



播放库

SDK 编程指南

(for Windows 8/Win 7/XP/2000/2003/Vista

32bit)

V6.5.x.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

1 产品简介.....	6
2 SDK 版本更新.....	7
版本号说明.....	7
Version 6.5.0.0.....	7
3 错误代码及说明.....	16
3.1 播放库错误码说明.....	16
3.2 鱼眼部分错误码说明.....	17
4 有关显示的说明.....	18
5 函数调用顺序.....	19
6 函数说明.....	21
系统操作及错误号获取.....	21
6.1. 获取播放库 SDK 版本号和 build 号 PlayM4_GetSdkVersion.....	21
6.2. 获取错误号 PlayM4_GetLastError.....	21
6.3. 测试播放所需系统功能 PlayM4_GetCaps.....	21
6.4. 初始化 Direct 表面 PlayM4_InitDDraw.....	22
6.5. 释放 Direct 表面 PlayM4_RealeseDDraw.....	22
6.6. 设置播放使用的定时器类型 PlayM4_SetTimerType.....	22
6.7. 获取播放使用的定时器类型 PlayM4_GetTimerType.....	23
6.8. 获取未使用的通道号 PlayM4_GetPort.....	23
6.9. 释放已使用的通道号 PlayM4_FreePort.....	23
文件操作.....	24
6.10. 打开文件 PlayM4_OpenFile.....	24
6.11. 关闭文件 PlayM4_CloseFile.....	24
流操作.....	25
6.12. 设置流播放模式 PlayM4_SetStreamOpenMode.....	25
6.13. 获取流播放模式 PlayM4_GetStreamOpenMode.....	25
6.14. 打开流 PlayM4_OpenStream.....	25
6.15. 关闭流 PlayM4_CloseStream.....	26
6.16. 输入流数据 PlayM4_InputData.....	26
6.17. 打开流（以音视频分开输入方式）PlayM4_OpenStreamEx.....	26
6.18. 关闭流（以音视频分开输入方式）PlayM4_CloseStreamEx.....	27
6.19. 打开流（以协议的方式）PlayM4_OpenStreamAdvanced.....	27
6.20. 输入视频流 PlayM4_InputVideoData.....	27
6.21. 输入音频流 PlayM4_InputAudioData.....	28
播放控制.....	29
6.22. 开启播放 PlayM4_Play.....	29
6.23. 关闭播放 PlayM4_Stop.....	29
6.24. 暂停/恢复播放 PlayM4_Pause.....	29
6.25. 快速播放 PlayM4_Fast.....	29
6.26. 慢速播放 PlayM4_Slow.....	30
6.27. 单帧前进 PlayM4_OneByOne.....	30

6.28. 单帧回退 PlayM4_OneByOneBack.....	30
6.29. 以独占方式打开声音 PlayM4_PlaySound.....	30
6.30. 关闭声音（独占方式） PlayM4_StopSound.....	31
6.31. 以共享方式打开声音 PlayM4_PlaySoundShare.....	31
6.32. 关闭声音（共享方式） PlayM4_StopSoundShare.....	31
6.33. 设置音量 PlayM4_SetVolume.....	31
6.34. 获取音量 PlayM4_GetVolume.....	32
6.35. 调整 WAVE 波形 PlayM4_AdjustWaveAudio.....	32
6.36. 设置图像质量 PlayM4_SetPicQuality.....	32
6.37. 获取图像质量 PlayM4_GetPictureQuality.....	32
6.38. 设置视频参数 PlayM4_SetColor.....	33
6.39. 获取视频参数 PlayM4_GetColor.....	33
6.40. 设置图像锐化 PlayM4_SetImageSharpen.....	33
6.41. 设置图像 Overlay 翻转模式 PlayM4_SetOverlayFlipMode.....	34
6.42. 设置图像旋转模式 PlayM4_SetRotateAngle.....	34
6.43. 设置文件当前播放位置（百分比） PlayM4_SetPlayPos.....	34
6.44. 获取文件当前播放位置（百分比） PlayM4_GetPlayPos.....	34
6.45. 设置文件当前播放时间（毫秒） PlayM4_SetPlayedTimeEx.....	35
6.46. 获取文件当前播放时间（毫秒） PlayM4_GetPlayedTimeEx.....	35
6.47. 设置文件当前播放帧号 PlayM4_SetCurrentFrameNum.....	35
6.48. 获取文件当前播放帧号 PlayM4_GetCurrentFrameNum.....	35
6.49. 去闪烁 PlayM4_SetDeflash.....	36
获取播放或解码信息.....	37
6.50. 获取文件总时间 PlayM4_GetFileTime.....	37
6.51. 获取文件总帧数 PlayM4_GetFileTotalFrames.....	37
6.52. 获取当前帧率 PlayM4_GetCurrentFrameRate.....	37
6.53. 获取已播放时间 PlayM4_GetPlayedTime.....	37
6.54. 获取已解码的视频帧数 PlayM4_GetPlayedFrames.....	37
6.55. 获取原始图像大小 PlayM4_GetPictureSize.....	38
6.56. 获取文件头长度 PlayM4_GetFileHeadLength.....	38
6.57. 获取当前显示帧的全局时间 PlayM4_GetSpecialData.....	39
解码操作及控制.....	40
6.58. 设置解码回调流类型 PlayM4_SetDecCBStream.....	40
6.59. 设置视频帧解码类型 PlayM4_SetDecodeFrameType.....	40
6.60. 解码回调 PlayM4_SetDecCallBack.....	40
6.61. 解码回调（增加用户传递参数） PlayM4_SetDecCallBackMend.....	41
6.62. 解码回调（带目标数据和数据大小） PlayM4_SetDecCallBackEx.....	42
6.63. 解码回调（带目标数据和数据大小和用户指针） PlayM4_SetDecCallBackExMend43	
6.64. 音频解码回调 PlayM4_SetAudioCallBack *.....	44
6.65. 文件结束消息 PlayM4_SetFileEndMsg.....	45
6.66. 文件结束回调 PlayM4_SetFileEndCallback.....	45
6.67. 分辨改变通知消息 PlayM4_SetEncChangeMsg.....	46
6.68. 分辨率改变通知回调 PlayM4_SetEncTypeChangeCallBack.....	46
6.69. 设置丢 B 帧个数 PlayM4_ThrowBFrameNum.....	46

6.70. 检查帧号是否连续 PlayM4_CheckDiscontinuousFrameNum.....	47
6.71. 设置水印回调函数 PlayM4_SetCheckWatermarkCallBack.....	47
6.72. 解码密钥 PlayM4_SetSecretKey.....	48
显示操作.....	49
6.73. 设置 OVERLAY 显示模式及关键色 PlayM4_SetOverlayMode.....	49
6.74. 获取当前是否采用 OVERLAY 显示模式 PlayM4_GetOverlayMode.....	49
6.75. 获取 OVERLAY 关键色 PlayM4_GetColorKey.....	50
6.76. 设置或增加显示区域 PlayM4_SetDisplayRegion.....	50
6.77. 刷新显示 PlayM4_RefreshPlay.....	50
6.78. 刷新显示（多显示区域） PlayM4_RefreshPlayEx.....	50
6.79. 设置显示模式 PlayM4_SetDisplayType.....	51
6.80. 获取显示模式 PlayM4_GetDisplayType.....	51
缓冲区操作.....	52
解码前源缓冲区.....	52
6.81. 获取源缓冲区剩余数据大小 PlayM4_GetSourceBufferRemain.....	52
6.82. 设置源缓冲区阈值及回调指针 PlayM4_SetSourceBufCallBack.....	52
6.83. 重置回调标志为有效状态 PlayM4_ResetSourceBufFlag.....	52
解码后播放缓冲区.....	53
6.84. 设置播放缓冲区最大缓冲帧数 PlayM4_SetDisplayBuf.....	53
6.85. 获取播放缓冲区最大缓冲帧数 PlayM4_GetDisplayBuf.....	53
源、播放缓冲区.....	53
6.86. 清空所有缓冲区的剩余数据 PlayM4_ResetSourceBuffer.....	53
6.87. 清空指定缓冲区的剩余数据 PlayM4_ResetBuffer.....	54
6.88. 获取指定缓冲区的大小 PlayM4_GetBufferValue.....	54
索引.....	55
6.89. 设置建立索引回调 PlayM4_SetFileRefCallBack.....	55
6.90. 查找指定位置之前的关键帧位置 PlayM4_GetKeyFramePos.....	55
6.91. 查找指定位置之后的关键帧位置 PlayM4_GetNextKeyFramePos.....	56
6.92. 获取文件索引 PlayM4_GetRefValue.....	56
6.93. 设置文件索引 PlayM4_SetRefValue.....	57
多屏显示.....	58
6.94. 枚举系统中的显示设备 PlayM4_InitDDrawDevice.....	58
6.95. 释放枚举显示设备过程分配的资源 PlayM4_ReleaseDDrawDevice.....	58
6.96. 获取显示设备（显卡）个数 PlayM4_GetDDrawDeviceTotalNums.....	58
6.97. 设置播放窗口使用的显示设备 PlayM4_SetDDrawDevice.....	59
6.98. 设置播放窗口使用的显示设备（多显示区域） PlayM4_SetDDrawDeviceEx *.....	59
6.99. 获取指定显卡和监视器信息 PlayM4_GetDDrawDeviceInfo.....	59
6.100. 获取指定显示设备的系统信息 PlayM4_GetCapsEx.....	60
抓图.....	61
6.101. 显示回调 PlayM4_SetDisplayCallBack.....	61
6.102. 显示回调（带用户指针） PlayM4_SetDisplayCallBackEx.....	61
6.103. 图像数据转为 BMP 格式 PLayerM4_ConvertToBmpFile.....	62
6.104. 图像数据转为 JPEG 格式 PlayM4_ConvertToJpegFile.....	62
6.105. 直接抓取 BMP 图像 PlayM4_GetBMP.....	63

6.106. 直接抓取 JPEG 图像 PlayM4_GetJPEG.....	63
6.107. 设置 JPEG 图像质量 PlayM4_SetJpegQuality.....	64
6.108. 抓取选择区域图片为 JPEG 格式 PlayM4_GetCropJPEG.....	64
6.109. 抓取选择区域图片为 BMP 格式 PlayM4_GetCropBMP.....	64
其他功能.....	66
6.110. 画图回调 PlayM4_RegisterDrawFun.....	66
6.111. 画图回调 PlayM4_RigisterDrawFun.....	66
6.112. 设置数据校验 PlayM4_SetVerifyCallBack *.....	67
6.113. 获取原始帧数据回调函数 PlayM4_GetOriginalFrameCallBack *.....	67
6.114. 获取文件最后的时标和帧号 PlayM4_GetFileSpecialAttr *.....	68
6.115. 码流错误直接跳至下一关键帧 PlayM4_SkipErrorData.....	69
倒放功能.....	70
6.116. 开始倒放 PlayM4_ReversePlay.....	70
6.117. 获取当前播放帧的全局时间 PlayM4_GetSystemTime.....	70
6.118. 设置流播放起始帧（毫秒） PlayM4_SetPlayedTimeEx.....	71
同步回放功能.....	72
6.119. 同步回放接口 PlayM4_SetSycGroup.....	72
6.120. 设置同步回放开始时间 PlayM4_SetSycStartTime*.....	72
鱼眼功能.....	73
6.121. 启用鱼眼接口 PlayM4_FEC_Enable.....	73
6.122. 关闭鱼眼模块接口 PlayM4_FEC_Disable.....	73
6.123. 获取鱼眼矫正处理子端口 PlayM4_FEC_GetPort.....	73
6.124. 删除鱼眼矫正处理子端口 PlayM4_FEC_DelPort.....	74
6.125. 设置鱼眼矫正参数 PlayM4_FEC_SetParam.....	74
6.126. 获取鱼眼矫正参数 PlayM4_FEC_GetParam.....	75
6.127. 设置显示窗口，可以随时切换 PlayM4_FEC_SetWnd.....	76
6.128. 设置鱼眼窗口的绘图回调 PlayM4_FEC_SetCallBack.....	77
倍帧功能.....	78
6.129. 倍帧 PlayM4_MotionFlow.....	78
图像增强功能.....	79
6.130. 设置关闭或开启模块 PlayM4_VIE_SetModuConfig.....	79
6.131. 设置图像增强区域 PlayM4_VIE_SetRegion.....	79
6.132. 获取开启图像增强模块 PlayM4_VIE_GetModuConfig.....	80
6.133. 设置增强参数 PlayM4_VIE_SetParaConfig.....	80
6.134. 获取开启模块的参数 PlayM4_VIE_GetParaConfig.....	81
7 注意事项.....	84

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 两种格式的播放截图。

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 V6.3.0.0 起，规定版本号定义如下：

V 主版本号 . 子版本号 . 修正版本号 . 预留版本号

- 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
- 预留版本号升级：内部使用

Version 6.5.0.0

Version 6.5.0.0 (2013-04-12)

- 添加倍帧功能接口
- 添加图像增强功能接口

Version 6.4.0.0 (2012-11-05)

- 添加鱼眼功能接口

Version 6.3.2.7 (2011-12-09)

- 添加流模式同步回放功能
- 添加设备抽场码流的支持
- 优化倒放快放速度
- 更新底层解码库，修正部分新码流解码绿屏的情况

Version 6.3.0.5 (2011-12-05)

- 新添加支持倒放的接口 PlayM4_ReversePlay、PlayM4_GetSystemTime 和 PlayM4_SetPlayedTimeEx
- 添加播放库头文件中有，而以前版本的播放库编程指南中没有的接口说明
- 修正显示方面的问题

Version 6.2.2.3

- 修正 RTP 快放，有音频问题
- 修正部分有水印设备，注册解码回调崩溃
- 支持标准 264 多参考帧解码
- 改善实时流延时问题

Version 6.2.1.1

- 新增支持 DS-65xx 系列、DS-66xx 系列 DVS 码流以及 RTP 封装码流的智能设备码流
- 支持 DS-91xx-ST 系列设备码流（标准 264 两场）
- 修正显示方面的几个问题

Version 6.1.1.21

- 修正 win7，service 下部分显卡关闭应用程序出现异常的情况
- 内部兼容多种水印加密
- 修正暂停情况下不能单帧操作
- 修正 hik 文件丢 B 帧引起快放的问题

Version 6.1.1.17

- 修正配置丢 B 帧后的抓图问题
- 增加对超长文件路径的支持
- 修正单帧前进/后退交错操作可能出现的播放帧序错误问题
- 减小预览延时
- 增加对纯音频流/文件的支持，与旧版本保持一致
- 修正音频解码回调的时标
- 增加对新版本水印的支持

注意事项

- 新版本是线程安全的，客户二次开发时不用关心多线程安全，但需要注意的，旧版本的一些调用方式，如：回调函数体内直接使用接口，会造成死锁。

Version 6.1.1.12

- 增加 [SkipErrorData](#) 和 [CheckDiscontinuousFrameNum](#) 的接口实现
- 补充接口 [AdjustWaveAudio](#) 的实现
- 再次修正 [GetSrcRemainData](#) 的返回值不正确的问题
- 解决某些屏保激活后导致的无显示问题
- 增加了消息和回调不能同时注册的保护
- 最大支持显示设备到 32 个，满足客户多屏上墙的需要
- 修正 2CIF 码流抓取 jpeg 图像导致的内存泄漏问题

Version 6.1.1.8 (2010-1-25)

- 修正 [GetNextKeyFramePos](#) 有时失败的问题
- 修正某些 8100 系列 DVR、4200 系列板卡产品播放画面出现块效应的问题

- 修正定位操作后视频解码回调出的数据不正确的问题
- 暂不支持纯音频流的播放

Version 6.1.1.7

- 修正某些 9000 系列混合 DVR 产品录像文件获取结束全局时间不正确的问题
- 增加对 AMR 音频的支持
- 增加对 PCM-L16 音频的支持

Version 6.1.1.6

- 增加对 65 系列 DVS 设备的支持
- 提高 4CIF 码流的解码效率
- 修正音视频解码回调的实现，与 4.9 版本保持一致

Version 6.1.1.5

- 修正了 GetSourceBufferRemain 返回值不正确的问题
- 修正了单帧前进有时失败的问题

Version 6.1.1.3

- 增加对 MJPEG 编码的支持
- 增加对 G726 音频的支持
- 修正锁屏后可能导致的无显示问题

Version 6.1.1.0

- 支持多屏显示的自适应。
- [PlayM4_GetFileSpecialAttr](#)、[PlayM4_GetOriginalFrameCallBack](#) 不再支持，相关功能由其他库实现
- 增加对 8100 系列 DVR、4200 系列板卡产品的支持
- 增加对 9000 系列混合 DVR 产品的密钥支持

Version 6.1.0.3

- 增加对 mpeg 音频的支持
- 修正播放变分辨率文件时导致的丢帧问题

Version 6.1.0.2

- 增加对部分 8100 系列 DVR 产品的支持

Version 6.0.0.1

- 增加对 9000 系列产品的支持
- 最多支持 500 路解码，取消 100 路限制

Version 5.00.00

- 将解码模块以动态库的形式独立出来，在解码过程中根据实际需要动态加载

Version 4.09.04 (2008-09-25)

- 增加获取和释放播放通道函数 [PlayM4_GetPort](#)、[PlayM4_FreePort](#)
- 增加对标准 mpeg4 码流的支持
- 增加密钥设置接口函数 [PlayM4_SetSecretKey](#)
- 增加设置文件播放结束回调函数 [PlayM4_SetFileEndCallback](#)

Version 4.08.00 (2006-08-13)

- 增加抓图函数 [PlayM4_GetBMP](#) 和 [PlayM4_GetJpeg](#)。用于修正在暂停状态下无法抓图的 bug；这两个函数在播放时可以随时调用
- 增加是否跳帧的判断函数 [PlayM4_CheckDiscontinuousFrameNum](#)。在帧号不连续时是否跳下一个 I 帧播放，默认情况下是跳下一个 I 帧播放
- 增加解码回调函数 [PlayM4_SetDecCallBackMend](#) 和以前解码回调函数区别在于增加了用户传递参数

Version 4.7

(build0711)

- 增加用于抓图存为 Jpeg 的函数 [PlayM4_ConvertToJpegFile](#) 和设置 Jpeg 图像质量的函数 [PlayM4_SetJpegQuality](#)
- 增加了一种错误宏类型 PLAYM4_JPEG_COMPRESS_ERROR，主要用于表明抓图存为 Jpeg 格式时，进行 Jpeg 压缩出错的返回
- 增加了设置是否去闪烁接口函数 [PlayM4_SetDeflash](#)

Version 4.6

(build1013)

- 修正 NTSC 制式下抓图失败
- 修正硬盘空间不足时抓图失败

(build0728)

- 优化了播放器 SDK 解码性能，降低播放器正常播放和快速播放文件时的 CPU 利用率
- 配合 HC 卡 4.0 版本发布，解码库版本升级，同时修正了旧版本中存在的一两个小 bug

Version 4.5

(build0407)

- 修正了上一版本调用函数 [PlayM4_SetPicQuality](#) 时，在某些显卡上出现花屏的 bug
- (build0303)
- 增加用于文件拼接的两个函数 [PlayM4_GetOriginalFrameCallBack](#) 和 [PlayM4_GetFileSpecialAttr](#)
 - 配合 HC 卡 3.2 版本发布，解码库升级，新版本向下兼容，可以解以前的码流，而老的解码库不能解 HC 卡 3.2 及以后版本的录象文件

Version 4.3

(build0901)

- 增加解码时图象格式发生改变通知用户的回调函数，用户可以改变界面大小
- 增加获取/调节视频参数接口，在回放时可以调节亮度、对比度、饱和度、色调等参数，以获得更好的播放效果
- 播放器的 DEMO 中增加了将解码后的 YUV 数据直接转成 AVI 格式文件的功能，注意目前只能对视频数据进行处理，转换后的 AVI 文件占用磁盘空间很大，一秒钟的数据量有 3.6M 左右，要转换的文件时间长度不要超过 500 秒，回放转换后的 AVI 文件需要安装 DivX5.2 版本的插件

(build0626)

- 修正了通过网络客户端显示图象时可能出现马赛克的问题
- 修正了上一版本用文件长度方式检索时可能出现的 bug

Version 4.2

(build0616)

- 支持图像格式的动态改变，如从 4CIF 到 CIF，播放器自动识别，不用重新启动
- 修改 [PlayM4_SetVerifyCallBack](#) 函数，可以检测文件丢帧或丢数据

Version 4.0

(build0420)

- 支持 4CIF 格式的解码，改正了在用户回调解码退出时可能出现的错误

Version 3.6

(build 1230)

- 支持图像格式的动态改变，如从 CIF 到 QCIF，播放器自动识别，不用重新启动
- 增加另外一个调整音量的接口 [PlayM4_AdjustWaveAudio](#)
- 增加水印校验接口 [PlayM4_SetVerifyCallBack](#)
- 增加解码后的音频数据回调，用户可用来分析音频数据 [PlayM4_SetAudioCallBack](#)

Version 3.4

(build 0626)

- 最多支持 100 路解码器，取消 16 路限制

- 增加 Set/Get 使用的定时器 [PlayM4_SetTimerType](#)、[PlayM4_GetTimerType](#)
- 增加 Reset/Get 播放器使用的缓冲区 [PlayM4_ResetBuffer](#)、[PlayM4_GetBufferValue](#)

Version 3.2

(build 0430)

- 增加音视频流分开解码
 - [PlayM4_OpenStreamEx](#)
 - [PlayM4_CloseStreamEx](#)
 - [PlayM4_InputVideoData](#)
 - [PlayM4_InputAudioData](#)
- 增加在使用 off-screen 表面时, 获得设备上下文, 可以在显示窗口打标记。Overlay 表面不需要这样做, 可直接在窗口上打标记 [PlayM4_RegisterDrawFun](#)
- 增加获取和设置索引信息 [PlayM4_GetRefValue](#)、[PlayM4_SetRefValue](#)

Version 3.0

(build 0325)

- 支持 DS-400XH 系列板卡生成的码流
- 可设置最多 4 个显示区域, 支持局部显示 (可实现局部放大)
 - [PlayM4_SetDisplayRegion](#)
 - [PlayM4_RefreshPlayEx](#)
 - [PlayM4_SetDDrawDeviceEx](#)
- 可以设置解码回调的流类型 [PlayM4_SetDecCBStream](#)
- 改正原来解码回调只回调 I 帧的 bug
- 改正 QCIF 并且 NTSC 模式下, 文件不能会放

Version 2.5

(build 1118)

- 修改 [PlayM4_OpenStream](#) 有时失败, 返回错误值 `PLAYM4_SYS_NOT_SUPPORT`

(build 1115)

- 增加设置显示模式 [PlayM4_SetDisplayType](#)、[PlayM4_GetDisplayType](#)
- 修改了在使用中发现的一些 bug

Version 2.4

(build 0911)

- 增加更多获取信息的接口
 - [PlayM4_GetStreamOpenMode](#)

[PlayM4_GetOverlayMode](#)

[PlayM4_GetColorKey](#)

[PlayM4_GetVolume](#)

[PlayM4_GetPictureQuality](#)

- 流模式下控制源缓冲区接口

[PlayM4_GetSourceBufferRemain](#)

[PlayM4_ResetSourceBuffer](#)

[PlayM4_SetSourceBufCallBack](#)

[PlayM4_ResetSourceBufFlag](#)

- 控制播放缓冲接口

[PlayM4_SetDisplayBuf](#)

[PlayM4_GetDisplayBuf](#)

- 文件准确定位及单帧回退接口，需要一定的时间生成索引

[PlayM4_SetFileRefCallBack](#)

[PlayM4_OneByOneBack](#)

[PlayM4_SetCurrentFrameNum](#)

[playM4_GetKeyFramePos](#)

[PlayM4_GetNextKeyFramePos](#)

- 可以抽帧播放，减小 CPU 利用率 [PlayM4_ThrowBFrameNum](#)

- 多显卡支持

[PlayM4_InitDDrawDevice](#)

[PlayM4_ReleaseDDrawDevice](#)

[PlayM4_GetDDrawDeviceTotalNums](#)

[PlayM4_SetDDrawDevice](#)

[PlayM4_GetDDrawDeviceInfo](#)

[PlayM4_GetCapsEx](#)

更新：

- 播放器效率大幅提高，在 Pentium4 1.5GHZ 下可同时播放 9 路复杂图像视频。简单图像的情况下，可以支持更多路。在多路播放情况下，需要显卡支持硬件缩放
- 在文件回放到文件尾时，不再强制退出，而由用户决定
- 获得的当前时间和帧号不再反映解码时状态，而是反映播放时状态。更符合用户习惯
- 调整快放和慢放范围，可以连续调用 4 次，即播放速度可以设置 16 倍速（实际速度视系统性能而定）
- 支持只有 MMX 指令集的 CPU(如 Pentium II)
- 可定位到文件最后一帧

修改：

- 在显卡不支持硬件缩放抓图时，图片数据错误

Version 2.2

(build 0703)

增加：

- 获得错误码接口 [PlayM4_GetLastError](#)
- 刷新显示接口 [PlayM4_RefreshPlay](#)
- 获得图像大小接口 [PlayM4_GetPictureSize](#)

- 设置 OVERLAY 模式显示接口 [PlayM4_SetOverlayMode](#)
- 设置图像质量接口 [PlayM4_SetPicQuality](#)
- 共享方式打开声音接口 [PlayM4_PlaySoundShare](#)
- 共享方式关闭声音接口 [PlayM4_StopSoundShare](#)

更新:

- 在暂停时可以单帧播放
- 在 STREAM_FILE 模式下增加了暂停, 快放, 慢放, 单帧播放操作
- 当显卡不支持 BLT 缩放时, 采用软件方法, 但占用 CPU 资源很高

修改:

- 在复合流抽帧情况下, 播放速度正常 (前一版本播放速度太快)

Version 2.0

(build 0607)

- 提高了图像质量, 消除了色斑

(build 0605)

- 抓图。
- 用户可以控制解码后音视频数据
- 设置流播放模式 (实时和文件模式)
- 获得当前播放时间 (单位毫秒)
- 设置当前播放位置到指定时间
- 获得文件中的总帧数和当前解码的帧号
- 获得当前帧率
- 获得当前版本号
- 支持 QCIF 格式播放
- 流播放的实时性和流畅性提高, 播放器性能提高

Version1.11

- 修改在非 VC++ 开发时死机问题

Version 1.1

- 播放器支持多路播放 (根据主机资源定, 以后我们效率会继续提高, 现在在 Pentium4 下支持 4 路) 最大支持 16 路。增加了流接口和一些其他功能

注意: CPU 必须是 Intel Pentium 3 以上。初始化和释放 DirectDraw 表面的接口不再需要, 每路可以相互独立。

Version 1.0

- 目前此播放器 sdk 只支持一路播放, 输入参数 nPort 必须是 0, 以后会增加支持多路的播放。目前此播放器的声音部分可能有不流畅的情况, 以后我们会改进。此播放器使用了显卡的某些功能, 所以如果不能显示图像, 请尝试以下操作: 1) 把屏幕颜色值设为 32 位。2) 更换显卡 (建议 ATI 系列,

因为 ATI 系列显卡对图形处理比较好)

3 错误代码及说明

[转到下一章节](#)

3.1 播放库错误码说明

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	多媒体时钟设置失败
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_CREATE_DDRAW_ERROR	9	创建 DirectDraw 失败
PLAYM4_CREATE_OFFSCREEN_ERROR	10	创建后端缓存失败
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	系统不支持，解码器只能工作在 Pentium 3 以上
PLAYM4_FILEHEADER_UNKNOWN	17	没有文件头
PLAYM4_VERSION_INCORRECT	18	解码器和编码器版本不对应
PALYM4_INIT_DECODER_ERROR	19	初始化解码器失败
PLAYM4_CHECK_FILE_ERROR	20	文件太短或码流无法识别
PLAYM4_INIT_TIMER_ERROR	21	初始化多媒体时钟失败
PLAYM4_BLT_ERROR	22	位拷贝失败
PLAYM4_UPDATE_ERROR	23	显示 overlay 失败
PLAYM4_OPEN_FILE_ERROR_MULTI	24	打开文件失败，流类型是复合流
PLAYM4_OPEN_FILE_ERROR_VIDEO	25	打开文件失败，流类型是视频流
PLAYM4_JPEG_COMPRESS_ERROR	26	JPEG 压缩失败
PLAYM4_EXTRACT_NOT_SUPPORT	27	文件不支持
PLAYM4_EXTRACT_DATA_ERROR	28	数据错误
PLAYM4_SECRET_KEY_ERROR	29	解码密钥错误

PLAYM4_DECODE_KEYFRAME_ERROR	30	解码关键帧失败
PLAYM4_NEED_MORE_DATA	31	数据不足
PLAYM4_INVALID_PORT	32	无效端口号
PLAYM4_NOT_FIND	33	查找失败
PLAYM4_NEED_LARGER_BUFFER	34	需要更大的缓冲区
PLAYM4_FAIL_UNKNOWN	99	未知的错误

3.2 鱼眼部分错误码说明

错误名称	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	非法子端口
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

[返回目录](#)

4 有关显示的说明

[转到下一章](#)

播放器的显示部分，主要采用了 DirectDraw 技术。目前使用了两种方式显示画面：

1、创建使用 off_screen 画面并 Blt（位转换货位传输，可做缩放、旋转、镜像等效果）到主画面；

2、创建使用 OVERLAY 画面。

这两种方式有以下特点：

第一种方式使用 Off_screen 画面，优点：多路播放可以相对对立，相互间不受影响，缺点：受显卡的影响比较大，如果显卡不支持缩放等操作，当用户需要缩放时（显示窗口和图像原始尺寸不同时），我们会使用软件方式进行缩放，如果放大 CPU 利用率会非常高，我们提供了一个接口 PlayM4_GetCaps，用户可以用它测试你的显卡是否支持 Blt 的缩放等操作；

第二种方式使用 OVERLAY 画面，优点：目前的绝大部分显卡都支持 OVERLAY 画面，OVERLAY 画面支持硬件的缩放，当第一种方式得不到显卡支持时，使用 OVERLAY 画面可以得到显卡支持，缺点：具有独占性，在一块显卡上同时只能有一个 OVERLAY 画面处于活动状态，因此同时只能有一路播放器使用 OVERLAY 画面，而且如果其他程序正在使用 OVERLAY 画面，那么播放器就不能再使用 OVERLAY 画面了，同样，如果播放器使用了 OVERLAY 画面，其他程序也不能再使用 OVERLAY 画面。

使用 2003 等服务器版操作系统的用户请注意：

Windows2003 等服务器版操作系统，由于并不是为了媒体娱乐而设置，而主要强调的是稳定，所以对于一些视频处理功能（硬件加速、directX）在默认配置下都会有所削弱或者不支持，需要手动开启。

而我们在视频显示这一块的处理，大部分还是需要依靠显卡的 BLT 功能（硬件缩放，所谓硬件缩放，简单的说是这样，例如图像的原始尺寸是 704*576，但是我们在显示器上看到的画面并不是此规格大小，可能小于这个尺寸，也可能大于这个尺寸，而这个缩放主要依靠显卡的缩放算法来进行，这个我们就叫做硬件缩放）。如果显卡或者显卡驱动不支持 BLT，则图像的缩放会切换为软件缩放（这个时候，图像的放大缩小会由 CPU 来进行算法处理），那么相对的，cpu 的利用率就会有所升高。

请按照以下步骤操作：

a、右键->属性->设置->高级->疑难解答->硬件加速，将“硬件加速”启动至全部；

b、开始->运行，输入 dxdiag，进入 DirectX 诊断，在“显示”选项内检查“DirectDraw”和“Direct3D”是否启用。

[返回目录](#)

5 函数调用顺序

[转到下一章节](#)



鱼眼相关接口调用方法：

PlayM4_OpenStream（数据流模式）/PlayM4_OpenFile（文件模式）
PlayM4_FEC_Enable（加载鱼眼模块并且初始化）
PlayM4_FEC_GetPort（获取鱼眼矫正的端口号）
PlayM4_FEC_SetParam（设置鱼眼相关必要的更新参数）
PlayM4_FEC_SetWnd（设置鱼眼显示窗口）
PlayM4_FEC_DelPort（删除鱼眼子端口号）
PlayM4_FEC_Disable（释放鱼眼模块）

安装方式和矫正方式定好之后，不能更改，除非将该端口号删除，进行重新的配置，在 PlayM4_FEC_GetPort 之后 PlayM4_FEC_DelPort 之前可以任意时刻的调用 PlayM4_FEC_SetParam，进行鱼眼矫正参数的更新，调用 PlayM4_FEC_SetWnd 可以将当前矫正图像呈现到任意指定的窗口上，可以随时切换。

图像增强接口调用方法：

图像增强接口调用需要先设置模块开启 PlayM4_VIE_SetModuConfig，再设置模块参数 PlayM4_VIE_SetParaConfig

[返回目录](#)

6 函数说明

[转到下一章节](#)

系统操作及错误号获取

6.1. 获取播放库 SDK 版本号和 build 号 **PlayM4_GetSdkVersion**

函 数:	DWORD PlayM4_GetSdkVersion()	
参 数:	无	
返回值:	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。	
说 明:	如果只是修改 bug，我们只升级后面两位版本号。	

[返回目录](#)

6.2. 获取错误号 **PlayM4_GetLastError**

函 数:	DWORD PlayM4_GetLastError(LONG nPort)	
参 数:	LONG 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。	

[返回目录](#)

6.3. 测试播放所需系统功能 **PlayM4_GetCaps**

函 数:	Int PlayM4_GetCaps()	
参 数:	无	
返回值:	0~8 位分别表示以下信息（位与是 TRUE 表示支持）:	
SUPPORT_DDRAW	支持 DIRECTDRAW；如果不支持，则播放器不能工作	
SUPPORT_BLT	显卡支持 BLT 操作；如果不支持，则播放器不能工作	
SUPPORT_BLTFOURCC	显卡 BLT 支持颜色转换；如果不支持，播放器会使用软件方式作 RGB 转换	
SUPPORT_BLTSHRINKX	显卡 BLT 支持 X 轴缩小；如果不支持，系统使用软件方式转换	
SUPPORT_BLTSHRINKY	显卡 BLT 支持 Y 轴缩小；如果不支持，系统使用软件方式转换	

SUPPORT_BLTSTRETCHX	显卡 BLT 支持 X 轴放大；如果不支持，系统使用软件方式转换
SUPPORT_BLTSTRETCHY	显卡 BLT 支持 Y 轴放大；如果不支持，系统使用软件方式转换
SUPPORT_SSE	CPU 支持 SSE 指令, Intel Pentium3 以上支持 SSE 指令
SUPPORT_MMX	CPU 支持 MMX 指令集
说 明:	测试播放器需要的一些系统功能
注 意:	如果显卡支持上面列出的显示功能，将会大大降低 CPU 利用率。如果显卡不支持放大缩小，建议将显示窗口的大小设置和解码图像的大小一致，例如解码后的图像是 352*288（PAL）大小，则建议显示窗口也设置为 352*288。

[返回目录](#)

6.4. 初始化 Direct 表面 **PlayM4_InitDDraw**

函 数:	BOOL PlayM4_InitDDraw(HWND hWnd)
参 数:	HWND hWnd 应用程序主窗口的句柄
返回值:	成功返回 TRUE；失败返回 FALSE
说 明:	在使用 vb, delphi 开发时请注意，它们生成的对话框具有 WS_CLIPCHILDREN 窗口风格，必须去掉这种风格，否则显示画面会被对话框上的控件覆盖。
注 意:	1.1 版本以上 SDK 不需要调用

[返回目录](#)

6.5. 释放 Direct 表面 **PlayM4_RealeseDDraw**

函 数:	BOOL PlayM4_RealeseDDraw()
参 数:	无
返回值:	成功返回 TRUE；失败返回 FALSE
注 意:	1.1 版本以上 SDK 不需要调用

[返回目录](#)

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

函 数:	BOOL PlayM4_SetTimerType(LONG nPort, DWORD nTimerType, DWORD nReserved)
参 数:	<div>LONG nPort 播放通道号</div> <div>DWORD nTimerType 时钟类型</div> <div>#define TIMER_1 1 1 TIMER_1 是多媒体时钟，一个进程中只能使用 16 个，时间精度高；</div> <div>#define TIMER_2 2 2 TIMER_2 是等待计时器，使用数目没有限制，但时间精度较低，不推荐在高倍速播放时使用。</div> <div>默认情况下 0~15 路使用 TIMER_1，其余使用 TIMER_2；</div> <div>DWORD nReserved 保留参数</div>
返回值:	成功返回 TRUE；失败返回 FALSE
注 意:	必须在 PlayM4_OpenFile 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。
--	--------------------------------------

[返回目录](#)

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

函 数:	BOOL PlayM4_GetTimerType(LONG nPort,DWORD *pTimerType, DWORD *pReserved)	
参 数:	LONG nPort	播放通道号
	[out]DWORD * pTimerType	TIMER_1 或 TIMER_2
	DWORD * pReserved	保留参数
返回值:	成功返回 TURE；失败返回 FALSE	

[返回目录](#)

6.8. 获取未使用的通道号 **PlayM4_GetPort**

函 数:	BOOL PlayM4_GetPort(LONG* nPort)	
参 数:	[out]LONG* nPort	播放通道号，指向用于获取端口号的 LONG 型变量指针
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	获取未使用的通道号,通道号的上限为 500。	

[返回目录](#)

6.9. 释放已使用的通道号 **PlayM4_FreePort**

函 数:	BOOL PlayM4_FreePort(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE；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。	

[返回目录](#)

文件操作

6.10.打开文件 **PlayM4_OpenFile**

函 数:	BOOL PlayM4_OpenFile(LONG nPort,LPSTR sFileName)	
参 数:	LONG nPort	播放通道号
	LPSTR sFileName	文件名
返回值:	成功返回 TURE; 失败返回 FALSE	
注 意:	文件不能超过 4GB 或小于 4KB。	

[返回目录](#)

6.11.关闭文件 **PlayM4_CloseFile**

函 数:	BOOL PlayM4_CloseFile(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

流操作

6.12. 设置流播放模式 `PlayM4_SetStreamOpenMode`

函 数:	BOOL PlayM4_SetStreamOpenMode(LONG nPort,DWORD nMode)	
参 数:	LONG nPort	播放通道号
	DWORD nMode	流播放模式: STREAME_REALTIME、STREAME_FILE
	<code>#define STREAME_REALTIME 0</code>	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；
	<code>#define STREAME_FILE 1</code>	此模式下则不然。（按帧率播放）
返回值:	成功返回 TURE；失败返回 FALSE	
注 意:	必须在播放之前设置。2.2 以上版本 SDK 以后可以做暂停，快放，慢放，单帧播放操作 PlayM4_SetStreamOpenMode 在 PlayM4_OpenStream 之前调用才有效。	

[返回目录](#)

6.13. 获取流播放模式 `PlayM4_GetStreamOpenMode`

函 数:	LONG PlayM4_GetStreamOpenMode(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	STREAME_REALTIME 或 STREAME_FILE	

[返回目录](#)

6.14. 打开流 `PlayM4_OpenStream`

函 数:	BOOL PlayM4_OpenStream(LONG nPort,PBYTE pFileHeadBuf,DWORD nSize,DWORD nBufPoolSize)	
参 数:	LONG nPort	播放通道号
	PBYTE pFileHeadBuf	文件头数据
	DWORD nSize	文件头长度
	DWORD 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。
	<code>#define SOURCE_BUF_MIN 1024*50</code>	
	<code>#define SOURCE_BUF_MAX 1024*100000</code>	

返回值:	成功返回 TURE; 失败返回 FALSE
------	-----------------------

[返回目录](#)

6.15.关闭流 PlayM4_CloseStream

函 数:	BOOL PlayM4_CloseStream(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

6.16.输入流数据 PlayM4_InputData

函 数:	BOOL PlayM4_InputData(LONG nPort,PBYTE pBuf,DWORD nSize)	
参 数:	LONG nPort	播放通道号
	PBYTE pBuf	流数据缓冲区地址
	DWORD nSize	流数据缓冲区大小
返回值:	TURE 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 一般是由于内部缓冲区满。 我们建议： 1.在实时流情况下，因数据满而无法送入播放库，可以选择丢弃，后果是画面可能跳帧或者花屏，如果无法接受，则 sleep 几毫秒后，再尝试多送几次 2.文件流回放模式下，数据满而导致无法送入播放库，需要不断尝试，直到送成功为止。	

[返回目录](#)

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

函 数:	BOOL PlayM4_OpenStreamEx(LONG nPort,PBYTE pFileHeadBuf,DWORD nSize,DWORD nBufPoolSize)	
参 数:	LONG nPort	播放通道号
	PBYTE pFileHeadBuf	文件头数据
	DWORD nSize	文件头长度
	DWORD 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。
	<pre>#define SOURCE_BUF_MIN 1024*50 #define SOURCE_BUF_MAX 1024*100000</pre>	
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	以音视频分开输入的方式打开流。	

[返回目录](#)

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

函 数:	BOOL PlayM4_CloseStreamEx(LONG nPort);	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

6.19.打开流（以协议的方式）PlayM4_OpenStreamAdvanced

函 数:	BOOL PlayM4_OpenStreamAdvanced(LONG nPort, int nProtocolType, PLAYM4_SESSION_INFO* pstSessionInfo, DWORD nBufPoolSize);	
参 数:	LONG nPort int nProtocolType PLAYM4_SESSION_INFO* pstSessionInfo DWORD nBufPoolSize	播放通道号 协议类型 协议信息结构体 协议数据大小
	#define PLAYM4_PROTOCOL_RTSP 1 #define PLAYM4_SESSION_INFO_SDP 1	RTSP 协议 nSessionInfoTypeSDP 信息
	typedef struct _PLAYM4_SESSION_INFO_ { int nSessionInfoType; int nSessionInfoLen; unsigned char* pSessionInfoData; } PLAYM4_SESSION_INFO;	交互信息结构 交互信息类型，比如 SDP，比如海康私有信息头 交互信息长度 交互信息数据指针
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	该接口是使用非海康 40 个字节开流，现支持用 SDP 开流	

[返回目录](#)

6.20.输入视频流 PlayM4_InputVideoData

函 数:	BOOL PlayM4_InputVideoData(LONG nPort,PBYTE pBuf,DWORD nSize)	
参 数:	LONG nPort PBYTE pBuf DWORD nSize	播放通道号 缓冲区地址 缓冲区大小
返回值:	TURE,表示已经输入数据。FALSE 表示失败，数据没有输入	
说 明:	输入视频流 (可以是复合流，但音频数据会被忽略); 打开流之后才能输入数据。返回 False 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，	

	<p>确保播放库不丢失数据。</p> <p>我们建议：</p> <p>1.在实时流情况下，因数据满而无法送入播放库，可以选择丢弃，后果是画面可能跳帧或者花屏，如果无法接受，则 sleep 几毫秒后，再尝试多送几次</p> <p>2.文件流回放模式下，数据满而导致无法送入播放库，需要不断尝试，直到送成功为止。</p>
--	---

[返回目录](#)

6.21.输入音频流 **PlayM4_InputAudioData**

函 数：	BOOL PlayM4_InputAudioData(LONG nPort,PBYTE pBuf,DWORD nSize)		
参 数：	LONG nPort	播放通道号	
	PBYTE pBuf	缓冲区地址	
	DWORD nSize	缓冲区大小	
返回值：	TURE,表示已经输入数据。FALSE 表示失败，数据没有输入		
说 明：	输入音频流；打开声音之后才能输入数据。返回 False 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。 我们建议： 1.在实时流情况下，因数据满而无法送入播放库，可以选择丢弃，后果是声音可能跳帧或者声音不连续，如果无法接受，则 sleep 几毫秒后，再尝试多送几次 2.文件流回放模式下，数据满而导致无法送入播放库，需要不断尝试，直到送成功为止。		

[返回目录](#)

播放控制

6.22.开启播放 **PlayM4_Play**

函 数:	BOOL PlayM4_Play(LONG nPort, HWND hWnd)	
参 数:	LONG nPort	播放通道号
	HWND hWnd	播放视频的窗口句柄
返回值:	成功返回 TURE; 失败返回 FALSE	
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, 只是改变当前播放速度为正常速度。	

[返回目录](#)

6.23.关闭播放 **PlayM4_Stop**

函 数:	BOOL PlayM4_Stop(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

6.24.暂停/恢复播放 **PlayM4_Pause**

函 数:	BOOL PlayM4_Pause(LONG nPort,DWORD nPause)	
参 数:	LONG nPort	播放通道号
	DWORD nPause	TURE 暂停, FALSE 恢复
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

6.25.快速播放 **PlayM4_Fast**

函 数:	BOOL PlayM4_Fast(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	每次调用将使当前播放速度加快一倍, 最多调用 4 次; 要恢复正常播放调用 PlayM4_Play(), 从当前位置开始正常播放; 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。	

[返回目录](#)


6.26.慢速播放 **PlayM4_Slow**

函 数:	BOOL PlayM4_Slow(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	每次调用将使当前播放速度慢一倍; 最多调用 4 次; 要恢复正常播放调用 PlayM4_Play(), 从当前位置开始正常播放。	

[返回目录](#)

6.27.单帧前进 **PlayM4_OneByOne**

函 数:	BOOL PlayM4_OneByOne(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	恢复正常播放需要调用 PlayM4_Play()。	

[返回目录](#)

6.28.单帧回退 **PlayM4_OneByOneBack**

函 数:	BOOL PlayM4_OneByOneBack(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	每调用一次倒退一帧。	
注 意:	只支持文件播放, 必须在文件索引生成之后才能调用	

[返回目录](#)

6.29.以独占方式打开声音 **PlayM4_PlaySound**

函 数:	BOOL PlayM4_PlaySound(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	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:	默认情况下声音是关闭的; PlayM4_PlaySound, PlayM4_StopSound 需要配对调用; 在播放声音过程中, 建议 PlayM4_PlaySound, PlayM4_PlaySoundShare 不要混用。	

[返回目录](#)



6.30.关闭声音（独占方式）PlayM4_StopSound

函 数:	BOOL PlayM4_StopSound()	
参 数:	无	
返回值:	成功返回 TURE; 失败返回 FALSE	
注 意:	PlayM4_PlaySound, PlayM4_StopSound 需要配对调用。	

[返回目录](#)

6.31.以共享方式打开声音 PlayM4_PlaySoundShare

函 数:	BOOL PlayM4_PlaySoundShare(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	以共享方式播放声音, 只管播放本路声音而不去关闭其他路的声音。	
注 意:	WIN98 及其之前版本操作系统不支持创建多个声音设备。如果声卡已经被使用, 那么此函数将返回 FALSE; PlayM4_PlaySoundShare, PlayM4_StopSoundShare 需要配对调用; 在播放声音过程中, 建议 PlayM4_PlaySound, PlayM4_PlaySoundShare 不要混用。	

[返回目录](#)

6.32.关闭声音（共享方式）PlayM4_StopSoundShare

函 数:	BOOL PlayM4_StopSoundShare(LONG nPort);	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
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:	在同一个进程中, 所有通道必须使用相同的方式播放或关闭声音; PlayM4_PlaySoundShare, PlayM4_StopSoundShare 需要配对调用。	

[返回目录](#)

6.33.设置音量 PlayM4_SetVolume

函 数:	BOOL PlayM4_SetVolume(LONG nPort,WORD nVolume)	
参 数:	LONG nPort	播放通道号
	WORD nVolume	音量值, 范围 0-0XFFFF
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	可以在播放之前设置, 返回值是 TRUE, 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。	

[返回目录](#)

6.34. 获取音量 **PlayM4_GetVolume**

函 数:	WORD PlayM4_GetVolume(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	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。	

[返回目录](#)

6.35. 调整 WAVE 波形 **PlayM4_AdjustWaveAudio**

函 数:	BOOL PlayM4_AdjustWaveAudio(LONG nPort, LONG nCoefficient)	
参 数:	LONG nPort	播放通道号
	LONG nCoefficient	调整参数, 0 是不调整, 范围从 MIN_WAVE_COEF 到 MAX_WAVE_COEF
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	调整 WAVE 波形, 可以改变声音的大小。 和 PlayM4_SetVolume 的不同在于, 它是调整声音数据, 只对该路起作用, 而 PlayM4_SetVolume 是调整声卡音量, 对整个系统起作用。	
注 意:	用这个函数会破坏音质, 除非想每路单独调整音量, 否则请谨慎使用。	

[返回目录](#)

6.36. 设置图像质量 **PlayM4_SetPicQuality**

函 数:	BOOL PlayM4_SetPicQuality(LONG nPort, BOOL bHighQuality)	
参 数:	LONG nPort	播放通道号
	BOOL bHighQuality	等于 1 时图像高质量, 等于 0 时低质量, 默认为 1。
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	当设置成高质量时画面效果好, 但 CPU 利用率高。 在支持多路播放时, 可以设为低质量, 以降低 CPU 利用率; 当某路放大播放时将该路设置成高质量, 以达到好的画面效果。	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetPictureQuality(LONG nPort, BOOL *bHighQuality)	
参 数:	LONG nPort	播放通道号
	[out] BOOL * bHighQuality	1 表示高质量, 0 表示低质量
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)



6.38.设置视频参数 **PlayM4_SetColor**

函 数:	BOOL PlayM4_SetColor(LONG nPort, DWORD nRegionNum, int nBrightness, int nContrast, int nSaturation, int nHue)	
参 数:	LONG nPort	播放通道号
	DWORD nRegionNum	显示区域, 参考 PlayM4_SetDisplayRegion; 如果只有一个显示区域(通常情况)设为 0
	int nBrightness	亮度 , 默认 64; 范围 0-128
	int nContrast	对比度, 默认 64; 范围 0-128
	int nSaturation	饱和度, 默认 64; 范围 0-128
	int nHue	色调 , 默认 64; 范围 0-128
返回值:	成功返回 TURE; 失败返回 FALSE	
注 意:	如果全部为默认值将不进行颜色调节	

[返回目录](#)

6.39.获取视频参数 **PlayM4_GetColor**

函 数:	BOOL PlayM4_GetColor(LONG nPort, DWORD nRegionNum, int *pBrightness, int *pContrast,int *pSaturation, int *pHue)	
参 数:	LONG nPort	播放通道号
	DWORD nRegionNum	显示区域
	[out]int * pBrightness	亮度 , 默认 64; 范围 0-128
	[out]int * pContrast	对比度, 默认 64; 范围 0-128
	[out]int * pSaturation	饱和度, 默认 64; 范围 0-128
	[out]int * pHue	色调 , 默认 64; 范围 0-128
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

6.40.设置图像锐化 **PlayM4_SetImageSharpen**

函 数:	BOOL PlayM4_SetImageSharpen(LONG nPort,DWORD nLevel)	
参 数:	LONG nPort	播放通道号
	DWORD nLevel	锐化级别 (1~6, 0 不做锐化, 1 低级, 6 高级) 默认不做锐化
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:		

[返回目录](#)

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

函 数:	BOOL PlayM4_SetOverlayFlipMode(LONG nPort, BOOL bTrue)	
参 数:	LONG nPort BOOL bTrue	播放通道号 是否设置翻转, 若为 TRUE 则为翻转; 为 FALSE 则不翻转
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	早期的版本指 overlay 表面缓冲区紧跟着一个或多个建立好的后缓冲区, 提高显示效率, 6.1.0.3 以上版本后没有功用。	

[返回目录](#)

6.42. 设置图像旋转模式 **PlayM4_SetRotateAngle**

函 数:	BOOL PlayM4_SetRotateAngle(LONG nPort, DWORD nRegionNum, DWORD dwType)	
参 数:	LONG nPort DWORD nRegionNum DWORD dwType	播放通道号 显示区域号 见宏定义
	#define R_ANGLE_0 -1 #define R_ANGLE_L90 0 #define R_ANGLE_R90 1 #define R_ANGLE_180 2	不旋转 向左旋转90度 向右旋转90度 旋转180度
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:		

[返回目录](#)

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

函 数:	BOOL PlayM4_SetPlayPos(LONG nPort, float fRelativePos)	
参 数:	LONG nPort float fRelativePos	播放通道号 播放位置, 范围 0-100%
返回值:	成功返回 TURE; 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, 为精确定位, 否则即为粗略定位 在 PlayM4_Play 之后调用才有效	

[返回目录](#)

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

函 数:	float PlayM4_GetPlayPos(LONG nPort)
------	-------------------------------------

参 数:	LONG 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 在 PlayM4_Play 之后调用才有效	

[返回目录](#)

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

函 数:	BOOL PlayM4_SetPlayedTimeEx(LONG nPort,DWORD nTime)	
参 数:	LONG nPort	播放通道号
	DWORD nTime	设置文件播放到指定时间，单位毫秒
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	根据时间设置文件播放位置	
注 意:	如果在建立文件索引的前提下使用，为精确定位，否则即为粗略定位 在 PlayM4_Play 之后调用才有效	

[返回目录](#)

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

函 数:	DWORD PlayM4_GetPlayedTimeEx(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	文件当前播放的时间，单位毫秒 有在 PlayM4_Play 之后调用才有效	

[返回目录](#)

6.47. 设置文件当前播放帧号 PlayM4_SetCurrentFrameNum

函 数:	BOOL PlayM4_SetCurrentFrameNum(LONG nPort,DWORD nFrameNum)	
参 数:	LONG nPort	播放通道号
	DWORD nFrameNum	帧号
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	设置当前播放播放位置到指定帧号；根据帧号来定位播放位置。	
注 意:	如果在建立文件索引的前提下使用，为精确定位，否则即为粗略定位 在 PlayM4_Play 之后调用才有效。	

[返回目录](#)

6.48. 获取文件当前播放帧号 PlayM4_GetCurrentFrameNum

函 数:	DWORD PlayM4_GetCurrentFrameNum(LONG nPort)	
参 数:	LONG 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。 有在 PlayM4_Play 之后调用才有效。

[返回目录](#)

6.49.去闪烁 **PlayM4_SetDeflash**

函 数:	BOOL PlayM4_SetDeflash(LONG nPort,BOOL bDeflash)	
参 数:	LONG nPort	播放通道号
	BOOL bDeflash	TRUE 表示设置去闪烁，FALSE 表示不设置，默认为不设置
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	设置是否去闪烁功能，原先在静止图像区域有噪声情况下，图像会产生闪烁现象（或称刷新或跳动），启动去闪烁功能后，闪烁效果可消除或减轻，同时也能降低噪声 此接口只在标准 264 下有效	

[返回目录](#)

获取播放或解码信息

6.50. 获取文件总时间 **PlayM4_GetFileTime**

函 数:	DWORD	PlayM4_GetFileTime(LONG nPort)
参 数:	LONG nPort	播放通道号
返回值:	文件总时间长度, 单位秒	
注 意:	不支持对正在写入的文件进行时间的读取。	

[返回目录](#)

6.51. 获取文件总帧数 **PlayM4_GetFileTotalFrames**

函 数:	DWORD	PlayM4_GetFileTotalFrames(LONG nPort)
参 数:	LONG nPort	播放通道号
返回值:	文件总帧数	
注 意:	不支持对正在写入的文件进行帧数的读取。	

[返回目录](#)

6.52. 获取当前帧率 **PlayM4_GetCurrentFrameRate**

函 数:	DWORD	PlayM4_GetCurrentFrameRate(LONG nPort)
参 数:	LONG nPort	播放通道号
返回值:	当前码流中当前编码帧率	
注 意:	如果码流中的帧率小于 1, 则返回 0	

[返回目录](#)

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

函 数:	DWORD	PlayM4_GetPlayedTime(LONG nPort)
参 数:	LONG nPort	播放通道号
返回值:	文件当前播放时间, 单位秒	

[返回目录](#)

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

函 数:	DWORD	PlayM4_GetPlayedFrames(LONG nPort)
参 数:	LONG nPort	播放通道号
返回值:	已经解码的视频帧数	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetPictureSize(LONG nPort,LONG *pWidth,LONG *pHeight)		
参 数:	LONG nPort,	播放通道号	
	[out]LONG * pWidth,	原始图像的宽度	
	[out]LONG * pHeight	原始图像的高度	
返回值:	成功返回 TURE； 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。		
注 意:	该函数获取的是刚刚播放那帧数据的长宽，因此只有当播放开始后才能获取准确的值。建议配合 PlayM4_SetEncTypeChangeCallBack/PlayM4_SetEncChangeMsg 使用		

[返回目录](#)

6.56.获取文件头长度 **PlayM4_GetFileHeadLength**

函 数:	DWORD 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:	<p>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):</p> <pre>CFile m_TestFile; void Start() { //获得文件头长度; DWORD nLength = PlayM4_GetFileHeadLength(); PBYTE pFileHead = new BYTE[nLength]; //打开文件; m_TestFile.Open("test.mp4 ", CFile::modeRead,NULL); m_TestFile.Read(pFileHead,nLength); //设置模式 PlayM4_SetStreamOpenMode(0,STREAME_FILE); //打开流接口 If (!PlayM4_OpenStream(0,pFileHead, nLength,1024*1000)) { m_strPlayFileName=""; MessageBox("文件打不开"); } }</pre>		

	<pre> } //播放 m_bPlaying = PlayM4_Play(0, m_hWnd); delete []pFileHead; } /// void InputData() { BYTE pBuf[4096]; m_TestFile.Read(pBuf,sizeof(pBuf)); while(!PlayM4_InputData(0,pBuf,sizeof(pBuf))) { if(!m_bPlaying) break;//如果已经停止播放，则退出； TRACE("SLEEEP\n"); Sleep(5); } }</pre>

[返回目录](#)

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

函 数:	DWORD 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](#)

解码操作及控制

6.58.设置解码回调流类型 **PlayM4_SetDecCBStream**

函 数:	BOOL PlayM4_SetDecCBStream(LONG nPort,DWORD nStream)	
参 数:	LONG nPort	播放通道号
	DWORD nStream	1 视频流, 2 音频流, 3 复合流
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	建议在解码回调函数前使用	

[返回目录](#)

6.59.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 TURE; 失败返回 FALSE	
说 明:		

[返回目录](#)

6.60.解码回调 **PlayM4_SetDecCallBack**

函 数:	BOOL PlayM4_SetDecCallBack(LONG nPort,void (CALLBACK* DecCBFun)(long nPort,char * pBuf,long nSize,FRAME_INFO * pFrameInfo, long nReserved1,long nReserved2))	
参 数:	LONG nPort	播放通道号
	void (CALLBACK* DecCBFun)	解码回调函数指针, 若不需要调用回调函数则置为 NULL, 否则不能为 NULL
	DecCBFun 回调函数参数说明	
	long nPort	播放器通道号
	char * pBuf	解码后的音视频数据指针
	long nSize	解码后的音视频数据 pBuf 的长度
	FRAME_INFO *pFrameInfo	图像和声音信息结构体指针
	long nReserved1	保留参数
	long nReserved2	保留参数

	<p>FRAME_INFO 结构说明</p> <pre>typedef struct{ long nWidth; long nHeight; long nStamp; long nType long nFrameRate; DWORD dwFrameNum; }FRAME_INFO;</pre> <p>宏定义说明</p> <p>T_AUDIO16</p> <p>T_RGB32</p> <p>T_UYVY</p> <p>T_YV12</p>	<p>画面宽，单位像素。如果是音频数据则为音频声道数；</p> <p>画面高。如果是音频数据则为样位率；</p> <p>时标信息，单位毫秒</p> <p>数据类型，T_AUDIO16, T_RGB32, T_YV12</p> <p>编码时产生的图像帧率，如果是音频数据则为采样率</p> <p>帧号</p> <p>音频数据;采样率 16khz, 单声道, 每个采样点 16 位表示。</p> <p>视频数据。每个像素 4 个字节, 排列方式与位图相似, “B-G-R-0 …”, 第一个像素位于图像左下角。</p> <p>视频数据, uyvy 格式。</p> <p>“U0-Y0-V0-Y1-U2-Y2-V2-Y3…”, 第一个像素位于图像左上角。</p> <p>视频数据, yv12 格式。排列顺序“Y0-Y1-……”, “V0-V1…”, “U0-U1-……”。</p> <p>关于图像格式详细信息, 用户可以查阅相关资料。</p>
返回值:	成功返回 TURE; 失败返回 FALSE	
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:	<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, 有关知识请参阅 directx 开发包。</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](#)

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

函 数:	BOOL PlayM4_SetDecCallBackMend(LONG nPort,void (CALLBACK* DecCBFun)(long nPort,char* pBuf,long nSize,FRAME_INFO* pFrameInfo, long nUser, long nReserved2), long nUser)	
参 数:	LONG nPort void (CALLBACK* DecCBFun)	播放通道号 解码回调函数, 若不需要调用解码回调则置

	<div><div>longnUser</div><div>DecCBFun 回调函数参数</div><div><div>longnPort</div><div>char *pBuf</div><div>longnSize</div><div>FRAME_INFO *pFrameInfo</div><div>longnUser</div><div>longnReserved2</div></div><div>FRAME_INFO 类型说明</div><div>typedef struct{<div><div>longnWidth</div><div>longnHeight</div><div>longnStamp</div><div>longnType</div><div>longnFrameRate</div></div></div><div>}FRAME_INFO;</div></div>	<div>为 NULL，否则不能为 NULL</div> <div>用户数据</div> <div></div> <div>播放通道号</div> <div>解码后的视音频数据指针</div> <div>解码后的视音频数据长度</div> <div>图像与声音信息结构体指针</div> <div>用户数据</div> <div>保留参数</div> <div></div> <div>画面宽，单位像素。如果是音频数据则为音频声道数；</div> <div>画面高。如果是音频数据为样位率</div> <div>时标信息，单位毫秒</div> <div>数据类型，T_AUDIO16，T_RGB32， T_YV12</div> <div>编码时产生的图像帧率，如果是音频数据则为采样率</div>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	<div>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，有关知识请参阅 directx 开发包。</div> <div>解码回调函数 PlayM4_SetDecCallBack 区别在于增加了用户传递参数</div> <div>当前解码的视频数据格式为 YV12，音频解码格式为 PCM。</div> <div>调用此回调函数后，则视频图像无显示。</div>	

[返回目录](#)

6.62.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:	<div><div>LONGnPort</div><div>Void (CALLBACK* DecCBFun)</div><div><div>char*pDest</div><div>longnDestSize</div></div></div>	<div>播放通道号</div> <div>解码回调函数，若不需要回调函数则置为 NULL，否则不能为 NULL</div> <div>目标数据（暂时不用）</div> <div>目标数据大小（暂时不用）</div>

	<p>DecCBFun 回调函数参数说明:</p> <p>Long nPort</p> <p>char * pBuf</p> <p>long nSize</p> <p>FRAME_INFO *pFrameInfo</p> <p>FRAME_INFO 类型说明:</p> <p>typedef struct{</p> <p>long nWidth</p> <p>long nHeight</p> <p>long nStamp</p> <p>long nType</p> <p>long nFrameRate</p> <p>}FRAME_INFO;</p> <p>long nReserved1</p> <p>long nReserved2</p>	<p>播放通道号</p> <p>解码后的视音频数据指针</p> <p>解码后的视音频数据长度</p> <p>图像与声音信息结构体指针</p> <p>画面宽，单位像素。如果是音频数据则为音频声道数；</p> <p>画面高。如果是音频数据为样位率</p> <p>时标信息，单位毫秒</p> <p>数据类型，T_AUDIO16，T_RGB32， T_YV12</p> <p>编码时产生的图像帧率，如果是音频数据则为采样率</p> <p>保留参数</p> <p>保留参数</p>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	<p>PlayM4_SetDecCallBackEx 与 PlayM4_SetDecCallBack 相比:</p> <p>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。</p> <p>其中参数 pDest 和 nDestSize 没有具体的用处,可以置为 NULL 和 0.</p> <p>当前解码的视频数据格式为 YV12，音频解码格式为 PCM。</p>	

[返回目录](#)

6.63.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:	<p>LONG nPort</p> <p>Void (CALLBACK* DecCBFun)</p> <p>char* pDest</p> <p>long nDestSize</p> <p>Long nUser</p> <p>DecCBFun 回调函数参数说明:</p>	<p>播放通道号</p> <p>解码回调函数，若不需要回调函数则置为 NULL,否则不能为 NULL</p> <p>目标数据</p> <p>目标数据大小</p> <p>用户指针</p>

	<div>LongnPort</div> <div>char *pBuf</div> <div>longnSize</div> <div>FRAME_INFO *pFrameInfo</div> <div>FRAME_INFO 类型说明: typedef struct{ longnWidth LongnHeight longnStamp longnType longnFrameRate }FRAME_INFO; longnUser LongnReserved2</div>	<div>播放通道号</div> <div>解码后的视音频数据指针</div> <div>解码后的视音频数据长度</div> <div>图像与声音信息结构体指针</div> <div>画面宽，单位像素。如果是音频数据则为音频声道数；</div> <div>画面高。如果是音频数据为样位率</div> <div>时标信息，单位毫秒</div> <div>数据类型，T_AUDIO16，T_RGB32，T_YV12</div> <div>编码时产生的图像帧率，如果是音频数据则为采样率</div> <div>用户数据</div> <div>保留参数</div>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	<div>PlayM4_SetDecCallBackExMend 与 PlayM4_SetDecCallBackMend 相比:</div> <div>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。</div> <div>其中参数 pDest 和 nDestSize 没有具体的用处。可以置为 NULL 和 0.</div> <div>当前解码的视频数据格式为 YV12，音频解码格式为 PCM。</div>	

[返回目录](#)

6.64.音频解码回调 **PlayM4_SetAudioCallBack ***

函 数:	BOOL PlayM4_SetAudioCallBack(LONG nPort, void (__stdcall * funAudio)(long nPort, char * pAudioBuf, long nSize, long nStamp, long nType, long nUser), long nUser)	
参 数:	<div>LONGnPort</div> <div>void (__stdcall * funAudio)</div> <div>longnUser</div> <div>funAudio 回调函数参数</div> <div>longnPort</div> <div>char *pAudioBuf</div> <div>longnSize</div> <div>longnStamp</div> <div>longnType</div> <div>longnUser</div>	<div>播放通道号</div> <div>音频解码回调函数，若不需要回调函数则置为 NULL, 否则不能为 NULL</div> <div>用户数据</div> <div>播放通道号</div> <div>wave 格式音频数据指针</div> <div>音频数据长度</div> <div>时标(ms)</div> <div>音频类型 T_AUDIO16, 采样率 16khz，单声道，每个采样点 16 位</div> <div>用户数据</div>

返回值:	成功返回 TURE; 失败返回 FALSE
注 意:	暂不支持

[返回目录](#)

6.65.文件结束消息 PlayM4_SetFileEndMsg

函 数:	BOOL PlayM4_SetFileEndMsg(LONG nPort,HWND hWnd,UINT nMsg)	
参 数:	LONG nPort	播放通道号
	HWND hWnd	消息发送的窗口
	UINT nMsg	用户自定义的输入的消息; 当播放到文件结束时用户从 hWnd 窗口过程中收到这个消息。此消息函数中的 wParam 参数返回 nPort 的值
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	设置文件结束时要发送的消息; 从 2.4 版本 SDK 起, 当文件播放完时, 解码线程将不会自动结束, 需要用户做停止工作。应用程序在收到这个消息后要调用播放结束函数 PlayM4_Stop(nPort)	
注 意:	回调与消息不能同时使用	

[返回目录](#)

6.66.文件结束回调 PlayM4_SetFileEndCallback

函 数:	BOOL PlayM4_SetFileEndCallback(LONG nPort, void(CALLBACK*FileEndCallback)(long nPort, void *pUser), void *pUser)	
参 数:	LONG 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	
	long nPort	播放通道号
	void * pUser	用户数据
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	设置文件播放结束回调函数。在 PlayM4_OpenSteam/PlayM4_OpenFile 之前调用才有效	
注 意:	关于回调函数。因为 vb 不支持多线程, 所以当回调函数是 VB 声明的函数时, 在 vc 的线程中调用 vb 的函数, 会有问题。 详见: Microsoft Knowledge Base Article - Q198607 “PRB: Access Violation in VB Run-Time Using AddressOf ”。 回调与消息不能同时使用。	

[返回目录](#)

6.67.分辨改变通知消息 **PlayM4_SetEncChangeMsg**

函 数:	BOOL PlayM4_SetEncChangeMsg(LONG nPort,HWND hWnd,UINT nMsg)	
参 数:	LONG nPort	播放通道号
	HWND hWnd	hWnd 消息发送的窗口
	UINT nMsg	用户输入的消息; 当设置解码时编码格式发生改变时要发送此定义的消息。消息函数中的 wParam 参数值是返回 nPort 的值
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	设置解码时编码分辨率发生改变时要发送的消息。	
注 意:	回调与消息不能同时使用。	

[返回目录](#)

6.68.分辨率改变通知回调 **PlayM4_SetEncTypeChangeCallBack**

函 数:	BOOL PlayM4_SetEncTypeChangeCallBack (LONG nPort,void(CALLBACK *funEncChange)(long nPort,long nUser),long nUser)	
参 数:	LONG 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
	long nUser	用户数据
	funEncChange 回调函数参数	
	long nPort	播放通道号
	long nUser	用户数据
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	需要在打开文件之前使用	
注 意:	不推荐回调与消息方式同时使用	

[返回目录](#)

6.69.设置丢 B 帧个数 **PlayM4_ThrowBFrameNum**

函 数:	BOOL PlayM4_ThrowBFrameNum(LONG nPort,DWORD nNum)	
参 数:	LONG nPort	播放通道号
	DWORD nNum	不解码 B 帧的帧数。范围 0~2
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	<p>设置不解码 B 帧帧数;</p> <p>不解码 B 帧, 可以减小 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。</p> <p>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。</p> <p>若编码类型中无 B 帧, 则调用此接口无效。</p>	

[返回目录](#)


6.70.检查帧号是否连续 **PlayM4_CheckDiscontinuousFrameNum**

函 数:	BOOL PlayM4_CheckDiscontinuousFrameNum(LONG nPort, BOOL bCheck)	
参 数:	LONG nPort BOOL bCheck	播放通道号 是否设置检查帧号连续性。 TURE: 检查帧号连续性。如果实际码流帧号不连续, 会跳到下一个 I 帧; FALSE: 不检查帧号连续性。如果数据能解码, 则播放, 不直接跳至下一个 I 帧。
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	检查帧号是否连续, 在码流不连续的情况下, 设置检查后, 会跳下一个关键帧解码。4.8build0813 6.1.1.17 版本开始支持。	

[返回目录](#)

6.71.设置水印回调函数 **PlayM4_SetCheckWatermarkCallBack**

函 数:	BOOL PlayM4_SetCheckWatermarkCallBack(LONG nPort,void(CALLBACK* funCheckWatermark)(long nPort,WATERMARK_INFO* pWatermarkInfo,DWORD nUser),DWORD nUser)	
参 数:	LONG nPort void(CALLBACK* funCheckWatermark) DWORD nUser funCheckWatermark 回调函数参数说明: long nPort WATERMARK_INFO* pWatermarkInfo WATERMARK_INFO 结构体说明: typedef struct{ char * pDataBuf; long nSize; long nFrameNum; BOOL bRsaRight; long nReserved; }WATERMARK_INFO; DWORD nUser	播放通道号 检查水印回调函数 用户指针 用户指针 水印信息结构体指针 数据指针 数据大小 帧号 是否是 RSA 加密 (暂时作为保留参数) 保留 用户指针
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:		

[返回目录](#)

6.72.解码密钥 **PlayM4_SetSecretKey**

函 数:	BOOL PlayM4_SetSecretKey(LONG nPort, LONG lKeyType, char *pSecretKey, LONG lKeyLen)	
参 数:	LONG nPort	播放通道号
	LONG 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
	LONG lKeyLen	密钥长度，单位为 bit（1 字节为 8 bit）
返回值:	成功返回 TURE；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 PlayM4_OpenSteam/PlayM4_OpenFile 之前调用才有效。	

[返回目录](#)

显示操作

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

函 数:	BOOL PlayM4_SetOverlayMode(LONG nPort,BOOL bOverlay,COLORREF colorKey)	
参 数:	LONG nPort	播放通道号
	BOOL bOverlay	TRUE 表示将首先尝试使用 OVERLAY 模式，如果不行再使用其他模式。FALSE 则不进行 OVERLAY 模式的尝试。
	COLORREF colorKey	用户设置的透明色，透明色相当于一层透视膜，显示画面只能穿过这种颜色，其他颜色将挡住显示的画面。用户应该在显示窗口中涂上这种颜色，才能看到显示画面。一般应该使用一种不常用的颜色作为透明色。这是一个双字节值 0x00bbgrr, 最高字节为 0，后三个字节分别表示 r, g, b 的值
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	<p>设置 OVERLAY 模式显示画面。在一块显卡中同一时刻只能有一个 OVERLAY 表面处于活动状态，如果此时系统中已经有程序使用了 OVERLAY，那么播放器就不能再创建 OVERLAY 表面，它将自动改用 Off_Screen 表面，并不返回 FALSE。一些常用的播放器，以及我们卡的预览都可能使用了 overlay 表面。同样，如果播放器使用了 OVERLAY 表面，那么，其他的程序将不能使用 OVERLAY 表面，特别注意，我们的卡在预览时可能也要使用 OVERLAY(用户可设置)，如果先打开播放器（并且使用了 OVERLAY），再启动预览，那么预览可能因为得不到 OVERLAY 而失败。</p> <p>使用 OVERLAY 模式的优点是：大部份的显卡都支持 OVERLAY，在一些不支持 BLT 硬件缩放和颜色转换的显卡上（如 SIS 系列显卡)使用 OVERLAY 模式（OVERLAY 模式下的缩放和颜色转换由显卡支持），可以大大减小 cpu 利用率并提高画面质量（相对于软件缩放和颜色转换）。</p> <p>缺点是：只能有一路播放器使用。该设置必须在 PLAY 之前使用，而且需要设置透明色。</p>	

[返回目录](#)

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

函 数:	LONG PlayM4_GetOverlayMode(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	0 表示没有使用 OVERLAY；1 表示使用了 OVERLAY 表面；-1 表示调用失败	
说 明:	检查当前播放器是否使用了 OVERLAY 模式	

[返回目录](#)

6.75. 获取 OVERLAY 关键色 **PlayM4_GetColorKey**

函 数:	COLORREF PlayM4_GetColorKey(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	颜色值	

[返回目录](#)

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

函 数:	BOOL PlayM4_SetDisplayRegion(LONG nPort,DWORD nRegionNum, RECT *pSrcRect,HWND hDestWnd, BOOL bEnable)		
参 数:	LONG	nPort	播放通道号
	DWORD	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。
	RECT *	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。
	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。
	BOOL	bEnable	打开（设置）或关闭显示区域。
返回值:	成功返回 TURE；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。 对于设置子窗口，RECT 宽或者高小于 16 个像素点，不能做放大处理，显示原图像大小。		

[返回目录](#)

6.77. 刷新显示 **PlayM4_RefreshPlay**

函 数:	BOOL PlayM4_RefreshPlay(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	刷新显示。当用户暂停时如果刷新了窗口, 则窗口中的图像因为刷新而消失, 此时调用这个接口可以重新把图像显示出来。只有在暂停和单帧播放时才会执行, 其它情况会直接返回。	

[返回目录](#)



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

函 数:	BOOL PlayM4_RefreshPlayEx(LONG nPort,DWORD nRegionNum)	
参 数:	LONG nPort	播放通道号
	DWORD nRegionNum	显示区域序号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	刷新显示，同 PlayM4_RefreshPlay。为支持 PlayM4_SetDisplayRegion 而增加一个参数。	

[返回目录](#)

6.79.设置显示模式 **PlayM4_SetDisplayType**

函 数:	BOOL PlayM4_SetDisplayType(LONG nPort,LONG nType)	
参 数:	LONG nPort	播放通道号
	LONG nType	几种模式:
	#define DISPLAY_NORMAL 0x00000001	正常分辨率数据送显卡显示。
	#define DISPLAY_QUARTER 0x00000002	1/4 分辨率数据送显卡显示。
	#define DISPLAY_YC_SCALE 0x00000004	YC 伸张
	#define DISPLAY_NOTEARING 0x00000008	无显示撕裂效果
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	<p>设置显示的模式，在小画面显示时，采用 DISPLAY_QUARTER 可以减小显卡工作量，从而支持更多路显示，但画面显示质量有下降。在正常和大画面显示时应该使用 DISPLAY_NORMAL。</p> <p>nType 是按位表示的，3 就是 1 和 2 同时设置</p> <p>正常显示 DISPLAY_NORMAL 和 1/4 显示 DISPLAY_QUARTER 不能同时设置，其他可以组合使用</p>	

[返回目录](#)

6.80.获取显示模式 **PlayM4_GetDisplayType**

函 数:	LONG PlayM4_GetDisplayType(LONG nPort)	
参 数:	LONG nPort	播放通道号
	#define DISPLAY_NORMAL 0x00000001	正常分辨率数据送显卡显示。
	#define DISPLAY_QUARTER 0x00000002	1/4 分辨率数据送显卡显示。
	#define DISPLAY_YC_SCALE 0x00000004	YC 伸张
	#define DISPLAY_NOTEARING 0x00000008	无显示撕裂效果
返回值:	显示模式，如 DISPLAY_NORMAL 或 DISPLAY_QUARTER	

[返回目录](#)

缓冲区操作

解码前源缓冲区

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

函 数:	DWORD PlayM4_GetSourceBufferRemain(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	获取源缓冲区剩余数据大小 (BYTE)	

[返回目录](#)

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

函 数:	BOOL PlayM4_SetSourceBufCallBack(LONG nPort,DWORD nThreShold, void (CALLBACK * SourceBufCallBack)(long nPort,DWORD nBufSize,DWORD dwUser,void*pResvered),DWORD dwUser,void *pReserved)	
参 数:	LONG nPort DWORD nThreShold Void (CALLBACK* SourceBufCallBack) DWORD dwUser void * pReserved SourceBufCallBack 回调函数参数说明 long nPort DWORD nBufSize DWORD dwUser void* pResvered	播放通道号 阈值 回调函数指针 用户数据 保留数据 播放器通道号 缓冲区中剩余数据 用户数据 保留数据
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	设置源缓冲区阈值和剩余数据小于等于阈值时的回调函数指针。只有在数据量从大到小"越过"指定阈值的时候, 才会触发回调; 一次触发后需要调用 PlayM4_ResetSourceBufFlag 接口重置参数, 使得再次有效	

[返回目录](#)

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

函 数:	BOOL PlayM4_ResetSourceBufFlag(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	重置回调标志位为有效状态。	

[返回目录](#)

解码后播放缓冲区

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

函 数:	BOOL PlayM4_SetDisplayBuf(LONG nPort,DWORD nNum)	
参 数:	LONG nPort DWORD nNum	播放通道号 播放缓冲区最大缓冲帧数。 #define MIN_DIS_FRAMES 1 #define MAX_DIS_FRAMES 50
返回值:	成功返回 TURE; 失败返回 FALSE	
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，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 10(帧)，如果用户追求最大延时最小，可以考虑适当减小这个值。 如果设置的只超过有效范围，内部自适应成默认值（6 个），并会返回成功 nNum 参数范围：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。	
注 意:	接口在 PlayM4_OpenStream 之后、PlayM4_Play 之前调用有效	

[返回目录](#)

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

函 数:	DWORD PlayM4_GetDisplayBuf(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	播放缓冲区最大缓冲帧数	

[返回目录](#)

源、播放缓冲区

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

函 数:	BOOL PlayM4_ResetSourceBuffer(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

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

函 数:	BOOL PlayM4_ResetBuffer(LONG nPort,DWORD nBufType)	
参 数:	LONG nPort	播放通道号
	DWORD nBufType	缓冲区类型
	缓冲区宏定义	
	BUF_VIDEO_SRC	视频数据源缓冲区, 缓冲解码之前视频数据, 只对流模式有效, 单位 byte。
	BUF_AUDIO_SRC	音频数据源缓冲区, 缓冲解码之前音频数据, 只对流模式有效, 单位 byte。且音视频数据分开送到情况下才有效
	BUF_VIDEO_RENDER	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:	成功返回 TURE; 失败返回 FALSE	

[返回目录](#)

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

函 数:	DWORD PlayM4_GetBufferValue(LONG nPort,DWORD nBufType)	
参 数:	LONG nPort	播放通道号
	DWORD nBufType	缓冲区类型
	缓冲区宏定义	
	BUF_VIDEO_SRC	视频数据源缓冲区, 缓冲解码之前视频数据, 只对流模式有效, 单位 byte。
	BUF_AUDIO_SRC	音频数据源缓冲区, 缓冲解码之前音频数据, 只对流模式有效, 单位 byte。且音视频数据分开送到情况下才有效
	BUF_VIDEO_RENDER	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)。这个接口可以帮助用户了解缓冲区中的数据, 从而在网络延时方面有所估计。

[返回目录](#)

索引

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

函 数:	BOOL PlayM4_SetFileRefCallBack(LONG nPort, void (__stdcall *pFileRefDone)(DWORD nPort, DWORD nUser), DWORD nUser)	
参 数:	<div>LONG nPort</div> <div>void (__stdcall * pFileRefDone)</div> <div>DWORD nUser</div> <div>pFileRefDone 回调函数参数说明</div> <div>DWORD nPort</div> <div>DWORD nUser</div>	<div>播放通道号</div> <div>索引回调函数</div> <div>用户数据</div> <div>播放器通道号</div> <div>用户数据</div>
返回值:	成功返回 TURE ; 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, 大约每秒处理 40M 左右的数据, 主要是因为从硬盘读数据比较慢。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>	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetKeyFramePos(LONG nPort, DWORD nValue, DWORD nType, PFRAME_POS pFramePos)	
参 数:	<div>LONG nPort</div> <div>DWORD nValue</div> <div>DWORD nType</div>	<div>播放通道号</div> <div>当前位置, 可以是时间或帧号, 类型由 nType 指定 nValue 类型。</div> <div>如果 nType 是 BY_FRAMENUM 则 nValue 表示帧号, 如果 nType 是 BY_FRAMTIME, 则 nValue 表示时间,</div>



	<p>PFRAME_POS pFramePos</p> <p>PFRAME_POS 类型说明</p> <pre>typedef struct{ long nFilePos long nFrameNum long nFrameTime }FRAME_POS,*PFRAME_POS;</pre>	<p>单位毫秒</p> <p>查找到的关键帧的文件位置、帧序号、时标信息</p> <p>文件位置</p> <p>帧序号</p> <p>帧时标（毫秒）</p>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	查找指定位置之前的关键帧位置。图像解码必须从关键帧开始，如果用户保存的文件不是从关键帧开始的，那么到下一个关键帧之前的数据会被忽略。如果用户要截取文件中的一段数据，则应该考虑从关键帧开始截取。结束位置则关系不大，最多丢失 3 帧数据。	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetNextKeyFramePos(LONG nPort,DWORD nValue, DWORD nType, PFRAME_POS pFramePos)	
参 数:	<p>LONG nPort</p> <p>DWORD nValue</p> <p>DWORD nType</p> <p>nValue 设置为:</p> <p>BY_FRAMENUM</p> <p>BY_FRAMETIME</p> <p>PFRAME_POS pFramePos</p> <p>PFRAME_POS 类型说明</p> <pre>typedef struct{ long nFilePos long nFrameNum long nFrameTime }FRAME_POS,*PFRAME_POS;</pre>	<p>播放通道号</p> <p>当前位置，可以是时间或帧号，类型由 nType 指定 nValue 类型。</p> <p>nValue 表示帧号， 则 nValue 表示时间，单位毫秒</p> <p>查找到的关键帧的文件位置、帧序号、时标信息</p> <p>文件位置</p> <p>帧序号</p> <p>帧时标（毫秒）</p>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	查找指定位置之后的关键帧位置。	

[返回目录](#)

6.92.获取文件索引 **PlayM4_GetRefValue**

函 数:	BOOL PlayM4_GetRefValue(LONG nPort,BYTE *pBuffer, DWORD *pSize)	
参 数:	LONG nPort	播放通道号
	[out]BYTE * pBuffer	索引信息
	DWORD * pSize	输入 pBuffer 的大小，输出索引信息大小。
返回值:	成功返回 TURE；失败返回 FALSE	
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:	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	

[返回目录](#)

6.93.设置文件索引 **PlayM4_SetRefValue**

函 数:	BOOL PlayM4_SetRefValue(LONG nPort,BYTE *pBuffer, DWORD nSize)	
参 数:	LONG nPort	播放通道号
	BYTE * pBuffer	索引信息
	DWORD nSize	索引信息的长度
返回值:	成功返回 TURE；失败返回 FALSE	
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](#)

多屏显示

这几个函数接口，是为支持多显卡而增加的。

Windows98,Windows2000 及 Windows2000 之后的操作系统才支持多显卡，并需要安装 DirectX6.0 或更高版本。如果用户不需支持多显卡环境，则这几个接口可以不予考虑。关于多显卡的编程请参考 Microsoft sdk 相关文档 “Multiple-Monitor Systems”。

注意：V6.1.1.0 及以上的版本对于多屏显示自适应处理，不需要再调用下面这部分函数。

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

函 数：	BOOL PlayM4_InitDDrawDevice()	
参 数：	无	
返回值：	成功返回 TURE；失败返回 FALSE	
注 意：	V6.1.1.0 版本及以后版本可以不用显式调用该函数。	

[返回目录](#)

6.95.释放枚举显示设备过程分配的资源

PlayM4_ReleaseDDrawDevice

函 数：	Void PlayM4_ReleaseDDrawDevice()	
参 数：	无	
返回值：	Void	
注 意：	V6.1.1.0 版本及以后版本可以不用显式调用该函数。	

[返回目录](#)

6.96. 获取显示设备（显卡）个数 **PlayM4_GetDDrawDeviceTotalNums**

函 数:	DWORD PlayM4_GetDDrawDeviceTotalNums()	
参 数:		
返回值:	成功返回 TURE；失败返回 FALSE 如果返回 0，则表示系统中只有主显示设备。如果返回 1，则表示系统中安装了多块显卡，但只有一块显卡与 Windows 桌面绑定。返回其他值，则表示系统中与桌面绑定的显卡数目。在多显卡的系统中可以通过设置显示属性，而指定任意一块显卡作为主显示设备。	
说 明:	获得系统中与 windows 桌面绑定的总的显示设备数目（这里主要是指显卡）	
注 意:	V6.1.1.0 版本及以后版本多屏显示可以自适应，无需调用该函数。	

[返回目录](#)

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

函 数:	BOOL PlayM4_SetDDrawDevice(LONG nPort,DWORD nDeviceNum)	
参 数:	LONG nPort	播放通道号
	DWORD nDeviceNum	显示设备的设备号，如果是 0，则表示使用主显示设备
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	设置播放窗口使用的显卡。注意该窗口必须在该显卡所对应的监视器上才能显示播放画面。	
注 意:	V6.1.1.0 版本及以后版本多屏显示可以自适应，无需调用该函数。	

[返回目录](#)

6.98. 设置播放窗口使用的显示设备（多显示区域）

PlayM4_SetDDrawDeviceEx *

函 数:	BOOL PlayM4_SetDDrawDeviceEx(LONG nPort,DWORD nRegionNum, DWORD nDeviceNum);	
参 数:	LONG nPort,	播放通道号
	DWORD nRegionNum	显示区域
	DWORD nDeviceNum	显示设备的设备号
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	设置播放窗口使用的显卡，功能同 PlayM4_SetDDrawDevice。 为支持 PlayM4_SetDisplayRegion 而增加一个参数。	
注 意:	不再支持，内部自适应多屏显示。	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetDDrawDeviceInfo(DWORD nDeviceNum, LPSTR lpDriverDescription,DWORD nDespLen,LPSTR lpDriverName , DWORD nNameLen,HMONITOR *hhMonitor)	
参 数:	DWORD nDeviceNum	显示设备的设备号，如果是 0，则表示主显示设备。
	LPSTR lpDriverDescription	表示 lpDriverDescription 已分配空间的大小，单位 byte。
	DWORD nDespLen	表示 lpDriverName 已分配空间的大小，单位 byte
	[out]LPSTR lpDriverName	显示设备的描述信息
	[out]DWORD nNameLen	显示设备的设备名
	[out]HMONITOR * hhMonitor	显示设备使用的监视器句柄，通过 Windows API 函数 GetMonitorInfo，可以得到详细信息，供用户定位窗口位置。
返回值:	成功返回 TURE；失败返回 FALSE	
注 意:	HMONITOR 类型 ， 当_WIN32_WINNT >= 0x0500 时，在“windef.h”头文件中定义，否则在“Multimon.H”中定义，如果用户在编译过程中找不到该类型定义，可以包含相应的头文件。 V6.1.1.0 版本及以后版本多屏显示可以自适应，该接口在 6.2.0.x 版本后不再起作用，无需调用。	

[返回目录](#)

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

函 数:	Int PlayM4_GetCapsEx(DWORD nDDrawDeviceNum)	
参 数:	DWORD nDDrawDeviceNum	指定显示设备的设备号，如果是 0，则表示主显示设备。
返回值:	指定显示设备的系统信息	
注 意:	V6. 1. 1. 0 版本及以后版本多屏显示可以自适应，该接口在 6. 2. 0. x 版本后不再起作用，无需调用。	

[返回目录](#)

抓图

6.101. 显示回调 **PlayM4_SetDisplayCallBack**

函 数:	BOOL PlayM4_SetDisplayCallBack(LONG nPort,void (CALLBACK* DisplayCBFun) (long nPort,char * pBuf,long nSize,long nWidth,long nHeight,long nStamp, long nType,long nReceved))	
参 数:	<div>LONG nPort void (CALLBACK* DisplayCBFun) DisplayCBFun 显示回调函数参数说明 long nPort char * pBuf long nSize long nWidth long nHeigh long nStamp long nType long nReceved</div>	<div>播放通道号 显示回调函数，若不需要回调函数则可以为 NULL，否则不能置为 NULL 通道号 返回图像数据的指针 返回图像数据大小 画面宽，单位像素 画面高 时标信息，单位毫秒 数据类型， T_YV12， T_RGB32， T_UYVY，详见 PlayM4_SetDecCallBack 宏定义 保留</div>
返回值:	成功返回 TURE；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 DisplayCBFun 设为 NULL。一旦设置回调函数，则一直有效，直到程序退出。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	

[返回目录](#)

6.102. 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:	<div>LONG nPort</div> <div>Void(CALLBACK* DisplayCBFun)</div> <div>Long nUser</div> <div>DisplayCBFun 回调函数参数说明</div> <div>DISPLAY_INFO * pstDisplayInfo</div> <div>DISPLAY_INFO 结构说明</div> <div>typedef struct</div> <div>{</div> <div> long nPort;</div> <div> char * pBuf;</div> <div> long nBufLen;</div> <div> long nWidth;</div> <div> long nHeight;</div> <div> long nStamp;</div> <div> long nType;</div> <div> long nUser;</div> <div>}DISPLAY_INFO</div>	<div>播放通道号</div> <div>显示回调函数，若不需要回调函数则可以为 NULL，否则不能置为 NULL</div> <div>用户指针</div> <div>当前显示信息结构体指针</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, 详见 PlayM4_SetDecCallBack 宏定义</div> <div>用户数据</div>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	当前获取到的图像数据类型为 YV12 格式	

[返回目录](#)

6.103. 图像数据转为 BMP 格式 **PLayM4_ConvertToBmpFile**

函 数:	BOOL PLayM4_ConvertToBmpFile (char* pBuf,long nSize,long nWidth,long nHeight,long nType,char *sFileName)	
参 数:	<div>char * pBuf</div> <div>long nSize</div> <div>long nWidth</div> <div>long nHeight</div> <div>long nType</div> <div>char * sFileName</div>	<div>抓图回调函数中图像缓冲区</div> <div>抓图回调函数中图像的大小</div> <div>抓图回调函数中图像宽度</div> <div>抓图回调函数中图像高度</div> <div>抓图回调函数中图像类型</div> <div>要保存的文件名。最好以 BMP 作为文件扩展名</div>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	将解码后数据保存成 BMP 文件。送入的图像数据必须是 YV12 数据，可从解码回	

	调或者显示回调中获取，可在播放库回调函数中使用，转换函数占用的 cpu 资源。 其他数据类型暂不支持 pbuf 所指向的内容在函数里面不能被修改
--	--

[返回目录](#)

6.104. 图像数据转为 JPEG 格式 **PlayM4_ConvertToJpegFile**

函 数:	BOOL PlayM4_ConvertToJpegFile(char *pBuf, long nSize, long nWidth, long nHeight, long nType, char *sFileName)	
参 数:	char * pBuf	抓图回调函数中图像缓冲区
	long nSize	抓图回调函数中图像的大小
	long nWidth	抓图回调函数中图像宽度
	long nHeight	抓图回调函数中图像高度
	long nType	抓图回调函数中图像类型 (当前的播放库获取的类型是 yv12)
	char * sFileName	要保存的文件名。最好以 JPG 作为文件扩展名
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	将解码后数据保存成 JPEG 文件，该函数可在回调函数中使用， 用法参见 PlayM4_ConvertToBmpFile 若图像不是 16 整数倍的，会裁剪至 16 的整数倍，如 176*120 抓图就会被裁剪到 176*112 pbuf 所指向的内容在函数里面不能被修改	

[返回目录](#)

6.105. 直接抓取 BMP 图像 **PlayM4_GetBMP**

函 数:	BOOL PlayM4_GetBMP(LONG nPort,PBYTE pBitmap,DWORD nBufSize, DWORD* pBmpSize);	
参 数:	LONG nPort	播放通道号
	PBYTE 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。
	DWORD nBufSize	申请的缓冲区大小
	DWORD* pBmpSize	获取到的实际 bmp 图像大小
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	获取的数据为一帧 bmp 数据，写成文件即可用图片浏览工具查看 pbuf 所指向的内容在函数里面不能被修改 抓取当前显示帧的图片，若当前不显示则抓取不到图片。	

[返回目录](#)

6.106. 直接抓取 JPEG 图像 **PlayM4_GetJPEG**

函 数:	BOOL PlayM4_GetJPEG(LONG nPort,PBYTE pJpeg,DWORD nBufSize, DWORD* pJpegSize)	
参 数:	LONG nPort	播放通道号
	PBYTE pJpeg	存放 JPEG 图像数据地址, 由用户分配, 不得小于 JPEG 图像大小, 建议大小 $w * h * 3/2$, 其中 w 和 h 分别为图像宽高。
	DWORD nBufSize	申请的缓冲区大小
	DWORD* pJpegSize	获取到的实际 JPEG 图像数据大小
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	<p>获取的数据为一帧 JPEG 数据, 写成文件即可用图片浏览工具查看</p> <p>若图像不是 16 整数倍的, 会裁剪至 16 的整数倍, 如 176*120 抓图就会被裁剪到 176*112</p> <p>pbuf 所指向的内容在函数里面不能被修改</p> <p>抓取当前显示帧的图片, 若当前不显示则抓取不到图片。</p>	

[返回目录](#)

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

函 数:	BOOL PlayM4_SetJpegQuality(long nQuality)	
参 数:	long 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 TURE; 失败返回 FALSE	
说 明:	<p>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 PlayM4_OpenFile 之前使用</p> <p>有效值范围 (0~100) 外的都取默认值 80, 且不报错</p>	

[返回目录](#)

6.108. 抓取选择区域图片为 JPEG 格式 **PlayM4_GetCropJPEG**

函 数:	BOOL PlayM4_GetCropJPEG(long nPort,CROP_PIC_INFO *pstPicInfo)	
参 数:	long nPort	播放通道号
	CROP_PIC_INFO *pstPicInfo	选择区域参数结构体指针
	<p>CROP_PIC_INFO 结构体参数说明:</p> <pre>typedef struct {</pre>	

[返回目录](#)

[返回目录](#)

其他功能

6.110. 画图回调 **PlayM4_RegisterDrawFun**

函 数:	BOOL PlayM4_RegisterDrawFun(LONG nPort, void (CALLBACK* DrawFun)(long nPort,HDC hDc,LONG nUser),LONG nUser)	
参 数:	LONG nPort void (CALLBACK* DrawFun) LONG nUser DrawFun 回调函数参数说明 long nPort HDC hDc LONG nUser	播放通道号 画图回调函数 用户数据 播放通道号 表面设备上下文，可以像操作显示窗口客户区 DC 那样操作它。 用户数据，就是上面输入的用户数据；
返回值:	成功返回 TURE；失败返回 FALSE	
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），就好像在窗口的客户区 DC 上绘图，但这个 DC 不是窗口客户区的 DC，而是 DirectDraw 里的 Off-Screen 表面的 DC。 注意，如果是使用 overlay 表面，这个接口无效，可以直接在窗口上绘图，只要不是透明色就不会被覆盖。 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	

[返回目录](#)

6.111. 画图回调 **PlayM4_RigisterDrawFun**

函 数:	BOOL PlayM4_RigisterDrawFun(LONG nPort,void (CALLBACK* DrawFun)(long nPort,HDC hDc,LONG nUser),LONG nUser)	
参 数:	LONG nPort (CALLBACK* DrawFun)	播放通道号 画图回调函数

	LONG nUser	用户数据
	DrawFun 回调函数参数说明:	播放通道号
	long nPort	表面设备上下文, 可以像操作显示窗口客户区 DC 那样操作它。
	HDC hDc	
	LONG nUser	用户数据, 就是上面输入的用户数据;
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	拼写错误, 与 PlayM4_RegisterDrawFun 相同	

[返回目录](#)

6.112. 设置数据校验 PlayM4_SetVerifyCallBack *

函 数:	BOOL PlayM4_SetVerifyCallBack(LONG nPort, DWORD nBeginTime, DWORD nEndTime, void (__stdcall* funVerify)(long nPort, FRAME_POS * pFilePos, DWORD blsVideo, DWORD nUser), DWORD nUser)	
参 数:	LONG nPort DWORD nBeginTime DWORD nEndTime void (__stdcall* funVerify) DWORD nUser funVerify 回调函数参数说明 long nPort FRAME_POS * pFilePos DWORD blsVideo DWORD nUser FRAME_POS 类型定义 long nFilePos long nFrameNum long nFrameTime long nErrorFrameNum SYSTEMTIME * pErrorTime long nErrorLostFrameNum long nErrorFrameSize	播放通道号 校验开始时间, 单位 ms 校验结束时间, 单位 ms 当发现数据被修改时回调的函数 用户数据 播放通道号 出错信息 是否视频数据, 1 视频, 0 音频 用户数据 出错的文件位置 出错后正常的帧号 出错后正常的时间 (相对此文件开始时间) 出错时的帧号 出错的绝对时间 (设备录象文件有用) 出错时丢失的帧数 出错时错误的数据大小
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	注册一个回调函数, 校验数据是否被修改, 实现水印功能。现在可以发现数据丢失情况。注意, 该校验在建立文件索引的时候进行, 所以必须建文件索引才能校验。在 PlayM4_OpenFile 之前使用。	
注 意:	暂不支持	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetOriginalFrameCallBack(LONG nPort, BOOL blsChange, BOOL bNormalSpeed,long nStartFrameNum, long nStartStamp, long nFileHeader, void(CALLBACK *funGetOrignalFrame)(long nPort,FRAME_TYPE *frameType, long nUser), long nUser)	
参 数:	<div><div><div>LONGnPort</div><div>BOOLblsChange</div><div>BOOLbNormalSpeed</div><div>longnStartFrameNum</div><div>longnStartStamp</div><div>[out]longnFileHeader</div><div>void(CALLBACK *funGetOrignalFrame)</div><div>longnUser</div><div>funGetOrignalFrame 回调函数参数:</div><div>longnPort</div><div>FRAME_TYPE * frameType</div><div>longnUser</div><div>FRAME_TYPE 类型说明</div><div>typedef struct{<div>char *pDataBuf</div><div>longnSize</div><div>longnFrameNum</div><div>BOOlblsAudio</div><div>longnReserved</div></div>}FRAME_TYPE;</div></div>	<div>播放通道号</div> <div>是否要改变每帧的参数</div> <div>是否要以正常速度得到原始帧</div> <div>如果要改变原始帧帧号, 则是此文件的开始帧号</div> <div>如果要改变原始帧时标, 则是此文件的开始时标</div> <div>文件头版本信息, 如果版本不匹配, 返回不成功</div> <div>原始帧回调函数</div> <div>用户数据</div> <div>播放通道号</div> <div>数据帧信息</div> <div>用户数据</div> <div>数据帧首地址</div> <div>数据帧的大小</div> <div>数据帧的个数</div> <div>是否音频帧</div> <div>保留</div>
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	创建得到原始数据的回调函数, 可以改变每帧的时标和帧号, 在文件打开之后调用, 用于将两个文件拼接在一起	
注 意:	不支持 , 相关功能可以使用其他库来实现	

[返回目录](#)

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

函 数:	BOOL PlayM4_GetFileSpecialAttr(LONG nPort, DWORD *pTimeStamp,DWORD *pFileNum ,DWORD *nFileHeader)	
参 数:	LONGnPort	播放通道号

	[out]DWORD * pTimeStamp	文件结束时标
	[out]DWORD * pFileNum	文件结束帧号
	[out]DWORD * nFileHeader	文件头信息
返回值:	成功返回 TRUE; 失败返回 FALSE	
说 明:	得到文件最后的时标和帧号, 在文件打开之后调用。与上个文件一起使用, 用于文件拼接	
注 意:	不支持, 相关功能可使用其他库来实现	

[返回目录](#)

6.115. 码流错误直接跳至下一关键帧 **PlayM4_SkipErrorData**

函 数:	BOOL PlayM4_SkipErrorData(LONG nPort, BOOL bSkip)	
参 数:	LONG nport	播放通道号
	BOOL bSkip	TRUE 码流错误时会直接跳下一个关键帧解码; FALSE 寻找下一帧数据解码
返回值:	成功返回 TRUE; 失败返回 FALSE	
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 TRUE	

[返回目录](#)

倒放功能

注意：倒放功能是一个扩展性功能，在 V6.3.0 开始支持。

文件方式倒放：

- 1 暂只支持海康基线设备码流
 - 2. 必须成功创建文件索引之后使用此功能
 - 3. 在 play 之后调用 PlayM4_ReversePlay，默认从文件起始开始播放
 - 4. 播放库 6.3.0 版本
 - 若源文件分辨率 1280*960 以下分辨率，I 帧间隔最大支持 100；
 - 若源文件分辨率 1280*960 及其以上分辨率，I 帧间距最大只支持 25；
- I 帧间距可以通过播放库其他的库（海康码流分析库）来获取。*
- 5. 不兼容码流时标和帧号异常反跳的情况。（过播放库其他的库（海康码流分析库）来确定帧号或时标反跳的情况）
- 文件模式倒放仅增加 [PlayM4_ReversePlay](#) 一个接口，实现从正放切换到倒放。未播放前倒放起点默认也是文件的起点，而不是文件结束位置，定位和其他播放操作同正放

流模式倒放要求：

- 1 在 play 之后调用 PlayM4_ReversePlay
- 2 送数据都需要完整的 GOP，可以分多次送
- 3 GOP 在时间上需要时逆序的，即从后面的时间到前面的时间连续送
- 4 流倒放送数据需要倒序，完整的逐个送入 GOP，GOP 数据要精确。
- 5 送完最后一个 GOP，即文件倒放结束前，需要调用 PlayM4_InputData(port, NULL,-1),保证最后一个送入的 GOP 能够正常播放出来；
- 6. 暂不支持流式单帧后退

GOP:Group of Picture ,图像组，一般指 I 帧及其后面依靠该 I 帧解码的其他帧。

6.116. 开始倒放 PlayM4_ReversePlay

函 数：	BOOL PlayM4_ReversePlay(LONG nPort)	
参 数：	LONG nport	播放通道号
返回值：	成功返回 TURE；失败返回 FALSE	
说 明：	6.3 版本开始支持该接口，该接口必须在 PlayM4_Play 之后调用；	
注 意：	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	

[返回目录](#)

6.117. 获取当前播放帧的全局时间 **PlayM4_GetSystemTime**

函 数:	BOOL PlayM4_GetSystemTime(LONG nPort, PLAYM4_SYSTEM_TIME *pstSystemTime)	
参 数:	LONG nport [out]PLAYM4_SYSTEM_TIME *pstSystemTime typedef struct PLAYM4_SYSTEM_TIME { DWORD dwYear; DWORD dwMon; DWORD dwDay; DWORD dwHour; DWORD dwMin; DWORD dwSec; DWORD dwMs; } PLAYM4_SYSTEM_TIME;	播放通道号 全局时间 年 月 日 时 分 秒 毫秒
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	6.3 版本开始有此接口，获取当前播放帧对应的全局时间	

[返回目录](#)

6.118. 设置流播放起始帧（毫秒） **PlayM4_SetPlayedTimeEx**

函 数:	BOOL PlayM4_SetPlayedTimeEx(LONG nPort,DWORD nTime)	
参 数:	LONG nPort DWORD nTime	播放通道号 设置播放起始帧位置，nTime 是指相对于送入的第一个 GOP 起始 I 帧的相对时标。 若需要从第一个 GOP 的 I 帧作为起始帧，则置为 0； 若将 I 帧后第一帧作为播放起始帧，则置为 40（帧率 25 为例,1 秒为 1000 毫秒，其中时间间隔为 1000 / 25 = 40 毫秒）； 若将第一个 GOP 的最后一帧作为播放起始帧，则置为-1
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	文件定位功能同 6.41 说明，流式播放定位	

[返回目录](#)

同步回放功能

6.119. 同步回放接口 **PlayM4_SetSycGroup**

函 数:	BOOL PlayM4_SetSycGroup(LONG nPort, DWORD dwGroupIndex)	
参 数:	LONG nPort	播放通道号
	DWORD dwGroupIndex	同步组序号, 取值在 0~3 之间
返回值:	成功返回 TURE; 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>PlayM4_SetSycGroup 在 PlayM4_Play 之前调用</p> <p>dwGroupIndex 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 PlayM4_FreePort 此播放通道才能退出同步组</p> <p>目前只支持海康设备码流, 且只支持录像文件以流方式模式回放。</p>	

[返回目录](#)

6.120. 设置同步回放开始时间 **PlayM4_SetSycStartTime***

函 数:	BOOL PlayM4_SetSycStartTime(LONG nPort, PLAYM4_SYSTEM_TIME *pstSystemTime)	
参 数:	LONG nPort	播放通道号
	PLAYM4_SYSTEM_TIME *pstSystemTime	同步的系统时间结构体指针
<p>PLAYM4_SYSTEM_TIME 结构体说明</p> <pre>typedef struct PLAYM4_SYSTEM_TIME { DWORD dwYear; DWORD dwMon; DWORD dwDay; DWORD dwHour; DWORD dwMin; DWORD dwSec; DWORD dwMs; } PLAYM4_SYSTEM_TIME;</pre>		
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	暂不支持	

[返回目录](#)

鱼眼功能

6.121. 启用鱼眼接口 **PlayM4_FEC_Enable**

函 数:	BOOL PlayM4_FEC_Enable(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:		

[返回目录](#)

6.122. 关闭鱼眼模块接口 **PlayM4_FEC_Disable**

函 数:	BOOL PlayM4_FEC_Disable(LONG nPort)	
参 数:	LONG nPort	播放通道号
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:		

[返回目录](#)

6.123. 获取鱼眼矫正处理子端口 **PlayM4_FEC_GetPort**

函 数:	BOOL PlayM4_FEC_GetPort(LONG nPort, unsigned int* nSubPort, FECPLACETYPE emPlaceType, FECCORRECTTYPE emCorrectType)	
参 数:	LONG nPort unsigned int* nSubPort FECPLACETYPE emPlaceType FECCORRECTTYPE emCorrectType	播放通道号 处理子端口 安装方式 矫正方式
	FECPLACETYPE 结构体说明: <pre>typedef enum tagFECPlaceType { FEC_PLACE_WALL = 0x1, FEC_PLACE_FLOOR = 0x2, FEC_PLACE_CEILING = 0x3, }FECPLACETYPE;</pre> FECCORRECTTYPE 结构体说明: <pre>typedef enum tagFECCorrectType { FEC_CORRECT_PTZ = 0x100, FEC_CORRECT_180 = 0x200,</pre>	壁装方式 (法线水平) 地面安装 (法线向上) 顶装方式 (法线向下) PTZ 180 度矫正 (对应 2P)

	FEC_CORRECT_360 = 0x300, }FECCORRECTTYPE;	360 全景矫正 （对应 1P）
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:		

[返回目录](#)

6.124. 删除鱼眼矫正处理子端口 **PlayM4_FEC_DelPort**

函 数:	BOOL PlayM4_FEC_DelPort(LONG nPort , unsigned int nSubPort)		
参 数:	LONG	nPort	播放通道号
	unsigned int	nSubPort	处理子端口
返回值:	成功返回 TURE；失败返回 FALSE		
说 明:			

[返回目录](#)

6.125. 设置鱼眼矫正参数 **PlayM4_FEC_SetParam**

函 数:	BOOL PlayM4_FEC_SetParam(LONG nPort , unsigned int nSubPort , FISHEYEPARAM * pPara)		
------	---	--	--

参 数:	<p>LONG nPort</p> <p>unsigned int nSubPort</p> <p>FISHEYEPARAM * pPara</p> <p>FISHEYEPARAM 结构体参数说明:</p> <pre>typedef struct tagFECParam { unsigned int nUpDataType; unsigned int nPlaceAndCorrect; PTZPARAM stPTZParam; CYCLEPARAM stCycleParam; float fZoom; float fWideScanOffset; int nResver[16]; }FISHEYEPARAM;</pre> <p>PTZPARAM 结构体说明:</p> <pre>typedef struct tagPTZParam { float fPTZPositionX; float fPTZPositionY; }PTZPARAM;</pre> <p>CYCLEPARAM 结构体说明:</p> <pre>typedef struct tagCycleParam { float fRadiusLeft; float fRadiusRight; float fRadiusTop; float fRadiusBottom; }CYCLEPARAM;</pre>	<p>播放通道号</p> <p>处理子端口</p> <p>更新的类型</p> <p>安装方式和矫正方式（只能用于获取，SetParam 的时候无效,该值表示安装方式和矫正方式的和）</p> <p>PTZ 校正的参数</p> <p>鱼眼图像圆心参数</p> <p>PTZ 显示的范围参数</p> <p>180 或者 360 度校正的偏移角度</p> <p>保留字段</p> <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>圆的最左边 X 坐标</p> <p>圆的最右边 X 坐标</p> <p>圆的最上边 Y 坐标</p> <p>圆的最下边 Y 坐标</p>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:		

[返回目录](#)

6.126. 获取鱼眼矫正参数 PlayM4_FEC_GetParam

函 数:	BOOL PlayM4_FEC_GetParam(LONG nPort , unsigned int nSubPort , FISHEYEPARAM * pPara)	
参 数:	<p>LONG nPort</p> <p>unsigned int nSubPort</p>	<p>播放通道号</p> <p>处理子端口</p>

	<div>FISHEYEPARAM * pPara</div> <div>FISHEYEPARAM 结构体参数说明: typedef struct tagFECParam { unsigned int nUpDateType; unsigned int nPlaceAndCorrect; PTZPARAM stPTZParam; CYCLEPARAM stCycleParam; float fZoom; float fWideScanOffset; int nResver[16]; }FISHEYEPARAM; PTZPARAM 结构体说明: typedef struct tagPTZParam { float fPTZPositionX; float fPTZPositionY; }PTZPARAM; CYCLEPARAM 结构体说明: typedef struct tagCycleParam { float fRadiusLeft; float fRadiusRight; float fRadiusTop; float fRadiusBottom; }CYCLEPARAM;</div>	<div>鱼眼矫正参数结构体参数指针</div> <div>更新的类型 安装方式和矫正方式（只能用于获取， SetParam 的时候无效,该值表示安装方式和 矫正方式的和） PTZ 校正的参数 鱼眼图像圆心参数 PTZ 显示的范围参数 180 或者 360 度校正的偏移角度 保留字段</div> <div>PTZ 显示的中心位置 X 坐标 PTZ 显示的中心位置 Y 坐标</div> <div>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</div>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:		

[返回目录](#)

6.127. 设置显示窗口，可以随时切换 **PlayM4_FEC_SetWnd**

函 数:	BOOL PlayM4_FEC_SetWnd(LONG nPort , unsigned int nSubPort , void * hWnd)		
参 数:	LONG	nPort	播放通道号
	unsigned int	nSubPort	处理子端口
	void *	hWnd	设置显示窗口句柄
返回值:	成功返回 TURE；失败返回 FALSE		
说 明:			

[返回目录](#)

6.128. 设置鱼眼窗口的绘图回调 **PlayM4_FEC_SetCallBack**

函 数:	BOOL PlayM4_FEC_SetCallBack(LONG nPort , unsigned int nSubPort , (FISHEYE_CallBack cbFunc)(void* pUser, unsigned int nSubPort , unsigned int nCBType , void * hDC , unsigned int nWidth , unsigned int nHeight) , void * pUser)	
参 数:	LONG nPort unsigned int nSubPort FISHEYE_CallBack cbFunc void * pUser FISHEYE_CallBack 回调函数 参数说明: void* pUser, unsigned int nSubPort , unsigned int nCBType , void * hDC , unsigned int nWidth, unsigned int nHeight	播放通道号 处理子端口 鱼眼窗口画图回调函数 用户指针 用户指针 处理子端口 鱼眼的画图回调类型（暂时不支持） DC 表面 DC 表面的宽 DC 表面的高
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:		

[返回目录](#)

倍帧功能

6.129. 倍帧 **PlayM4_MotionFlow**

函 数:	BOOL PlayM4_MotionFlow(LONG nPort, DWORD dwAdjustType)	
参 数:	LONG nPort	播放通道号
	DWORD dwAdjustType	倍帧类型
	#define MOTION_FLOW_NONE 0	不开启倍帧
	#define MOTION_FLOW_CPU 1	CPU 倍帧
	#define MOTION_FLOW_GPU 2	GPU 倍帧
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	倍帧适用于帧率 20 以上的码流，若帧率在 20 以下的码流调用倍帧接口无效果	

[返回目录](#)

图像增强功能

6.130. 设置关闭或开启模块 **PlayM4_VIE_SetModuConfig**

函 数:	BOOL PLAYM4_VIE_SetModuConfig(LONG lPort,int nModuFlag,BOOL bEnable)	
参 数:	LONG lPort	播放通道号
	int nModuFlag	图像增强模块标记, nModuFlag 对应 PLAYM4_VIE_MODULES 宏, 可组合
	BOOL bEnable	是否开启, TRUE 为开启; FALSE 为关闭
	PLAYM4_VIE_MODULES 宏定义说明 typedef enum _PLAYM4_VIE_MODULES { PLAYM4_VIE_MODU_ADJ = 0x00000001, 图像基本调节 PLAYM4_VIE_MODU_EHAN = 0x00000002, 局部增强模块 PLAYM4_VIE_MODU_DEHAZE = 0x00000004, 去雾模块 PLAYM4_VIE_MODU_DENOISE = 0x00000008, 去噪模块 PLAYM4_VIE_MODU_SHARPEN = 0x00000010, 锐化模块 PLAYM4_VIE_MODU_DEBLOCK = 0x00000020, 去块滤波模块 PLAYM4_VIE_MODU_CRB = 0x00000040, 色彩平衡模块 PLAYM4_VIE_MODU_LENS = 0x00000080, 镜头畸变矫正模块 }PLAYM4_VIE_MODULES;	
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	先设置模块开启, 再设置模块参数; 期间采用默认的参数; 关闭模块后, 上次设置的参数清空; 其他接口调用, 必须在该接口开启模块后; 否则, 返回错误	

[返回目录](#)

6.131. 设置图像增强区域 **PlayM4_VIE_SetRegion**

函 数:	BOOL PLAYM4_VIE_SetRegion(LONG lPort,LONG lRegNum,, RECT* pRect)	
参 数:	LONG lPort	播放通道号
	LONG lRegNum	图像增强区域个数
	RECT* pRect	图像增强区域
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	设置图像增强区域, NULL 表示对全图进行增强; 若区域设置超过全图, 采用全图; 最小设置区域为 16*16 像素。 最多可设置 4 个支持设置区域; 6.5.版本只支持一个。多个区域要求不能重叠, 有重叠就报错	

注 意:	设置区域必须是 16 的倍数 如果设定区域的 4 个值不是偶数，内部会处理成偶数，即 /2 * 2;处理后的长宽需要是 16 的倍数。
-------------	--

[返回目录](#)

6.132. 获取开启图像增强模块 **PlayM4_VIE_GetModuConfig**

函 数:	BOOL PLAYM4_VIE_GetModuConfig(LONG IPort,int* dwModuFlag)	
参 数:	<div>LONG IPort</div> <div>[out]int* dwModuFlag</div> <div>PLAYM4_VIE_MODULES 宏定义说明</div> <div>typedef enum _PLAYM4_VIE_MODULES</div> <div>{</div> <div>PLAYM4_VIE_MODU_ADJ = 0x00000001,</div> <div>PLAYM4_VIE_MODU_EHAN = 0x00000002,</div> <div>PLAYM4_VIE_MODU_DEHAZE = 0x00000004,</div> <div>PLAYM4_VIE_MODU_DENOISE = 0x00000008,</div> <div>PLAYM4_VIE_MODU_SHARPEN = 0x00000010,</div> <div>PLAYM4_VIE_MODU_DEBLOCK = 0x00000020,</div> <div>PLAYM4_VIE_MODU_CRB = 0x00000040,</div> <div>PLAYM4_VIE_MODU_LENS = 0x00000080,</div> <div>}PLAYM4_VIE_MODULES;</div>	<div>播放通道号</div> <div>图像增强模块标记</div> <div>图像基本调节</div> <div>局部增强模块</div> <div>去雾模块</div> <div>去噪模块</div> <div>锐化模块</div> <div>去块滤波模块</div> <div>色彩平衡模块</div> <div>镜头畸变矫正模块</div>
返回值:	成功返回 TURE；失败返回 FALSE	
说 明:	获取开启模块	

[返回目录](#)

6.133. 设置增强参数 **PlayM4_VIE_SetParaConfig**

函 数:	BOOL PLAYM4_VIE_SetParaConfig(LONG IPort,PLAYM4_VIE_PARACONFIG* pParaConfig)	
参 数:	<div>LONG IPort</div> <div>PLAYM4_VIE_PARACONFIG*</div> <div>pParaConfig</div> <div>typedef struct</div> <div>_PLAYM4_VIE_DYNPARAM_</div> <div>{</div> <div>int moduFlag;</div> <div>//PLAYM4_VIE_MODU_ADJ</div> <div>int brightVal;</div>	<div>播放通道号</div> <div>图像增强参数设置结构体指针</div> <div>启用的算法处理模块</div> <div>亮度调节值，[-255, 255]</div>

	<pre>int contrastVal; int colorVal; //PLAYM4_VIE_MODU_EHAN int toneScale; int toneGain; int toneOffset; int toneColor; //PLAYM4_VIE_MODU_DEHAZE int dehazeLevel; int dehazeTrans; int dehazeBright; //PLAYM4_VIE_MODU_DENOISE int denoiseLevel; //PLAYM4_VIE_MODU_SHARPEN int usmAmount; int usmRadius; int usmThreshold; //PLAYM4_VIE_MODU_DEBLOCK int deblockLevel; //PLAYM4_VIE_MODU_LENS int lensWarp; int lensZoom; //PLAYM4_VIE_MODU_CRB } PLAYM4_VIE_PARACONFIG;</pre>	<p>对比度调节值, [-256, 255] 饱和度调节值, [-256, 255]</p> <p>滤波范围, [0, 100] 对比度调节, 全局对比度增益值, [-256, 255] 亮度调节, 亮度平均值偏移, [-255, 255] 颜色调节, 颜色保真值, [-256, 255]</p> <p>去雾强度, [0, 255] 透射值, [0, 255] 亮度补偿, [0, 255]</p> <p>去噪强度, [0, 255]</p> <p>锐化强度, [0, 255] 锐化半径, [1, 15] 锐化阈值, [0, 255]</p> <p>去块强度, [0, 100]</p> <p>畸变量, [-256, 255] 缩放量, [-256, 255]</p>
返回值:	成功返回 TURE; 失败返回 FALSE	
说 明:	<p>若未开启模块, 则参数设置被忽略 //如果有一个参数设置错误, or 有参数设置被忽略, 是否要返回失败; 其他的参数设置是否要生效? 如在 PLAYM4_VIE_MODULES 中定义如 PLAYM4_VIE_MODU_ADJ PLAYM4_VIE_MODU_EHAN 模块启用后, 则必须设置相应的参数;</p>	

[返回目录](#)

6.134. 获取开启模块的参数 **PlayM4_VIE_GetParaConfig**

函 数:	BOOL PLAYM4_VIE_GetParaConfig(LONG lPort,PLAYM4_VIE_PARACONFIG* pParaConfig)
------	--

参 数:	<p>LONG IPort</p> <p>[out]PLAYM4_VIE_PARACONFIG* pParaConfig</p> <p>typedef struct _PLAYM4_VIE_DYNPARAM_ {</p> <p>int moduFlag;</p> <p>//PLAYM4_VIE_MODU_ADJ</p> <p>int brightVal;</p> <p>int contrastVal;</p> <p>int colorVal;</p> <p>//PLAYM4_VIE_MODU_EHAN</p> <p>int toneScale;</p> <p>int toneGain;</p> <p>int toneOffset;</p> <p>int toneColor;</p> <p>//PLAYM4_VIE_MODU_DEHAZE</p> <p>int dehazeLevel;</p> <p>Int dehazeTrans;</p> <p>int dehazeBright;</p> <p>//PLAYM4_VIE_MODU_DENOISE</p> <p>int denoiseLevel;</p> <p>//PLAYM4_VIE_MODU_SHARPEN</p> <p>Int usmAmount;</p> <p>int usmRadius;</p> <p>int usmThreshold;</p> <p>//PLAYM4_VIE_MODU_DEBLOCK</p> <p>int deblockLevel;</p> <p>//PLAYM4_VIE_MODU_LENS</p> <p>int lensWarp;</p> <p>int lensZoom;</p> <p>//PLAYM4_VIE_MODU_CRB</p> <p>} PLAYM4_VIE_PARACONFIG;</p>	<p>播放通道号</p> <p>图像增强参数设置结构体指针</p> <p>启用的算法处理模块</p> <p>亮度调节值, [-255, 255]</p> <p>对比度调节值, [-256, 255]</p> <p>饱和度调节值, [-256, 255]</p> <p>滤波范围, [0, 100]</p> <p>对比度调节, 全局对比度增益值, [-256, 255]</p> <p>亮度调节, 亮度平均值偏移, [-255, 255]</p> <p>颜色调节, 颜色保真值, [-256, 255]</p> <p>去雾强度, [0, 255]</p> <p>透射值, [0, 255]</p> <p>亮度补偿, [0, 255]</p> <p>去噪强度, [0, 255]</p> <p>锐化强度, [0, 255]</p> <p>锐化半径, [1, 15]</p> <p>锐化阈值, [0, 255]</p> <p>去块强度, [0, 100]</p> <p>畸变量, [-256, 255]</p> <p>缩放量, [-256, 255]</p>
返回值:	成功返回 TURE; 失败返回 FALSE	

说 明:	
------	--

[返回目录](#)

7 注意事项

1. 回调函数的实现函数中不能调用带 nPort 为参数的接口。

例如：

//声明解码回调函数

```
NAME(PlayM4_SetDecCallBackEx)(m_lPort, DecCBFun, NULL, 0);
```

//解码回调函数实现体：

```
void CALLBACK DecCBFun(long nPort, char * pBuf, long nSize,
                        FRAME_INFO * pFrameInfo,
                        long nReserved1, long /*nReserved2*/)
{
    //不能调用类似于 PlayM4_XXXXX(long nPort,...)的接口;
}
```

2. 文件不能超过 4G 或小于 4K;
3. 若需要对文件进行定位建议先建立索引，可实现精确定位；
4. 播放库 V6.1.1.0 以及以上的版本对于多屏显示实现了自适应，不需要再调用播放库中的相关的接口。
5. 文件倒放功能必须先建议索引，流式倒放 GOP 需要逆序，完整送入播放库。
6. 用远程协助，远程登录方式，使用 pc 显卡资源有限，软件层会转 RGB 去显示，消耗 cpu，这属于正常现象
7. 辅屏主屏配置修改后，需要释放播放库后，重新加载播放库。
8. 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。
9. 倍帧适用于帧率高于 20 的码流，若帧率低于 20 的码流则调用倍帧接口无效果；
10. 图像增强接口调用需要先设置模块开启，再设置模块参数；期间采用默认的参数，其他接口调用，必须在该接口开启模块后；否则，返回错误。

[返回目录](#)

科技呵护未来

First Choice for Security Professionals