



播放库

SDK 编程指南

(for iOS)

V7.0.2.X

HIKVISION

杭州海康威视数字技术股份有限公司

<http://www.hikvision.com>

技术热线：400-700-5998

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.....	2
1 产品简介.....	6
2 SDK 版本更新.....	7
版本号说明.....	7
Version 7.0.2.X.....	7
3 错误代码及说明.....	8
4 有关显示的说明.....	9
5 函数调用顺序.....	10
6 函数说明.....	11
系统操作及错误号获取.....	11
6.1. 获取错误号 PlayM4_GetLastError.....	11
6.2. 测试播放所需系统功能 PlayM4_GetCaps*.....	11
6.3. 初始化 DDrawPlayM4_InitDDraw*.....	11
6.4. 释放 Direct 表面 PlayM4_RealseDDraw*.....	11
6.5. 设置播放使用的定时器类型 PlayM4_SetTimerType.....	12
6.6. 获取播放使用的定时器类型 PlayM4_GetTimerType.....	12
6.7. 获取未使用的通道号 PlayM4_GetPort.....	12
6.8. 释放已使用的通道号 PlayM4_FreePort.....	13
文件操作.....	14
6.9. 打开文件 PlayM4_OpenFile.....	14
6.10. 关闭文件 PlayM4_CloseFile.....	14
流操作.....	15
6.11. 设置流播放模式 PlayM4_SetStreamOpenMode.....	15
6.12. 获取流播放模式 PlayM4_GetStreamOpenMode.....	15
6.13. 打开流 PlayM4_OpenStream.....	15
6.14. 关闭流 PlayM4_CloseStream.....	16
6.15. 输入流数据 PlayM4_InputData.....	16
6.16. 打开流（以音视频分开输入方式）PlayM4_OpenStreamEx*.....	16
6.17. 关闭流（以音视频分开输入方式）PlayM4_CloseStreamEx*.....	16
6.18. 输入视频流 PlayM4_InputVideoData*.....	17
6.19. 输入音频流 PlayM4_InputAudioData*.....	17
播放控制.....	18
6.20. 开启播放 PlayM4_Play.....	18
6.21. 关闭播放 PlayM4_Stop.....	18
6.22. 暂停/恢复播放 PlayM4_Pause.....	18
6.23. 快速播放 PlayM4_Fast.....	18
6.24. 慢速播放 PlayM4_Slow.....	19
6.25. 单帧前进 PlayM4_OneByOne.....	19

6.26. 单帧回退 PlayM4_OneByOneBack.....	19
6.27. 以独占方式打开声音 PlayM4_PlaySound.....	19
6.28. 关闭声音（独占方式） PlayM4_StopSound.....	20
6.29. 以共享方式打开声音 PlayM4_PlaySoundShare.....	20
6.30. 关闭声音（共享方式） PlayM4_StopSoundShare.....	20
6.31. 设置音量 PlayM4_SetVolume.....	20
6.32. 获取音量 PlayM4_GetVolume.....	21
6.33. 调整 WAVE 波形 PlayM4_AdjustWaveAudio*.....	21
6.34. 音视频同步 PlayM4_SyncToAudio*.....	21
6.35. 设置图像质量 PlayM4_SetPicQuality*.....	21
6.36. 获取图像质量 PlayM4_GetPictureQuality*.....	22
6.37. 设置视频参数 PlayM4_SetColor*.....	22
6.38. 获取视频参数 PlayM4_GetColor*.....	22
6.39. 设置播放模式 PlayM4_SetPlayMode*.....	22
6.40. 设置文件当前播放位置（百分比） PlayM4_SetPlayPos.....	23
6.41. 获取文件当前播放位置（百分比） PlayM4_GetPlayPos.....	23
6.42. 设置文件当前播放时间（毫秒） PlayM4_SetPlayedTimeEx.....	23
6.43. 获取文件当前播放时间（毫秒） PlayM4_GetPlayedTimeEx.....	23
6.44. 设置文件当前播放帧号 PlayM4_SetCurrentFrameNum.....	24
6.45. 获取文件当前播放帧号 PlayM4_GetCurrentFrameNum.....	24
6.46. 去闪烁 PlayM4_SetDeflash*.....	24
获取播放或解码信息.....	25
6.47. 获取文件总时间 PlayM4_GetFileTime.....	25
6.48. 获取文件开始和结束的全局时间 PlayM4_GetFileTimeEx.....	25
6.49. 获取文件总帧数 PlayM4_GetFileTotalFrames.....	25
6.50. 获取当前帧率 PlayM4_GetCurrentFrameRate.....	25
6.51. 获取当前帧率 PlayM4_GetCurrentFrameRateEx.....	26
6.52. 获取已播放时间 PlayM4_GetPlayedTime.....	26
6.53. 获取已解码的视频帧数 PlayM4_GetPlayedFrames.....	26
6.54. 获取原始图像大小 PlayM4_GetPictureSize.....	26
6.55. 获取文件头长度 PlayM4_GetFileHeadLength.....	27
6.56. 获取当前显示帧的全局时间 PlayM4_GetSpecialData.....	27
6.57. 设置检查水印回调函数 PlayM4_SetCheckWatermarkCallBack*.....	27
6.58. 获取 Abs 帧信息 PlayM4_GetAbsFrameNum*.....	28
解码操作及控制.....	29
6.59. 设置解码回调流类型 PlayM4_SetDecCBStream.....	29
6.60. 设置视频帧解码类型 PlayM4_SetDecodeFrameType.....	29
6.61. 解码回调 PlayM4_SetDecCallBack.....	29
6.62. 解码回调（增加用户传递参数） PlayM4_SetDecCallBackMend.....	30
6.63. 解码回调（带目标数据和数据大小） PlayM4_SetDecCallBackEx.....	31
6.64. 解码回调（带目标数据和数据大小和用户指针） PlayM4_SetDecCallBackExMend32.....	31
6.65. 音频解码回调 PlayM4_SetAudioCallBack*.....	33
6.66. 文件结束消息 PlayM4_SetFileEndMsg*.....	33
6.67. 文件结束回调 PlayM4_SetFileEndCallback.....	33

6.68. 分辨改变通知消息 PlayM4_SetEncChangeMsg*	34
6.69. 分辨率改变通知回调 PlayM4_SetEncTypeChangeCallBack	34
6.70. 设置丢 B 帧个数 PlayM4_ThrowBFrameNum	34
6.71. 获取丢 B 帧数回调 PlayM4_GetThrowBFrameCallBack*	35
6.72. 检查帧号是否连续 PlayM4_CheckDiscontinuousFrameNum*	35
6.73. 解码密钥 PlayM4_SetSecretKey	35
显示操作	36
6.74. 设置 OVERLAY 显示模式及关键色 PlayM4_SetOverlayMode*	36
6.75. 获取当前是否采用 OVERLAY 显示模式 PlayM4_GetOverlayMode*	36
6.76. 获取 OVERLAY 关键色 PlayM4_GetColorKey*	36
6.77. 设置图像的锐度 PlayM4_SetImageSharpen*	36
6.78. 设置图像 Overlay 翻转模式 PlayM4_SetOverlayFlipMode*	37
6.79. 设置或增加显示区域 PlayM4_SetDisplayRegion	37
6.80. 设置显示窗口 PlayM4_SetVideoWindow	37
6.81. 刷新显示 PlayM4_RefreshPlay*	38
6.82. 刷新显示（多显示区域）PlayM4_RefreshPlayEx*	38
6.83. 设置显示模式 PlayM4_SetDisplayType*	38
6.84. 获取显示模式 PlayM4_GetDisplayType*	38
缓冲区操作	39
解码前源缓冲区	39
6.85. 获取源缓冲区剩余数据大小 PlayM4_GetSourceBufferRemain	39
6.86. 设置源缓冲区阈值及回调指针 PlayM4_SetSourceBufCallBack	39
6.87. 重置回调标志为有效状态 PlayM4_ResetSourceBufFlag*	39
解码后播放缓冲区	40
6.88. 设置播放缓冲区最大缓冲帧数 PlayM4_SetDisplayBuf	40
6.89. 获取播放缓冲区最大缓冲帧数 PlayM4_GetDisplayBuf	40
源、播放缓冲区	40
6.90. 清空所有缓冲区的剩余数据 PlayM4_ResetSourceBuffer	40
6.91. 清空指定缓冲区的剩余数据 PlayM4_ResetBuffer	41
6.92. 获取指定缓冲区的大小 PlayM4_GetBufferValue	41
6.93. 注册用户数据回调 PlayM4_SetGetUserDataCallBack*	42
索引	43
6.94. 设置建立索引回调 PlayM4_SetFileRefCallBack	43
6.95. 查找指定位置之前的关键帧位置 PlayM4_GetKeyFramePos*	43
6.96. 查找指定位置之后的关键帧位置 PlayM4_GetNextKeyFramePos*	44
6.97. 获取文件索引 PlayM4_GetRefValue	44
6.98. 设置文件索引 PlayM4_SetRefValue	44
显示	45
6.99. 枚举系统中的显示设备 PlayM4_InitDDrawDevice*	45
6.100. 释放枚举显示设备过程分配的资源 PlayM4_ReleaseDDrawDevice*	45
6.101. 获取显示设备（显卡）个数 PlayM4_GetDDrawDeviceTotalNums*	45
6.102. 设置播放窗口使用的显示设备 PlayM4_SetDDrawDevice*	46
6.103. 设置播放窗口使用的显示设备（多显示区域）PlayM4_SetDDrawDeviceEx *	46
6.104. 获取指定显卡和监视器信息 PlayM4_GetDDrawDeviceInfo*	46

6.105. 获取指定显示设备的系统信息 PlayM4_GetCapsEx*.....	46
抓图.....	47
6.106. 抓图回调 PlayM4_SetDisplayCallBack.....	47
6.107. 显示回调（带用户指针） PlayM4_SetDisplayCallBackEx.....	47
6.108. 图像数据转为 BMP 格式 PPlayM4_ConvertToBmpFile.....	48
6.109. 图像数据转为 JPEG 格式 PlayM4_ConvertToJpegFile.....	48
6.110. 直接抓取 BMP 图像 PlayM4_GetBMP.....	49
6.111. 直接抓取 JPEG 图像 PlayM4_GetJPEG.....	49
6.112. 设置 JPEG 图像质量 PlayM4_SetJpegQuality.....	50
其他功能.....	51
6.113. 画图回调 PlayM4_RegisterDrawFun.....	51
6.114. 画图回调 PlayM4_RigisterDrawFun.....	51
6.115. 设置数据校验 PlayM4_SetVerifyCallBack*.....	52
6.116. 获取原始帧数据回调函数 PlayM4_GetOriginalFrameCallBack*.....	52
6.117. 获取文件最后的时标和帧号 PlayM4_GetFileSpecialAttr *.....	52
6.118. 跳过错误数据 PlayM4_SkipErrorData.....	52

1 产品简介

海康威视播放库 SDK（以下简称“播放库 SDK”）是海康威视嵌入式网络硬盘录像机、视频服务器、IP 设备的配套产品的播放相关的二次开发包，适用于以下编码产品数据流的解码与播放：

DS-90xx 系列、DS-76xx 系列混合硬盘录像机

DS-91xx 系列、DS-81xx/71xx/72xxHV 系列、DS-80xx/70xx/72xxH 系列、DS-78xx/88xx 系列、专用系列（ATM、车载、环境监测、审讯）网络硬盘录像机

DS-95xx/96xx 系列、DS-76xx 系列 NVR

DS-60xx 系列、DS-61xx、DS-63xx、DS-64xx、DS-65xx 系列视频服务器、编/解码器；

DS-40xx/41xx/42xx 系列板卡

IP 设备：包含 IP 模块、IP 摄像机、IP 快球等

播放库 SDK 主要功能：

主要用于实时码流预览，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。

2 SDK 版本更新

[转到下一章节](#)

版本号说明

播放库 SDK 版本号自 V7.0.2.X 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 7.0.2.X

Version 7.0.2.X (2012-12-14)

版本说明：（**ver 7.0.2.X**）

- 增加对 mpeg2 视频的支持
- 增加对 g726, amr, mpeg, aac 音频的支持
- IOS 视频解码只对标准 264 进行过优化，其他为进行优化

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
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_FAIL_UNKNOWN	99	错误

[返回目录](#)

4 有关显示的说明

注意:

- 1、**DirectDraw**是windows平台下的图像显示技术，iOS平台下PlayM4_InitDDraw , PlayM4_RealseDDraw , PlayM4_InitDDrawDevice, PlayM4_ReleaseDDrawDevice, PlayM4_SetDDrawDevice, PlayM4_SetDDrawDeviceEx, PlayM4_GetDDrawDeviceTotalNums, , PlayM4_GetDDrawDeviceInfo, PlayM4_GetCaps, PlayM4_GetCapsEx, PlayM4_SetDDrawDeviceEx接口无效;
- 2、**iOS平台overlay渲染技术尚未实现**，PlayM4_GetOverlayMode, PlayM4_SetOverlayFlipMode, PlayM4_SetOverlayMode, PlayM4_GetColorKey接口暂不支持;
- 3、**iOS平台向窗口传消息的机制尚未实现**，PlayM4_SetEncChangeMsg, PlayM4_SetFileEndMsg, PlayM4_SetEncChangeMsg接口暂不支持;
- 4、PlayM4_RefreshPlayEx, PlayM4_SetDeflash, PlayM4_InputVideoData, PlayM4_InputAudioData, PlayM4_SetPicQuality, PlayM4_GetPictureQuality, PlayM4_SetSourceBufCallBack, PlayM4_ResetSourceBufFlag, PlayM4_GetKeyFramePos, PlayM4_GetNextKeyFramePos, PlayM4_CheckDiscontinuousFrameNum, PlayM4_SetDisplayType, PlayM4_GetDisplayType, PlayM4_SetColor, PlayM4_GetColor, PlayM4_AdjustWaveAudio, PlayM4_SetAudioCallBack, PlayM4_OpenStreamEx, PlayM4_CloseStreamEx, PlayM4_SetPicQuality, PlayM4_GetPictureQuality, PlayM4_ResetSourceBufFlag, PlayM4_GetKeyFramePos, PlayM4_GetNextKeyFramePos, PlayM4_SetDisplayType, PlayM4_GetDisplayType, PlayM4_SyncToAudio, PlayM4_RefreshPlayEx, PlayM4_AdjustWaveAudio, PlayM4_SetPlayMode, PlayM4_GetAbsFrameNum, PlayM4_SetColor, PlayM4_GetColor, PlayM4_SetImageSharpen, PlayM4_SetCheckWatermarkCallBack, PlayM4_SetGetUserDataCallBack, PlayM4_GetThrowBFrameCallBack, PlayM4_SetAudioCallBack等接口在该版本中暂不支持;
- 5、PlayM4_GetOriginalFrameCallBack, PlayM4_GetFileSpecialAttr接口等和基线保持一致, 不再支持;
- 6、某些数据结构如PLAYM4_HWND, PLAYM4_HDC等, 在不同平台有不同的定义, 具体请参考相关头文件。

5 函数调用顺序

应用程序初始化

文件模式下

PlayM4_GetPort
PlayM4_SetFileRefCallBack
PlayM4_OpenFile
PlayM4_Play
PlayM4_Stop
PlayM4_CloseFile
PlayM4_FreePort

流模式下

PlayM4_GetPort
PlayM4_SetStreamOpenMode
PlayM4_OpenStream
PlayM4_SetDisplayBuf
PlayM4_Play
PlayM4_InputData
PlayM4_Stop
PlayM4_CloseStream
PlayM4_FreePort

应用程序结束

6 函数说明

系统操作及错误号获取

6.1. 获取错误号 **PlayM4_GetLastError**

函 数: unsigned int PlayM4_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](#)

6.2. 测试播放所需系统功能 **PlayM4_GetCaps***

函 数: int PlayM4_GetCaps()

参 数:

返回值:

说 明:

注 意: iOS 平台不支持该接口。

[返回目录](#)

6.3. 初始化 DDraw**PlayM4_InitDDraw***

函 数: int PlayM4_InitDDraw(PPLAYM4_HWND hWnd)

参 数:

返回值:

说 明:

注 意: iOS 平台不支持该接口。

[返回目录](#)

6.4. 释放 Direct 表面 **PlayM4_RealeseDDraw***

函 数: int PlayM4_RealeseDDraw()

参 数: 无

返回值:

说 明:

注 意: 该接口 iOS 平台无效

[返回目录](#)

6.5. 设置播放使用的定时器类型 **PlayM4_SetTimerType**

函 数: int PlayM4_SetTimerType(int nPort,unsigned int nTimerType, unsigned int nReserved)

参 数: int nPort 播放通道号
 unsigned int TIMER_1 时间精度比较高;
 nTimerType TIMER_2 相比于 TIMER_1, 时间精度低于 TIMER_1。
 unsigned int 保留参数
 nReserved

返回值: 成功返回 1; 失败返回 0

说 明:

注 意: 必须在 Open 之前调用。建议在实时预览时设置成 Timer2, 在文件回放时设置为 Timer1。

TIMER_1 和 TIMER_2 不同的系统时钟精度不相同, TIMER_1 精度高于 TIMER_2.

[返回目录](#)

6.6. 获取播放使用的定时器类型 **PlayM4_GetTimerType**

函 数: int PlayM4_GetTimerType(int nPort,unsigned int *pTimerType, unsigned int *pReserved)

参 数: int nPort 播放通道号
 [out] unsigned int * pTimerType TIMER_1 或 TIMER_2
 unsigned int * pReserved 保留参数

返回值: 成功返回 1; 失败返回 0

说 明:

注 意:

[返回目录](#)

6.7. 获取未使用的通道号 **PlayM4_GetPort**

函 数: int PlayM4_GetPort(int * nPort)

参 数: [out] int * nPort 播放通道号, 指向用于获取端口号的 int 型变量指针

返回值: 成功返回 1; 失败返回 0

说 明: 获取未使用的通道号,通道号的上限为 500。

注 意:

[返回目录](#)



6.8. 释放已使用的通道号 **PlayM4_FreePort**

函 数: `int PlayM4_FreePort(int nPort)`

参 数: `int nPort` 播放通道号

返回值: 成功返回 1; 失败返回 0

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

6.9. 打开文件 **PlayM4_OpenFile**

函 数: int PlayM4_OpenFile(int nPort,char* sFileName)

参 数: int nPort 播放通道号

char* sFileName 文件名

返回值: 成功返回 1; 失败返回 0

说 明:

注 意: 文件不能超过 4G 或小于 4K

[返回目录](#)

6.10.关闭文件 **PlayM4_CloseFile**

函 数: int PlayM4_CloseFile(int nPort)

参 数: 无

返回值: 成功返回 1; 失败返回 0

说 明:

注 意:

[返回目录](#)

流操作

6.11.设置流播放模式 **PlayM4_SetStreamOpenMode**

函 数: int PlayM4_SetStreamOpenMode(int nPort,unsigned int nMode)

参 数: int nPort 播放通道号

unsigned int 流播放模式: STREAME_REALTIME、STREAME_FILE

nMode STREAME_REALTIME 模式（默认）下, 会尽量保正实时性, 防止数据阻塞; 而且数据检查严格; (适合播放网络实时数据, 解码器会立刻解码)

STREAME_FILE 文件模式, 文件流模式, 适合于文件回放方式, 偏重于流畅性而不注重实时性; (按帧率播放)

返回值: 成功返回 1; 失败返回 0

说 明:

注 意: 必须在播放之前设置。

[返回目录](#)

6.12.获取流播放模式 **PlayM4_GetStreamOpenMode**

函 数: Int PlayM4_GetStreamOpenMode(int nPort)

参 数: int nPort 播放通道号

返回值: STREAME_REALTIME 或 STREAME_FILE

说 明:

注 意:

[返回目录](#)

6.13.打开流 **PlayM4_OpenStream**

函 数: int PlayM4_OpenStream(int nPort,unsigned char* pFileHeadBuf,unsigned int nSize, unsigned int nBufPoolSize)

参 数: int nPort 播放通道号

unsigned char* pFileHeadBuf 文件头数据（用户从回调中得到的文件头数据）

unsigned int nSize 文件头长度

unsigned int nBufPoolSize 设置播放器中存放数据流的缓冲区大小。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。其中

#define SOURCE_BUF_MAX 1024*100000

#define SOURCE_BUF_MIN 1024*50

返回值： 成功返回 1；失败返回 0

说 明：

注 意：

[返回目录](#)

6.14.关闭流 PlayM4_CloseStream

函 数： int PlayM4_CloseStream(int nPort)

参 数： Int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明：

注 意：

[返回目录](#)

6.15.输入流数据 PlayM4_InputData

函 数： int PlayM4_InputData(int nPort,unsigned char* pBuf,unsigned int nSize)

参 数： int nPort 播放通道号
 unsigned char* pBuf 流数据缓冲区地址
 unsigned int nSize 流数据缓冲区大小

返回值： 1 表示已经输入数据。0 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 0，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： 实时流模式下建议尝试几次后，为了保证实时性，可丢弃数据。

[返回目录](#)

6.16.打开流（以音视频分开输入方式） PlayM4_OpenStreamEx*

函 数： int PlayM4_OpenStreamEx(int nPort,unsigned char* pFileHeadBuf,unsigned int nSize,unsigned int nBufPoolSize)

参 数：

返回值：

说 明：

注 意： iOS 平台该接口暂不支持

[返回目录](#)

6.17.关闭流（以音视频分开输入方式） PlayM4_CloseStreamEx*

函 数： int PlayM4_CloseStreamEx(int nPort)

参 数：

返回值:

说 明:

注 意: iOS 平台该接口暂不支持

[返回目录](#)

6.18.输入视频流 **PlayM4_InputVideoData***

函 数: int PlayM4_InputVideoData(int nPort,unsigned char* pBuf,unsigned int nSize)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口暂不支持

[返回目录](#)

6.19.输入音频流 **PlayM4_InputAudioData***

函 数: int PlayM4_InputAudioData(int nPort,unsigned char* pBuf,unsigned int nSize)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口暂不支持

[返回目录](#)

播放控制

6.20.开启播放 **PlayM4_Play**

函 数: int PlayM4_Play(int nPort, PLAYM4_HWND hWnd)

参 数: int nPort 播放通道号
PLAYM4_HWND hWnd 播放视频的窗口句柄

返回值: 成功返回 1; 失败返回 0

说 明: 播放开始, 播放视频画面大小将根据 hWnd 窗口调整, 要全屏显示, 只要把 hWnd 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](#)

6.21.关闭播放 **PlayM4_Stop**

函 数: int PlayM4_Stop(int nPort)

参 数: int nPort 播放通道号

返回值: 成功返回 1; 失败返回 0

说 明:

注 意:

[返回目录](#)

6.22.暂停/恢复播放 **PlayM4_Pause**

函 数: int PlayM4_Pause(int nPort, unsigned int nPause)

参 数: int nPort 播放通道号
unsigned int nPause 1 暂停, 0 恢复

返回值: 成功返回 1; 失败返回 0

说 明:

注 意:

[返回目录](#)

6.23.快速播放 **PlayM4_Fast**

函 数: int PlayM4_Fast(int nPort)

参 数: int nPort 播放通道号

返回值: 成功返回 1; 失败返回 0

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](#)

6.24.慢速播放 PlayM4_Slow

函 数： int PlayM4_Slow(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

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](#)

6.25.单帧前进 PlayM4_OneByOne

函 数： int PlayM4_OneByOne(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明： 恢复正常播放需要调用 PlayM4_Play()

注 意：

[返回目录](#)

6.26.单帧回退 PlayM4_OneByOneBack

函 数： int PlayM4_OneByOneBack(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明： 单帧回退。

注 意： 每调用一次回退一帧，此函数必须在文件索引生成之后才能调用

[返回目录](#)

6.27.以独占方式打开声音 PlayM4_PlaySound

函 数： int PlayM4_PlaySound(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明： 打开声音；同一时刻只能有一路声音。

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

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](#)



6.28.关闭声音（独占方式） PlayM4_StopSound

函 数： int PlayM4_StopSound()

参 数： 无

返回值： 成功返回 1；失败返回 0

说 明： 关闭最后一次使用 PlayM4_PlaySound 接口打开声音的播放通道

注 意： PlayM4_PlaySound，PlayM4_StopSound 需要配对调用。

[返回目录](#)

6.29.以共享方式打开声音 PlayM4_PlaySoundShare

函 数： int PlayM4_PlaySoundShare(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明： 以共享方式播放声音，只播放本路声音而不去关闭其他路的声音。

注 意： 共享和非共享的音频接口不建议混用

[返回目录](#)

6.30.关闭声音（共享方式） PlayM4_StopSoundShare

函 数： int PlayM4_StopSoundShare(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明： 以共享方式关闭声音。

Playm4_PlaySound 和 PlayM4_StopSound 是以独占方式播放声音的。

注 意： 共享和非共享的音频接口不建议混用

[返回目录](#)

6.31.设置音量 PlayM4_SetVolume

函 数： int PlayM4_SetVolume(int nPort,unsigned short nVolume)

参 数： int nPort 播放通道号
unsigned short nVolume 音量值，范围 0-0XFFFF

返回值： 成功返回 1；失败返回 0

说 明： 可以在播放之前设置，返回值是 0，但设置的值被保存，并作为启动声音时的初始音量。这里的音量是只声卡输出的主音量，会影响到其他的声音应用。

注 意：

[返回目录](#)

6.32. 获取音量 **PlayM4_GetVolume**

函 数: unsigned short PlayM4_GetVolume(int nPort)

参 数: int nPort 播放通道号

返回值: 获取当前的音量值。这里的音量是只声卡输出的主音量，会影响到其他的声音应用。

范围为 0~0xffff

说 明:

注 意:

[返回目录](#)

6.33. 调整 WAVE 波形 **PlayM4_AdjustWaveAudio***

函 数: int PlayM4_AdjustWaveAudio(int nPort, int nCoefficient)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.34. 音视频同步 **PlayM4_SyncToAudio***

函 数: int PlayM4_SyncToAudio(int nPort, int bSyncToAudio)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.35. 设置图像质量 **PlayM4_SetPicQuality***

函 数: int PlayM4_SetPicQuality(int nPort, int bHighQuality)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.36. 获取图像质量 **PlayM4_GetPictureQuality***

函 数: `int PlayM4_GetPictureQuality(int nPort, int *bHighQuality)`

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.37. 设置视频参数 **PlayM4_SetColor***

函 数: `int PlayM4_SetColor(int nPort, unsigned int nRegionNum, int nBrightness, int nContrast, int nSaturation, int nHue)`

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.38. 获取视频参数 **PlayM4_GetColor***

函 数: `int PlayM4_GetColor(int nPort, unsigned int nRegionNum, int *pBrightness, int *pContrast, int *pSaturation, int *pHue)`

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.39. 设置播放模式 **PlayM4_SetPlayMode***

函 数: `int PlayM4_SetPlayMode(int nPort, int bNormal)`

参 数:

返回值:

说 明: 设置播放模式为按帧率播放还是时间戳播放

注 意: iOS 平台该接口无效。

[返回目录](#)

6.40. 设置文件当前播放位置（百分比） **PlayM4_SetPlayPos**

函 数: `int PlayM4_SetPlayPos(int nPort, float fRelativePos)`
参 数: `int nPort` 播放通道号
`float fRelativePos` 播放位置, 范围 0-100%
返回值: 成功返回 1; 失败返回 0
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, 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。

注 意:

[返回目录](#)

6.41. 获取文件当前播放位置（百分比） **PlayM4_GetPlayPos**

函 数: `float PlayM4_GetPlayPos(int nPort)`
参 数: `int nPort` 播放通道号
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](#)

6.42. 设置文件当前播放时间（毫秒） **PlayM4_SetPlayedTimeEx**

函 数: `int PlayM4_SetPlayedTimeEx(int nPort, unsigned int nTime)`
参 数: `int nPort` 播放通道号
`unsigned int nTime` 设置文件播放到指定时间, 单位毫秒
返回值: 成功返回 1; 失败返回 0
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](#)

6.43. 获取文件当前播放时间（毫秒） **PlayM4_GetPlayedTimeEx**

函 数: `unsigned int PlayM4_GetPlayedTimeEx(int nPort)`
参 数: `int nPort` 播放通道号
返回值: 文件当前播放的时间, 单位毫秒
说 明:
注 意:

[返回目录](#)

6.44. 设置文件当前播放帧号 **PlayM4_SetCurrentFrameNum**

函 数: `int PlayM4_SetCurrentFrameNum(int nPort,unsigned int nFrameNum)`

参 数: `int nPort` 播放通道号
`unsigned int nFrameNum` 帧号

返回值: 成功返回 1; 失败返回 0

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](#)

6.45. 获取文件当前播放帧号 **PlayM4_GetCurrentFrameNum**

函 数: `unsigned int PlayM4_GetCurrentFrameNum(int nPort)`

参 数: `int nPort` 播放通道号

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](#)

6.46. 去闪烁 **PlayM4_SetDeflash***

函 数: `int PlayM4_SetDeflash(int nPort,int bDeflash)`

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口

[返回目录](#)

获取播放或解码信息

6.47. 获取文件总时间 `PlayM4_GetFileTime`

函 数: `unsigned int PlayM4_GetFileTime(int nPort)`

参 数: `int nPort` 播放通道号

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](#)

6.48. 获取文件开始和结束的全局时间 `PlayM4_GetFileTimeEx`

函 数: `Int PlayM4_GetFileTimeEx(int nPort, unsigned int* pStart, unsigned int* pStop, unsigned int* pRev)`

参 数: `int nPort` 播放通道号
`[out]unsigned int* pStart` 文件开始时间
`[out]unsigned int* pStop` 文件结束时间
`[out]unsigned int* pRev` 保留

返回值: 成功返回 1, 失败返回 0;

说 明:

注 意:

[返回目录](#)

6.49. 获取文件总帧数 `PlayM4_GetFileTotalFrames`

函 数: `unsigned int PlayM4_GetFileTotalFrames(int nPort)`

参 数: `int nPort` 播放通道号

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](#)

6.50. 获取当前帧率 `PlayM4_GetCurrentFrameRate`

函 数: `unsigned int PlayM4_GetCurrentFrameRate(int nPort)`

参 数: `int nPort` 播放通道号

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、若返回 0, 表明该接口不支持

[返回目录](#)

6.51. 获取当前帧率 **PlayM4_GetCurrentFrameRateEx**

函 数: unsigned int PlayM4_GetCurrentFrameRate(int nPort)
参 数: int nPort 播放通道号
返回值: 当前码流中当前编码帧率
说 明:
注 意: 主要是为增加对帧率小于 1 的情况下获取帧率而增加的

[返回目录](#)

6.52. 获取已播放时间 **PlayM4_GetPlayedTime**

函 数: unsigned int PlayM4_GetPlayedTime(int nPort)
参 数: int nPort 播放通道号
返回值: 文件当前播放时间, 单位秒
说 明:
注 意:

[返回目录](#)

6.53. 获取已解码的视频帧数 **PlayM4_GetPlayedFrames**

函 数: unsigned int PlayM4_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](#)

6.54. 获取原始图像大小 **PlayM4_GetPictureSize**

函 数: int PlayM4_GetPictureSize(int nPort, int *pWidth, int *pHeight)
参 数: int nPort, 播放通道号
[out] int * pWidth, 原始图像的宽度
[out] int * pHeight 原始图像的高度
返回值: 成功返回 1; 失败返回 0
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。

注 意： 建议配合 PlayM4_SetEncTypeChangeCallBack/PlayM4_SetEncChangeMsg 使用

[返回目录](#)

6.55. 获取文件头长度 PlayM4_GetFileHeadLength

函 数： unsigned int PlayM4_GetFileHeadLength()

参 数： 无

返回值： 返回用于交互的海康媒体文件头或海康媒体信息头的大小

说 明： 得到当前版本播放器能播放的文件的文件头长度。 主要应用在流播放器的 STREAME_FILE 模式下。以便读出文件头作为 PlayM4_OpenStream()的输入参数
例程(以播放通道为 0 的通道为例)：

注 意：

[返回目录](#)

6.56. 获取当前显示帧的全局时间 PlayM4_GetSpecialData

函 数： Unsigned int PlayM4_GetSpecialData(LONG nPort)

参 数： LONG nPort 播放通道号

返回值： 失败返回-1，成功则返回一个全局时间的压缩值，精确到秒

```
#define GET_YEAR(_time_)      (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)    (((_time_)>>22) & 15)
#define GET_DAY(_time_)      (((_time_)>>17) & 31)
#define GET_HOUR(_time_)     (((_time_)>>12) & 31)
#define GET_MINUTE(_time_)   (((_time_)>>6) & 63)
#define GET_SECOND(_time_)   (((_time_)>>0) & 63)
```

说 明： 有正常图像显示后才能调用此接口

[返回目录](#)

6.57. 设置检查水印回调函数 PlayM4_SetCheckWatermarkCallBack*

函 数： unsigned int PlayM4_SetCheckWatermarkCallBack(int nPort,void(CALLBACK*
funCheckWatermark)(int nPort,WATERMARK_INFO* pWatermarkInfo,unsigned int
nUser),unsigned int nUser)

参 数：

返回值：

说 明： iOS 平台下暂不支持

[返回目录](#)



6.58. 获取 Abs 帧信息 `PlayM4_GetAbsFrameNum*`

函 数: `unsigned int PlayM4_GetAbsFrameNum(int nPort)`

参 数:

返回值:

说 明: `iOS` 平台下暂不支持

[返回目录](#)

解码操作及控制

6.59.设置解码回调流类型 **PlayM4_SetDecCStream**

函 数: int PlayM4_SetDecCStream(int nPort,unsigned int nStream)
参 数: int nPort 播放通道号
unsigned int nStream 1 视频流, 2 音频流, 3 复合流
返回值: 成功返回 1; 失败返回 0
说 明: 建议在解码回调函数前使用
注 意:

[返回目录](#)

6.60.设置视频帧解码类型 **PlayM4_SetDecodeFrameType**

函 数: BOOL PlayM4_SetDecodeFrameType(LONG nPort,DWORD nFrameType)
参 数: LONG nPort 播放通道号
DWORD nFrameType 解码帧类型说明
#define DECODE_NORMAIL 0 正常解码
#define DECODE_KEY_FRAME 1 只解关键帧
#define DECODE_NONE 2 不解视频帧
返回值: 成功返回 1; 失败返回 0
说 明:

[返回目录](#)

6.61.解码回调 **PlayM4_SetDecCallBack**

函 数: int PlayM4_SetDecCallBack(int nPort,void (CALLBACK* DecCBFun)(int nPort,char
* pBuf, int nSize,FRAME_INFO * pFrameInfo, int nReserved1, int nReserved2))
参 数: int nPort 播放通道号
void (CALLBACK* DecCBFun) 解码回调函数指针, 不能为 NULL

DecCBFun 回调函数参数说明

int	nPort	播放器通道号
char *	pBuf	解码后的音视频数据
int	nSize	解码后的音视频数据 pBuf 的长度
FRAME_INFO *	pFrameInfo	图像和声音信息
int	nReserved1	保留参数
int	nReserved2	保留参数

FRAME_INFO 结构说明

typedef struct{

int	nWidth;	画面宽，单位像素。如果是音频数据则为音频声道数；
int	nHeight;	画面高。如果是音频数据则为样位率；
int	nStamp;	时标信息，单位毫秒
int	nType	数据类型，T_AUDIO16， T_YV12，
int	nFrameRate;	编码时产生的图像帧率，如果是音频数据则为采样率
unsigned int		帧号
	dwFrameNum;	
}FRAME_INFO;		

宏定义说明

T_AUDIO16	音频数据;采样率 16khz，单声道，每个采样点 16 位表示。
T_YV12	视频数据,yv12 格式。排列顺序“Y0-Y1-.....”，“V0-V1....”，“U0-U1-.....”。 关于图像格式详细信息，用户可以查阅相关资料。

返 回 成功返回 1；失败返回 0

值:

说 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 **明:** PlayM4_Play 之前调用，在 PlayM4_Close 时自动失效，下次调用 PlayM4_Play 之前需要重新设置。

注 解码部分不控制**播放速度**，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。
意: 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。

[返回目录](#)

6.62.解码回调（增加用户传递参数） **PlayM4_SetDecCallBackMend**

函 数:	int	PlayM4_SetDecCallBackMend(int	nPort,void (CALLBACK* DecCBFun)
		(int	nPort,char * pBuf, int nSize,FRAME_INFO * pFrameInfo, int nUser, int nReserved2), int nUser)
参 数:	int	nPort	播放通道号
	void (CALLBACK* DecCBFun)		解码回调函数，不能为 NULL
	int	nUser	用户数据

DecCBFun 回调函数参数

int	nPort	播放通道号
char *	pBuf	解码后的视音频数据
int	nSize	解码后的视音频数据长度
FRAME_INFO *	pFrameInfo	图像与声音信息
int	nUser	用户数据

int nReserved2 保留参数

FRAME_INFO 类型说明

```
typedef struct{
    int nWidth          画面宽，单位像素。如果是音频数据则为音频声道数；
    int nHeight         画面高。如果是音频数据为样位率
    int nStamp          时标信息，单位毫秒
    int nType           数据类型，T_AUDIO16， T_YV12
    int nFrameRate      编码时产生的图像帧率，如果是音频数据则为采样率
}FRAME_INFO;
```

返回值： 成功返回 1；失败返回 0

说 明： 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 **PlayM4_Play** 之前调用，在 **PlayM4_Stop** 时自动失效，下次调用 **PlayM4_Play** 之前需要重新设置。注意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。和解码回调函数 **PlayM4_SetDecCallBack** 区别在于增加了用户传递参数

注 意：

[返回目录](#)

6.63.解码回调（带目标数据和数据大小） **PlayM4_SetDecCallBackEx**

函 数： BOOL PlayM4_SetDecCallBackEx(LONG nPort,void (CALLBACK* DecCBFun)(long nPort,char* pBuf,long nSize,FRAME_INFO* pFrameInfo, long nReserved1,long nReserved2), char* pDest, long nDestSize)

参 数： LONG nPort 播放通道号

Void (CALLBACK* DecCBFun) 解码回调函数，若不需要回调函数则置为 NULL，否则不能为 NULL

char* pDest 目标数据

long nDestSize 目标数据大小

DecCBFun 回调函数参数说明:

Long nPort 播放通道号

char * pBuf 解码后的视音频数据指针

long nSize 解码后的视音频数据长度

FRAME_INFO *pFrameInfo 图像与声音信息结构体指针

FRAME_INFO 类型说明:

```
typedef struct{
    long nWidth          画面宽，单位像素。如果是音频数据则为音频声道数；
```


		道数;
Long	nHeight	画面高。如果是音频数据为样位率
long	nStamp	时标信息, 单位毫秒
long	nType	数据类型, T_AUDIO16, T_RGB32, T_YV12
long	nFrameRate	编码时产生的图像帧率, 如果是音频数据则为采样率
}FRAME_INFO;		
long	nReserved1	保留参数
long	nReserved2	保留参数

返回值: 成功返回 TURE; 失败返回 FALSE

说 明: PlayM4_SetDecCallBackEx 与 PlayM4_SetDecCallBack 相比:
PlayM4_SetDecCallBackEx 会解码并显示, 并将解码的数据以回调函数的方式给用户, 但是 PlayM4_SetDecCallBack 仅仅将解码后的数据给用户, 并不显示。
其中参数 pDest 和 nDestSize 没有具体的用处,可以置为 NULL 和 0。
当前解码的视频数据格式为 YV12, 音频解码格式为 PCM。

[返回目录](#)

6.64.解 码 回 调 （ 带 目 标 数 据 和 数 据 大 小 和 用 户 指 针 ）

PlayM4_SetDecCallBackExMend

函 数:	BOOL PlayM4_SetDecCallBackExMend(LONG nPort, void (CALLBACK* DecCBFun)(long nPort, char* pBuf, long nSize, FRAME_INFO* pFrameInfo, long nUser, long nReserved2), char* pDest, long nDestSize, long nUser)	
参 数:	LONG	nPort 播放通道号
	Void (CALLBACK* DecCBFun)	解码回调函数, 若不需要回调函数则置为 NULL,否则不能为 NULL
	char*	pDest 目标数据
	long	nDestSize 目标数据大小
	Long	nUser 用户指针

DecCBFun 回调函数参数说明:

Long	nPort	播放通道号
char *	pBuf	解码后的视音频数据指针
long	nSize	解码后的视音频数据长度
FRAME_INFO *	pFrameInfo	图像与声音信息结构体指针

FRAME_INFO 类型说明:

typedef struct{		
long	nWidth	画面宽, 单位像素。如果是音频数据则为音频声道数;

Long

nHeight

画面高。如果是音频数据为样位率

long

nStamp

时标信息，单位毫秒

long

nType

数据类型，T_AUDIO16，T_RGB32，T_YV12

long

nFrameRate

编码时产生的图像帧率，如果是音频数据则为采样率

}FRAME_INFO;

long

nUser

用户数据

Long

nReserved2

保留参数

返回值：成功返回 1；失败返回 0

说明：PlayM4_SetDecCallBackExMend 与 PlayM4_SetDecCallBackMend 相比：PlayM4_SetDecCallBackExMend 会解码并显示，并将解码的数据以回调函数的方式给用户，但是 PlayM4_SetDecCallBackMend 仅仅将解码后的数据给用户，并不显示。

其中参数 pDest 和 nDestSize 没有具体的用处。可以置为 NULL 和 0。当前解码的视频数据格式为 YV12，音频解码格式为 PCM。

[返回目录](#)

6.65.音频解码回调 PlayM4_SetAudioCallBack*

函数：int PlayM4_SetAudioCallBack(int nPort,
void (__stdcall * funAudio)(int nPort, char * pAudioBuf, int nSize, int nStamp, int nType, int nUser), int nUser)

参数：

返回值：

说明：

注意：iOS 平台暂不支持

[返回目录](#)

6.66.文件结束消息 PlayM4_SetFileEndMsg*

函数：int PlayM4_SetFileEndMsg(int nPort,PLAYM4_HWND hWnd,unsigned int nMsg)

参数：

返回值：

说明：

注意：iOS 平台暂不支持。

[返回目录](#)

6.67.文件结束回调 PlayM4_SetFileEndCallback

函数：int PlayM4_SetFileEndCallback(int nPort,
void(CALLBACK*FileEndCallback)(int nPort, void *pUser), void *pUser)

参数：int nPort 播放通道号

```
void(CALLBACK* FileEndCallback)    文件结束回调函数
void * pUser                        用户数据
```

FileEndCallback 回调函数参数

```
int nPort        播放通道号
void * pUser      用户数据
```

返回值: 成功返回 1; 失败返回 0

说明: 设置文件播放结束回调函数。在 PlayM4_OpenFile 之前调用才有效

注意:

[返回目录](#)

6.68.分辨改变通知消息 PlayM4_SetEncChangeMsg*

函数: int PlayM4_SetEncChangeMsg(int nPort,PLAYM4_HWND hWnd,unsigned int nMsg)

参数:

返回值:

说明:

注意: iOS 平台暂不支持。

[返回目录](#)

6.69.分辨率改变通知回调 PlayM4_SetEncTypeChangeCallBack

函数: int PlayM4_SetEncTypeChangeCallBack
(int nPort,void(CALLBACK *funEncChange)(int nPort,int nUser), int nUser)

```
参数: int nPort        播放通道号
void(CALLBACK *funEncChange) 分辨率发生改变时通知用户的回调函数
int nUser                用户数据
```

funEncChange 回调函数参数

```
int nPort        播放通道号
int nUser        用户数据
```

返回值: 成功返回 1; 失败返回 0

说明: 需要在打开文件之前使用

注意:

[返回目录](#)

6.70.设置丢 B 帧个数 PlayM4_ThrowBFrameNum

函数: int PlayM4_ThrowBFrameNum(int nPort,unsigned int nNum)

参数: int nPort 播放通道号

返回值: `unsigned int nNum` 不解码 B 帧的帧数。范围 0~2
成功返回 1；失败返回 0

说明: 设置不解码的 B 帧个数；不解码 B 帧，可以减小解码时的 CPU 利用率；
如果码流中没有 B 帧，那么设置这个值将不会有作用。
如在快速播放，和支持多路而 CPU 利用率太高的情况下可以考虑使用。

注意:

[返回目录](#)

6.71. 获取丢 B 帧数回调 `PlayM4_GetThrowBFrameCallBack*`

函数: `int PlayM4_GetThrowBFrameCallBack(int nPort, void(CALLBACK* funThrowBFrame)(int nPort, unsigned int nBFrame, unsigned int nUser), unsigned int nUser)`

参数:

返回值:

说明:

注意: iOS 平台不支持此接口。

[返回目录](#)

6.72. 检查帧号是否连续 `PlayM4_CheckDiscontinuousFrameNum*`

函数: `int PlayM4_CheckDiscontinuousFrameNum(int nPort, int bCheck)`

参数:

返回值:

说明:

注意: iOS 平台不支持此接口。

[返回目录](#)

6.73. 解码密钥 `PlayM4_SetSecretKey`

函数: `int PlayM4_SetSecretKey(int nPort, int lKeyType, char *pSecretKey, int lKeyLen)`

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）
`char *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 1；失败返回 0

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 `PlayM4_OpenStream/PlayM4_OpenFile` 之前调用才有效。

注意:

[返回目录](#)

显示操作

6.74. 设置 OVERLAY 显示模式及关键色 **PlayM4_SetOverlayMode***

函 数: `int PlayM4_SetOverlayMode(int nPort, int bOverlay, COLORREF colorKey)`

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

6.75. 获取当前是否采用 OVERLAY 显示模式 **PlayM4_GetOverlayMode***

函 数: `int PlayM4_GetOverlayMode(int nPort)`

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

6.76. 获取 OVERLAY 关键色 **PlayM4_GetColorKey***

函 数: `COLORREF PlayM4_GetColorKey(int nPort)`

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

6.77. 设置图像的锐度 **PlayM4_SetImageSharpen***

函 数: `Int PlayM4_SetImageSharpen(int nPort, unsigned int nLevel)`

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

6.78. 设置图像 Overlay 翻转模式 **PlayM4_SetOverlayFlipMode***

函 数: `Int PlayM4_SetOverlayFlipMode(int nPort,int bTrue)`

参 数:

返回值:

说 明:

注 意: **iOS 平台不支持此接口。**

[返回目录](#)

6.79. 设置或增加显示区域 **PlayM4_SetDisplayRegion**

函 数: `int PlayM4_SetDisplayRegion(int nPort,unsigned int nRegionNum, HKRECT *pSrcRect, PLAYM4_HWND hDestWnd, int bEnable)`

参 数: `int nPort` 播放通道号
`unsigned int nRegionNum` 显示区域序号, 0~(MAX_DISPLAY_WND-1)。
如果 `nRegionNum` 为 0, 表示对主要显示窗口(PlayM4_Play 中设置的窗口)进行设置, 将忽略 `hDestWnd` 和 `bEnable` 的设置。
`HKRECT *pSrcRect` 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。
`PLAYM4_HWND hDestWnd` 设置显示窗口。如果该区域的窗口已经设置过 (打开过), 那么该参数被忽略。
打开 (设置) 或关闭显示区域。
`int bEnable`

返回值: 成功返回 1; 失败返回 0

说 明: 设置或增加显示区域。可以做局部放大显示。

注 意:

[返回目录](#)

6.80. 设置显示窗口 **PlayM4_SetVideoWindow**

函 数: `int PlayM4_SetVideoWindow(int nPort, unsigned int nRegionNum, PLAYM4_HWND hWnd)`

参 数: `int nPort` 播放库端口号
`unsigned int nRegionNum` 显示区域序号
`PLAYM4_HWND hWnd` 显示的窗口句柄

返回值: 成功返回 1, 失败返回 0

说 明:

注 意:

[返回目录](#)

6.81.刷新显示 **PlayM4_RefreshPlay***

函 数: int PlayM4_RefreshPlay(int nPort)

参 数: int nPort 播放库端口

返回值: 成功返回 1, 失败返回 0

说 明:

注 意:

[返回目录](#)

6.82.刷新显示（多显示区域） **PlayM4_RefreshPlayEx***

函 数: int PlayM4_RefreshPlayEx(int nPort,unsigned int nRegionNum)

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

6.83.设置显示模式 **PlayM4_SetDisplayType***

函 数: int PlayM4_SetDisplayType(int nPort,int nType)

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

6.84.获取显示模式 **PlayM4_GetDisplayType***

函 数: int PlayM4_GetDisplayType(int nPort)

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)



缓冲区操作

解码前源缓冲区

6.85. 获取源缓冲区剩余数据大小 **PlayM4_GetSourceBufferRemain**

函 数: unsigned int PlayM4_GetSourceBufferRemain(int nPort)

参 数: int nPort 播放库端口号

返回值: 返回源缓冲区生育数据大小（单位：BYTE）

说 明: 成功返回 1；失败返回 0

注 意:

[返回目录](#)

6.86. 设置源缓冲区阈值及回调指针 **PlayM4_SetSourceBufCallBack**

函 数: int PlayM4_SetSourceBufCallBack(int nPort,unsigned int nThreShold,
void (CALLBACK * SourceBufCallBack)(int nPort,unsigned int nBufSize,unsigned int
dwUser,void*pResvered),unsigned int dwUser,void *pReserved)

参 数: int nPort 播放库端口号
unsigned int nThreShold 缓冲区阈值
CALLBACK * SourceBufCallBack 缓冲区阈值回调函数
unsigned int dwUser 用户数据
void * pReserved 保留参数

SourceBufCallBack 参数说明:

int nPort 播放库端口号
unsigned int nBufSize 缓冲区阈值大小
unsigned int dwUser 用户数据
void* pResvered 保留数据

返回值: 成功返回 1，失败返回 0

说 明:

注 意:

[返回目录](#)

6.87. 重置回调标志为有效状态 **PlayM4_ResetSourceBufFlag***

函 数: int PlayM4_ResetSourceBufFlag(int nPort)

参 数:

返回值:

说 明:

注 意： iOS 平台不支持此接口。

[返回目录](#)

解码后播放缓冲区

6.88. 设置播放缓冲区最大缓冲帧数 **PlayM4_SetDisplayBuf**

函 数： int PlayM4_SetDisplayBuf(int nPort, unsigned 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 1；失败返回 0

说 明： 设置播放缓冲区（即解码后的图像缓冲区）大小；
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。
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 15（帧），在 25 帧/秒的情况下即 0.6 秒的数据。
在播放流时我们的默认值是 15(帧)，如果用户追求较小的延时，可以考虑适当减小这个值。

注 意： 建议在 PlayM4_OpenStream 之前配置，如果要改变这个配置，必须 CloseStream

[返回目录](#)

6.89. 获取播放缓冲区最大缓冲帧数 **PlayM4_GetDisplayBuf**

函 数： unsigned int PlayM4_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：

注 意：

[返回目录](#)

源、播放缓冲区

6.90. 清空所有缓冲区的剩余数据 **PlayM4_ResetSourceBuffer**

函 数： int PlayM4_ResetSourceBuffer(int nPort)

参 数： int nPort 播放通道号

返回值： 成功返回 1；失败返回 0

说 明： 清空数据缓冲（同时清除帧解析缓存）的剩余数据

注 意：

[返回目录](#)

6.91.清空指定缓冲区的剩余数据 **PlayM4_ResetBuffer**

函 数： int PlayM4_ResetBuffer(int nPort,unsigned int nBufType)

参 数： int nPort 播放通道号

unsigned int 缓冲区类型

nBufType

缓冲区宏定义

BUF_VIDEO_SRC 视频数据源缓冲区（同时清除帧解析缓存），只对流模式有效，单位 byte。

BUF_AUDIO_SRC 音频数据源缓冲区（同时清除帧解析缓存），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 视频播放缓冲区剩余数据，单位帧数。

BUF_AUDIO_RENDER 音频播放缓冲区剩余数据，单位帧数。

返回值： 成功返回 1；失败返回 0

说 明：

注 意：

[返回目录](#)

6.92.获取指定缓冲区的大小 **PlayM4_GetBufferValue**

函 数： unsigned int PlayM4_GetBufferValue(int nPort,unsigned int nBufType)

参 数： int nPort 播放通道号

unsigned int 缓冲区类型

nBufType

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。且音视频数据分开送到情况下才有效（对应 **PlayM4_OpenStreamEx**）

BUF_VIDEO_RENDER 视频播放缓冲区剩余数据，单位帧数。

BUF_AUDIO_RENDER 音频播放缓冲区剩余数据，单位帧数，音频 40ms 数据定为一帧。

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](#)

6.93.注册用户数据回调 **PlayM4_SetGetUserDataCallBack***

函 数: Int PlayM4_SetGetUserDataCallBack(int nPort,void(CALLBACK* funGetUserData)(int nPort,unsigned char *pUserBuf,unsigned int nBufLen,unsigned int nUser),unsigned int nUser)

参 数:

返回值:

说 明:

注 意: iOS 平台不支持此接口。

[返回目录](#)

索引

6.94.设置建立索引回调 **PlayM4_SetFileRefCallBack**

函 数:	int PlayM4_SetFileRefCallBack(int nPort, void (__stdcall *pFileRefDone) (unsigned int nPort, unsigned int nUser), unsigned int nUser)	
参 数:	int nPort void (__stdcall * pFileRefDone) unsigned int nUser pFileRefDone 回调函数参数说明 unsigned int nPort unsigned int nUser	播放通道号 索引回调函数 用户数据 播放器通道号 用户数据
返回值:	成功返回 1；失败返回 0	
说 明:	文件索引建立后回调。 为了能在文件播放中精确定位，在文件打开的时候建议生成文件索引。（目前测试的情况是 100M-200M 的录像文件，建议索引时间约 1s），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。	
注 意:		

[返回目录](#)

6.95.查找指定位置之前的关键帧位置 **PlayM4_GetKeyFramePos***

函 数: int PlayM4_GetKeyFramePos(int nPort,unsigned int nValue, unsigned int nType,
PFRAME_POS pFramePos)

参 数:

返回值:

说 明:

注 意: iOS 平台上该接口暂不支持。

[返回目录](#)

6.96.查找指定位置之后的关键帧位置 **PlayM4_GetNextKeyFramePos***

函 数: int PlayM4_GetNextKeyFramePos(int nPort,unsigned int nValue, unsigned int
nType, PFRAME_POS pFramePos)

参 数:

返回值:

说 明:

注 意: iOS 平台上该接口暂不支持。

[返回目录](#)

6.97. 获取文件索引 **PlayM4_GetRefValue**

函 数: `int PlayM4_GetRefValue(int nPort, unsigned char *pBuffer, unsigned int* pSize)`

参 数: `int nPort` 播放通道号
`[out] unsigned` 索引信息
`char * pBuffer` 输入 `pBuffer` 的大小, 输出索引信息大小。
`unsigned int* pSize` 注意: 可以在第一次指定 `pSize=0, pBuffer=NULL`, 从 `pSize` 的返回值获得需要的缓冲区大小。然后分配足够的缓冲, 再调用一次

返回值: 成功返回 1; 失败返回 0

说 明: 获取文件索引信息, 以便下次打开同一个文件时直接使用这个信息。必须在索引建成后才能获得信息。

注 意:

[返回目录](#)

6.98. 设置文件索引 **PlayM4_SetRefValue**

函 数: `int PlayM4_SetRefValue(int nPort, unsigned char *pBuffer, unsigned int nSize)`

参 数: `int nPort` 播放通道号
`unsigned char * pBuffer` 索引信息
`unsigned int nSize` 索引信息的长度

返回值: 成功返回 1; 失败返回 0

说 明: 设置文件索引。如果已经有了文件索引信息, 可以不再调用生成索引的回调函数 `PlayM4_SetFileRefCallBack`, 直接输入索引信息。

注 意: 索引信息及其长度必须准确, 在 `OpenFile` 之后调用

[返回目录](#)

显示

这几个函数接口，是为支持而增加的。

6.99.枚举系统中的显示设备 **PlayM4_InitDDrawDevice***

函 数： int PlayM4_InitDDrawDevice()
参 数：
返回值：
说 明：
注 意： iOS 平台该接口无效。

[返回目录](#)

6.100. 释放枚举显示设备过程分配的资源

PlayM4_ReleaseDDrawDevice*

函 数： Void PlayM4_ReleaseDDrawDevice()
参 数：
返回值：
说 明：
注 意： iOS 平台该接口无效。

[返回目录](#)

6.101. 获取显示设备（显卡）个数

PlayM4_GetDDrawDeviceTotalNums*

函 数： unsigned int PlayM4_GetDDrawDeviceTotalNums()
参 数：
返回值：
说 明：
注 意： iOS 平台该接口无效。

[返回目录](#)

6.102. 设置播放窗口使用的显示设备 **PlayM4_SetDDrawDevice***

函 数：	int PlayM4_SetDDrawDevice(int nPort,unsigned int nDeviceNum)	
参 数：		
返回值：		

说 明:	
注 意:	iOS 平台该接口无效。

[返回目录](#)

6.103. 设置播放窗口使用的显示设备（多显示区域）

PlayM4_SetDDrawDeviceEx *

函 数: int PlayM4_SetDDrawDeviceEx(int nPort,unsigned int nRegionNum,
unsigned int nDeviceNum)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.104. 获取指定显卡和监视器信息 PlayM4_GetDDrawDeviceInfo*

函 数: int PlayM4_GetDDrawDeviceInfo(unsigned int nDeviceNum,
char* lpDriverDescription,unsigned int nDespLen,char* lpDriverName ,
unsigned int nNameLen,HMONITOR *hhMonitor)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

6.105. 获取指定显示设备的系统信息 PlayM4_GetCapsEx*

函 数: int PlayM4_GetCapsEx(unsigned int nDDrawDeviceNum)

参 数:

返回值:

说 明:

注 意: iOS 平台该接口无效。

[返回目录](#)

抓图

6.106. 抓图回调 **PlayM4_SetDisplayCallBack**

函 数: int PlayM4_SetDisplayCallBack(int nPort,void (CALLBACK* DisplayCBFun)
(int nPort,char * pBuf,int nSize, int nWidth, int nHeight, int nStamp,
int nType, int nReceaved))

参 数: int nPort 播放通道号
void (CALLBACK* DisplayCBFun) 抓图回调函数，可以为 NULL

DisplayCBFun 抓图回调函数参数说明

int	nPort	播放端口号
char *	pBuf	返回图像数据
int	nSize	返回图像数据大小
int	nWidth	画面宽，单位像素
int	nHeigh	画面高
int	nStamp	时标信息，单位毫秒
int	nType	数据类型， T_YV12， T_UYVY， 详见 PlayM4_SetDecCallBack 宏定义
int	nReceaved	保留

返回值: 成功返回 1；失败返回 0

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 DisplayCBFun 设为 NULL。一旦设置回调函数，则一直有效，直到 **PlayM4_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](#)

6.107. 显示回调（带用户指针） **PlayM4_SetDisplayCallBackEx**

函 数: BOOL PlayM4_SetDisplayCallBackEx(LONG nPort,void (CALLBACK*
DisplayCBFun)(DISPLAY_INFO *pstDisplayInfo), long nUser)

参 数: LONG nPort 播放通道号
Void(CALLBACK* DisplayCBFun) 抓图回调函数，若不需要回调函数则可以为
NULL，否则不能置为 NULL
Long nUser 用户指针

DisplayCBFun 回调函数参数说明

DISPLAY_INFO *pstDisplayInfo 当前显示信息结构体指针

DISPLAY_INFO 结构说明

typedef struct

{

long nPort;	通道号
char * pBuf;	返回图像数据指针
long nBufLen;	返回图像数据大小
long nWidth;	画面宽，单位像素
long nHeight;	画面高
long nStamp;	时标信息，单位毫秒
long nType;	数据类型， T_YV12, T_RGB32, T_UYVY， 详见 PlayM4_SetDecCallBack 宏定义
long nUser;	用户数据

}DISPLAY_INFO

返回值：成功返回 1；失败返回 0

说明：当前获取到的图像数据类型为 YV12 格式[返回目录](#)6.108. 图像数据转为 BMP 格式 **PlayM4_ConvertToBmpFile**

函数：int PlayM4_ConvertToBmpFile (char * pBuf,int nSize, int nWidth, int nHeight, int nType,char *sFileName)

参数：	char * pBuf	抓图回调函数中图像缓冲区
	int nSize	抓图回调函数中图像的大小
	int nWidth	抓图回调函数中图像宽度
	int nHeight	抓图回调函数中图像高度
	int nType	抓图回调函数中图像类型 (T_YV12)
	char * sFileName	要保存的文件名。最好以 BMP 作为文件扩展名

返回值：成功返回 1；失败返回 0

说明：将抓图得到的图像数据保存成 BMP 文件。转换函数占用的 cpu 资源，如果不需要保存图片，则不要调用。**注意：**2CIF 分辨率的图像抓图后的分辨率为 4CIF。[返回目录](#)6.109. 图像数据转为 JPEG 格式 **PlayM4_ConvertToJpegFile**

函数：int PlayM4_ConvertToJpegFile(char *pBuf, int nSize, int nWidth, int nHeight, int nType, char *sFileName)

参数：	char * pBuf	图像数据缓存
	int nSize	图像大小
	int nWidth	图像宽度
	int nHeight	图像高度
	int nType	图像类型 (T_YV12)

char * sFileName 保存路径

返回值: 成功返回 1; 失败返回 0

说明: 抓图存为 JPEG 文件, 该函数可在显示回调函数中使用,
用法参见 PlayM4_ConvertToBmpFile

注意: 2CIF 分辨率的图像抓图后的分辨率为 4CIF。

[返回目录](#)

6.110. 直接抓取 BMP 图像 PlayM4_GetBMP

函数: int PlayM4_GetBMP(int nPort,unsigned char* pBitmap,unsigned int nBufSize,
unsigned int* pBmpSize)

参数: int nPort 播放通道号
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 1; 失败返回 0

说明:

注意: 2CIF 分辨率的图像抓图后的分辨率为 4CIF。

[返回目录](#)

6.111. 直接抓取 JPEG 图像 PlayM4_GetJPEG

函数: int PlayM4_GetJPEG(int nPort,unsigned char* pJpeg,unsigned int nBufSize,
unsigned int* pJpegSize)

参数: int nPort 播放通道号
unsigned char* pJpeg 存放 JPEG 图像数据地址, 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。
unsigned int nBufSize 申请的缓冲区大小
unsigned int * pJpegSize 获取到的实际 JPEG 图像数据大小

返回值: 成功返回 1; 失败返回 0

说明:

注意: 2CIF 分辨率的图像抓图后的分辨率为 4CIF。

[返回目录](#)



6.112. 设置 JPEG 图像质量 **PlayM4_SetJpegQuality**

函 数: `int PlayM4_SetJpegQuality(int nQuality)`

参 数: `int nQuality` 图像质量参数, 范围 0~100
0: 图像质量最差, 但抓取的图像大小最小
100: 图像质量最好, 但抓取的图像大小最大

返回值: 成功返回 1; 失败返回 0

说 明: 设置抓取的 jpeg 图像质量, 设置范围 0~100, 建议使用 75~90, 若不调用该函数, 则采用默认图像质量, 目前默认为 80。需要在抓图进行前调用, 建议在 `openfile` 之前使用

注 意:

[返回目录](#)

其他功能

6.113. 画图回调 `PlayM4_RegisterDrawFun`

函 数: `int PlayM4_RegisterDrawFun(int nPort,`
`void (CALLBACK* DrawFun)(int nPort,PLAYM4_HDC hDc,int nUser), int nUser)`

参 数: `int nPort` 播放通道号
`void (CALLBACK* DrawFun)` 画图回调函数
`int nUser` 用户数据

`DrawFun` 回调函数参数说明

`int nPort` 播放通道号
`PLAYM4_HDC hDc` 表面设备上下文
`int nUser` 用户数据，就是上面输入的用户数据；

返回值: 成功返回 1；失败返回 0

说 明: 注册一个回调函数，获得当前表面的 `device context`,可以在这个 DC 上画图（或写字），类似于在窗口的客户区 DC 上绘图。

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](#)

6.114. 画图回调 `PlayM4_RigisterDrawFun`

函 数: `int PlayM4_RigisterDrawFun(int nPort,`
`void (CALLBACK* DrawFun)(int nPort,PLAYM4_HDC hDc,int nUser), int nUser)`

参 数: `int nPort` 播放通道号
`void (CALLBACK* DrawFun)` 画图回调函数
`int nUser` 用户数据

`DrawFun` 回调函数参数说明

`int nPort` 播放通道号
`PLAYM4_HDC hDc` 表面设备上下文
`int nUser` 用户数据，就是上面输入的用户数据；

返回值: 成功返回 1；失败返回 0

说 明: 注册一个回调函数，获得当前表面的 `device context`,可以在这个 DC 上画图（或写字），类似于在窗口的客户区 DC 上绘图。

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: `PlayM4_RegisterDrawFun` 拼写错误

[返回目录](#)



6.115. 设置数据校验 **PlayM4_SetVerifyCallBack***

函 数: `int PlayM4_SetVerifyCallBack(int nPort, unsigned int nBeginTime, unsigned int nEndTime, void (__stdcall* funVerify)(int nPort, FRAME_POS * pFilePos, unsigned int blsVideo, unsigned int nUser), unsigned int nUser)`

参 数:

返回值:

说 明:

注 意: **iOS 平台该接口暂不支持。**

[返回目录](#)

6.116. 获取原始帧数据回调函数 **PlayM4_GetOriginalFrameCallBack***

函 数: `int PlayM4_GetOriginalFrameCallBack(int nPort, int blsChange, int bNormalSpeed, int nStartFrameNum, int nStartStamp, int nFileHeader, void(CALLBACK *funGetOriginalFrame)(int nPort, FRAME_TYPE *frameType, int nUser), int nUser)`

参 数:

返回值:

说 明:

注 意: **不再支持**，相关功能可以使用其他库来实现

[返回目录](#)

6.117. 获取文件最后的时标和帧号 **PlayM4_GetFileSpecialAttr ***

函 数: `int PlayM4_GetFileSpecialAttr(int nPort, unsigned int *pTimeStamp, unsigned int *pFileNum, unsigned int *nFileHeader)`

参 数:

返回值:

说 明:

注 意: **不再支持**，相关功能可以使用其他库来实现

[返回目录](#)

6.118. 跳过错误数据 **PlayM4_SkipErrorData**

函 数: `int PlayM4_PlaySkipErrorData(int nPort, int bSkip)`

参 数: `int nport` 播放通道号
`int bSkip` 1 码流错误时会直接跳下一个关键帧解码;
0 寻找下一帧数据解码

返回值: 成功返回 1; 失败返回 0

说 明: 如果在码流有错误的情况下, 不允许出现花屏, 建议可以设置成 1

注 意：

[返回目录](#)

科技呵护未来

First Choice for Security Professionals