即解码后的图像缓冲区）大小，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>在播放文件时候，我们默认是 6 帧；播放流时默认是 6 帧。</p> <p>若设置的值超过有效范围，播放库会内部自适应为 6，且会返回成功。</p> <p>在 play 之间调用有效</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 openStream 之后，play 之前调用。	

[返回目录](#)

### 5.8.5 获取显示缓冲区缓冲帧数 **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

### 5.8.6 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。

	<div>BUF_AUDIO_SRC</div> <div>BUF_VIDEO_RENDER</div> <div>BUF_AUDIO_RENDER</div> <div>BUF_VIDEO_DECODED</div> <div>BUF_AUDIO_DECODED</div>	<div>音频数据源缓冲区大小，只对流模式有效，单位 <code>byte</code>。且音视频数据分开送到情况下才有效</div> <div>解码后视频缓冲区总节点个数，以帧为单位。</div> <div>解码后音频缓冲区总节点个数，以帧为单位。</div> <div>解码后视频缓冲区数据节点个数，以帧为单位</div> <div>解码后音频缓冲区数据节点个数，以帧为单位</div>
返回值：	根据参数不同，返回缓冲区值，源缓冲区返回 <code>byte</code> ,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 <code>byte</code> ）。BUF_VIDEO_RENDER 可在网络延时方面有所估计。	
注 意：		

[返回目录](#)

## 5.9 索引

### 5.9.1 索引回调 **setFileRefCB**

函 数:	boolean setFileRefCB(int nPort, PlayerFileRefCB fileRefCB)	
参 数:	<div>int nPort PlayerFileRefCB fileRefCB</div> <div>回调函数 PlayerFileRefCB public void onFileRefDone(int nPort);</div> <div>参数说明: int nPort</div>	<div>播放通道号 索引回调函数, 可以为 NULL</div> <div>通道号</div>
返回值:	成功返回 true; 失败返回 false	
说 明:	<p>建立文件索引回调。</p> <p>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。</p> <p>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。</p>	
注 意:	在进行精准定位时, 由于需要解码一个 GOP 的数据, 在解码性能不足的情况下, 画面会停顿几秒再继续播放。	

[返回目录](#)

## 5.10 抓图

### 5.10.1 显示回调 **setDisplayCB**

函 数:	boolean setDisplayCB(int nPort, PlayerDisplayCB displayCB)	
参 数:	<div>int nPort</div> <div>PlayerDisplayCB displayCB</div> <div>回调函数 PlayerDisplayCB</div> <div>public void onDisplay(int nPort, byte[] data, int nDataLen, int nWidth, int nHeight, int nFrameTime, int nDataType, int Reserved);</div> <div>参数说明:</div> <div>int nPort</div> <div>byte[] data</div> <div>int nDataLen</div> <div>int nWidth,</div> <div>int nHeight</div> <div>int nFrameTime</div> <div>int nDataType</div> <div>int Reserved</div>	<div>播放通道号</div> <div>显示回调函数，可以为 NULL</div> <div>通道号</div> <div>返回图像数据指针</div> <div>返回图像数据大小</div> <div>画面宽，单位像素</div> <div>画面高</div> <div>时标信息，单位毫秒</div> <div>数据类型， T_YV12, T_RGB32, T_UYVY,</div> <div>用户数据</div>
返回值:	成功返回 true；失败返回 false	
说 明:	设置显示回调函数；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 <b>closeStream</b> 。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:	setDisplayCB 采用的每次回调都会进行 attach 和 dettach 操作，因此不需要设置为 NULL,但是这种方式可能会导致多路 JNI 回调的性能下降。	

[返回目录](#)

### 5.10.2 带全局时间显示回调 **setDisplayCBEx**

函 数:	boolean setDisplayCBEx(int nPort, PlayerCallBack.PlayerDisplayCBEx displayCBEx)	
参 数:	<div>int nPort</div> <div>PlayerDisplayCBEx displayCBEx</div> <div>回调函数 PlayerDisplayCBEx</div> <div>onDisplayEx(int nPort, byte[] data, int nDataLen, int nWidth, int nHeight, int nFrameTime, int nDataType, int Reserved, int nYear, int nMon, int nDay, int nHour, int</div>	<div>播放通道号</div> <div>显示回调函数，可以为 NULL</div>

	<div>nMin, int nSec, int nMs);</div> <div>参数说明:</div> <div><div><div>int</div><div>nPort</div></div><div>byte[] data</div><div><div>int</div><div>nDataLen</div></div><div><div>int</div><div>nWidth</div></div><div><div>int</div><div>nHeight</div></div><div><div>int</div><div>nFrameTime</div></div><div><div>int</div><div>nDataType</div></div><div><div>int</div><div>Reserved</div></div><div><div>int</div><div>nYear</div></div><div><div>int</div><div>nMon</div></div><div><div>int</div><div>nDay</div></div><div><div>int</div><div>nHour</div></div><div><div>int</div><div>nMin</div></div><div><div>int</div><div>nSec</div></div><div><div>int</div><div>nMs</div></div></div> <div><div>通道号</div><div>返回图像数据指针</div><div>返回图像数据大小</div><div>画面宽，单位像素</div><div>画面高</div><div>时标信息，单位毫秒</div><div>数据类型， T_YV12, T_RGB32, T_UYVY,</div><div>保留</div><div>年</div><div>月</div><div>日</div><div>时</div><div>分</div><div>秒</div><div>毫秒</div></div>
返回值:	成功返回 true; 失败返回 false
说 明:	<div>设置显示回调函数; 注意要尽快返回, 如果要停止回调, 可以把回调函数指针 displayCBEx 为 NULL。一旦设置回调函数, 则一直有效, 直到 stop。</div> <div>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。</div> <div>回调函数中获取的年、月、日、时、分、秒、毫秒信息是当前显示数据码流中解析出来的全局时间, 是 DSP 编码码流时打入的。</div> <div>SetDisplayCBEx 设置的带全局时间显示回调, 在调用 stop/closefile/closestream/freeport 的任何一个调用都会关闭当前设置的回调, 下次要获取全局时间显示回调, 用户必须重新设置。</div>
注 意:	SetDisplayCBEx 采用的 attach 和 Dettach 一次操作机制, 可以提升 JNI 回调性能。

[返回目录](#)

5.10.3 截取当前画面 BMP 图片 **getBMP**

函 数:	boolean getBMP(int nPort, byte[] pBmp, int nBufSize, MPIInteger pBmpSize)	
参 数:	<div><div><div>int</div><div>nPort</div></div><div>[out]unsigned char* pBitmap</div><div><div>unsigned int</div><div>nBufSize</div></div><div>[out]unsigned int *</div></div> <div><div>播放端口</div><div>存放 BMP 图像数据地址, 由用户分配, 不得小于 bmp 图像大小, 分硬解码和软解两种:</div><div>软 解 : sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER)+ w * h * 4, 其中 w 和 h 即分别为图像宽高。</div><div>硬解下: sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER)+backWidth*backHeight *4 其中 backWidth 和 backHeight 为窗口后缓存宽</div></div>	

	pBmpSize	高，跟窗口大小有关、跟图像宽高无关。 申请的缓冲区大小 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:	硬解下，用户可以通过 pBmp 传入 null 值，来获取所需分配缓存大小。 软解下，pBmp 传入 null 值，会返回错误码 2。	
注意:	2CIF 分辨率的图像抓图后的分辨率为 4CIF,抓图时分配的缓存 w、h 需按照 4CIF 分辨率来取值。	

[返回目录](#)

#### 5.10.4 私有信息 BMP 抓图 getBmpEx

函数:	public boolean getBmpEx(int nPort, byte[] pBitMap, int nBufferSize, MPInteger nBitMapSize)	
参数:	int nPort [out]byte[] pBitMap unsigned int nBufSize [out] MPInteger nBitMapSize	播放通道号 存放 BMP 图像数据地址，由用户分配，不得小于 $\text{sizeof}(\text{BITMAPFILEHEADER}) + \text{sizeof}(\text{BITMAPINFOHEADER}) + \text{backWidth} * \text{backHeight} * 4$ 其中 backWidth 和 backHeight 为窗口后缓存宽高，跟窗口大小有关、跟图像宽高无关。 申请的缓冲区大小 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:	用户可以通过 pBitMap 传入 null 值，来获取实际需要分配缓存大小。	
注意:	getbmpex 图片大小跟窗口分辨率有关，跟码流分辨率没有关系	

[返回目录](#)

#### 5.10.5 截取当前画面 JPEG 图片 getJPEG

函数:	boolean getJPEG(int nPort, byte[] pJpeg, int nBufSize, MPInteger pJpegSize)	
参数:	int nPort [out]byte[] pJpeg unsigned int nBufSize [out]unsigned int * pJpegSize	播放通道号 jpeg 图片数据缓存，由用户分配，因为无法预算出编码后的数据大小，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](#)

## 5.11 硬解码

### 5.11.1 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:	<p>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。</p> <p>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。</p> <p>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。</p>	
注 意:	<p>软硬解跟 home 交互操作:</p> <ul style="list-style-type: none"> <li>➤ 硬解状态下, 按 home 键切换到后台, 此时 surfaceview 被销毁, 由于 surfaceview 被销毁, 硬解码解码无法显示, 播放库内部会 try catch 硬解码接口调用异常, 自动切换为软解, 要想切换后台后继续为硬解码, 在切换至后台前, 需要关闭播放库, 切换到前台重新开启播放库即可。</li> <li>➤ 在软解下, 按 home 键切换到后台, 此时 surfaceview 被销毁, 由于 surfaceview 被销毁, 软解码解码无法显示, 此时切换会前台, 需要调用 setvideowindow(surfaceview.getholder()) 重新创建窗口, 创建完画面即可正常播放。</li> </ul> <p>新版播放库设置硬解码针对不同 API 分为两种硬解码方式, 并分别提供两种 so 库供 APP 调用:</p> <ul style="list-style-type: none"> <li>➤ AIP&lt;21(android5.0 以下), 采用老硬解码方式(跟 7.3.3 一致), 在 APP 运行时会自动加载 libPlayCtrl_L.so。</li> <li>➤ API&gt;=21(android5.0 及以上版本), 采用新硬解码方式, 支持电子放大、抓图功能, 在 APP 运行时会自动加载 libPlayCtrl.so。</li> </ul> <p>启用硬解码后不支持的功能接口有如下:</p> <ul style="list-style-type: none"> <li>➤ 私有数据显示 (renderPrivateData/renderPrivateDataEx)</li> <li>➤ 设置字体路径(setOverlayPriInfoFlag)</li> <li>➤ 二代鱼眼 (setDisEffect、setDisParam、getDisParam、setImageCorrection、verticalFlip)</li> <li>➤ 三代鱼眼 (enableFEC、disableFEC、getPortFEC、delPortFEC、setParamFEC、getParamFEC、setWnd、setWndEx、setCurrentPTZPort、setCurrentPTZPort、setPTZoutLineShowMode、setPTZToWindow、setFish3DRotate、getFishCapPicSize、getFishCapPic)</li> <li>➤ 创建或销毁窗口(setVideoWindow)</li> <li>➤ ivs 回调 (setIVSDrawFunCallback)</li> <li>➤ 倒放 (reversePlay)</li> <li>➤ 老硬解码方式不支持 getJPG、getBMP、setDisplayRegion、getBMPEX</li> <li>➤ 解码回调(setDecodeCB/setDecodeCBEx)硬解码可以设置成功, 但是不会起作用。建议硬解码下不要设置解码回调。</li> </ul>	

	<ul style="list-style-type: none"> <li>➤ 显示回调(setDisplayCB/setDisplayCBEx),硬解码下只起通知作用。</li> <li>➤ 新的硬解码方式只支持在原始窗口做电子放大。</li> <li>➤ 在开启硬解码下, 低帧率下(&lt;=1.0f)直接切换为软解。</li> <li>➤ 在开启硬解码下, 低分辨率下(W&lt;200 &amp;&amp; H&lt;200)直接切换为软解。</li> </ul>
--	--

[返回目录](#)

### 5.11.2 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:	<ul style="list-style-type: none"> <li>➤ 硬解码只支持 4.1 以上系统, 仅支持 H.264、H.265、MPEG4 编码数据, 且不支持电子放大、抓图、鱼眼矫正、倒放、图像垂直翻转、私有数据显示、同步播放等功能。</li> <li>➤ 设置硬解码最大路数, 对 switchToHard 调用没有限制, 例如: 当前预览 3 路, 开了 2 路硬解码(设置最大硬解码路数为 2), 另外一路软解也可以调用 switchToHard 切换到硬解。</li> </ul>	

[返回目录](#)

### 5.11.3 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:	老硬解码方式返回 1, 新的硬解码返回 2; 软解码返回 0	
说 明:	在码流解码正常播放后调用, 否则返回的解码类型无意义	
注 意:	<p>老的硬解码只支持 4.1 以上系统, 仅支持 H.264、H.265、MPEG4 编码数据, 且不支持电子放大、抓图、鱼眼矫正、倒放、图像垂直翻转、私有数据显示、同步播放等功能。</p> <p>5.0 系统及以上 都采用新的硬解码, 新的硬解码方式 支持电子放大、BMP 抓图</p>	

[返回目录](#)

### 5.11.4 切换到软解 **switchToSoft**

函 数:	boolean switchToSoft(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:	➤ 此接口主要是用来手动切换将硬解切换为软解; 如果已经是软解, 则调用	

	<p>switchToSoft 返回错误码 2。</p> <ul style="list-style-type: none"><li>➤ switchtosoft 后需要调用 play（不调用 play 可能会停止），switchtosoft 后不调用 play 条件下再调用 switchtoHard，会报顺序错误。</li><li>➤ 若开启了电子放大，调用此接口后电子放大效果消失。</li></ul> <p>软硬解跟 home 交互操作：</p> <ul style="list-style-type: none"><li>➤ 硬解状态下，按 home 键切换到后台，此时 surfaceview 被销毁，由于 surfaceview 被销毁，硬解码解码无法显示，播放库内部会 try catch 硬解码接口调用异常，自动切换为软解，要想切换后台后继续为硬解码，在切换至后台前，需要关闭播放库，切换到前台重新开启播放库即可。</li><li>➤ 在软解下，按 home 键切换到后台，此时 surfaceview 被销毁，由于 surfaceview 被销毁，软解码解码无法显示，此时切换会前台，需要调用 setvideowindow(surfaceview.getholder())重新创建窗口，创建完画面即可正常播放。</li></ul>
--	---

[返回目录](#)

### 5.11.5 切换到硬解 **switchToHard**

函 数：	boolean switchToHard(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：	<ul style="list-style-type: none"><li>➤ 此接口主要是用来手动切换将软解切换为硬解；如果已经是硬解，则调用 switchToHard 返回错误码 2。</li><li>➤ 当前版本切换硬解码只需要调用 switchToHard</li><li>➤ 软解切硬解时，需要销毁解码器和重新等待 1 帧，需要等到 2-3 秒左右时间画面播放。</li><li>➤ 硬解码只能原窗口电子放大。</li><li>➤ 若开启了电子放大，调用此接口后电子放大效果消失。</li><li>➤ 若码流的帧率小于 1，则默认切换为软解。</li></ul> <p>以下几种情况下调用 switchToHard 会返回不支持</p> <ul style="list-style-type: none"><li>➤ 编码方式不支持硬解码(除标准 H264(包含 smart264)、标准 H265(包含 smart265)、MPEG4 编码)</li><li>➤ 内部自动切换为软解的码流，主要包含：场编码、SVC 标记码流、走廊码流(视频图像宽&lt;高)，低分辨率码流(宽&lt;200 且高&lt;200)</li><li>➤ 设置多路同步播放</li><li>➤ 开启电子放大</li><li>➤ 开启三代鱼眼</li></ul> <p>软硬解跟 home 交互操作：</p> <ul style="list-style-type: none"><li>➤ 硬解状态下，按 home 键切换到后台，此时 surfaceview 被销毁，由于 surfaceview 被销毁，硬解码解码无法显示，播放库内部会 try catch 硬解码接口调用异常，自动切换</li></ul>	

	<p>为软解，要想切换后台后继续为硬解码，在切换至后台前，需要关闭播放库，切换到前台重新开启播放库即可。</p> <p>➤ 在软解下，按 home 键切换到后台，此时 <code>surfaceview</code> 被销毁，由于 <code>surfaceview</code> 被销毁，软解解码无法显示，此时切换会前台，需要调用 <code>setvideowindow(surfaceview.getholder())</code> 重新创建窗口，创建完画面即可正常播放。</p>
--	---

[返回目录](#)

## 5.12 预录像功能

### 5.12.1 设置预录像数据回调 **setPreRecordCallBack**

函 数:	boolean setPreRecordCallBack(int nPort, PlayerPreRecordCB PreRecordCB)	
参 数:	<div>int                    nPort</div> <div>PlayerPreRecordCB    PreRecordCB</div> <div>PlayerPreRecordCB 回调函数</div> <div>public void onPreRecord(int nPort, byte[] data, int nDataLen)</div> <div>回调函数参数说明</div> <div>int                    nPort</div> <div>byte[]                data</div> <div>int                    nDataLen</div>	<div>播放通道号</div> <div>数据回调函数</div> <div>用户指针</div> <div>播放通道号</div> <div>音视频数据</div> <div>数据长度</div>
返回值:	成功返回 1；失败返回 0	
说 明:	<p>需要在play之后，正常显示第一帧之后调用。</p> <p>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。</p> <p>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,然后通过回调函数中给用户，方便用户保存成文件查看。</p> <p>当前版本 setPreRecordCallBack 接口要在第一个 I 帧显示之后调用才行</p> <p>所以建议播放出图像再调用。</p> <p>播放的窗口句柄不能为NULL，必须是有效的显示窗口句柄</p>	
注 意:	<p>预录像回调中，回调出来的数据需要及时处理，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。因为若不及时处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。</p> <p>预录像 I/O 操作过慢的情况下会对实时性有影响。</p> <p>预录像回调接口均需要在设置预录像开关以后才能设置，否则返回错误码 2.</p>	
示例代码	<pre>int setPreRecordCallBack(int nPort, PlayerPreRecordCB);  //声明预录像回调函数 void onPreRecordCBfun(int nPort,byte[] buffer,int nLen);  setPreRecordCallBack(lPort, PreRecordCBfun,NULL);  switch (dwDataType) { case NET_DVR_SYSHEAD:     bFlag =getPort(&amp;lPort);</pre>	

	<pre>bFlag =setStreamOpenMode(lPort,1); bFlag =openStream(lPort,pBuffer,dwBufSize,4*1024*1024);  bFlag =setPreRecordFlag(lPort,1); bFlag =setDecodeCB(lPort,DecCBFun);  bFlag =play(lPort,pDlg-&gt;m_ctrlVideoPic.m_hWnd); break; case NET_DVR_STREAMDATA:  bFlag =inputData(lPort,pBuffer,dwBufSize);  if ((g_bFlag== 0)&amp;&amp;(g_bDecFlag == 1)) { Sleep(100); OutputDebugString("zytest:: set dec callback success !"); g_bFlag =setPreRecordCallBack(lPort,fPreRecordCBfun,pUser); if (g_bFlag == 0) { unsigned int dwErr =GetLastError(lPort); } else { OutputDebugString("zytest:: set pre record callback success !"); } } break;  }</pre>
--	--

[返回目录](#)

5.12.2 设置预录像数据回调 **setPreRecordCallBackEx**

函 数:	boolean setPreRecordCallBackEx(int nPort, PlayerPreRecordCBEx PreRecordCBEx)	
参 数:	int nPort PlayerPreRecordCBEx PreRecordCBEx  PlayerPreRecordCBEx 回调函数 public void onPreRecordEx(int nPort, byte[] data, int nDataLen, int nYear, int nMon, int nDay, int nHour, int nMin, int nSec, int nMs, int nType, int	播放通道号 数据回调函数 用户指针

	<div>nFrameNum, int nStampTime);</div> <div>回调函数参数说明</div> <table><tr><td>int</td><td>nPort</td><td>播放通道号</td></tr><tr><td>byte[]</td><td>data</td><td>音视频数据</td></tr><tr><td>int</td><td>nDataLen</td><td>数据长度</td></tr><tr><td>int</td><td>nYear,</td><td>年</td></tr><tr><td>int</td><td>nMon</td><td>月</td></tr><tr><td>int</td><td>nDay</td><td>日</td></tr><tr><td>int</td><td>nHour</td><td>时</td></tr><tr><td>int</td><td>nMin</td><td>分</td></tr><tr><td>int</td><td>nSec</td><td>秒</td></tr><tr><td>int</td><td>nMs</td><td>毫秒</td></tr><tr><td>int</td><td>nType</td><td>帧类型</td></tr><tr><td>int</td><td>nFrameNum</td><td>帧号</td></tr><tr><td>int</td><td>nStampTime</td><td>时间戳</td></tr></table>	int	nPort	播放通道号	byte[]	data	音视频数据	int	nDataLen	数据长度	int	nYear,	年	int	nMon	月	int	nDay	日	int	nHour	时	int	nMin	分	int	nSec	秒	int	nMs	毫秒	int	nType	帧类型	int	nFrameNum	帧号	int	nStampTime	时间戳	
int	nPort	播放通道号																																							
byte[]	data	音视频数据																																							
int	nDataLen	数据长度																																							
int	nYear,	年																																							
int	nMon	月																																							
int	nDay	日																																							
int	nHour	时																																							
int	nMin	分																																							
int	nSec	秒																																							
int	nMs	毫秒																																							
int	nType	帧类型																																							
int	nFrameNum	帧号																																							
int	nStampTime	时间戳																																							
返回值:	成功返回 1；失败返回 0																																								
说 明:	<p>需要在play之后，正常显示第一帧之后调用。</p> <p>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。</p> <p>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，然后通过回调函数中给用户，方便用户保存成文件查看。</p> <p>当前版本 setPreRecordCallBack 接口要在第一个 I 帧显示之后调用才行</p> <p>所以建议播放出图像再调用。</p> <p>播放的窗口句柄不能为NULL，必须是有效的显示窗口句柄</p>																																								
注 意:	<p>预录像回调中，回调出来的数据需要及时处理，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。因为若不及时处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。</p> <p>预录像 I/O 操作过慢的情况下会对实时性有影响。</p> <p>预录像回调接口均需要在设置预录像开关以后才能设置，否则返回错误码 2.</p>																																								
示例代码																																									

[返回目录](#)

5.12.3 设置预录像开关 **setPreRecordFlag**

函 数:	boolean setPreRecordFlag(int nPort, int bFlag)							
参 数:	<table><tr><td>int</td><td>nPort</td><td>播放通道号</td></tr><tr><td>bool</td><td>bFlag</td><td>设置预录像开关: 1 表示开启, 0 表示关闭</td></tr></table>	int	nPort	播放通道号	bool	bFlag	设置预录像开关: 1 表示开启, 0 表示关闭	
int	nPort	播放通道号						
bool	bFlag	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:	<p>需要在 play 之后，正常显示第一帧之后调用。</p> <p>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</p>							



	数返回给用户，方便用户保存成文件查看。
<b>注 意：</b>	因缓存原因录像会有 2~3 秒延时

[返回目录](#)

## 5.13 鱼眼功能

### 5.13.1 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](#)

### 5.13.2 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:	<ul style="list-style-type: none"><li>➤ 鱼眼矫正必须在已启用播放的窗口上进行，可以通过 <a href="#">setDisplayRegion</a>、<a href="#">setVideoWindow</a> 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)。</li><li>➤ <b>enDisplayEffect</b> 参数取值范围 0-8；若设定为 9（此版本不支持红蓝 3D 效果）则返回错误 16，不支持；若设定为其他值，则返回 0，且没有矫正效果。</li></ul>	

[返回目录](#)

### 5.13.3 获取鱼眼矫正参数 **getFECDisplayParam**

函 数:	boolean getFECDisplayParam(int nPort, int nRegionNum, MPVR_FISH_PARAM stFishParam)	
参 数:	<div>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; }</div>	<div>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</div>
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](#)

### 5.13.4 设置鱼眼矫正参数 **setFECDisplayParam**

函 数:	boolean setFECDisplayParam(int nPort, int nRegionNum, MPVR_FISH_PARAM stFishParam)	
参 数:	<div>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; }</div>	<div>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) PTZ 矫正中心 X 坐标, 取值范围: [0.0, 1.0] PTZ 矫正中心 Y 坐标, 取值范围: [0.0, 1.0]</div>
返回值:	成功返回 true; 失败返回 false	

<b>说 明:</b>	接口需要设置具体鱼眼效果后才可调用。 参数需要满足以下条件: $xLeft < xRight$ ; $yTop < yBottom$ ; 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。
<b>注 意:</b>	需要新的窗口实现鱼眼效果需要先使用电子放大创建对应的新的显示窗口

[返回目录](#)

### 5.13.5 开启鱼眼功能 **enableFEC**

<b>函 数:</b>	boolean enableFEC(int nPort)	
<b>参 数:</b>	int nPort	播放通道号;
<b>返回值:</b>	成功返回 true; 失败返回 false	
<b>说 明:</b>		
<b>注 意:</b>	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 此接口为7.3.2版本新添加的接口, 建议使用此接口。	

[返回目录](#)

### 5.13.6 关闭鱼眼功能 **disableFEC**

<b>函 数:</b>	boolean disableFEC(int nPort)	
<b>参 数:</b>	int nPort	播放通道号;
<b>返回值:</b>	成功返回 true; 失败返回 false	
<b>说 明:</b>		
<b>注 意:</b>	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 此接口为7.3.2版本新添加的接口, 建议使用此接口。	

[返回目录](#)

### 5.13.7 获取鱼眼矫正处理子端口 **getPortFEC**

<b>函 数:</b>	boolean getPortFEC(int nPort, int nPlaceType, int nCorrectType)	
<b>参 数:</b>	int nPort int nPlaceType 安装方式说明: <code>public static class FEC_PLACETYPE</code> { <code>public static final int FEC_NULL</code> = 0; <code>public static final int FEC_PLACE_WAL</code> = 1; <code>public static final int FEC_PLACE_FLOOR</code> = 2;	播放通道号; 安装方式  无安装 壁装方式(法线水平) 地面安装(法线向上)

	<pre>public static final int FEC_PLACE_CEILING = 3; } int nCorrectType 矫正方式说明: public static class FEC_CORRECTTYPE {     public static final int FEC_CORRECT_PTZ = 0;     public static final int FEC_CORRECT_180 = 1;     public static final int FEC_CORRECT_360 = 2;     public static final int FEC_CORRECT_LAT = 3;     public static final int FEC_CORRECT_SEM = 4;     public static final int FEC_CORRECT_CYC = 5;     public static final int FEC_CORRECT_PLA = 6;     public static final int FEC_CORRECT_CYC_SPL = 7; }</pre>	<p>顶装方式(法线向下)</p> <p>矫正方式</p> <p>PTZ</p> <p>180 度矫正 (对应 2P)</p> <p>360 全景矫正 (对应 1P)</p> <p>维度展开</p> <p>半球</p> <p>圆柱</p> <p>小行星</p> <p>剪开圆柱</p>
返回值:	成功返回子端口号, 取值范围为[2~5]; 失败返回 false	
说 明:	<p>➤ 三代鱼眼支持 4 路展开, 子端口取值为[2~5]。</p> <p>➤ 移动平台渲染和鱼眼依赖的是同一个渲染库资源。从设计上考虑, 鱼眼 ISubPort 从 2 开始。</p> <p>➤ 不同安装方式对应的矫正方式如下面表格所示:</p>	
注 意:	<p>➤ 7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用 7.3.2 及其以后版本的鱼眼接口。</p> <p>➤ 一个 Port 只对应一种安装方式。</p> <p>➤ 此接口为7.3.2版本新添加的接口, 建议使用此接口。</p>	

不同安装方式对应的矫正方式:

安 装 方 式	支持矫正方式			
	FEC_CORRECT_PTZ	FEC_CORRECT_180	FEC_CORRECT_360	FEC_CORRECT_LAT
FEC_ NULL	VR_ET_NULL	VR_ET_NULL	VR_ET_NULL	VR_ET_NULL
FEC_ PLAC E_W ALL (壁 装)	VR_ET_FISH_PTZ_WALL	不支持	VR_ET_FISH_LATIT UDE_WALL	VR_ET_FISH_LATIT UDE_WALL
FEC_ PLAC E_FL OOR (底 装)	VR_ET_FISH_PTZ_FLOOR FEC_CORRECT_SEM FEC_CORRECT_CYC EC_CORRECT_PLA EC_CORRECT_CYC_SPL	VR_ET_FISH_PANORAMA_FLOOR180 FEC_CORRECT_SEM FEC_CORRECT_CYC EC_CORRECT_PLA EC_CORRECT_CYC_SPL	VR_ET_FISH_PANORAMA_FLOOR360 FEC_CORRECT_SEM FEC_CORRECT_CYC EC_CORRECT_PLA EC_CORRECT_CYC_SPL	VR_ET_FISH_LATIT UDE_WALL FEC_CORRECT_SEM FEC_CORRECT_CYC EC_CORRECT_PLA EC_CORRECT_CYC_SPL

			SPL	
FEC_P	VR_ET_FISH_PTZ_CEILING	VR_ET_FISH_PANORAMA	VR_ET_FISH_PANORA	VR_ET_FISH_LATITUDE
LACE	FEC_CORRECT_SEM	_CEILING180	MA_CEILING360	_WALL
_CEILI	FEC_CORRECT_CYC	FEC_CORRECT_SEM	FEC_CORRECT_SEM	FEC_CORRECT_SEM
NG(	EC_CORRECT_PLA	FEC_CORRECT_CYC	FEC_CORRECT_CYC	FEC_CORRECT_CYC
顶装)	EC_CORRECT_CYC_SPL	EC_CORRECT_PLA	EC_CORRECT_PLA	EC_CORRECT_PLA
		EC_CORRECT_CYC_SPL	EC_CORRECT_CYC_SPL	EC_CORRECT_CYC_SPL

[返回目录](#)

### 5.13.8 删除鱼眼矫正处理子端口 **delPortFEC**

函 数:	boolean delPortFEC(int nPort, int nSubPort)	
参 数:	int nPort int nSubPort	播放通道号; 删除子通道号 (子端口取值为[2~5])
返回值:	成功返回 true; 失败返回 false。	
说 明:	三代鱼眼支持4 路展开, 子端口取值为[2~5]	
注 意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 此接口为7. 3. 2版本新添加的接口, 建议使用此接口。	

[返回目录](#)

### 5.13.9 设置鱼眼矫正参数 **setParamFEC**

函 数:	boolean setParamFEC(int nPort, int nSubPort, FISHEYE_PARAM stParam)	
参 数:	int nPort int nSubPort FISHEYE_PARAM stParam	播放通道号; 处理子端口 鱼眼矫正参数结构体
	FISHEYE_PARAM 结构体说明: <pre>public static class FISHEYE_PARAM {     public int updateType;     public int placeAndCorrect;     public float zoom;      public float wideScanOffset;      public PTZ_PARAM ptzParam;     public CYCLE_PARAM cycleParam; }</pre>	
	updateType 说明: public static final int FEC_UPDATE_RADIUS	更新类型 安装、矫正方式 PTZ显示的范围参数 (取值范围: [0.00001, 1.0]) 180或360矫正的偏移角度(取值 范围 (0.0, 360.0) ) PTZ矫正参数 鱼眼图像圆心参数 更新圆心相关参数

	<pre> public static final int FEC_UPDATE_PTZZOOM          0x2 public static final int FEC_UPDATE_WIDESCANOFFSET  0x4 public static final int FEC_UPDATE_PTZPARAM        0x8 </pre> <p>更新zoom值 更新矫正偏移角度 更新ptz相关参数</p> <p>PTZ_PARAM 结构体说明:</p> <pre> public static class PTZ_PARAM {     public float ptzPositionX;     public float ptzPositionY; } </pre> <p>PTZ显示的中心位置 X坐标 PTZ显示的中心位置 Y坐标</p> <p>CYCLE_PARAM 结构体说明:</p> <pre> public static class CYCLE_PARAM {     public float radiusLeft;     public float radiusRight;     public float radiusTop;     public float radiusBottom; } </pre> <p>圆的最左边X坐标 圆的最右边X坐标 圆的最上边Y坐标 圆的最下边Y坐标</p>
返回值:	成功返回 true; 失败返回 false
说 明:	三代鱼眼支持4路展开, 子端口取值为[2~5]
注 意:	<p>➤ 7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用 7.3.2 及其以后版本的鱼眼接口。</p> <p>➤ 此接口为7.3.2版本新添加的接口, 建议使用此接口。</p> <p>➤ stParam参数中fZoom及fWideScanOffset取值必须满足绝对值<math>\geq 0.00001</math>, 此时参数设置才能生效; 否则虽然接口不报错, 但是实际没有效果。鱼眼不支持翻转。</p> <p>参数设置说明:</p> <p>➤ 设置鱼眼参数时会根据更新数据类型顺序处理; 优先的更新类型矫正错误时接口直接返回错误, 后面的更新类型不做处理;</p> <p>➤ 优先的更新类型矫正正确, 但是后面的更新类型矫正错误, 画面会显示优先的矫正效果, 返回后面矫正类型对应的错误码。</p> <p>➤ 更新类型判断顺序: 同时更新ptz和zoom&gt;只更新ptz不更新zoom&gt;更新圆心相关参数&gt;更新偏移角度&gt;只更新zoom不更新ptz。</p> <p>➤ 更新类型判断顺序: 同时更新ptz和zoom&gt;只更新ptz不更新zoom&gt;更新圆心相关参数&gt;更新偏移角度&gt;只更新zoom不更新ptz。</p> <p>➤ 圆心参数nLeft&lt;0.0f、nRight&gt;1.0f、nTop&lt;0.0f、nBottom&gt;1.0f设置成功但无效果。</p> <p>➤ 360度全景矫正画面修改偏移角度后, PTZ框消失。</p> <p>➤ 此接口对3D鱼眼设置无效果。</p> <p>例如: 参数设置了更新PTZ和圆心参数; 情况1、ptz更新错误, 则直接退出, 后面的圆心参数不生效; 情况2、ptz更新正确, 而圆心参数更新错误, 此时有ptz更新效果, 但是接口报错。</p>

	<p>setParamFEC 如果设置为updateType: 0x8</p> <pre>ptzParam { fPTZPositionX: 1.0 fPTZPositionY: 1.0}}</pre> <p>则由渲染库设置有效半径范围为(0.0001f, 0.4f), 鱼眼图像中默认圆心(0.5f, 0.5f), 计算出&gt;0.4f时 直接返回参数错误码非PTZ107越界错误码。</p> <p>PTZ中心坐标 参数限制如下:</p> <pre>float r = (float)sqrt((pFishPara-&gt;stPTZParam.fPTZPositionY - 0.5)*(pFishPara-&gt;stPTZParam.fPTZPositionY - 0.5) + (pFishPara-&gt;stPTZParam.fPTZPositionX - 0.5)*(pFishPara-&gt;stPTZParam.fPTZPositionX - 0.5)); if (r &gt; 0.4f) {     return VR_E_PARAM; }</pre> <p>圆心参数范围:</p> <pre>pFishPara-&gt;stCycleParam.fRadiusLeft [0.0,1.0] pFishPara-&gt;stCycleParam.fRadiusTop [0.0,1.0] pFishPara-&gt;stCycleParam.fRadiusRight [0.0,1.0] pFishPara-&gt;stCycleParam.fRadiusBottom [0.0,1.0]</pre> <p>PTZ矫正放大系数:</p> <pre>pFishPara-&gt;fZoom [0.00001, 1.0]</pre> <p>180或360角度中心角:</p> <pre>pFishPara-&gt;fWideScanOffset 没限制</pre>
--	--

[返回目录](#)

5.13.10 获取鱼眼矫正参数 **getParamFEC**

函 数:	boolean getParamFEC(int nPort, int nSubPort, FISHEYE_PARAM stParam)	
参 数:	<div>int nPort</div> <div>int nSubPort</div> <div>FISHEYE_PARAM stParam</div> <div>FISHEYE_PARAM 结构体说明:</div> <div>public static class FISHEYE_PARAM</div> <div>{</div> <div>public int updateType;</div> <div>public int placeAndCorrect;</div> <div>public float zoom;</div> <div>public float wideScanOffset;</div> <div>public PTZ_PARAM ptzParam;</div>	<div>播放通道号;</div> <div>处理子端口</div> <div>鱼眼矫正参数结构体</div> <div>更新类型</div> <div>安装、矫正方式</div> <div>PTZ显示的范围参数 (取值范围: [0.00001, 1.0])</div> <div>180或360矫正的偏移角度(取值范围(0.0, 360.0))</div> <div>PTZ矫正参数</div>





```
stFEPara1.stPTZParam.fPTZPositionX = 0.2; //PTZ 显示的中心坐标
stFEPara1.stPTZParam.fPTZPositionY = 0.3;
stFEPara1.fZoom = 0.1; //PTZ 显示的范围参数

//stCycleParam 用于圆心对当前鱼眼所有子端口都有影响
setParamFEC (m_lPort,m_nSubPort1,&stFEPara1);

//子端口 2 顶装，PTZ 矫正 获取鱼眼矫正处理子端口
getPortFEC (m_lPort,&m_nSubPort2,FEC_PLACE_CEILING,FEC_CORRECT_PTZ);

//更新位置
stFEPara2.nUpDateType = FEC_UPDATE_PTZPARAM;
stFEPara2.stPTZParam.fPTZPositionX = 0.7;
stFEPara2.stPTZParam.fPTZPositionY = 0.8;
setParamFEC (m_lPort,m_nSubPort2,&stFEPara2);

//Zoom
stFEPara3.nUpDateType = FEC_UPDATE_PTZZOOM;
stFEPara3.fZoom = 0.9; //(0,1)

setParamFEC (m_lPort,m_nSubPort2,&stFEPara3);

//更新位置
stFEPara3.nUpDateType = FEC_UPDATE_PTZPARAM;
stFEPara3.stPTZParam.fPTZPositionX = 0.7;
stFEPara3.stPTZParam.fPTZPositionY = 0.8;
stFEPara3.fZoom = 0.9;
setParamFEC (m_lPort,m_nSubPort2,&stFEPara3))

//Zoom
stFEPara3.nUpDateType = FEC_UPDATE_PTZZOOM;
setParamFEC (m_lPort,m_nSubPort2,&stFEPara3);

setWnd (m_lPort,m_nSubPort1,hWnd1);
```

[返回目录](#)

5.13.12 设置显示窗口 **setWndEx**

函 数:	boolean setWndEx(int nPort, int nSubPort, SurfaceTexture texture)		
参 数:	int	nPort	播放通道号
	int	nSubPort	处理子端口
	SurfaceTexture	texture	设置显示窗口句柄

返回值:	成功返回 true; 失败返回 false
说明:	三代鱼眼支持4 路展开, 子端口取值为[2~5]
注意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 此接口为7.3.2版本新添加的接口

[返回目录](#)

### 5.13.13 获取当前选择鱼眼子端口 **getCurrentPTZPort**

函数:	int getCurrentPTZPort(int nPort, int nPanorama, float fPositionX, float fPositionY)	
参 数:	int nPort	播放通道号
	int nPanorama	0表示操作原图; 1则表示操作全景图鼠标x位置 (接触屏幕点x坐标)
	float fPositionX	鼠标y位置 (接触屏幕点y坐标)
	float fPositionY	
返回值:	成功返回子端口号; 失败返回-1	
说明:		
注意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持4 路展开, 子端口取值为[2~5]。 此接口为7.3.2新增加接口;	

[返回目录](#)

### 5.13.14 设置当前选择鱼眼子端口 **setCurrentPTZPort**

函 数:	boolean setCurrentPTZPort(int nPort, int nSubPort)	
参 数:	int nPort	播放通道号
	int nSubPort	子端口号
返回值:	成功返回 true; 失败返回 false	
说明:		
注意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持4 路展开, 子端口取值为[2~5]。 此接口为7.3.2新增加接口;	

[返回目录](#)

### 5.13.15 画矩形或者不规则框选择 **setPTZoutLineShowMode**

函 数:	boolean setPTZoutLineShowMode(int nPort, int nShowmode)	
参 数:	int nPort	播放通道号
	int nShowmode	展现模式
	FEC_SHOWMODE 结构体说明:	

	<pre>public static class FEC_SHOWMODE {     public static final int FEC_PTZ_OUTLINE_NULL    = 0;     public static final int FEC_PTZ_OUTLINE_RECT   = 1;     public static final int FEC_PTZ_OUTLINE_RANGE  = 2; }</pre>	不显示 矩形显示 (暂不支持) 真实区域显示
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持4 路展开, 子端口取值为[2~5]。 此接口为7.3.2新增加接口;	

[返回目录](#)

5.13.16 获取当前子端口号对应的当前 PTZ 坐标值 **setPTZ2Window**

函 数:	boolean setPTZ2Window(int nPort, int nSubPort, PTZ_PARAM stPTZRefOrigin, PTZ_PARAMM stPTZRefWindow, PTZ_PARAM stPTZWindow, MPFloat stPosX, MPFloat stPosY)	
参 数:	<div><div><div>int</div><div>nPort</div></div><div>int</div><div>nSubPort</div><div>PTZ_PARAM</div><div>stPTZRefOrigin</div><div>PTZ_PARAM</div><div>stPTZRefWindow</div><div>PTZ_PARAM</div><div>stPTZWindow</div><div>MPFloat</div><div>stPosX,</div><div>MPFloat</div><div>stPosY</div></div> <div>PTZ_PARAM 结构体说明: <pre>public static class PTZ_PARAM {     public float ptzPositionX;     public float ptzPositionY; }</pre></div>	<div>播放通道号</div> <div>设置期望的子port号</div> <div>当前PTZ坐标值</div> <div>手指或鼠标触碰控制区域的初始坐标值</div> <div>手指或鼠标触碰控制区域的当前坐标值</div> <div>计算出新的PTZ坐标X值</div> <div>计算出新的PTZ坐标Y值</div> <div> PTZ显示的中心位置 X坐标</div> <div>PTZ显示的中心位置 Y坐标</div>
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持4 路展开, 子端口取值为[2~5]。 此接口为7.3.2新增加接口;	

[返回目录](#)

### 5.13.17 获取当前子端口号对应的当前 PTZ 坐标值

#### setFECDisplayPTZ

函 数:	boolean setFECDisplayPTZ(int nPort, int nSubPort, MPVR_FISH_PTZ stOriPtz, MPVR_FISH_PTZ stWndPtz, MPVR_FISH_PTZ stWnd,MPFloat stX,MPFloat stY)	
参 数:	<div> <div>int</div> <div>nPort</div> </div> <div> <div>int</div> <div>nSubPort</div> </div> <div> <div>PTZ_PARAM</div> <div>stPTZRefOrigin</div> </div> <div> <div>PTZ_PARAM</div> <div>stPTZRefWindow</div> </div> <div> <div>PTZ_PARAM</div> <div>stPTZWindow</div> </div> <div> <div>MPFloat</div> <div>stPosX,</div> </div> <div> <div>MPFloat</div> <div>stPosY</div> </div> <div>           PTZ_PARAM 结构体说明:  <pre>public static class PTZ_PARAM {     public float ptzPositionX;     public float ptzPositionY; }</pre> </div>	<div>播放通道号</div> <div>设置期望的子port号</div> <div>当前PTZ坐标值</div> <div>手指或鼠标触碰控制区域的初始坐标值</div> <div>手指或鼠标触碰控制区域的当前坐标值</div> <div>计算出新的PTZ坐标X值</div> <div>计算出新的PTZ坐标Y值</div> <div> PTZ显示的中心位置 X坐标</div> <div>PTZ显示的中心位置 Y坐标</div>
返回值:	成功返回 true；失败返回 false	
说 明:		
注 意:	7.3.1 及之前版本和7.3.2 及以后版本鱼眼接口不可同时使用。建议使用7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持4 路展开，子端口取值为[2~5]。 此接口为7.3.2新增加接口；	

[返回目录](#)

### 5.13.18 设置鱼眼 3D 旋转（此设置只针对 3D 鱼眼起作用）

#### setFish3DRotate

函 数:	public boolean setFish3DRotate(int nPort, int nSubPort, SRTRANS_PARAM srTransParam )	
参 数:	<div> <div>int</div> <div>nPort</div> </div> <div> <div>int</div> <div>nSubPort</div> </div> <div> <div>SRTRANS_PARAM</div> <div>srTransParam</div> </div> <div>           SRTRANS_PARAM 结构体说明:  <pre>public static class SRTRANS_PARAM</pre> </div>	<div>播放通道号</div> <div>设置期望的子port号</div> <div>3D旋转参数设置结构体</div>

	<pre> {     public float ptzPositionX;     public float ptzPositionY;     public  SRTRANS_ELEMENT srtransElement;     public  int nTransCount; }  SRTRANS_ELEMENT 结构体说明: public static class SRTRANS_ELEMENT {     public float    fAxisX;     public         float fAxisY;     public float    fAxisZ;     public float    fValue; } </pre>	<p>PTZ显示的中心位置 X坐标</p> <p>PTZ显示的中心位置 Y坐标</p> <p>未用到</p> <p>绕x轴旋转值</p> <p>绕Y轴旋转值</p> <p>绕Z轴旋转值(未使用)</p> <p>缩放系数, 此值为正数时设置后画面缩小</p>
返回值:	成功返回 <b>true</b> ; 失败返回 <b>false</b>	
说 明:		
注 意:	<p>参数fAxisX、fAxisY、fValue取值范围根据3D鱼眼类型取值不同, 具体如下:</p> <p>1.绕X轴旋转范围:</p> <p>    小行星: 3.14~6.28</p> <p>    切开圆柱: <math>-\pi/4 \sim 0</math></p> <p>    其他无限制;</p> <p>2.绕Y轴旋转无限制</p> <p>3.缩放</p> <p>    对于半球: -0.7~900</p> <p>    对于圆柱: 0~900</p> <p>    对于小行星: -1.238~0.216</p> <p>    对于切开圆柱: 不支持</p> <p>    其他: 无限制</p> <p>参数超出范围时, 对于切开圆柱下绕X轴是限位(如范围是0~1, 你设置-1就限制到0, 设置2就限制到1), 其余效果设置时参数越界报错。</p> <p>旋转时参数设置顺序为Y,X, Value, 后面的参数错误不影响前面正确参数旋转效果(如绕Y轴旋转正确时, 但X参数错误, 此时调用接口返回报错但是画面会按Y参数设置旋转)。</p> <ul style="list-style-type: none"> <li>● 圆柱剪开、3D鱼眼矫正时, Y周旋转取值范围为[0.0,1.0)。</li> <li>● 设置鱼眼3D, 临界取值时, 由于float类型在各个手机平台精度不一致, 所设置的值会有所不同, 如上层设置-900.7 在三星s6上取值为-900.700012(小数点最后5位是不准确取值), 而其他平台可能会-900.700000。因此计算出来有所差别。导致在不同手机上可能会出现返回值不一致问题。</li> </ul> <p>鱼眼3D旋转设置默认值:</p> <p>VR_ET_FISH_PLANET_CEILING</p> <pre> m_fScale = 0.0f m_fRotateX = 3.14f m_fRotateY = 0.0f </pre>	

	<div>VR_ET_FISH_CYLINDER_CEILING</div> <div>m_fScale = 5.0f</div> <div>m_fRotateX = 0.0f</div> <div>m_fRotateY = 0.0f</div>
	<div>VR_ET_FISH_SEMISPHERE_CEILING</div> <div>m_fScale = 0.0f</div> <div>m_fRotateX = 0.0f</div> <div>m_fRotateY = 0.0f</div>
	<div>VR_ET_FISH_PLANET_FLOOR</div> <div>m_fScale = 0.0f</div> <div>m_fRotateX = 3.14f</div> <div>m_fRotateY = 0.0f</div>
	<div>VR_ET_FISH_CYLINDER_FLOOR</div> <div>m_fScale = 5.0f</div> <div>m_fRotateX = 0.0f</div> <div>m_fRotateY = 0.0f</div>
	<div>VR_ET_FISH_SEMISPHERE_FLOOR</div> <div>m_fScale = 0.0f</div> <div>m_fRotateX = 0.0f</div> <div>m_fRotateY = 0.0f</div>
	<div>VR_ET_FISH_CYLINDER_SPLIT_FLOOR</div> <div>m_fScale = 2.72f</div> <div>m_fRotateX = 0.0f</div> <div>m_fRotateY = 0.0f</div>
	<div>VR_ET_FISH_CYLINDER_SPLIT_CEILING</div> <div>m_fScale = 2.72f</div> <div>m_fRotateX = 0.0f</div> <div>m_fRotateY = 0.0f</div>

[返回目录](#)

5.13.19 获取鱼眼抓图所需缓存大小 **getFishCapPicSize**

函 数:	public boolean getFishCapPicSize(int nPort, int nSubPort, MPInteger nCapPicSize)		
参 数:	int	nPort	播放通道号
	int	nSubPort	设置期望的子port号
	[out]MPInteger	nCapPicSize	实际抓图图片大小
返回值:	成功返回 true; 失败返回 false		

说 明:	
注 意:	

[返回目录](#)

5.13.20 鱼眼抓图 **getFishCapPic**

函 数:	public boolean getFishCapPic(int nPort, int nSubPort, int nType, byte[] pFishPic, int nFishPicSize)		
参 数:	int	nPort	播放通道号
	int	nSubPort	设置期望的子port号
	int	nType	数据类型，（暂只支持1：BMP）
	byte[]	pFishPic	图片缓冲区（由用户申请）
	int	nFishPicSize	图片大小
返回值:	成功返回 true；失败返回 false		
说 明:	nFishPicSize获取的值由getFishCapPicSize提供，且pFishPic分配的缓存需要>=nFishPicSize		
注 意:	需要先调用getFishCapPicSize获取nFishPicSize，然后调用此接口实现鱼眼抓图。		

[返回目录](#)





	<div>PLAYM4_PRIDATA_RENDER 类型说明</div> <div><pre>public static class PRIVATE_RENDER {     public static final int RENDER_ANA_INTEL_DATA = 1;     public static final int RENDER_MD = 2;     public static final int RENDER_ADD_POS = 4;     public static final int RENDER_ADD_PIC = 8;     public static final int RENDER_FIRE_DETCTET = 16;     public static final int RENDER_TEM = 32; }</pre></div> <div><div>int nSubType</div><div>int bTrue</div></div> <div><div>智能分析</div><div>移动侦测</div><div>POS 信息后叠加</div><div>图片叠加信息</div><div>热成像信息</div><div>温度信息</div><div>子类型</div><div>是否显示</div></div>
返回值:	成功返回 true; 失败返回 false
说 明:	<div>当nIntelType 为PLAYM4_RENDER_FIRE_DETCTET，则子类</div> <div>型取值为PLAYM4_FIRE_ALARM;</div> <div><pre>public static class FIRE_ALARM {     public static final int FIRE_FRAME_DIS = 1; // 火点框显示     public static final int FIRE_MAX_TEMP = 2; // 最高温度     public static final int FIRE_MAX_TEMP_POSITION = 4; // 最高温度位置显示     public static final int FIRE_DISTANCE = 8; //最高温度距离 }</pre></div> <div>当nIntelType 为PLAYM4_RENDER_TEM，则子类型取值为PLAYM4_TEM_FLAG。</div> <div><pre>public static class TEM_FLAG {     public static final int TEM_REGION_BOX = 1; // 框测温     public static final int TEM_REGION_LINE = 2; // 线测温     public static final int TEM_REGION_POINT = 4; // 点测温 }</pre></div>
注 意:	<div><div>➤ 此接口适用于带有智能信息的码流，若无智能信息的码流, 则调用此接口没有效果。</div><div>➤ 只有智能信息和移动侦测是默认开启的，POS信息叠加和图片叠加暂不支持。热成像默认开启，温度信息默认关闭。</div><div>➤ 热成像设备开启电子放大无效。</div><div>➤ 私有数据不支持图片叠加和POS信息叠加。</div><div>➤ 热成像信息和温度信息中的字符，不支持叠加。</div></div> <div>参数设定说明如下:</div> <div><div>➤ 打开: 先打开大开关nIntelType，再打开小开关nSubType，如测温-&gt;线测温，先开大开关温度，再开小开关线;</div><div>➤ 关闭: 大开关nIntelType控制全部，如关闭测温，则关闭所有测温; 小开关nSubType</div></div>

	关闭具体的，如关闭线测温： ➤ 关闭后再打开： 保留小开关nSubType信息。 如关闭大开关nIntelType测温时，线测温是打开的，再次开启大开关nIntelType测温，线测温也是开的。 ➤ 小开关nSubType的关闭不受大开关nIntelType开启和关闭限制。
--	---

[返回目录](#)

5.14.5 私有数据回调 **setAdditionalCallback**

函 数:	boolean setAdditionalCallback(int nPort, int nSyncType, PlayerAdditionalCB additionCB)		
参 数:	int	nPort	播放通道号
	int	nSyncType	同步类型
	PlayerAdditionalCB	additionCB	私有数据回调函数
	PlayerAdditionalCB 回调函数		
	public void onAdditional (int nPort, int nType, int nStrVersion, int nTimeStamp, int nLength, byte[] data )		
	回调函数参数说明		
	int	nPort	播放通道号
	int	nType	私有数据类型
	int	nStrVersion	数据返回的结构体版本，主要是为了兼容性
	int	nTimeStamp	私有数据时间戳
int	nLength	私有数据长度	
byte[]	data	私有数据指针	
返回值:	成功返回 true；失败返回 false		
说 明:	此接口适用于带有智能信息的码流，若无智能信息，则回调函数不响应。（车载信息等）		
注 意:			

[返回目录](#)

5.14.6 私有数据回调 **setIVSDrawFunCallback**

函 数:	boolean setIVSDrawFunCallback(int nPort, PlayerIVSDrawFunCB drawFunCB)	
参 数:	<div><div>intnPort</div><div>PlayerIVSDrawFunCBdrawFunCB</div><div>PlayerIVSDrawFunCB 回调函数</div><div>public void onIVSDrawFun(int nPort, int nWidth, int nHeight, int nStamp, int nType, int nFrameRate, int nFrameNum,int nDataType,int nDataLen,byte[]</div></div>	<div>播放通道号</div> <div>私有数据回调函数</div>

	data)	
	回调函数参数说明	
	int nPort	播放通道号
	int nWidth	图像宽
	int nHeight	图像高
	int nStamp	当前视频帧时间戳
	int nType	当前帧类型（当前只支持 T_YV12）
	int nFrameRate	视频帧率
	int nFrameNum	视频帧号
	int nDataType	和码流数据同步的附属信息类型，目前有：智能信息，车载信息
	int nDataLen	数据长度
	byte[] data	数据指针
返回值：	成功返回 true；失败返回 false	
说 明：	此接口需要开启私有数据显示(默认开启)。客户通过此接口获取到私有信息后自己实现私有信息显示。 硬解码不支持此项功能。	
注 意：	此接口适用于带有智能信息的码流。（点、线、框相关）。	

[返回目录](#)

### 5.14.7 设置字体库路径 **setOverlayPriInfoFlag**

函 数：	boolean setOverlayPriInfoFlag(int nPort, int nIntelType, int bTrue, String filePath)	
参 数：	int nPort	播放通道号
	int nIntelType	保留类型，暂时没用到
	int bTrue	是否设置
	String filePath	字体库路径
返回值：	成功返回 true；失败返回 false	
说 明：	此接口需要开启私有数据显示(默认开启)。客户通过此接口获取到私有信息后自己实现私有信息显示。	
注 意：	此接口主要用于设置字体路径，进行叠字需要同时开启pos信息叠加 renderPrivateData(nPort,4,1)。	

[返回目录](#)

## 5.15 倒放

### 5.15.1 倒放 **reversePlay**

函 数:	boolean reversePlay(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:	<ul style="list-style-type: none"><li>➤ 倒放不播放音频。分辨率大于等于1080P时，倒放的GOP只能存放25帧，所以I帧间隔大于25时，会有卡顿。</li><li>➤ 文件模式(openFile)下倒放前需要创建文件索引，支持最大倒放速度比正放最大速度要慢。暂不支持TS封装文件。</li><li>➤ 流播放模式(openStream)时，实时流数据播放不支持倒放。非实时流数据，即录像文件以读取数据方式然后送入到播放库中以流模式播放，可以实现倒放，但是应用层送入解码的数据必须是一个GOP，且是逆序送入的数据，即将录像文件先解析出GOP，然后根据GOP的时序，逆序的送入到播放库中。如本地PC上的录像文件，客户自己读取录像文件数据分析出各个GOP(一般指I帧机器后面依靠该I帧解码的其他帧)，文件时间顺序为 I1PPPPPPP..., I2PPPPPPP..., I3PPPPPPPP...,....., InPPPPPPPP..., 则送入到播放库中倒放需要按照 InPPPPPP..., In-1PPPPPP...,....., I3PPPPPPPP..., I2PPPPPPPP..., I1PPPPPPPP...的顺序依次送入。</li></ul>	
注 意:		

[返回目录](#)

## 6. 见问题排查

### 关于流模式接口

#### Question 1 送入到播放库 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 解码回调和显示回调函数的区别

#### 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 Android 版本的播放库问题

### Question 22 Android 版本的播放库不能正常预览

#### Answer 22

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；