

# CPhoneC 中间件接口函数开发手册

## 2 次开发接口 API 描述

介绍 CPhoneC 中间件的 API 功能，如：控制 USB 设备，以便第三方软件（如：CRM 系统，办公软件等）能更好利用中间件的功能，快速实现无缝对接

jiujiu

QQ: 969571843 [7\*24 小时]

版本: 2.0

2013/11/6

版权所有 © 北京西风软件工作室 2013。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。



## 目 录

1 前言 .....	11
1.1 通用说明 .....	11
1.2 开发时需要注意的细节 .....	11
1.3 安装设备驱动和插件 .....	12
2 接口函数 .....	12
2.1 打开设备 (CPC_OpenDevice) .....	12
2.1.1 ODT_LUSBDEV 打开本地 USB 设备模块 .....	12
2.1.2 ODT_SOUND 打开声卡模块 .....	13
2.1.3 ODT_SIP 打开 SIP 网络模块 .....	13
2.2 关闭设备(CPC_CloseDevice) .....	14
2.2.1 ODT_LUSBDEV 关闭本地 USB 设备模块 .....	15
2.2.2 ODT_SOUND 关闭声卡模块 .....	15
2.2.3 ODT_SIP 关闭 SIP 网络模块 .....	15
2.2.4 ODT_CHANNEL 关闭指定的通道 .....	16
2.3 获取设备信息 (CPC_DevInfo) .....	16
2.3.1 CPC_DEVINFO_GETCHIPTYPE 获取 USB 模块类型 .....	16
2.3.2 CPC_DEVINFO_GETCHIPS 获取 USB 模块数量 .....	17
2.3.3 CPC_DEVINFO_GETTYPE 获取通道类型 .....	17
2.3.4 CPC_DEVINFO_GETMODULE 获取通道功能模块 .....	18
2.3.5 CPC_DEVINFO_GETCHIPCHID 获取通道所在 USB 芯片的传输 ID .....	18
2.3.6 CPC_DEVINFO_GETSERIAL 获取通道序列号 .....	18
2.3.7 CPC_DEVINFO_GETCHANNELS 获取通道数量 .....	19
2.3.8 CPC_DEVINFO_GETDEVID 获取通道所在的 USB 模块 ID .....	19
2.3.9 CPC_DEVINFO_GETDLLVER 获取开发包 SDK 里的 DLL 版本号 .....	20
2.3.10 CPC_DEVINFO_GETCHIPCHANNEL 获取 USB 模块第一个传输 ID 的通道 ID .....	20
2.3.11 CPC_DEVINFO_GETCHIPCHANNELS 获取 USB 模块第二个传输 ID 的通道 ID .....	20
2.3.12 CPC_DEVINFO_GETCHANNELTYPE 获取该通道接入的线路类型 .....	21
2.3.13 CPC_DEVINFO_FILEVERSION 获取库文件版本字符 .....	21
2.3.14 CPC_DEVINFO_FILEVERSIONL 获取库文件版本数字 .....	21
2.4 通道参数设置 (CPC_SetParam) .....	22
2.4.1 CPC_PARAM_BUSY 忙音检测灵敏度 .....	22
2.4.2 CPC_PARAM_DTMFLEVEL 检测 DTMF 时允许的性噪比 .....	23
2.4.3 CPC_PARAM_DTMFVOL 检测 DTMF 时允许的最小能量值 .....	23
2.4.4 CPC_PARAM_DTMFNUM 检测 DTMF 时允许的最小持续时长 .....	24
2.4.5 CPC_PARAM_DTMFLOWINHIGH 检测 DTMF 时低频不能超过高频的倍数 .....	24
2.4.6 CPC_PARAM_DTMFHIGHINLOW 检测 DTMF 时高频不能超过低频的倍数 .....	25
2.4.7 CPC_PARAM_DTMFCALLIDLEVEL 检测 DTMF 来电号码时允许的性噪比 .....	25
2.4.8 CPC_PARAM_DTMFCALLIDVOL 检测 DTMF 来电号码时允许的最小能量值 .....	26
2.4.9 CPC_PARAM_DTMFCALLIDNUM 检测 DTMF 来电号码时允许的最小持续时长 .....	26



2.4.10	CPC_PARAM_DIALSPEED	DTMF 拨号时的时间长度	26
2.4.11	CPC_PARAM_DIALSILENCE	DTMF 拨号时两个号码之间静音长度	27
2.4.12	CPC_PARAM_DIALVOL	DTMF 拨号音量大小	27
2.4.13	CPC_PARAM_RINGSILENCE	来电响铃后多少时间静音提示为未接电话	28
2.4.14	CPC_PARAM_CONNECTSILENCE	通话时线路上连续多少时间静音后通知	28
2.4.15	CPC_PARAM_RINGBACKNUM	摘机后拨几个号码以上开始检测回铃声	29
2.4.16	CPC_PARAM_FLASHELAPSE	软件拍插簧间隔时间	29
2.4.17	CPC_PARAM_FLASHENDELAPSE	软拍插簧后等待多少时间回调	30
2.4.18	CPC_PARAM_RINGELAPSE	内线 Phone 震铃响的时间长度	30
2.4.19	CPC_PARAM_RINGSILENCEELAPSE	内线 Phone 震铃后的静音时间长度	31
2.4.20	CPC_PARAM_RINGTIMEOUT	内线 Phone 震铃超时次数	31
2.4.21	CPC_PARAM_RINGCALLIDTYPE	内线震铃时发送号码的方式	32
2.4.22	CPC_PARAM_REFUSEELAPSE	调用拒接来电时间间隔时间长度	32
2.4.23	CPC_PARAM_DIALTONETIMEOUT	摘机后检测拨号音超时时间	32
2.4.24	CPC_PARAM_MINCHKFLASHELAPSE	检测拍插簧时挂机至少维持时间	33
2.4.25	CPC_PARAM_MAXCHKFLASHELAPSE	检测拍插簧时挂机最大维持时间	33
2.4.26	CPC_PARAM_HANGUPELAPSE	检测电话机挂机时至少维持时间	34
2.4.27	CPC_PARAM_OFFHOOKELAPSE	检测电话机摘机时至少维持时间	34
2.4.28	CPC_PARAM_RINGHIGHELAPSE	检测震铃响时至少维持的时间	35
2.4.29	CPC_PARAM_RINGLOWELAPSE	检测震铃静音时至少维持的时间	35
2.4.30	CPC_PARAM_DIALTONERESULT	检测拨号音超后处理方式	36
2.4.31	CPC_PARAM_ECHOTHRESHOLD	回音抵消为全回音的抵消倍数	36
2.4.32	CPC_PARAM_ECHODECVALUE	回音抵消后为全回音后再缩小倍数	37
2.4.33	CPC_PARAM_SIGSILENCEAM	静音检测幅度大小	37
2.4.34	CPC_PARAM_AM_MIC	麦克风音量	38
2.4.35	CPC_PARAM_AM_SPKOUT	耳机音量	38
2.4.36	CPC_PARAM_AM_LINEIN	线路录音音量	38
2.4.37	CPC_PARAM_AM_LINEOUT	播放到线路语音音量	39
2.4.38	CPC_PARAM_AM_DOPLAY	喇叭音量	39
2.4.39	CPC_PARAM_CITYCODEL	电话线所在的城市区号数值	40
2.4.40	CPC_PARAM_CITYCODE	电话线所在的城市区号字符	40
2.5	获取通道参数 (CPC_GetParam)		41
2.5.1	CPC_PARAM_BUSY	忙音检测灵敏度	41
2.5.2	CPC_PARAM_DTMFLEVEL	检测 DTMF 时允许的性噪声比	41
2.5.3	CPC_PARAM_DTMFVOL	检测 DTMF 时有效的 DTMF 最小能量值	42
2.5.4	CPC_PARAM_DTMFNUM	检测 DTMF 时允许的最小持续时长	42
2.5.5	CPC_PARAM_DTMFLOWINHIGH	检测 DTMF 时低频不能超过高频的倍数	42
2.5.6	CPC_PARAM_DTMFHIGHINLOW	检测 DTMF 时高频不能超过低频的倍数	43
2.5.7	CPC_PARAM_DIALSPEED	DTMF 拨号时的时间长度	43
2.5.8	CPC_PARAM_DIALSILENCE	DTMF 拨号时两个号码之间静音长度	43
2.5.9	CPC_PARAM_DIALVOL	DTMF 拨号音量大小	43
2.5.10	CPC_PARAM_RINGSILENCE	来电响铃后多少时间静音提醒为未接电话	44
2.5.11	CPC_PARAM_CONNECTSILENCE	通话时线路上连续多少时间静音后通知	44
2.5.12	CPC_PARAM_RINGBACKNUM	摘机后拨几个号码以上开始检测回铃声	44



2.5.13 CPC_PARAM_FLASHELAPSE	使用软件拍插簧间隔时间	45
2.5.14 CPC_PARAM_FLASHENDELAPSE	软拍插簧后等待多少时间回调	45
2.5.15 CPC_PARAM_RINGELAPSE	内线 Phone 震铃响的时间长度	45
2.5.16 CPC_PARAM_RINGSILENCEELAPSE	内线 Phone 震铃后的静音时间长度	45
2.5.17 CPC_PARAM_RINGTIMEOUT	内线 Phone 震铃超时次数	46
2.5.18 CPC_PARAM_RINGCALLIDTYPE	内线震铃时发送号码的方式	46
2.5.19 CPC_PARAM_REFUSEELAPSE	调用拒接来电时间间隔时间长度	46
2.5.20 CPC_PARAM_DIALTONETIMEOUT	摘机后检测拨号音超时时间	47
2.5.21 CPC_PARAM_MINCHKFLASHELAPSE	检测拍插簧时挂机至少维持时间	47
2.5.22 CPC_PARAM_MAXCHKFLASHELAPSE	检测拍插簧时挂机最多维持时间	47
2.5.23 CPC_PARAM_HANGUPELAPSE	检测电话机挂机时至少维持时间	47
2.5.24 CPC_PARAM_OFFHOOKELAPSE	检测电话机摘机时至少维持时间	48
2.5.25 CPC_PARAM_RINGHIGHELAPSE	检测震铃响时至少维持的时间	48
2.5.26 CPC_PARAM_RINGLOWELAPSE	检测震铃静音时至少维持的时间	48
2.5.27 CPC_PARAM_DIALTONERESULT	检测拨号音超时后处理方式	49
2.5.28 CPC_PARAM_ECHOTHRESHOLD	回音抵消为全回音的抵消倍数	49
2.5.29 CPC_PARAM_ECHODECVALUE	回音抵消后为全回音后再缩小倍数	49
2.5.30 CPC_PARAM_SIGSILENCEAM	静音幅度大小	49
2.5.31 CPC_PARAM_AM_MIC	麦克风音量	50
2.5.32 CPC_PARAM_AM_SPKOUT	耳机音量	50
2.5.33 CPC_PARAM_AM_LINEIN	线路录音音量	50
2.5.34 CPC_PARAM_AM_LINEOUT	播放到线路语音音量	51
2.5.35 CPC_PARAM_AM_DOPLAY	喇叭音量	51
2.5.36 CPC_PARAM_CITYCODEL	电话线所在的城市区号	51
2.6 USB 设备控制 (CPC_SetDevCtrl)		51
2.6.1 CPC_CTRL_DOSHARE	通道共享模式	52
2.6.2 CPC_CTRL_DOHOOK	软件摘挂机	52
2.6.3 CPC_CTRL_DOPHONE	断开电话机跟 PSTN 线路连接	53
2.6.4 CPC_CTRL_DOPLAY	打开/关闭喇叭	53
2.6.5 CPC_CTRL_DOLINETOSPK	打开/关闭线路声音到耳机	53
2.6.6 CPC_CTRL_DOPLAYTOSPK	打开/关闭播放的语音到耳机	54
2.6.7 CPC_CTRL_DOMICTOLINE	打开/关闭麦克风到电话线	54
2.6.8 CPC_CTRL_ECHO	打开/关闭回音抵消	55
2.6.9 CPC_CTRL_RECVFSK	打开/关闭接收 FSK 制式的来电号码	55
2.6.10 CPC_CTRL_RECVDTMF	打开/关闭接收 DTMF	56
2.6.11 CPC_CTRL_RECVSIGN	打开/关闭信号音检测	56
2.6.12 CPC_CTRL_WATCHDOG	打开/关闭设备看门狗	57
2.6.13 CPC_CTRL_PLAYMUX	选择到喇叭的语音线路	57
2.6.14 CPC_CTRL_PLAYTOLINE	打开/关闭播放的语音到线路	58
2.6.15 CPC_CTRL_SELECTLINEIN	选择输入的线路通道	59
2.6.16 CPC_CTRL_SELECTADCIN	选择输入的为线路语音还是麦克风语音	59
2.6.17 CPC_CTRL_PHONEPOWER	打开/关闭给电话机独立供电	60
2.6.18 CPC_CTRL_RINGPOWER	打开/关闭内线供电震铃	60
2.6.19 CPC_CTRL_LEDPOWER	打开/关闭 LED 指示灯	61



2.6.20 CPC_CTRL_LINEOUT	打开/关闭线路输出功能	61
2.6.21 CPC_CTRL_UPLOAD	打开/关闭设备数据上传功能	61
2.6.22 CPC_CTRL_DOWNLOAD	打开/关闭设备数据下载功能	62
2.6.23 CPC_CTRL_POLARITY	打开/关闭极性反转检测	62
2.6.24 CPC_CTRL_READFRAMENUM	设置一次读取 USB 帧数量	63
2.7	获取 USB 设备控制状态 (CPC_GetDevCtrl)	63
2.7.1 CPC_CTRL_DOSHARE	通道共享模式	64
2.7.2 CPC_CTRL_DOHOOK	软件摘挂机	64
2.7.3 CPC_CTRL_DOPHONE	电话机跟 PSTN 线路连接是否断开	64
2.7.4 CPC_CTRL_DOPLAY	打开/关闭喇叭	65
2.7.5 CPC_CTRL_DOLINETOSPK	打开/关闭线路声音到耳机	65
2.7.6 CPC_CTRL_DOPLAYTOSPK	打开/关闭播放的语音到耳机	65
2.7.7 CPC_CTRL_DOMICTOLINE	打开/关闭麦克风到电话线	66
2.7.8 CPC_CTRL_ECHO	打开/关闭回音抵消	66
2.7.9 CPC_CTRL_RECVFSK	打开/关闭接收 FSK 制式的来电号码	66
2.7.10 CPC_CTRL_RECVDTMF	打开/关闭接收 DTMF	67
2.7.11 CPC_CTRL_RECVSIGN	打开/关闭信号音检测	67
2.7.12 CPC_CTRL_WATCHDOG	打开/关闭设备看门狗	67
2.7.13 CPC_CTRL_PLAYMUX	选择到喇叭的语音线路	68
2.7.14 CPC_CTRL_PLAYTOLINE	打开/关闭播放的语音到线路	68
2.7.15 CPC_CTRL_SELECTLINEIN	选择输入的线路通道	68
2.7.16 CPC_CTRL_SELECTADCIN	选择输入的为线路语音还是麦克风语音	68
2.7.17 CPC_CTRL_PHONEPOWER	打开/关闭给电话机独立供电	69
2.7.18 CPC_CTRL_RINGPOWER	打开/关闭内线供电震铃	69
2.7.19 CPC_CTRL_LEDPOWER	打开/关闭 LED 指示灯	69
2.7.20 CPC_CTRL_LINEOUT	打开/关闭线路输出功能	70
2.7.21 CPC_CTRL_UPLOAD	打开/关闭设备数据上传功能	70
2.7.22 CPC_CTRL_DOWNLOAD	打开/关闭设备数据下载功能	70
2.7.23 CPC_CTRL_POLARITY	打开/关闭极性反转检测	71
2.7.24 CPC_CTRL_POLARITY	获取一次 USB 请求的帧数量	71
2.7.25 CPC_CTRL_PHONE	检测本地电话机摘挂机状态	71
2.7.26 CPC_CTRL_MICIN	检测麦克风插入状态	72
2.7.27 CPC_CTRL_RINGTIMES	获取当前来电已经响铃的次数	72
2.7.28 CPC_CTRL_RINGSTATE	检测来电响铃/静音状态	72
2.8	播放语音文件(CPC_PlayFile)	73
2.8.1 CPC_PLAY_FILE_START	开始播放语音文件	73
2.8.2 CPC_PLAY_FILE_SETCALLBACK	设置播放文件回调函数	74
2.8.3 CPC_PLAY_FILE_SETVOLUME	设置播放文件音量	75
2.8.4 CPC_PLAY_FILE_GETVOLUME	获取播放文件音量	76
2.8.5 CPC_PLAY_FILE_PAUSE	暂停播放文件	76
2.8.6 CPC_PLAY_FILE_RESUME	恢复播放文件	77
2.8.7 CPC_PLAY_FILE_ISPAUSE	检测是否已暂停播放	77
2.8.8 CPC_PLAY_FILE_SETREPEAT	设置是否循环播放	78
2.8.9 CPC_PLAY_FILE_ISREPEAT	检测是否在循环播放	78



2.8.10 CPC_PLAY_FILE_SEEKTO 跳转到某个时间开始播放.....	79
2.8.11 CPC_PLAY_FILE_SETREPEATTIMEOUT 设置循环播放超时次数.....	80
2.8.12 CPC_PLAY_FILE_GETREPEATTIMEOUT 获取循环播放超时次数.....	80
2.8.13 CPC_PLAY_FILE_SETPLAYTIMEOUT 设置播放超时时长.....	81
2.8.14 CPC_PLAY_FILE_GETPLAYTIMEOUT 获取播放超时时长.....	82
2.8.15 CPC_PLAY_FILE_TOTALLEN 播放的语音文件总时长.....	82
2.8.16 CPC_PLAY_FILE_CURSEEK 当前播放时间的离文件开始偏移时长.....	83
2.8.17 CPC_PLAY_FILE_ELAPSE 已经总共播放语音文件的时长.....	83
2.8.18 CPC_PLAY_FILE_ISPLAY 检查播放句柄是否有效.....	84
2.8.19 CPC_PLAY_FILE_ENABLEAGC 打开/关闭自动增益播放.....	85
2.8.20 CPC_PLAY_FILE_IENABLEAGC 检测是否使用了自动增益播放.....	85
2.8.21 CPC_PLAY_FILE_STOP 停止播放指定文件.....	86
2.8.22 CPC_PLAY_FILE_GETCOUNT 获取有效的播放文件句柄数量.....	86
2.8.23 CPC_PLAY_FILE_STOPALL 停止播放所有文件.....	87
2.9 播放语音流缓冲(CPC_PlayBuf).....	87
2.9.1 CPC_PLAY_BUF_START 开始缓冲播放.....	88
2.9.2 CPC_PLAY_BUF_SETCALLBACK 设置语音流缓冲播放回调函数.....	89
2.9.3 CPC_PLAY_BUF_SETWAVEFORMAT 指定缓冲播放的语音流的编码格式.....	90
2.9.4 CPC_PLAY_BUF_WRITEDATA 写语音流数据到缓冲.....	91
2.9.5 CPC_PLAY_BUF_SETVOLUME 设置播放语音流缓冲的音量.....	92
2.9.6 CPC_PLAY_BUF_GETVOLUME 获取播放语音流缓冲的音量.....	92
2.9.7 CPC_PLAY_BUF_SETUSERVALUE 设置本次语音流播放的用户自定义数值.....	93
2.9.8 CPC_PLAY_BUF_GETUSERVALUE 获取本次语音流播放的用户自定义数值.....	94
2.9.9 CPC_PLAY_BUF_ENABLEAGC 打开/关闭语音流自动增益处理.....	94
2.9.10 CPC_PLAY_BUF_IENABLEAGC 检测是否使用语音流自动增益处理.....	95
2.9.11 CPC_PLAY_BUF_PAUSE 暂停播放缓冲.....	95
2.9.12 CPC_PLAY_BUF_RESUME 恢复播放缓冲.....	96
2.9.13 CPC_PLAY_BUF_ISPAUSE 检测是否已暂停播放.....	97
2.9.14 CPC_PLAY_BUF_STOP 停止播放语音流缓冲.....	97
2.9.15 CPC_PLAY_BUF_FREESIZE 获取当前缓冲空闲字节长度.....	98
2.9.16 CPC_PLAY_BUF_DATASIZE 获取当前缓冲语音数据长度.....	98
2.9.17 CPC_PLAY_BUF_TOTALSAMPLES 获取本次语音流播放已经总共播放的采样数.....	99
2.9.18 CPC_PLAY_BUF_SETJITTERBUFSIZE 设置动态缓冲播放的长度.....	99
2.9.19 CPC_PLAY_BUF_GETJITTERBUFSIZE 获取动态缓冲播放的长度.....	100
2.9.20 CPC_PLAY_BUF_GETCOUNT 获取有效的语音流播放句柄数量.....	100
2.9.21 CPC_PLAY_BUF_STOPALL 停止所有语音流播放.....	101
2.10 多文件连续播放(CPC_PlayMultiFile).....	102
2.10.1 CPC_PLAY_MULTIFILE_START 开始多文件连续播放.....	102
2.10.2 CPC_PLAY_MULTIFILE_PAUSE 暂停多文件连续播放.....	103
2.10.3 CPC_PLAY_MULTIFILE_RESUME 恢复多文件连续播放.....	104
2.10.4 CPC_PLAY_MULTIFILE_ISPAUSE 检测是否暂停了多文件连续播放.....	104
2.10.5 CPC_PLAY_MULTIFILE_SETVOLUME 设置多文件连续播放的音量.....	105
2.10.6 CPC_PLAY_MULTIFILE_GETVOLUME 获取多文件连续播放的音量.....	105
2.10.7 CPC_PLAY_MULTIFILE_ISSTART 检测是否启动了多文件连续播放.....	106





2. 10. 8	CPC_PLAY_MULTIFILE_STOP 停止多文件连续播放 .....	106
2. 10. 9	CPC_PLAY_MULTIFILE_STOPALL 停止全部多文件连续播放 .....	107
2. 11	字符串播放(CPC_PlayString).....	107
2. 11. 1	CPC_PLAY_STRING_INITLIST 初始化字符播放列表 .....	108
2. 11. 2	CPC_PLAY_STRING_START 开始字符播放 .....	109
2. 11. 3	CPC_PLAY_STRING_PAUSE 暂停字符播放 .....	109
2. 11. 4	CPC_PLAY_STRING_RESUME 恢复字符播放 .....	110
2. 11. 5	CPC_PLAY_STRING_ISPAUSE 检测是否暂停了字符播放 .....	110
2. 11. 6	CPC_PLAY_STRING_SETVOLUME 设置字符播放音量 .....	111
2. 11. 7	CPC_PLAY_STRING_GETVOLUME 获取字符播放音量 .....	111
2. 11. 8	CPC_PLAY_STRING_ISSTART 检测是否启动了字符播放 .....	112
2. 11. 9	CPC_PLAY_STRING_STOP 停止字符播放 .....	113
2. 11. 10	CPC_PLAY_STRING_STOPALL 停止全部字符播放 .....	113
2. 12	文件录音 (CPC_RecordFile).....	114
2. 12. 1	CPC_RECORD_FILE_START 开始文件录音 .....	114
2. 12. 2	CPC_RECORD_FILE_PAUSE 暂停文件录音 .....	115
2. 12. 3	CPC_RECORD_FILE_RESUME 恢复文件录音 .....	116
2. 12. 4	CPC_RECORD_FILE_ISPAUSE 检测是否暂停文件录音 .....	116
2. 12. 5	CPC_RECORD_FILE_ELAPSE 获取已经文件录音的时长 .....	117
2. 12. 6	CPC_RECORD_FILE_SETVOLUME 设置文件录音音量 .....	117
2. 12. 7	CPC_RECORD_FILE_GETVOLUME 获取文件录音音量 .....	118
2. 12. 8	CPC_RECORD_FILE_PATH 获取当前录音文件的路径 .....	118
2. 12. 9	CPC_RECORD_FILE_STOP 停止指定文件录音 .....	119
2. 12. 10	CPC_RECORD_FILE_STOPALL 停止全部文件录音 .....	120
2. 13	缓冲语音流录音 (CPC_RecordBuf).....	120
2. 13. 1	CPC_RECORD_BUF_HWND_START 开始缓冲录音窗口回调 .....	121
2. 13. 2	CPC_RECORD_BUF_HWND_STOP 停止某个缓冲录音窗口回调 .....	121
2. 13. 3	CPC_RECORD_BUF_HWND_STOPALL 停止全部缓冲录音窗口回调 .....	122
2. 13. 4	CPC_RECORD_BUF_CALLBACK_START 开始缓冲录音回调函数 .....	122
2. 13. 5	CPC_RECORD_BUF_CALLBACK_STOP 停止缓冲录音回调函数 .....	123
2. 13. 6	CPC_RECORD_BUF_CALLBACK_STOPALL 停止全部缓冲录音回调函数 .....	124
2. 13. 7	CPC_RECORD_BUF_SETCBSAMPLES 设置回调采样数 .....	124
2. 13. 8	CPC_RECORD_BUF_GETCBSAMPLES 获取设置的回调采样数 .....	125
2. 13. 9	CPC_RECORD_BUF_ENABLEECHO 打开/关闭回音抵消后的语音数据 .....	126
2. 13. 10	CPC_RECORD_BUF_ISENABLEECHO 检测是否使用了回音抵消后的语音数据 .....	127
2. 13. 11	CPC_RECORD_BUF_PAUSE 暂停缓冲录音 .....	127
2. 13. 12	CPC_RECORD_BUF_ISPAUSE 检测是否暂停缓冲录音 .....	128
2. 13. 13	CPC_RECORD_BUF_RESUME 恢复缓冲录音 .....	129
2. 13. 14	CPC_RECORD_BUF_SETVOLUME 设置缓冲录音音量 .....	130
2. 13. 15	CPC_RECORD_BUF_GETVOLUME 获取缓冲录音音量 .....	131
2. 13. 16	CPC_RECORD_BUF_SETWAVEFORMAT 设置录音回调的语音编码格式 .....	131
2. 13. 17	CPC_RECORD_BUF_GETWAVEFORMAT 获取录音回调的语音编码格式 .....	132
2. 14	通道语音会议控制 (CPC_Conference).....	133
2. 14. 1	CPC_CONFERENCE_CREATE 创建会议 .....	134



2. 14. 2	CPC_CONFERENCE_ADDTOCONF 增加通道到指定会议中	134
2. 14. 3	CPC_CONFERENCE_GETCONFID 获取某个通道所在的会议 ID	135
2. 14. 4	CPC_CONFERENCE_SETSPKVOLUME 设置会议中某个通道播放音量	136
2. 14. 5	CPC_CONFERENCE_GETSPKVOLUME 获取会议中某个通道播放音量	136
2. 14. 6	CPC_CONFERENCE_SETMICVOLUME 设置会议中某个通道说话音量	137
2. 14. 7	CPC_CONFERENCE_GETMICVOLUME 获取会议中某个通道说话音量	137
2. 14. 8	CPC_CONFERENCE_PAUSE 暂停某个会议	138
2. 14. 9	CPC_CONFERENCE_RESUME 恢复某个会议	139
2. 14. 10	CPC_CONFERENCE_ISPAUSE 检测某个会议是否被暂停了	139
2. 14. 11	CPC_CONFERENCE_ENABLESPK 打开/关闭会议中某个通道听功能	140
2. 14. 12	CPC_CONFERENCE_IENABLESPK 检测会议听功能是否打开	140
2. 14. 13	CPC_CONFERENCE_ENABLEMIC 打开/关闭会议者说功能	141
2. 14. 14	CPC_CONFERENCE_IENABLEMIC 检测会议者说功能是否打开	141
2. 14. 15	CPC_CONFERENCE_ENABLEAGC 打开/关闭会议中的通道自动增益	142
2. 14. 16	CPC_CONFERENCE_IENABLEAGC 检测是否打开了自动增益	143
2. 14. 17	CPC_CONFERENCE_DELETECHANNEL 把通道从会议中删除	143
2. 14. 18	CPC_CONFERENCE_DELETECONF 删除一个会议	144
2. 14. 19	CPC_CONFERENCE_DELETEALLCONF 删除所有会议	144
2. 14. 20	CPC_CONFERENCE_GETCONFCOUNT 获取会议数量	145
2. 14. 21	CPC_CONFERENCE_RECORD_START 开始会议录音	145
2. 14. 22	CPC_CONFERENCE_RECORD_PAUSE 暂停会议录音	146
2. 14. 23	CPC_CONFERENCE_RECORD_RESUME 恢复会议录音	147
2. 14. 24	CPC_CONFERENCE_RECORD_ISPAUSE 检测是否暂停会议录音	147
2. 14. 25	CPC_CONFERENCE_RECORD_ISSTART 检测是否启动了会议录音	148
2. 14. 26	CPC_CONFERENCE_RECORD_FILEPATH 获取会议录音文件路径	148
2. 14. 27	CPC_CONFERENCE_RECORD_STOP 停止指定会议录音	149
2. 14. 28	CPC_CONFERENCE_RECORD_STOPALL 停止全部会议录音	150
2. 15	事件获取 (CPC_Event)	150
2. 15. 1	CPC_EVENT_POP 获取事件缓冲里最早的一个事件	151
2. 15. 2	CPC_EVENT_POPEX 获取事件缓冲里最早的一个事件	151
2. 15. 3	CPC_EVENT_TYPE 获取事件类型	152
2. 15. 4	CPC_EVENT_HANDLE 获取事件句柄	152
2. 15. 5	CPC_EVENT_RESULT 获取事件数值	153
2. 15. 6	CPC_EVENT_PARAM 获取事件参数	154
2. 15. 7	CPC_EVENT_DATA 获取事件数据	154
2. 15. 8	CPC_EVENT_DATAEX 获取事件附加数据	155
2. 15. 9	CPC_EVENT_REMOVE 删除事件队列里最早的一个事件	156
2. 15. 10	CPC_EVENT_REMOVEALL 删除事件队列里所有的事件	156
2. 15. 11	CPC_EVENT_REGWND 注册事件接收窗口	157
2. 15. 12	CPC_EVENT_UNREGWND 删除事件接收窗口	157
2. 15. 13	CPC_EVENT_REGCBFUNC 注册接收事件的回调函数	158
2. 15. 14	CPC_EVENT_REGCBFUNC_EX 注册接收事件的回调函数	158
2. 15. 15	CPC_EVENT_UNREGCBFUNC 删除接收事件的回调函数	159
2. 16	通用功能 (CPC_General)	160





2. 16. 1 CPC_GENERAL_STARTDIAL 开始拨号 .....	160
2. 16. 2 CPC_GENERAL_SENDDIAL 接通线路后二次拨号 .....	161
2. 16. 3 CPC_GENERAL_REDIAL 重拨最后一次呼叫的号码 .....	161
2. 16. 4 CPC_GENERAL_STOPDIAL 终止拨号/二次拨号 .....	162
2. 16. 5 CPC_GENERAL_ISDIALING 检测是否正在拨号/二次拨号 .....	163
2. 16. 6 CPC_GENERAL_STARTRING 开始给接在 phone 口设备模拟间隔震铃 .....	163
2. 16. 7 CPC_GENERAL_STOPRING 停止给接在 phone 口设备模拟间隔震铃 .....	164
2. 16. 8 CPC_GENERAL_ISRINGING 检测是否给接在 phone 口设备模拟震铃 .....	164
2. 16. 9 CPC_GENERAL_STARTFLASH 开始拍插簧 .....	164
2. 16. 10 CPC_GENERAL_STOPFLASH 停止拍插簧 .....	165
2. 16. 11 CPC_GENERAL_ISFLASHING 检测是否正在拍插簧 .....	166
2. 16. 12 CPC_GENERAL_STARTREFUSE 拒接当前正在呼入的来电 .....	166
2. 16. 13 CPC_GENERAL_STOPREFUSE 停止拒接 .....	167
2. 16. 14 CPC_GENERAL_ISREFUSEING 检测是否正拒接来电 .....	167
2. 16. 15 CPC_GENERAL_GETCALLIDTYPE 获取本次来电号码的获取类型 .....	168
2. 16. 16 CPC_GENERAL_GETCALLID 获取本次呼入的来电号码 .....	168
2. 16. 17 CPC_GENERAL_GETTELDIALCODE 获取本次话机已经拨出的号码 .....	169
2. 16. 18 CPC_GENERAL_GETTELDIALCODEEX 获取本次话机已经拨出的号码 .....	169
2. 16. 19 CPC_GENERAL_RESETTELDIALBUF 清空话机已经拨出的号码 .....	170
2. 16. 20 CPC_GENERAL_GETTELDIALLEN 获取本次话机已经拨出的号码长度 .....	170
2. 16. 21 CPC_GENERAL_STARTSHARE 启动设备共享服务模块 .....	170
2. 16. 22 CPC_GENERAL_STOPSHARE 停止设备共享服务模块 .....	171
2. 16. 23 CPC_GENERAL_ISSHARE 检测是否启动了共享服务模块 .....	171
2. 16. 24 CPC_GENERAL_ENABLECALLIN 禁止/允许外线呼入 .....	172
2. 16. 25 CPC_GENERAL_IENABLECALLIN 检测是否允许外线呼入 .....	172
2. 16. 26 CPC_GENERAL_ISLINEHOOK 检测线路是否处于摘机状态 .....	172
2. 16. 27 CPC_GENERAL_ISLINEFREE 检测线路是否处于空闲状态 .....	173
2. 16. 28 CPC_GENERAL_RESETRINGBACK 复位检测到的回铃音,重新启动检测 .....	173
2. 16. 29 CPC_GENERAL_CHECKCHANNELID 检测通道 ID 是否合法 .....	174
2. 16. 30 CPC_GENERAL_CHECKDIALTONE 获取系统是否已经检测到拨号音了 .....	174
2. 16. 31 CPC_GENERAL_CHECKSILENCE 开始检测静音 .....	175
2. 16. 32 CPC_GENERAL_CHECKVOICE 开始检测声音 .....	176
2. 16. 33 CPC_GENERAL_CHECKLINESTATE 检测线路状态是否可用/接反 .....	176
2. 16. 34 CPC_GENERAL_SETUSERVALUE 设置用户自定义通道数值 .....	177
2. 16. 35 CPC_GENERAL_GETUSERVALUE 获取保存的用户自定义通道数值 .....	177
2. 16. 36 CPC_GENERAL_SETUSERSTRING 设置用户自定义通道字符串 .....	178
2. 16. 37 CPC_GENERAL_GETUSERSTRING 获取保存的用户自定义通道字符串 .....	178
2. 16. 38 CPC_GENERAL_READPARAM 从配置文件里读取所有参数 .....	179
2. 16. 39 CPC_GENERAL_WRITEPARAM 把当前的所有参数都写入到配置文件 .....	179
2. 17 PSTN 线路呼叫日志信息(CPC_CallLog) .....	180
2. 17. 1 CPC_CALLLOG_BEGINTIME 呼叫开始时间 .....	180
2. 17. 2 CPC_CALLLOG_RINGBACKTIME 呼出时检测到回铃时间 .....	181
2. 17. 3 CPC_CALLLOG_CONNECTEDTIME 呼入/呼出时检测到接通时间 .....	181
2. 17. 4 CPC_CALLLOG_ENDTIME 呼叫结束时间 .....	182



2. 17. 5 CPC_CALLLOG_CALLTYPE 呼入/呼出类型 .....	182
2. 17. 6 CPC_CALLLOG_CALLRESULT 呼入/呼出结果 .....	182
2. 17. 7 CPC_CALLLOG_CALLID 呼入/呼出号码 .....	183
2. 17. 8 CPC_CALLLOG_CALLRECFILE 获取录音文件路径 .....	183
2. 17. 9 CPC_CALLLOG_DELRECFILE 删除录音文件 .....	183
2. 17. 10 CPC_CALLLOG_RESET 复位所有呼叫信息 .....	184
2. 18 设备存储读写功能 (CPC_Storage) .....	184
2. 18. 1 CPC_STORAGE_PUBLIC_READ 读取共享区域数据 .....	185
2. 18. 2 CPC_STORAGE_PUBLIC_READSTR 读取共享区域的字符串数据 .....	185
2. 18. 3 CPC_STORAGE_PUBLIC_WRITE 写入数据到共享区域 .....	186
2. 18. 4 CPC_STORAGE_PUBLIC_SETREADPWD 设置读取共享区域数据的密码 .....	187
2. 18. 5 CPC_STORAGE_PUBLIC_SETWRITEPWD 设置写入共享区域数据的密码 .....	187
2. 18. 6 CPC_STORAGE_PUBLIC_GETSPACESIZE 获取共享区域的总共空间长度 .....	188
2. 19 通用工具类 (CPC_Tool) .....	188
2. 19. 1 CPC_TOOL_PSTNEND 检测中国大陆的号码长度是否结束 .....	189
2. 19. 2 CPC_TOOL_CODETYPE 检测中国大陆的号码类型 .....	189
2. 19. 3 CPC_TOOL_LOCATION 获取中国大陆的号码所在地信息 .....	190
2. 19. 4 CPC_TOOL_DISKFREESPACE 获取该硬盘剩余空间 .....	190
2. 19. 5 CPC_TOOL_DISKTOTALSPACE 获取该硬盘总共空间 .....	191
2. 19. 6 CPC_TOOL_DISKLIST 获取该硬盘所有的盘符列表 .....	191
2. 19. 7 CPC_TOOL_CONVERTFMT 转换 wav 语音文件格式 .....	192
2. 19. 8 CPC_TOOL_SELECTDIRECTORY 弹出一个选择目录的对话框 .....	192
2. 19. 9 CPC_TOOL_SELECTFILE 弹出一个选择文件的对话框 .....	193
2. 19. 10 CPC_TOOL_APMQUERYSUSPEND 是否允许 PC 系统进入待机/休眠状态 .....	194
2. 19. 11 CPC_TOOL_SLEEP 让调用该方法的线程堵塞一定时间 .....	194
2. 20 WEB 服务器操作(CPC_Remote) .....	195
2. 20. 1 CPC_REMOTE_UPLOADFILE 使用 http 协议上传文件到 WEB 服务器 .....	195
2. 20. 2 CPC_REMOTE_DOWNLOADFILE 使用 http 协议下载文件 .....	196
2. 20. 3 CPC_REMOTE_UPLOADDATA 使用 http 上传数据 .....	197



## 1 前言

### 1.1 通用说明

- 以下提到的‘PSTN’为公共电话交换网，也就为接在USB设备Line口的线路。
- 以下提到的话机/电话机为接在USB设备Phone口的设备(如：普通模拟电话机/模拟交换机)。
- 以下提到的内线也就为接在Phone的设备(如：普通模拟电话机/模拟交换机)。
- 以下提到的PC为接着USB设备的电脑。
- 以下提到的本地USB设备为接在本程序所在的计算机上的USB设备
- 设备的通道号从 0 开始，也就是说要控制第1个通道时通道号用 0,以次类推
- 以下提到的参数类型
 

CPCINT64	-	表示有符号64bit(8字节)
CPCINT32	-	表示有符号32bit(4字节)
CPCUINT32	-	表示无符号32bit(4字节)
CPCINT16	-	表示有符号16bit(2字节)
CPCUINT16	-	表示有符号16bit(2字节)
CPCBYTE8	-	表示无符号8bit(1字节)
CPCCHAR8	-	表示有符号8bit(1字节)
CPCPCCHAR8	-	表示字符串指针(ANSI)
CPCPSHORT16	-	表示字符串指针(UNICODE)
- 以下提到的[in]表示该参数为输入。
- 以下提到的[out]表示该参数为输出。
- 以下提到的[in, out]表示该参数为输入也可以做输出。
- 以下提到的音量设置里的默认音量都为 100，如果设置为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音，如果为 10000 就表示把音量放大 100 倍。
- 以下使用的操作类型名称实际为一个对应的数字，具体值可以查阅 参数定义文档，开发时也可以用数字代替，如：CPC\_OpenDevice(ODT\_SOUND, 0, 0); 用 CPC\_OpenDevice(1, 0, 0);因为 ODT\_SOUND 对应的数值是 1，使用字母的形式是为提高代码可读性。
- /\*\*/只是注释信息，可以忽略掉

### 1.2 开发时需要注意的细节

- 1.不同的编译器对 int,long 的字长定义不一样，  
如：有些 int 为 32 位,有些 int 为 16 位  
有些 long 为 32 位,有些 long 为 64 位  
- 使用前请仔细查阅该编译器说明
- 2.使用保留的参数值时都必须使用 0,保留的字符参数为空字符，如：""
- 3.最后一次函数执行的错误可以使用 CPC\_GetLastError 来获取,返回 32 位的错误值，具体错误原因查看中间件错误文档定义。  
函数原型: CPCINT32 BRISDKLIBAPI CPC\_GetLastError();



## 1.3 安装设备驱动和插件

方式一：插入设备，windows 会提示有新硬件插入，取消该窗口，运行开发包 Plugins\_Driver\_Setup.exe 进行安装

方式二：运行开发包 Plugins\_Driver\_Setup.exe 进行安装，然后插入设备

备注：

如果是在 win7 或 win8 系统安装，点击 Plugins\_Driver\_Setup.exe，选择 ‘管理员身份运行’，安装后系统就自动安装了驱动和支持所有浏览器的插件。

## 2 接口函数

### 2.1 打开设备 (CPC\_OpenDevice)

#### 功能描述 (Description)

打开 CPhoneC 支持的各种 USB 设备模块/声卡模块/网络通讯模块等

#### 函数原型 (Function)

```
CPCINT32 CPC_OpenDevice (/*[in]*/ CPCUINT32 uDevType,  
                        /*[in]*/ CPCUINT32 uValue,  
                        /*[in]*/ CPCCHAR8 * pValue);
```

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 参数描述 (Parameters description)

#### 2.1.1 ODT\_LUSBDEV 打开本地 USB 设备模块

##### 功能描述 (Description)

打开本地 USB 设备模块

##### 参数介绍 (Parameters)

参数名	参数值	描述
uDevType	ODT_LUSBDEV	



nValue	0 - 打开所有本地 USB 设备 其它 - 打开指定数量的本地 USB 设备	打开本地 USB 设备的参数
pValue	保留	

**返回值 (Return Values)**

小于 0 表示失败, 其它返回打开的通道数量

**备注 (Remarks)**

使用其它所有设备 API 时必须打开设备

**例如 (Example)**

```
CPC_OpenDevice(ODT_LUSBDEV, 0, 0);
```

- 打开所有本地 USB 设备

```
CPC_OpenDevice(ODT_LUSBDEV, 2, 0);
```

- 打开最前面 2 个本地 USB 设备, 如果设备少于 2 个就返回实际共打开的数量

## 2.1.2 ODT\_SOUND 打开声卡模块

**功能描述 (Description)**

打开本地声卡模块

**参数介绍 (Parameters)**

参数名	参数值	描述
uDevType	ODT_SOUND	
nValue	保留	
pValue	保留	

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

要使用本地声卡功能时必须先打开声卡模块

**例如 (Example)**

```
CPC_OpenDevice(ODT_SOUND, 0, 0);
```

## 2.1.3 ODT\_SIP 打开 SIP 网络模块

**功能描述 (Description)**

打开网络即时消息/P2P 语音/文件传输的 sip 网络通信模块

**参数介绍 (Parameters)**

参数名	参数值	描述
uDevType	ODT_SIP	
nValue	保留	
pValue	指定 SIP 模块的 license 字符	默认使用: <a href="http://www.cphonec.com/v1">http://www.cphonec.com/v1</a>

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败



**备注 (Remarks)**

- 非特殊需求的用户使用“http://www.cphonec.com/v1”为 license

**例如 (Example)**

```
CPC_OpenDevice(ODT_SIP,0,“http://www.cphonec.com/v1”);
```

## 2.1.4 ODT\_PLUGIN 打开设备插入检测模块

**功能描述 (Description)**

开始启动设备插入模块检测，如果检测到设备插入后，系统将收到 CPCEvent\_PlugIn 事件，然后自动停止检测。

**参数介绍 (Parameters)**

参数名	参数值	描述
uDevType	ODT_PLUGIN	
nValue	保留	
pValue	保留	

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败

**备注 (Remarks)**

- 打开设备失败后或者检测到设备被拔出后可以启动来自动检测设备插入检测。

**例如 (Example)**

```
CPC_OpenDevice(ODT_PLUGIN, 0, “”);
```

## 2.2 关闭设备(CPC\_CloseDevice)

**功能描述 (Description)**

关闭指定的设备模块

**函数原型 (Function)**

```
CPCINT32 CPC_CloseDevice (/*[in]*/ CPCUINT32 uDevType,  
/*[in]*/ CPCUINT32 uValue);
```

**返回值 (Return Values)**

有符号 32bit 位,大于 0 表示成功, 其它表示失败.

**参数描述 (Parameters description)**



## 2.2.1 ODT\_LUSBDEV 关闭本地 USB 设备模块

### 功能描述 (Description)

关闭本地 USB 设备模块

### 参数介绍 (Parameters)

参数名	参数值	描述
uDevType	ODT_LUSBDEV	
nValue	保留	

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

设备关闭后将不能在使用设备相关的 API.

### 例如 (Example)

```
CPC_CloseDevice(ODT_LUSBDEV, 0);
```

## 2.2.2 ODT\_SOUND 关闭声卡模块

### 功能描述 (Description)

关闭声卡模块

### 参数介绍 (Parameters)

参数名	参数值	描述
uDevType	ODT_SOUND	
nValue	保留	

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_CloseDevice(ODT_SOUND,0);
```

## 2.2.3 ODT\_SIP 关闭 SIP 网络模块

### 功能描述 (Description)

关闭网络即时消息/P2P 语音/文件传输的网络模块

### 参数介绍 (Parameters)

参数名	参数值	描述
uDevType	ODT_SIP	
nValue	保留	

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_CloseDevice(ODT_SIP, 0);
```



## 2.2.4 ODT\_CHANNEL 关闭指定的通道

### 功能描述 (Description)

关闭指定的通道

### 参数介绍 (Parameters)

参数名	参数值	描述
uDevType	ODT_CHANNEL	
nValue	通道 ID, 从 0 - N	0 表示设备第一个通道

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_CloseDevice(ODT_CHANNEL,1);  
- 关闭通道 1(第 2 个通道)
```

## 2.3 获取设备信息 (CPC\_DevInfo)

### 功能描述 (Description)

获取 USB 设备相关信息

### 函数原型 (Function)

```
CPCINT32 CPC_DevInfo (/*[in]*/CPCINT32 nChannelID,  
/*[in]*/CPCUINT32 uDevInfoType);
```

### 返回值 (Return Values)

有符号 32bit 位, 小于等于 0 表示失败, 其它表示成功

### 参数描述 (Parameters description)

## 2.3.1 CPC\_DEVINFO\_GETCHIPTYPE 获取 USB 模块类型

### 功能描述 (Description)

获取 USB 模块类型

### 参数介绍 (Parameters)

参数名	参数值	描述
-----	-----	----



nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETCHIPTYPE	

**返回值 (Return Values)**

小于等于 0 表示失败, 其它表示 USB 的模块类型

**备注 (Remarks)**

该功能保留

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHIPTYPE);
```

## 2.3.2 CPC\_DEVINFO\_GETCHIPS 获取 USB 模块数量

**功能描述 (Description)**

获取 USB 模块数量

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	保留	
uDevInfoType	CPC_DEVINFO_GETCHIPS	

**返回值 (Return Values)**

小于 0 表示失败, 其它返回打开的设备里总共有几个 USB 芯片

**备注 (Remarks)**

- 该值等同于最后一个通道的设备 ID+1.(CPC\_DEVINFO\_GETDEVID)
- 某些设备的一个 USB 芯片具有 2 个通道, 当一个 USB 芯片只有一个通道时, 设备 ID 就是通道 ID

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHIPS);
```

## 2.3.3 CPC\_DEVINFO\_GETTYPE 获取通道类型

**功能描述 (Description)**

获取通道类型

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETTYPE	

**返回值 (Return Values)**

小于 0 表示失败, 其它返回设备的类型 ID, 具体 ID 所指定的信息查看相关定义

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETTYPE);
```



## 2.3.4 CPC\_DEVINFO\_GETMODULE 获取通道功能模块

### 功能描述 (Description)

获取通道拥有的功能模块

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETMODULE	

### 返回值 (Return Values)

小于 0 表示失败，返回设备拥有的功能模块，多个功能按逻辑与组合

### 备注 (Remarks)

功能模块如：(详细信息查看专门的设备功能模块定义)

DEVMODULE\_DOPLAY

...

DEVMODULE\_POLARITY

### 例如 (Example)

```
CPC_DevInfo(0, CPC_DEVINFO_GETMODULE);
```

## 2.3.5 CPC\_DEVINFO\_GETCHIPCHID 获取通道所在 USB 芯片的传输 ID

### 功能描述 (Description)

获取通道所在 USB 芯片的传输 ID

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETCHIPCHID	

### 返回值 (Return Values)

小于 0 表示失败，其它返回该通道在 USB 芯片里的传输 ID.(0 或者 1)

### 备注 (Remarks)

一个 USB 芯片模块有可能拥有两个传输通道。

- 0 表示第一个传输通道

- 1 表示第二个传输通道

### 例如 (Example)

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHIPCHID);
```

## 2.3.6 CPC\_DEVINFO\_GETSERIAL 获取通道序列号

### 功能描述 (Description)

获取通道序列号

### 参数介绍 (Parameters)

参数名	参数值	描述
-----	-----	----





nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETSERIAL	

**返回值 (Return Values)**

小于 0 表示失败, 返回通道的设备序列号值

**备注 (Remarks)**

- 序列号为一个 32bit 的数值,不是字符串
- 如果该通道没有序列号返回 0

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETSERIAL);
```

## 2.3.7 CPC\_DEVINFO\_GETCHANNELS 获取通道数量

**功能描述 (Description)**

获取打开的 USB 总共通道数量

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETCHANNELS	

**返回值 (Return Values)**

小于 0 表示失败, 返回打开的总共的通道数量

**备注 (Remarks)**

- 有些一个设备拥有 2 个通道
- 该通道在 USB 里传输 ID(CPC\_DEVINFO\_GETCHIPID)可能为 0 或者 1

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHANNELS);
```

## 2.3.8 CPC\_DEVINFO\_GETDEVID 获取通道所在的 USB 模块 ID

**功能描述 (Description)**

- ◆ 获取通道所在的 USB 模块 ID

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道 ID	0 表示设备第一个通道
uDevInfoType	CPC_DEVINFO_GETDEVID	

**返回值 (Return Values)**

小于 0 表示失败, 返回该通道所在的 USB 芯片 ID

**备注 (Remarks)**

- 最后一个通道的 USB 模块 ID 也就为总共打开了几个 USB 芯片
- 某些设备的一个 USB 芯片具有 2 个通道, 当一个 USB 芯片只有一个通道时, 设备 ID 就是通道 ID

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETDEVID);
```



### 2.3.9 CPC\_DEVINFO\_GETDLLVER 获取开发包 SDK 里的 DLL 版本号

#### 功能描述 (Description)

获取开发包 SDK 里的 DLL 版本号

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	保留	
uDevInfoType	CPC_DEVINFO_GETDLLVER	

#### 返回值 (Return Values)

小于 0 表示失败，返回使用的 SDK 的 DLL 版本的数值

#### 备注 (Remarks)

当 DLL 新版本改进后该版本 ID 会发生变化，这样会有新的功能在低版本的 DLL 里不能使用，必须在指定的 DLL 版本以上，

#### 例如 (Example)

```
CPC_DevInfo(0, CPC_DEVINFO_GETDLLVER);
```

### 2.3.10 CPC\_DEVINFO\_GETCHIPCHANNEL

#### 获取 USB 模块第一个传输 ID 的通道 ID

#### 功能描述 (Description)

获取 USB 模块第一个传输 ID 所在的通道 ID

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	USB 模块 ID	
uDevInfoType	CPC_DEVINFO_GETCHIPCHANNEL	

#### 返回值 (Return Values)

小于 0 表示失败，返回通道号

#### 备注 (Remarks)

- 如果您想控制某个 USB 模块，但不知道通道 ID 时可以使用该功能先获取通道 ID

#### 例如 (Example)

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHIPCHANNEL);
```

### 2.3.11 CPC\_DEVINFO\_GETCHIPCHANNELS

#### 获取 USB 模块第二个传输 ID 的通道 ID

#### 功能描述 (Description)

获取 USB 模块第二个传输 ID 所在的通道 ID

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	USB 模块 ID	
uDevInfoType	CPC_DEVINFO_GETCHIPCHANNELS	

**返回值 (Return Values)**

小于 0 表示失败，返回通道号

**备注 (Remarks)**

- 如果想控制某个 USB 模块，但不知道通道 ID 时可以使用该功能先获取通道 ID

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHIPCHANNELS);
```

**2.3.12 CPC\_DEVINFO\_GETCHANNELTYPE 获取该通道接入的线路类型****功能描述 (Description)**

获取该通道接入的线路类型

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道 ID	
uDevInfoType	CPC_DEVINFO_GETCHANNELTYPE	

**返回值 (Return Values)**

小于 0 - 表示失败

- 1 - 该通道是接电话机，只能对电话机独立操作
- 2 - 该通道是接电话线，只能对电话线独立操作，检测不到电话机摘挂拨号
- 3 - 该通道即有电话线，也有电话机，可以检测到电话机摘挂拨号

**备注 (Remarks)**

- 无

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_GETCHANNELTYPE);
```

**2.3.13 CPC\_DEVINFO\_FILEVERSION 获取库文件版本字符****功能描述 (Description)**

获取该接口函数所在的 dll(npcphonec.dll)文件版本字符串格式的信息

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	保留	
uDevInfoType	CPC_DEVINFO_FILEVERSION	

**返回值 (Return Values)**

返回 npcphonec.dll 文件的版本信息字符的内存地址,如: 1,11,8,12

**备注 (Remarks)**

- 该版本信息跟查看该 DLL 文件属性中的文件版本是一样的

**例如 (Example)**

```
CPC_DevInfo(0, CPC_DEVINFO_FILEVERSION);
```

**2.3.14 CPC\_DEVINFO\_FILEVERSIONL 获取库文件版本数字****功能描述 (Description)**



获取该接口函数所在的 dll(npcphonec.dll)文件版本数值格式的信息

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	保留	
uDevInfoType	CPC_DEVINFO_FILEVERSIONL	

#### 返回值 (Return Values)

返回 npcphonec.dll 文件的版本信息字符的内存地址,如: 113812

#### 备注 (Remarks)

- 该版本信息跟查看该 DLL 文件属性中的文件版本是一样的

#### 例如 (Example)

```
CPC_DevInfo(0, CPC_DEVINFO_FILEVERSIONL);
```

## 2.4 通道参数设置 (CPC\_SetParam)

### 功能描述 (Description)

设置通道参数(设置参数必须先打开设备)

### 函数原型 (Function)

```
CPCINT32 CPC_SetParam(/*[in]*/CPCINT32 nChannelID,  
                      /*[in]*/CPCUINT32 uParamType,  
                      /*[in]*/CPCINT32 nValue);
```

### 返回值 (Return Values)

有符号 32bit 位,大于 0 表示成功, 其它表示失败

### 参数描述 (Parameters description)

### 2.4.1 CPC\_PARAM\_BUSY 忙音检测灵敏度

#### 功能描述 (Description)

检测到连续几个忙音后回调

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_BUSY	
nValue	忙音个数	范围(1-10), 建议 6 个左右

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 检测忙音个数越少, 检测的灵敏度就越高, 检测时间也越少, 对方挂机后就可以更快的检测到挂机或忙音事件,但在复杂的环境下有可能误测。

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_BUSY,5);
```

## 2.4.2 CPC\_PARAM\_DTMFLEVEL 检测 DTMF 时允许的性噪比

**功能描述 (Description)**

检测 DTMF 时允许的性噪比,该值越小, 检测灵敏度越高

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFLEVEL	
nValue	性噪比等级	范围(0-7), 建议 4 左右

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 检测灵敏度越高, 对 DTMF 的噪声质量要求就越低, 越不容易漏掉接收 DTMF, 但在复杂的环境会提高误测的概率。
- 如果发现有误差 DTMF 码, 可以提高该值来降低灵敏度
- 该参数只对摘机状态下的检测 DTMF 有效, 对 DTMF 来电号码信息检测无效
- 非特殊情况不需要修改该值, 要修改该值建议设置在 1-4。

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_DTMFLEVEL, 3);
```

## 2.4.3 CPC\_PARAM\_DTMFVOL 检测 DTMF 时允许的最小能量值

**功能描述 (Description)**

检测 DTMF 时允许的最小能量值

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFVOL	
nValue	有效 DTMF 的最小能量值	范围(1-100), 建议 20 左右

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 该值越大, 对 DTMF 的声音大小要求就越大, 也就是说如果该值设置的比较大, 但实际接收到的 DTMF 音量比较小, 系统就会丢弃该 DTMF
- 如果发现有音量比较小的检测 DTMF 不到, 可以降低该值来提供灵敏度
- 该参数只对摘机状态下的检测 DTMF 有效, 对 DTMF 来电号码信息检测无效





- 非特殊情况不要修改该值，要修改该值，一般设置在 5-20

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFVOL,10);
```

## 2.4.4 CPC\_PARAM\_DTMFNUM 检测 DTMF 时允许的最小持续时长

功能描述 (Description)

检测 DTMF 时允许的最小持续时长

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFNUM	
nValue	最小持续时长	范围(1-10)，建议 6 左右

返回值 (Return Values)

大于 0 表示成功，其它表示失败

备注 (Remarks)

- 该值越大，对 DTMF 的持续时间要求就越长，也就是说如果该值设置的比较大，但实际接收到的 DTMF 时间比较短，系统就会丢弃该 DTMF。
- 实际需要时间为该值\*8ms，也就是如果该值设置为 6，就需要 DTMF 信号持续时间不能少于 6\*8ms=48ms(毫秒)。
- 中国电信规定标准的 DTMF 持续时间不能少于 45ms.一般话机等设备产生 DTMF 持续时间为 75ms 左右。
- 该参数只对摘机状态下的检测 DTMF 有效，对 DTMF 来电号码信息检测无效
- 建议需要在通话时收 DTMF 码，该值设置在 6-9 左右。
- 该值越小(一般在<6 的时候)在通话中可能会引起误测接收到 DTMF。

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFNUM,7);
```

## 2.4.5 CPC\_PARAM\_DTMFLOWINHIGH

### 检测 DTMF 时低频不能超过高频的倍数

功能描述 (Description)

检测 DTMF 时低频能量不能超过高频能量的倍数

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFLOWINHIGH	
nValue	倍数	范围(2-12)，建议 9 左右

返回值 (Return Values)

大于 0 表示成功，其它表示失败

备注 (Remarks)

- 该值越大，对 DTMF 信号传输后对频率衰减的要求的就越低，误测试概率就会提高



- 默认为 9，非特殊情况用户不需要修改。

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFLOWINHIGH ,7);
```

## 2. 4. 6 CPC\_PARAM\_DTMFHIGHINLOW

### 检测 DTMF 时高频不能超过低频的倍数

功能描述 (Description)

检测 DTMF 时高频能量不能超过低频能量的倍数

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFHIGHINLOW	
nValue	倍数	范围(2-12)，建议 7 左右

返回值 (Return Values)

大于 0 表示成功，其它表示失败

备注 (Remarks)

- 该值越大，对 DTMF 信号传输后对频率衰减的要求的就越低，误测试概率就会提高
- 默认为 7，非特殊情况用户不需要修改
- 非特殊情况不要修改该值

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFHIGHINLOW ,4);
```

## 2. 4. 7 CPC\_PARAM\_DTMFCALLIDLEVEL

### 检测 DTMF 来电号码时允许的性噪比

功能描述 (Description)

检测 DTMF 来电号码时允许的性噪比,该值越小，检测灵敏度越高

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFCALLIDLEVEL	
nValue	性噪比等级	范围(0-7)，建议 4 左右

返回值 (Return Values)

大于 0 表示成功，其它表示失败

备注 (Remarks)

- 同 CPC\_PARAM\_DTMFLEVEL
- 该参数只对挂机状态下的检测 DTMF 来电号码有效，对通话中检测 DTMF 无效
- 非特殊情况不要修改该值

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFCALLIDLEVEL, 3);
```



## 2. 4. 8 CPC\_PARAM\_DTMFCALLIDVOL

### 检测 DTMF 来电号码时允许的最小能量值

#### 功能描述 (Description)

检测 DTMF 来电号码时允许的最小能量值

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFCALLIDVOL	
nValue	有效 DTMF 的最小能量值	范围(1-100)，建议 5 左右

#### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

#### 备注 (Remarks)

- 同 CPC\_PARAM\_DTMFVOL
- 该参数只对挂机状态下的检测 DTMF 来电号码有效，对通话中检测 DTMF 无效
- 非特殊情况不要修改该值

#### 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFCALLIDVOL, 10);
```

## 2. 4. 9 CPC\_PARAM\_DTMFCALLIDNUM

### 检测 DTMF 来电号码时允许的最小持续时长

#### 功能描述 (Description)

检测 DTMF 来电号码时允许的最小持续时长

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFCALLIDNUM	
nValue	最小持续时长	范围(1-10)，建议 3 左右

#### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

#### 备注 (Remarks)

- 同 CPC\_PARAM\_DTMFNUM
- 该参数只对挂机状态下的检测 DTMF 来电号码有效，对通话中检测 DTMF 无效
- 非特殊情况不要修改该值

#### 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DTMFCALLIDNUM, 7);
```

## 2. 4. 10 CPC\_PARAM\_DIALSPEED DTMF 拨号时的时间长度

#### 功能描述 (Description)



DTMF 拨号时的时间长度

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALSPEED	
nValue	时间长度(单位:毫秒)	范围(45-128),建议 100 左右

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 该值越小, 拨号的速度就越快, 但太快时可能在某些交换机下引起不能正常拨号
- 默认为 75ms,非特殊需求, 建议用户不要修改该值
- 中国电信规定该值必须大于 45ms

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_DIALSPEED, 100);
```

## 2. 4. 11 CPC\_PARAM\_DIALSILENCE

### DTMF 拨号时两个号码之间静音长度

**功能描述 (Description)**

DTMF 拨号时两个号码之间静音长度

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALSILENCE	
nValue	时间长度(单位:毫秒)	范围(45-128),建议 100 左右

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 该值越小, 拨号的速度就越快, 但太快时可能在某些交换机下引起不能正常拨号
- 默认为 75ms,非特殊需求, 建议用户不要修改该值
- 中国电信规定该值必须大于 45ms

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_DIALSILENCE, 100);
```

## 2. 4. 12 CPC\_PARAM\_DIALVOL DTMF 拨号音量大小

**功能描述 (Description)**

DTMF 拨号拨号音量大小

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALVOL	



nValue	音量大小	范围(50-200),建议 100 左右
--------	------	----------------------

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音
- 非特殊需求一般用户不要修改该音量

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_DIALVOL, 100);
```

## 2. 4. 13 CPC\_PARAM\_RINGSILENCE

### 来电响铃后多少时间静音提示为未接电话

**功能描述 (Description)**

来电响铃后多少时间静音不再响铃提醒为未接电话

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGSILENCE	
nValue	来电响铃静音时长(单位: 毫秒)	范围(1000-10000),建议 6000 左右

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认时长为 5000ms
- 中国电信标准时间为 4000ms,所以一般设置为 4000ms 以上,实际时长可根据交换机的震铃最大间隔来调整。
- 当您在一个打入电话里误测到产生一个未接电话时可能是交换机震铃间隔时间超过设置的值, 就需要提高该时长
- 在这个时间内有继续响铃了就不会提示未接电话

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_RINGSILENCE, 5000);
```

## 2. 4. 14 CPC\_PARAM\_CONNECTSILENCE

### 通话时线路上连续多少时间静音后通知

**功能描述 (Description)**

通话时线路上连续多少时间静音后事件回调 CPCEvent\_Silence 事件通知用户

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_CONNECTSILENCE	





nValue	静音时长(单位: 毫秒)	范围(1000-600000)
--------	--------------	-----------------

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

-该事件可用在来电留言等无人值班时可用判断对方不响应,可以挂机释放线路

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_CONNECTSILENCE, 30000);
```

- 设置通道 0 通话时线路上连续 30 秒静音后事件回调通知用户 (CPCEvent\_Silence 事件)

## 2. 4. 15 CPC\_PARAM\_RINGBACKNUM

### 摘机后拨几个号码以上开始检测回铃声

**功能描述 (Description)**

摘机后拨几个号码以上开始检测回铃声

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGBACKNUM	
nValue	号码数量	范围(1-128)

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认为 2, 一般出局号码是一个数字, 使用 2 个开始检测, 避免拨第一个号码后检测到交换机的回铃而认为拨号结束而把后面拨的号码被忽略。  
- 非特殊情况用户不需要修改

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_RINGBACKNUM, 2);
```

## 2. 4. 16 CPC\_PARAM\_FLASHELAPSE 软件拍插簧间隔时间

**功能描述 (Description)**

使用软件拍插簧(CPC\_GENERAL\_STARTFLASH)间隔时间

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_FLASHELAPSE	
nValue	拍插簧间隔时间	范围((100-2000ms)

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 拍插簧实际为当线路在摘机状态时挂机后等待一定时间再摘机,这个挂机和摘机之



间的等待时间就为间隔时间，该时间必须在一定范围内交换机才能识别。

- 默认为 600ms
- 一般交换机的拍插簧检测范围在 300-1500ms 左右

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_FLASHELAPSE ,600);
```

## 2. 4. 17 CPC\_PARAM\_FLASHENDELAPSE 软拍插簧后等待多少时间回调

功能描述 (Description)

软件拍插簧结束后等待多少时间回调通知用户(BriEvent\_FlashEnd)

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_FLASHENDELAPSE	
nValue	等待延迟时间	范围(1-10000ms)

返回值 (Return Values)

大于 0 表示成功，其它表示失败

备注 (Remarks)

- 用户一般在接收到该消息后执行二次拨号转分机的功能
- 拍插簧结束后等待一定时间后交换机才会发送拨号音，这个时候才可以二次拨号，如果交换机还没拨号音，可能会引起二次拨号失败
- 如果有特殊交换机环境需要延迟比较长时间，根据实际情况设置
- 该值的效果也可以在二次拨的分机前面加 ‘,’ 逗号来代替。

如：本来交换机在拍插簧后需要延迟 2 秒才能拨号

方法一：设置该延迟时间为 2 秒

方法二：不设置该值，而是在二次拨号前面加两个逗号,如：“,,1011”也会起到同样的效果。

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_FLASHENDELAPSE,2000);
```

## 2. 4. 18 CPC\_PARAM\_RINGELAPSE 内线 Phone 震铃响的时间长度

功能描述 (Description)

内线 Phone 震铃(CPC\_GENERAL\_STARTRING)响的时间长度

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGELAPSE	
nValue	震铃时长	范围(100-2000ms)

返回值 (Return Values)

大于 0 表示成功，其它表示失败

备注 (Remarks)



-中国电信的标准震铃为 1 秒响 4 秒静音,所以该值标准应该为 1000.

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_RINGELAPSE,1000);
```

## 2. 4. 19 CPC\_PARAM\_RINGSILENCEELAPSE

### 内线 Phone 震铃后的静音时间长度

功能描述 (Description)

内线 Phone 震铃(CPC\_GENERAL\_STARTRING)后的静音时间长度

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGSILENCEELAPSE	
nValue	震铃静音时长	范围(100-10000ms)

返回值 (Return Values)

大于 0 表示成功, 其它表示失败

备注 (Remarks)

-中国电信的标准震铃为 1 秒响 4 秒静音,所以该值标准应该为 4000

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_RINGSILENCEELAPSE,4000);
```

## 2. 4. 20 CPC\_PARAM\_RINGTIMEOUT 内线 Phone 震铃超时次数

功能描述 (Description)

内线 Phone 震铃超时次数

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGTIMEOUT	
nValue	超时次数	范围(1-20)

返回值 (Return Values)

大于 0 表示成功, 其它表示失败

备注 (Remarks)

- 如果设置的震铃时间为 1 秒响 4 秒静音,那么一次就为 5 秒, 设置 12 次超时的话也就是在 12\*5=60 秒后就自动停止, 用户接收到(BriEvent\_RingTimeOut)事件
- 该值默认为 12.

例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_RINGTIMEOUT,12);
```



## 2. 4. 21 CPC\_PARAM\_RINGCALLIDTYPE 内线震铃时发送号码的方式

### 功能描述 (Description)

内线震铃时发送号码的方式

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGCALLIDTYPE	
nValue	DIALTYPE_DTMF 或 DIALTYPE_FSK	具体是一个数值

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

发送的号码可以在支持来电显示的话机上显示号码

### 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_RINGCALLIDTYPE, DIALTYPE_DTMF);  
- 设置通道 0 Phone 内线震铃内线震铃时使用 dtmf 方式发送号码
```

## 2. 4. 22 CPC\_PARAM\_REFUSEELAPSE 调用拒接来电时间间隔时间长度

### 功能描述 (Description)

调用拒接来电时间间隔时间长度

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_REFUSEELAPSE	
nValue	间隔时间	范围(100-5000ms)

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 当有来电时自动摘机后等待一段时间自动挂机完成拒接功能, 这个等待时间就为拒接间隔时长
- 间隔时间太短有可能拒接失败
- 非特殊需求用户不需要修改该参数

### 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_REFUSEELAPSE, 1000);  
- 设置通道 0 拒接来电时间间隔为 1 秒, 也就在摘机后等待 1 秒后自动挂机
```

## 2. 4. 23 CPC\_PARAM\_DIALNETIMEOUT 摘机后检测拨号音超时时间

### 功能描述 (Description)

摘机后检测拨号音超时自动开始拨号时间



## 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALTONETIMEOUT	
nValue	超时时间	范围(1000-5000ms)

## 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

## 备注 (Remarks)

-无

## 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_DIALTONETIMEOUT, 3000);
```

- 设置通道 0 摘机后 3 秒内没有检测拨号音就自动开始拨号

## 2. 4. 24 CPC\_PARAM\_MINCHKFLASHLAPSE

## 检测拍插簧时挂机至少维持时间

## 功能描述 (Description)

检测电话机是否拍插簧时, 电话机挂机后再再摘机前至少维持的时间

## 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_MINCHKFLASHLAPSE	
nValue	最小维持时间	范围(100-5000ms)

## 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

## 备注 (Remarks)

- 非特殊需求用户不需要修改该参数

## 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_MINCHKFLASHLAPSE, 500);
```

- 设置通道 0 检测电话机拍插簧时,电话机挂机后至少要维持 500ms 再摘机才算有效拍插簧, 也就是说挂机时间小于该值就不算拍插簧

## 2. 4. 25 CPC\_PARAM\_MAXCHKFLASHLAPSE

## 检测拍插簧时挂机最大维持时间

## 功能描述 (Description)

检测电话机是否拍插簧时, 电话机挂机后再摘机最大维持的时间

## 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_MAXCHKFLASHLAPSE	



nValue	最大维持时间	范围(100-5000ms)
--------	--------	----------------

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 非特殊需求用户不需要修改该参数

**例如 (Example)**

CPC\_SetParam(0, CPC\_PARAM\_MAXCHKFLASHLAPSE, 1500);

- 设置通道 0 检测电话机拍插簧时,电话机挂机后最多维持 1500ms 再摘机才算有效拍插簧，也就是说挂机时间大于该值就不算拍插簧

## 2. 4. 26 CPC\_PARAM\_HANGUPELAPSE 检测电话机挂机时至少维持时间

**功能描述 (Description)**

检测电话机挂机时至少需要维持的时间长度

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_HANGUPELAPSE	
nValue	最小维持时间	范围(100-5000ms)

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 建议该参数设置的不要比最大拍插簧时间小，避免检测到挂机后又检测到拍插簧
- 非特殊情况用户不需要修改该参数

**例如 (Example)**

CPC\_SetParam(0, CPC\_PARAM\_HANGUPELAPSE, 200);

- 设置通道 0 检测电话机是否挂机时，挂机状态至少维持 200ms 后才算挂机

## 2. 4. 27 CPC\_PARAM\_OFFHOOKELAPSE

### 检测电话机摘机时至少维持时间

**功能描述 (Description)**

检测电话机摘机时至少需要维持的时间长度

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_OFFHOOKELAPSE	
nValue	最小维持时间	范围(100-5000ms)

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 如果有话机出现误检测到摘机的情况，可以设置调高该值



- 非特殊情况用户不需要修改该参数

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_OFFHOOKELAPSE, 200);
```

- 设置通道 0 检测电话机是否摘机时, 摘机状态至少维持 200ms 后才算摘机
- 非特殊情况用户不需要修改该参数

**2. 4. 28 CPC\_PARAM\_RINGHIGHELAPSE 检测震铃响时至少维持的时间****功能描述 (Description)**

检测来电震铃时一次震铃时间至少需要维持的时间长度

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGHIGHELAPSE	
nValue	最小维持时间	范围(100-2000ms)

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 中国电信的震铃标准为 1 秒响 4 秒不响, 但有交换机把 1 秒响铃分成 3 部分, 如: 第一部分响 400ms, 在停 200ms, 再响 400ms, 这个时候就需要设置震铃检测小于 400ms 才能检测到。具体情况看不同的交换机有所区别,
- 默认为 200ms, 可以适应目前绝大部分交换机
- 非特殊情况用户不需要修改该参数

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_RINGHIGHELAPSE, 300);
```

- 设置通道 0 检测检测来电震铃时, 震铃状态至少维持 300ms 后才算一次震铃

**2. 4. 29 CPC\_PARAM\_RINGLOWELAPSE 检测震铃静音时至少维持的时间****功能描述 (Description)**

检测来电震铃时一次震铃结束后不响的时间至少需要维持的时间长度

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGLOWELAPSE	
nValue	最小维持时间	范围(100-2000ms)

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认为 100ms, 非特殊情况用户不需要修改该参数

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_RINGLOWELAPSE, 100);
```





- 设置通道 0 检测来电震铃时，一次震铃结束后不响的时间维持 100ms 回调用户一次震铃结束

## 2. 4. 30 CPC\_PARAM\_DIALTONERESULT 检测拨号音超时处理方式

### 功能描述 (Description)

检测拨号音超时后强制拨号还是提示拨号 dostartdial 失败

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALTONERESULT	
nValue	CHECKDIALTONE_TIMEOUTDIAL CHECKDIALTONE_FAILED	-强制拨号 -报告拨号失败不进行拨号

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

- 默认为 2(CHECKDIALTONE\_TIMEOUTDIAL)
- 系统在拨号时收到的 BriEvent\_DoStartDial 的 IResult 就为该值

设置为 3 时，当系统拨号前检测拨号音失败后系统将触发 BriEvent\_DoStartDial 事件，同时该事件里的 IResult 值为 3，停止拨号，用户接收到该事件后应该对应做后续的处理，如：挂机释放线路(系统拨号前会自动摘机)

设置为 2 时，当系统拨号前检测拨号音失败后系统将触发 BriEvent\_DoStartDial 事件，同时该事件里的 IResult 值为 2，系统立即强制开始拨号

### 例如 (Example)

```
CPC_SetParam(0,CPC_PARAM_DIALTONERESULT,3);
```

- 设置通道 0 拨号前如果检测拨号音失败就停止拨号

## 2. 4. 31 CPC\_PARAM\_ECHOTHRESHOLD 回音抵消为全回音的抵消倍数

### 功能描述 (Description)

回音抵消时,如果抵消倍数达到该值以山表示该语言为全回音,可考虑进行进一步减少增益

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_ECHOTHRESHOLD	
nValue	最低倍数	

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

- 默认为 20
- 如果设置为 1 表示不启用该功能



- 非特殊需求用户不需要修改该参数

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_ECHOTHRESHOLD, 10);
```

- 设置通道 0 回音抵消时抵消倍数达到 10 倍以上就表示该语音该全回音

## 2. 4. 32 CPC\_PARAM\_ECHODECVALUE

### 回音抵消后为全回音后再缩小倍数

**功能描述 (Description)**

回音抵消时,如果为全回音时把抵消后的语音再缩小一定倍数,尽量达到全静音效果

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_ECHODECVALUE	
nValue	缩小倍数	范围(1-N)

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认为 16
- 非特殊需求用户不需要修改该参数

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_ECHODECVALUE, 4);
```

- 设置通道 0 回音抵消时如果为全回音, 把抵消结果再缩小 4 倍.

## 2. 4. 33 CPC\_PARAM\_SIGSILENCEAM 静音检测幅度大小

**功能描述 (Description)**

分析信号音时如果语音幅度小于设置的该值就识别为静音

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_SIGSILENCEAM	
nValue	语音幅度大小	范围(1-32767)

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认为 64
- 非特殊需求用户不需要修改该参数

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_SIGSILENCEAM, 128);
```



## 2. 4. 34 CPC\_PARAM\_AM\_MIC 麦克风音量

### 功能描述 (Description)

设置插在设备上的麦克风音量等级大小

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_MIC	
nValue	音量等级大小	.范围(0-7)

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

音量等级越大，麦克风通话时对方听到您说话的的声音越大。

### 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_AM_MIC, 5);
```

- 设置通道 0 麦克风音量等级为 5.

## 2. 4. 35 CPC\_PARAM\_AM\_SPKOUT 耳机音量

### 功能描述 (Description)

设置插在设备上的耳机音量等级大小

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_SPKOUT	
nValue	音量等级大小	.范围(0-15)

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

- 音量越大，耳机听到的声音越大

### 例如 (Example)

```
CPC_SetParam(0, CPC_PARAM_AM_SPKOUT, 11);
```

- 设置通道 0 耳机输出音量等级为 11.

## 2. 4. 36 CPC\_PARAM\_AM\_LINEIN 线路录音音量

### 功能描述 (Description)

设置采集线路语音的音量等级大小

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	



uParamType	CPC_PARAM_AM_LINEIN	
nValue	音量等级大小	.范围(0-7)

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 音量等级越大，采集的声音越大,声音越大对回音抵消有一定影响
- 非特殊需求，用户不要调节该音量等级

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_AM_LINEIN, 3);
```

- 设置通道 0 采集线路语音音量等级为 3.

## 2. 4. 37 CPC\_PARAM\_AM\_LINEOUT 播放到线路语音音量

**功能描述 (Description)**

设置播放语音到线路的音量等级大小

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_LINEOUT	
nValue	音量等级大小	.范围(0-15)

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 音量等级越大，通话时播放语音到对方的声音越大，但对回音抵消效果有一定的影响
- 非特殊需求，用户不要调节该增益

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_AM_LINEOUT, 11);
```

## 2. 4. 38 CPC\_PARAM\_AM\_DOPLAY 喇叭音量

**功能描述 (Description)**

设置喇叭的音音量等级大小

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_DOPLAY	
nValue	音量等级大小	.范围(0-15)

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 音量等级越大，喇叭里的声音越大

**例如 (Example)**

```
CPC_SetParam(0, CPC_PARAM_AM_DOPLAY, 11);
```

- 设置通道 0 喇叭输出增益为 11.

## 2. 4. 39 CPC\_PARAM\_CITYCODEL 电话线所在的城市区号数值

**功能描述 (Description)**

设置接入在 USB 设备 LINE 口上的电话线所在的城市区号

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_CITYCODEL	
nValue	城市区号数值形式	如: "010"就为 10

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 只适合中国大陆
- 当前接入到 USB 设备上的线路所在的城市区号代码
- 如果您没有设置本地区号时:
  - 当您所在的城市区号为 010,这时如果您调用 API 拨打 01082888888 时, 当地电信会提示拨号错误, 因为在拨打本地号码时不能加本地的区号
- 如果您所在城市区号为 010, 并且您设置该本地区号为 "010":
  - 当您拨打 01082888888 时, 系统将自动过滤前面的 010, 直接拨打 82888888 就不会拨号失败
- 建议每个使用者准确设置该本地区号

**例如 (Example)**

```
CPC_SetParam(0,CPC_PARAM_CITYCODE,10);
```

- 设置通道 0 本地区号为"010"

## 2. 4. 40 CPC\_PARAM\_CITYCODE 电话线所在的城市区号字符

**功能描述 (Description)**

设置接入在 USB 设备 LINE 口上的电话线所在的城市区号

**参数介绍 (Parameters)**

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_CITYCODE	
nValue	城市区号的内存地址	

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 同 CPC\_PARAM\_CITYCODEL,建议使用 CPC\_PARAM\_CITYCODEL 来设置

**例如 (Example)**

```
CPC_SetParam(0,CPC_PARAM_CITYCODE,(CPCINT32)"010");  
-设置通道 0 本地区号为"010"
```

## 2.5 获取通道参数 (CPC\_GetParam)

### 功能描述 (Description)

获取通道参数

### 函数原型 (Function)

```
CPCINT32  CPC_GetParam(/*[in]*/CPCINT32  nChannelID,  
                        /*[in]*/CPCUINT32  uParamType);
```

### 返回值 (Return Values)

有符号 32bit 位,小于 0 返回失败,其它返回相应的参数值

### 参数描述 (Parameters description)

### 2.5.1 CPC\_PARAM\_BUSY 忙音检测灵敏度

#### 功能描述 (Description)

获取检测到连续几个忙音后回调

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_BUSY	

#### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_BUSY);
```

### 2.5.2 CPC\_PARAM\_DTMFLEVEL 检测 DTMF 时允许的性噪声比

#### 功能描述 (Description)

获取检测 DTMF 时允许的性噪声比,该值越小,检测灵敏度越高

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFLEVEL	



例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DTMFLEVEL);
```

### 2.5.3 CPC\_PARAM\_DTMFVOL 检测 DTMF 时有效的 DTMF 最小能量值

功能描述 (Description)

获取检测 DTMF 时允许的能量

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFVOL	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DTMFVOL);
```

### 2.5.4 CPC\_PARAM\_DTMFNUM 检测 DTMF 时允许的最小持续时长

功能描述 (Description)

获取检测 DTMF 时允许的持续时间次数

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFNUM	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DTMFNUM);
```

### 2.5.5 CPC\_PARAM\_DTMFLOWINHIGH

检测 DTMF 时低频不能超过高频的倍数

功能描述 (Description)

获取检测 DTMF 时低频能量不能超过高频能量的倍数

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFLOWINHIGH	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DTMFLOWINHIGH);
```





## 2.5.6 CPC\_PARAM\_DTMFHIGHINLOW

### 检测 DTMF 时高频不能超过低频的倍数

#### 功能描述 (Description)

获取检测 DTMF 时高频能量不能超过低频能量的倍数

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DTMFHIGHINLOW	

#### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DTMFHIGHINLOW);
```

## 2.5.7 CPC\_PARAM\_DIALSPEED DTMF 拨号时的时间长度

#### 功能描述 (Description)

获取 DTMF 拨号时的时间长度(单位:毫秒 ms)

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALSPEED	

#### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DIALSPEED);
```

## 2.5.8 CPC\_PARAM\_DIALSILENCE DTMF 拨号时两个号码之间静音长度

#### 功能描述 (Description)

获取 DTMF 拨号时两个号码之间静音长度(单位:毫秒 ms)

#### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALSILENCE	

#### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DIALSILENCE);
```

## 2.5.9 CPC\_PARAM\_DIALVOL DTMF 拨号音量大小

#### 功能描述 (Description)

获取拨号音量大小

#### 参数介绍 (Parameters)

参数名	参数值	描述
-----	-----	----



nChannelID	通道号	
uParamType	CPC_PARAM_DIALVOL	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DIALVOL);
```

## 2. 5. 10 CPC\_PARAM\_RINGSILENCE

### 来电响铃后多少时间静音提醒为未接电话

功能描述 (Description)

获取来电响铃后多少时间静音提醒为未接电话,中国电信标准时间为 4000ms

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGSILENCE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGSILENCE);
```

## 2. 5. 11 CPC\_PARAM\_CONNECTSILENCE

### 通话时线路上连续多少时间静音后通知

功能描述 (Description)

获取通话时线路上连续多少时间静音后事件回调通知用户

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_CONNECTSILENCE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_CONNECTSILENCE);
```

## 2. 5. 12 CPC\_PARAM\_RINGBACKNUM

### 摘机后拨几个号码以上开始检测回铃声

功能描述 (Description)

获取摘机后拨完几个数字以上开始检测回铃声

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGBACKNUM	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGBACKNUM);
```



## 2. 5. 13 CPC\_PARAM\_FLASHELAPSE 使用软件拍插簧间隔时间

### 功能描述 (Description)

获取软拍插簧间隔时间

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_FLASHELAPSE	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_FLASHELAPSE);
```

## 2. 5. 14 CPC\_PARAM\_FLASHENDELAPSE 软拍插簧后等待多少时间回调

### 功能描述 (Description)

获取软拍插簧结束后等待多少时间回调通知用户(BriEvent\_FlashEnd)

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_FLASHENDELAPSE	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_FLASHENDELAPSE, 1000);
```

## 2. 5. 15 CPC\_PARAM\_RINGELAPSE 内线 Phone 震铃响的时间长度

### 功能描述 (Description)

获取内线 Phone 震铃时时间长度

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGELAPSE	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGELAPSE);
```

## 2. 5. 16 CPC\_PARAM\_RINGSILENCEELAPSE

### 内线 Phone 震铃后的静音时间长度

### 功能描述 (Description)

获取内线 Phone 震铃后静音时间长度

### 参数介绍 (Parameters)

参数名	参数值	描述
-----	-----	----



nChannelID	通道号	
uParamType	CPC_PARAM_RINGSILENCEELAPSE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGSILENCEELAPSE);
```

## 2. 5. 17 CPC\_PARAM\_RINGTIMEOUT 内线 Phone 震铃超时次数

功能描述 (Description)

获取内线 Phone 震铃超时次数

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGTIMEOUT	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGTIMEOUT);
```

## 2. 5. 18 CPC\_PARAM\_RINGCALLIDTYPE 内线震铃时发送号码的方式

功能描述 (Description)

获取内线震铃时发送号码的方式

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGCALLIDTYPE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGCALLIDTYPE);
```

## 2. 5. 19 CPC\_PARAM\_REFUSEELAPSE 调用拒接来电时间间隔时间长度

功能描述 (Description)

获取调用拒接来电时间间隔时间长度

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_REFUSEELAPSE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_REFUSEELAPSE);
```



## 2. 5. 20 CPC\_PARAM\_DIALTONETIMEOUT 摘机后检测拨号音超时时间

### 功能描述 (Description)

获取摘机后检测拨号音超时自动开始拨号

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALTONETIMEOUT	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DIALTONETIMEOUT);
```

## 2. 5. 21 CPC\_PARAM\_MINCHKFLASHLAPSE

### 检测拍插簧时挂机至少维持时间

### 功能描述 (Description)

获取检测电话机是否拍插簧时，电话机挂机后再摘机至少维持的时间

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_MINCHKFLASHLAPSE	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_MINCHKFLASHLAPSE);
```

## 2. 5. 22 CPC\_PARAM\_MAXCHKFLASHLAPSE

### 检测拍插簧时挂机最多维持时间

### 功能描述 (Description)

获取检测电话机是否拍插簧时，电话机挂机后再摘机最多维持的时间

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_MAXCHKFLASHLAPSE	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_MAXCHKFLASHLAPSE);
```

## 2. 5. 23 CPC\_PARAM\_HANGUPELAPSE 检测电话机挂机时至少维持时间

### 功能描述 (Description)

获取检测电话机挂机时至少需要维持的时间长度

### 参数介绍 (Parameters)

参数名	参数值	描述
-----	-----	----



nChannelID	通道号	
uParamType	CPC_PARAM_HANGUPELAPSE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_HANGUPELAPSE);
```

## 2. 5. 24 CPC\_PARAM\_OFFHOOKELAPSE

### 检测电话机摘机时至少维持时间

功能描述 (Description)

获取检测电话机摘机时至少需要维持的时间长度

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_OFFHOOKELAPSE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_OFFHOOKELAPSE);
```

## 2. 5. 25 CPC\_PARAM\_RINGHIGHELAPSE 检测震铃响时至少维持的时间

功能描述 (Description)

获取检测来电震铃时一次震铃时间至少需要维持的时间长度

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGHIGHELAPSE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGHIGHELAPSE);
```

## 2. 5. 26 CPC\_PARAM\_RINGLOWELAPSE 检测震铃静音时至少维持的时间

功能描述 (Description)

获取检测来电震铃时一次震铃结束后不响的时间至少需要维持的时间长度

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_RINGLOWELAPSE	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_RINGLOWELAPSE);
```



## 2. 5. 27 CPC\_PARAM\_DIALTONERESULT 检测拨号音超时后处理方式

### 功能描述 (Description)

获取检测拨号音超时后是强制拨号还是提示拨号 dostartdial 失败

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_DIALTONERESULT	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_DIALTONERESULT);
```

## 2. 5. 28 CPC\_PARAM\_ECHOTHRESHOLD 回音抵消为全回音的抵消倍数

### 功能描述 (Description)

获取回音抵消时,如果抵消倍数达到该值以山表示该语音为全回音

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_ECHOTHRESHOLD	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_ECHOTHRESHOLD);
```

## 2. 5. 29 CPC\_PARAM\_ECHODECVALUE

### 回音抵消后为全回音后再缩小倍数

### 功能描述 (Description)

获取回音抵消时,如果为全回音时把抵消后的语音再缩小倍数

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_ECHODECVALUE	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_ECHODECVALUE);
```

## 2. 5. 30 CPC\_PARAM\_SIGSILENCEAM 静音幅度大小

### 功能描述 (Description)

获取分析信号音时如果语音幅度小于设置的该值就识别为静音

### 参数介绍 (Parameters)

参数名	参数值	描述
-----	-----	----





nChannelID	通道号	
uParamType	CPC_PARAM_SIGSILENCEAM	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_SIGSILENCEAM);
```

### 2. 5. 31 CPC\_PARAM\_AM\_MIC 麦克风音量

功能描述 (Description)

获取插在设备上的麦克风增益大小

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_MIC	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_AM_MIC);
```

### 2. 5. 32 CPC\_PARAM\_AM\_SPKOUT 耳机音量

功能描述 (Description)

获取插在设备的耳机增益大小

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_SPKOUT	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_AM_SPKOUT);
```

### 2. 5. 33 CPC\_PARAM\_AM\_LINEIN 线路录音音量

功能描述 (Description)

获取采集线路语音的增益大小

参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_LINEIN	

例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_AM_LINEIN);
```



## 2. 5. 34 CPC\_PARAM\_AM\_LINEOUT 播放到线路语音音量

### 功能描述 (Description)

获取播放语音到线路的增益大小

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_LINEOUT	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_AM_LINEOUT);
```

## 2. 5. 35 CPC\_PARAM\_AM\_DOPLAY 喇叭音量

### 功能描述 (Description)

获取喇叭的增益大小

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_AM_DOPLAY	

### 例如 (Example)

```
CPC_GetParam(0, CPC_PARAM_AM_DOPLAY);
```

## 2. 5. 36 CPC\_PARAM\_CITYCODEL 电话线所在的城市区号

### 功能描述 (Description)

获取设置的本地城市区号

### 参数介绍 (Parameters)

参数名	参数值	描述
nChannelID	通道号	
uParamType	CPC_PARAM_CITYCODEL	

### 例如 (Example)

```
CPC_GetParam(0,CPC_PARAM_CITYCODEL);
```

- 返回的数值最前面是不带 0，如果设置的 10，返回的也为 10,表示设置的区号为"010"

## 2. 6 USB 设备控制 (CPC\_SetDevCtrl)

### 功能描述 (Description)

控制 USB 设备状态参数



## 函数原型 ( Function )

```
CPCINT32  CPC_SetDevCtrl (/*[in]*/CPCINT32  nChannelID,  
                          /*[in]*/CPCUINT32  uCtrlType,  
                          /*[in]*/CPCINT32  nValue);
```

## 返回值 ( Return Values )

有符号 32bit 位,大于 0 表示成功,其它表示失败.

## 参数描述 (Parameters description)

### 2. 6. 1 CPC\_CTRL\_DOSHARE 通道共享模式

#### 功能描述 ( Description )

设置通道共享模式

#### 参数介绍 ( Parameters )

[in] *nChannelID*

控制的通道

[in] *nValue*

共享模式值

#### 返回值 ( Return Values )

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

该功能保留,目前用户不要使用

#### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_DOSHARE, SM_SENDVOICE);
```

### 2. 6. 2 CPC\_CTRL\_DOHOOK 软件摘挂机

#### 功能描述 ( Description )

软件摘挂机控制

#### 参数介绍 ( Parameters )

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 软挂机

1 — 软摘机

#### 返回值 ( Return Values )

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

无



例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_DOHOOK,1);
```

## 2. 6. 3 CPC\_CTRL\_DOPHONE 断开电话机跟 PSTN 线路连接

功能描述 (Description)

控制电话机是否断开跟 PSTN 线路连接

参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 电话机跟 PSTN 线路断开

1 — 电话机跟 PSTN 线路连接

返回值 (Return Values)

大于 0 表示成功, 其它表示失败

备注 (Remarks)

- 断开后话机将不能正常通话,来电时话机也不会震铃了
- 建议用户不要在正在拨号或正在收发传真时调用该功能

例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_DOPHONE,1);
```

## 2. 6. 4 CPC\_CTRL\_DOPLAY 打开/关闭喇叭

功能描述 (Description)

打开/关闭喇叭

参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭喇叭

1 — 打开喇叭

返回值 (Return Values)

大于 0 表示成功, 其它表示失败

备注 (Remarks)

- 关闭喇叭后所有线路或者播放的声音都不会从喇叭出来
- 打开喇叭后根据选择进入喇叭语音的通道的语音从喇叭出来

例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_DOPLAY,1);
```

## 2. 6. 5 CPC\_CTRL\_DOLINETOSPK 打开/关闭线路声音到耳机

功能描述 (Description)



打开/关闭线路声音到耳机

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭线路声音到耳机

1 — 打开线路声音到耳机

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 关闭后电话线上的声音将不会到耳机，也就通话时耳机听不到对方的声音
- 用耳机通话时必须打开
- 打开后耳机就可以听到线路上的声音，也就通话时可以听到对方的声音

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_DOLINETOSPK,1);
```

## 2. 6. 6 CPC\_CTRL\_DOPLAYTOSPK 打开/关闭播放的语音到耳机

**功能描述 (Description)**

打开/关闭播放的语音到耳机

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭播放的语音到耳机

1 — 打开播放的语音到耳机

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 关闭后耳机将听不到从电脑上播放的所有语音,如: 播放的语音文件，拨号音
- 打开后耳机将听到从电脑上播放的所有语音

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_DOPLAYTOSPK,1);
```

## 2. 6. 7 CPC\_CTRL\_DOMICTOLINE 打开/关闭麦克风到电话线

**功能描述 (Description)**

打开/关闭麦克风到电话线

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道



[in] *nValue*

- 0 — 关闭麦克风到电话线
- 1 — 打开麦克风到电话线

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

关闭后麦克风说话的声音将不会到线路上，也就在麦克风通话时对方听不到麦克风的说话声

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_DOMICTOLINE, 1);
```

## 2.6.8 CPC\_CTRL\_ECHO 打开/关闭回音抵消

**功能描述 (Description)**

打开/关闭回音抵消

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

- 0 — 关闭回音抵消
- 1 — 打开回音抵消

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 如果关闭该功能，所有回音抵消数据相关的参数都会无效，比如：文件录音时使用录回音抵消后的数据就会无效，录到的语音为对方声音和电脑播放的声音。
- 在系统放音状态时关闭可以节省 CPU 资源
- 非特殊需求一般用户不要关闭该功能

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_ECHO, 1);
```

## 2.6.9 CPC\_CTRL\_RECVFSK 打开/关闭接收 FSK 制式的来电号码

**功能描述 (Description)**

打开/关闭接收 FSK 制式的来电号码

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

- 0 — 关闭接收 FSK 制式的来电号码
- 1 — 打开接收 FSK 制式的来电号码

**返回值 (Return Values)**



大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 如果关闭该功能，系统将接收不到 FSK 制式的来电号码
- 关闭后可以节省 CPU 资源
- 非特殊需求一般用户不要关闭该功能.

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_RECVFSK,1);
```

## 2. 6. 10 CPC\_CTRL\_RECVDTMF 打开/关闭接收 DTMF

**功能描述 (Description)**

打开/关闭接收 DTMF

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭接收 DTMF

1 — 打开接收 DTMF

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 如果关闭该功能，包括 DTMF 制式来电号码和通话中按键
- 关闭后可以节省 CPU 资源
- 非特殊需求一般用户不要关闭该功能

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_RECVDTMF,1);
```

## 2. 6. 11 CPC\_CTRL\_RECVSIGN 打开/关闭信号音检测

**功能描述 (Description)**

打开/关闭信号音检测

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭信号音检测

1 — 打开信号音检测

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 如果关闭该功能，系统将检测不到拨号音，回铃音，忙音等所有通过语音检测的信号音





- 关闭后可以节省 CPU 资源
- 非特殊需求一般用户不要关闭该功能.

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_RECVSIGN, 1);
```

## 2. 6. 12 CPC\_CTRL\_WATCHDOG 打开/关闭设备看门狗

**功能描述 (Description)**

打开/关闭设备看门狗

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭设备看门狗

1 — 打开设备看门狗

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 关闭看门狗后, 如果 PC 跟设备一定时间内没有通信, [设备不会自动复位](#)
- 建议用户在调试程序状态时关闭看门狗, 因为当程序在调试断点状态时, PC 不会再跟设备交互数据, 会引起设备自动复位而导致设备不能再被控制, 需要重新打开设备后才能控制
- 打开看门狗后, 如果 PC 跟设备一定时间内没有通信, [设备将自动复位](#)
- 建议用户在发布的程序启动后打开看门狗, 避免程序或系统出现故障后引起设备不能正常工作

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_WATCHDOG, 1);
```

## 2. 6. 13 CPC\_CTRL\_PLAYMUX 选择到喇叭的语音线路

**功能描述 (Description)**

选择到喇叭的语音线路

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

1 — 选择设备的第 1 个通道的电话线的语音到喇叭

- 选择该喇叭语音通道后, 设备的第 1 个通道所有在电话线上的声音将全部到喇叭里 (喇叭被打开的情况下), 其它声音不会到喇叭

2 — 选择设备的第 1 个通道 PC 播放的语音到喇叭

- 选择该喇叭语音通道后, 设备的第 1 个通道所有从



PC 上播放的语音将全部到喇叭里,(喇叭被打开的情况下),其它声音不会到喇叭

0 一选择设备的第 2 个通道的电话线的语音到喇叭

- 选择该喇叭语音通道后,设备的第 2 个通道所有在电话线上的声音将全部到喇叭里 (喇叭被打开的情况下),其它声音不会到喇叭.

备注:该类型只对一个芯片有 2 通道时有用,也就是对该设备通道具有断开电话机功能(DOPHONE)的设备有效

如:IA1/IB1,IA2/IB2,IA3/IB3

3 一选择设备的第 2 个通道 PC 播放的语音到喇叭

- 选择该喇叭语音通道后,设备的第 2 个通道所有从 PC 上播放的语音将全部到喇叭里,(喇叭被打开的情况下),其它声音不会到喇叭

备注:该类型只对一个芯片有 2 通道时有用,也就是对该设备通道具有断开电话机功能(DOPHONE)的设备有效

如:IA1/IB1,IA2/IB2,IA3/IB3

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_PLAYMUX, 2);
```

## 2.6.14 CPC\_CTRL\_PLAYTOLINE 打开/关闭播放的语音到线路

#### 功能描述 (Description)

打开/关闭播放的语音到线路

#### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 一 关闭播放的语音到线路

1 一 打开播放的语音到线路

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

#### 备注 (Remarks)

- 关闭后所有 PC 播放的声音将不会到线路上,引起不能正常软件拨号/不能通话时放音给对方。

- 打开后 PC 上所有播放的语音就可以到线路上。(线路已经软摘机的情况下)

- 非特殊需求一般用户不要关闭该功能

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_PLAYTOLINE ,1);
```

## 2. 6. 15 CPC\_CTRL\_SELECTLINEIN 选择输入的线路通道

**功能描述 (Description)**

选择输入的线路通道

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

用户不要使用该功能，设置后可能会引起设备工作不正常

**例如 (Example)**

无

## 2. 6. 16 CPC\_CTRL\_SELECTADCIN 选择输入的为线路语音还是麦克风语音

**功能描述 (Description)**

选择输入的为线路语音还是麦克风语音

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 选择麦克风语音输入到 PC

1 — 选择线路语音输入到 PC

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 选择麦克风语音输入到 PC 参数后，PC 上采集的全部为来自设备麦克风插口上的语音，所有线路的语音都将被丢弃，也就检测不到线路上的任何信号音和来电号码，当您需要独立对麦克风录音时使用，其它时候不要使用。如：需要临时使用设备上的 麦克风录个提示语音文件。

- 选择线路语音输入到 PC 参数后，PC 上采集的全部为来自线路上，如果使用麦克风通话时，打开麦克风后，麦克风声音自动到线路上，这时从线路上采集到的也同时有麦克风声音。

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_SELECTADCIN ,1);
```



## 2. 6. 17 CPC\_CTRL\_PHONEPOWER 打开/关闭给电话机独立供电

### 功能描述 (Description)

打开/关闭给电话机独立供电

### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭给电话机独立供电

1 — 打开给电话机独立供电

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 不给电话机供电,使用 `CPC_DoPhone` 断开电话机跟线路连接后,话机将没有任何供电来源,所有对电话机的操作都会无效,也检测不到电话机摘挂机。
- 给电话机独立供电后,使用 `CPC_DoPhone` 断开电话机跟线路连接后,设备将自动给电话机供电了,让话机处于工作状态,这时电话机已经不能进行普通的线路通话,但还能检测到电话机的摘挂机,还可以对电话机进行独立语音录放,也就是可以从话机话柄录音,PC 放音到话机话筒上。

### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_PHONEPOWER,1);
```

## 2. 6. 18 CPC\_CTRL\_RINGPOWER 打开/关闭内线供电震铃

### 功能描述 (Description)

打开/关闭内线供电震铃

### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭内线供电震铃

1 — 打开内线供电震铃

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 关闭内线供电震铃后,内线 `phone` 口停止模拟响铃。
- 打开内线供电震铃后,内线 `phone` 口会模拟来电响
- 非特殊需求建议用户不要使用该功能
- 如果要内线模拟响铃建议使用 `CPC_GENERAL_STARTRING`

### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_RINGPOWER,1);
```



## 2. 6. 19 CPC\_CTRL\_LEDPOWER 打开/关闭 LED 指示灯

### 功能描述 (Description)

打开/关闭 LED 指示灯

### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭 LED 指示灯

1 — 打开 LED 指示灯

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_LEDPOWER,1);
```

## 2. 6. 20 CPC\_CTRL\_LINEOUT 打开/关闭线路输出功能

### 功能描述 (Description)

打开/关闭线路输出功能

### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭线路输出功能

1 — 打开线路输出功能

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 关闭后所有播放到线路的语音和麦克风声音到线路都将失效。
- 非特殊需求一般用户不要关闭该功能
- 打开后所有播放到线路的语音((同时打开播放语音到线路)和麦克风声音(同时打开麦克风语音到线路)才能到线路上

### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_LINEOUT,1);.
```

## 2. 6. 21 CPC\_CTRL\_UPLOAD 打开/关闭设备数据上传功能

### 功能描述 (Description)

打开/关闭设备数据上传功能



#### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭设备数据上传功能

1 — 打开设备数据上传功能

#### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

#### 备注 (Remarks)

- 关闭后将接收不到设备语音数据。
- 非特殊需求一般用户不要关闭该功能

#### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_UPLOAD, 1);
```

## 2.6.22 CPC\_CTRL\_DOWNLOAD 打开/关闭设备数据下载功能

#### 功能描述 (Description)

打开/关闭设备数据下载功能

#### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭设备数据下载功能

1 — 打开设备数据下载功能

#### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

#### 备注 (Remarks)

- 关闭后将不能发送语音/拨号/所有控制到设备。
- 非特殊需求一般用户不要关闭该功能。

#### 例如 (Example)

```
CPC_SetDevCtrl(0, CPC_CTRL_DOWNLOAD, 1);
```

## 2.6.23 CPC\_CTRL\_POLARITY 打开/关闭级性反转检测

#### 功能描述 (Description)

打开/关闭级性反转检测呼叫后对方摘挂机

#### 参数介绍 (Parameters)

[in] *nChannelID*

控制的通道

[in] *nValue*

0 — 关闭级性反转摘挂机检测



## 1 一 打开级性反转摘挂机检测

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 对支持该功能的设备才有效
- 反极信号是模拟线路中唯一可靠的判断对方摘机的方法，只是一方面需要开通才有该功能，另一方面在有些地方，即使申请也未必受理，因为该功能原本是针对公用电话计费的。
- 没有该功能，利用信号音做判断依据，但可靠性不能做到 100%，尤其是在被叫方为彩铃时，检测率很低

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_POLARITY, 1);
```

## 2.6.24 CPC\_CTRL\_READFRAMENUM 设置一次读取 USB 帧数量

**功能描述 (Description)**

一次请求读取的 USB 帧数量

**参数介绍 (Parameters)**

[in] *nChannelID*

控制的通道

[in] *nValue*

帧数(范围 1-30)

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 帧数越大占用 CPU 会越小，来电、摘机等事件延迟也就越大，一帧为 4ms,最大 30 帧
- 默认为 4 帧

**例如 (Example)**

```
CPC_SetDevCtrl(0, CPC_CTRL_READFRAMENUM, 8);
```

## 2.7 获取 USB 设备控制状态 (CPC\_GetDevCtrl)

**功能描述 (Description)**

获取控制 CC301 设备控制状态

**函数原型 (Function)**

```
CPCINT32 CPC_GetDevCtrl (/*[in]*/CPCINT16 nChannelID,
                          /*[in]*/CPCUINT32 uCtrlType);
```



### 返回值 (Return Values)

---

有符号 32bit 位,小于 0 获取失败,其它返回状态信息。

### 参数描述 (Parameters description)

---

## 2.7.1 CPC\_CTRL\_DOSHARE 通道共享模式

### 功能描述 (Description)

检测通道是否共享

### 返回值 (Return Values)

- 小于 0 — 检测失败
- 0 — 通道没有共享
- 1 — 通道已经共享

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_DOSHARE);
```

## 2.7.2 CPC\_CTRL\_DOHOOK 软件摘挂机

### 功能描述 (Description)

检测软件摘挂机状态

### 返回值 (Return Values)

- 小于 0 — 检测失败
- 0 — 通道没有软摘机
- 1 — 通道已经软摘机

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_DOHOOK);
```

## 2.7.3 CPC\_CTRL\_DOPHONE 电话机跟 PSTN 线路连接是否断开

### 功能描述 (Description)

检测电话机是否跟 PSTN 线路连接

### 返回值 (Return Values)

- 小于 0 — 检测失败
- 0 — 电话机跟 PSTN 线路已断开
- 1 — 电话机跟 PSTN 线路已连接



**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_DOPHONE );
```

## 2. 7. 4 CPC\_CTRL\_DOPLAY 打开/关闭喇叭

**功能描述 (Description)**

检测喇叭是否打开

**返回值 (Return Values)**

小于 0	— 检测失败
0	— 喇叭已关闭
1	— 喇叭已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_DOPLAY);
```

## 2. 7. 5 CPC\_CTRL\_DOLINETOSPK 打开/关闭线路声音到耳机

**功能描述 (Description)**

检测线路声音到耳机是否打开

**返回值 (Return Values)**

小于 0	— 检测失败
0	— 线路声音到耳机已关闭
1	— 线路声音到耳机已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_DOLINETOSPK );
```

## 2. 7. 6 CPC\_CTRL\_DOPLAYTOSPK 打开/关闭播放的语音到耳机

**功能描述 (Description)**

检测播放的语音到耳机是否打开

**返回值 (Return Values)**

小于 0	— 检测失败
0	— 播放的语音到耳机已关闭
1	— 播放的语音到耳机已打开

**备注 (Remarks)**

无

**例如 (Example)**



```
CPC_GetDevCtrl(0, CPC_CTRL_DOPLAYTOSPK );
```

## 2. 7. 7 CPC\_CTRL\_DOMICTOLINE 打开/关闭麦克风到电话线

### 功能描述 (Description)

检测麦克风语音到线路是否打开

### 返回值 (Return Values)

- 小于 0 — 检测失败
- 0 — 麦克风语音到线路已关闭
- 1 — 麦克风语音到线路已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_DOMICTOLINE );
```

## 2. 7. 8 CPC\_CTRL\_ECHO 打开/关闭回音抵消

### 功能描述 (Description)

检测回音抵消是否打开

### 返回值 (Return Values)

- 小于 0 — 检测失败
- 0 — 回音抵消已关闭
- 1 — 回音抵消已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_ECHO );
```

## 2. 7. 9 CPC\_CTRL\_RECVFSK 打开/关闭接收 FSK 制式的来电号码

### 功能描述 (Description)

检测接收 FSK 制式的来电号码是否打开

### 返回值 (Return Values)

- 小于 0 — 检测失败
- 0 — 接收 FSK 制式的来电号码已关闭
- 1 — 接收 FSK 制式的来电号码已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_RECVFSK );
```



## 2. 7. 10 CPC\_CTRL\_RECVDTMF 打开/关闭接收 DTMF

### 功能描述 (Description)

检测接收 DTMF 是否打开

### 返回值 (Return Values)

小于 0    — 检测失败  
0           — 接收 DTMF 已关闭  
1           — 接收 DTMF 已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_RECVDTMF );
```

## 2. 7. 11 CPC\_CTRL\_RECVSIGN 打开/关闭信号音检测

### 功能描述 (Description)

检测信号音分析是否打开

### 返回值 (Return Values)

小于 0    — 检测失败  
0           — 信号音分析已关闭  
1           — 信号音分析已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_RECVSIGN );
```

## 2. 7. 12 CPC\_CTRL\_WATCHDOG 打开/关闭设备看门狗

### 功能描述 (Description)

检测看门狗是否打开

### 返回值 (Return Values)

小于 0    — 检测失败  
0           — 看门狗已关闭  
1           — 看门狗已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_WATCHDOG );
```



## 2. 7. 13 CPC\_CTRL\_PLAYMUX 选择到喇叭的语音线路

### 功能描述 (Description)

获取进入喇叭的语音线路

### 返回值 (Return Values)

小于 0     — 检测失败  
其它       — 进入喇叭的语音线路

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_PLAYMUX);
```

## 2. 7. 14 CPC\_CTRL\_PLAYTOLINE 打开/关闭播放的语音到线路

### 功能描述 (Description)

检测播放语音到线路是否打开

### 返回值 (Return Values)

小于 0     — 检测失败  
0           — 播放语音到线路已关闭  
1           — 播放语音到线路已打开

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_PLAYTOLINE);
```

## 2. 7. 15 CPC\_CTRL\_SELECTLINEIN 选择输入的线路通道

### 功能描述 (Description)

获取输入的语音线路

### 返回值 (Return Values)

小于 0     — 检测失败  
其它       — 输入的语音线路

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_SELECTLINEIN);
```

## 2. 7. 16 CPC\_CTRL\_SELECTADCIN 选择输入的为线路语音还是麦克风语音

### 功能描述 (Description)

检测输入的为线路 PSTN 线路还是麦克风

**返回值 (Return Values)**

- 小于 0 — 检测失败
- 0 — 麦克风语音输入
- 1 — 线路语音输入

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_SELECTADCIN );
```

## 2. 7. 17 CPC\_CTRL\_PHONEPOWER 打开/关闭给电话机独立供电

**功能描述 (Description)**

检测给电话机独立供电是否打开

**返回值 (Return Values)**

- 小于 0 — 检测失败
- 0 — 电话机独立供电已关闭
- 1 — 电话机独立供电已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_PHONEPOWER );
```

## 2. 7. 18 CPC\_CTRL\_RINGPOWER 打开/关闭内线供电震铃

**功能描述 (Description)**

检测内线供电震铃是否打开

**返回值 (Return Values)**

- 小于 0 — 检测失败
- 0 — 内线供电震铃已关闭
- 1 — 内线供电震铃已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_RINGPOWER );
```

## 2. 7. 19 CPC\_CTRL\_LEDPOWER 打开/关闭 LED 指示灯

**功能描述 (Description)**

检测 LED 指示灯是否打开

**返回值 (Return Values)**

- 小于 0 — 检测失败
- 0 — LED 指示灯已关闭



1 — LED 指示灯已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_LEDPOWER);
```

## 2.7.20 CPC\_CTRL\_LINEOUT 打开/关闭线路输出功能

**功能描述 (Description)**

检测线路输出功能是否打开

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 线路输出功能已关闭

1 — 线路输出功能已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_LINEOUT);
```

## 2.7.21 CPC\_CTRL\_UPLOAD 打开/关闭设备数据上传功能

**功能描述 (Description)**

检测设备数据上传功能是否打开

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 设备数据上传功能已关闭

1 — 设备数据上传功能已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_UPLOAD);
```

## 2.7.22 CPC\_CTRL\_DOWNLOAD 打开/关闭设备数据下载功能

**功能描述 (Description)**

检测设备数据下载功能是否打开

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 设备数据下载功能已关闭

1 — 设备数据下载功能已打开

**备注 (Remarks)**

无



**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_DOWNLOAD );
```

## 2. 7. 23 CPC\_CTRL\_POLARITY 打开/关闭级性反转检测

**功能描述 (Description)**

获取一次请求读取的 USB 帧数量

**返回值 (Return Values)**

小于 0 — 获取失败

其它 — 一次请求的 USB 帧数量

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_READFRAMENUM );
```

## 2. 7. 24 CPC\_CTRL\_POLARITY 获取一次 USB 请求的帧数量

**功能描述 (Description)**

级性反转摘挂机检测是否打开

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 级性反转摘挂机检测已关闭

1 — 级性反转摘挂机检测已打开

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_POLARITY);
```

## 2. 7. 25 CPC\_CTRL\_PHONE 检测本地电话机摘挂机状态

**功能描述 (Description)**

检测本地电话机摘挂机状态

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 电话机挂机状态

1 — 电话机摘机状态

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_GetDevCtrl(0, CPC_CTRL_PHONE );
```



## 2. 7. 26 CPC\_CTRL\_MICIN 检测麦克风插入状态

### 功能描述 (Description)

检测麦克风插入状态

### 返回值 (Return Values)

小于 0    — 检测失败  
0           — 麦克风没有插入  
1           — 麦克风已插入

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_MICIN);
```

## 2. 7. 27 CPC\_CTRL\_RINGTIMES 获取当前来电已经响铃的次数

### 功能描述 (Description)

获取当前来电已经响铃的次数

### 返回值 (Return Values)

小于 0    — 检测失败  
其它       — 来电已经响铃的次数

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_RINGTIMES);
```

## 2. 7. 28 CPC\_CTRL\_RINGSTATE 检测来电响铃/静音状态

### 功能描述 (Description)

检测来电响铃状态

### 返回值 (Return Values)

小于 0    — 检测失败  
0           — 来电正在静音状态  
1           — 来电正在响铃状态

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_GetDevCtrl(0, CPC_CTRL_RINGSTATE);
```





## 2.8 播放语音文件(CPC\_PlayFile)

### 功能描述 (Description)

播放语音文件到设备(建议用户不要在等待接收来电号码前播放语音到线路上,避免影响接收来电号码)

### 函数原型 (Function)

```
CPCINT32 CPC_PlayFile (/*[in]*/CPCINT16 nChannelID,  
/*[in]*/CPCUINT32 uPlayType,  
/*[in]*/CPCINT32 nValue,  
/*[in]*/ CPCINT32 nValueEx,  
/*[in]*/ CPCCHAR8* pValue);
```

### 返回值 (Return Values)

有符号 32bit 位

### 参数描述 (Parameters description)

### 2.8.1 CPC\_PLAY\_FILE\_START 开始播放语音文件

#### 功能描述 (Description)

开始播放语音文件

#### 参数介绍 (Parameters)

[in] nChannelID

播放语音文件的通道

[in] nValue — 保留

[in] nValueEx

语音文件的播放属性

PLAYFILE\_MASK\_REPEAT - 表示是否重复播放

PLAYFILE\_MASK\_PAUSE - 表示启动后先暂停

[in] pValue

需要播放的 wav 语音文件所在路径。只支持系统 wave 格式的语音文件, 不支持 mp3/wmv/rm 等其它语音格式

- 使用绝对路径。如: "c:\a.wav"。

- 使用相对路径。如: "a.wav", 系统将自动定为该文件为本进程所在目录下的 a.wav。

- 该播放文件路径支持网络共享文件路径:

如: [\\192.168.0.1\wav\a.wav](http://192.168.0.1/wav/a.wav)

#### 返回值 (Return Values)

- 小于等于 0 表示播放失败, 其它返回本次播放的唯一句柄



- 以后所有对该播放控制都必须使用该句柄,所以用户必须保存该句柄

**备注 (Remarks)**

- 测试语音文件是否可以支持播放系统 wave 格式, 可以使用 windows 自带的录音机进行播放, 如果录音机支持, 那该文件就为系统 wave 文件。
- 一个通道可以同时进行任意多个文件播放, 多个文件一起播放时系统自动进行混音后播放
- 建议用户使用绝对路径的方式来播放
- 如果想播放声音到设备的喇叭出声, 请打开喇叭 `CPC_CTRL_DOPLAY`
- 如果想播放声音到线路上, 请调用软摘机功能 `CPC_CTRL_DOHOOK`
- 目前不支持播放远程 HTTP 协议和 FTP 协议语音流, 所以如下格式的语音文件路径不能被播放: 如: `http://192.168.0.1/a.wav`

**例如 (Example)**

绝对路径:

```
CPC_SetDevCtrl(1,CPC_CTRL_DOPLAY,1);
```

- 打开喇叭, 让播放声音从喇叭出来

```
CPCINT32 i32PlayFileHandle =CPC_PlayFile(1,
    CPC_PLAY_FILE_START,
    0,
    PLAYFILE_MASK_REPEAT| PLAYFILE_MASK_PAUSE,
    "c:\a.wav");
```

- 通道 1 循环播放 `c:\a.wav`, 并且启动后 `PLAYFILE_MASK_PAUSE` 标记先暂停播放, 等待调用 `CPC_PLAY_FILE_RESUME` 开始播放, 如果不用该标记, 一启动就开始播放

相对路径:

```
CPCINT32 i32PlayFileHandle= CPC_PlayFile(1,
    CPC_PLAY_FILE_START,
    0,
    0,
    "a.wav");
```

- 通道 1 开始播放本程序目录下的 `a.wav`。

## 2.8.2 CPC\_PLAY\_FILE\_SETCALLBACK 设置播放文件回调函数

**功能描述 (Description)**

设置播放文件回调函数

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 `CPC_PLAY_FILE_START` 返回的句柄

[in] *nValueEx*

回调函数的用户自定义参数, 在回调时返回给用户层

[in] *pValue*



回调函数指针地址.查看 PCallBack\_PlayFile 的原型介绍

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 在 C++中用户自定义参数一般使用类指针(this)
- 回调函数的原型请查看回调函数(PCallBack\_PlayFile)定义

**例如 (Example)**

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_SETCALLBACK,  
              i32PlayFileHandle,  
              (CPCINT32)this,  
              PlayFileCallbackFunc);
```

- PlayFileCallbackFunc:为 PCallBack\_PlayFile 类型回调函数地址。
- this 为用户自定义参数, 在 C++一般使用类指针(this)。

## 2. 8. 3 CPC\_PLAY\_FILE\_SETVOLUME 设置播放文件音量

**功能描述 (Description)**

设置播放文件音量

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx*

音量值。范围(0-10000)

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

**例如 (Example)**

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_SETVOLUME,  
              i32PlayFileHandle,  
              200,  
              0);
```



## 2. 8. 4 CPC\_PLAY\_FILE\_GETVOLUME 获取播放文件音量

### 功能描述 (Description)

获取播放文件音量

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，其它返回播放文件的音量值

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_SETVOLUME,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 5 CPC\_PLAY\_FILE\_PAUSE 暂停播放文件

### 功能描述 (Description)

暂停播放文件

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_PAUSE,
              i32PlayFileHandle,
              0,
              0);
```



- 暂停通道 1 播放 i32PlayFileHandle 句柄的文件

## 2.8.6 CPC\_PLAY\_FILE\_RESUME 恢复播放文件

### 功能描述 (Description)

恢复播放文件

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayFile(1,  
  
              CPC_PLAY_FILE_RESUME,  
              i32PlayFileHandle,  
              0,  
              0);
```

- 恢复通道 1 播放 i32PlayFileHandle 句柄的文件

## 2.8.7 CPC\_PLAY\_FILE\_ISPAUSE 检测是否已暂停播放

### 功能描述 (Description)

检测是否已暂停播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 — 检测失败

0 — 表示没有暂停播放

1 — 表示已经暂停播放

### 备注 (Remarks)

无

**例如 (Example)**

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_ISPAUSE,
              i32PlayFileHandle,
              0,
              0);
```

## 2.8.8 CPC\_PLAY\_FILE\_SETREPEAT 设置是否循环播放

**功能描述 (Description)**

设置是否循环播放

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx*

是否循环播放

0 — 不使用循环播放

1 — 使用循环播放

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

该设置必须在文件播放完毕之前设置

**例如 (Example)**

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_SETREPEAT,
              i32PlayFileHandle,
              0,
              0);
```

- 设置通道 1 播放 i32PlayFileHandle 句柄进行循环播放

## 2.8.9 CPC\_PLAY\_FILE\_ISREPEAT 检测是否在循环播放

**功能描述 (Description)**

检测是否在循环播放

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄



[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 表示没有使用循环播放

1 — 表示已经使用循环播放

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_ISREPEAT,  
              i32PlayFileHandle,  
              0,  
              0);
```

## 2. 8. 10 CPC\_PLAY\_FILE\_SEEKTO 跳转到某个时间开始播放

**功能描述 (Description)**

跳转到某个时间开始播放

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx*

跳转到某个播放时间，从文件开始计算的毫秒数

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败

**备注 (Remarks)**

该功能一般配合 CPC\_PLAY\_FILE\_TOTALLEN 获取总共时间后再使用

**例如 (Example)**

```
CPCINT32 iTotLen=CPC_PlayFile(1,  
                               CPC_PLAY_FILE_TOTALLEN,  
                               i32PlayFileHandle,  
                               0,  
                               0);
```

- 获取语音文件总共时长,单位: 毫秒

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_SEEKTO,  
              i32PlayFileHandle,  
              iTotLen/2,  
              0);
```



- 把通道 1 播放的 i32PlayFileHandle 文件跳转到中间时刻(iTotalLen/2) 开始播放

## 2.8.11 CPC\_PLAY\_FILE\_SETREPEATTIMEOUT 设置循环播放超时次数

### 功能描述 (Description)

设置循环播放超时次数

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx*

循环播放超时次数.范围(0-65535)

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

- 该参数只有当使用循环播放后(*CPC\_PLAY\_FILE\_ISREPEAT* 为 1 时)才有效
- 默认该值为 0,表示循环次数不限制
- 播放循环到该次数后系统就会立即停止播放
- 一般该参数要在播放语音开始后立即设置

### 例如 (Example)

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_SETREPEATTIMEOUT,  
              i32PlayFileHandle,  
              100,  
              0);
```

- 把通道 1 播放的 i32PlayFileHandle 循环播放的次数 100 次为超时

## 2.8.12 CPC\_PLAY\_FILE\_GETREPEATTIMEOUT 获取循环播放超时次数

### 功能描述 (Description)

获取循环播放超时次数

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)





小于 0 表示失败, 其它返回循环播放的超时次数

**备注 (Remarks)**

默认该值为 0, 表示循环次数不限制

**例如 (Example)**

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_GETREPEATTIMEOUT,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 13 CPC\_PLAY\_FILE\_SETPLAYTIMEOUT 设置播放超时时长

**功能描述 (Description)**

设置播放超时时长

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx*

播放超时的时长, 范围(0-0xFFFFFFFF) 单位: 毫秒

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 默认该值为 0, 表示没有播放时长超时限制
- 当总共播放的时长达到该值时, 系统自动停止播放。
- 当该超时时间跟循环播放次数超时一起使用时, 哪个先达到超时条件后就立即停止。如: 播放一个语音 10 秒的文件, 如果启动了循环播放, 而且设置次数为 10 次, 同时设置了播放超时为 50 秒, 那么系统将在总共播放 50 秒后停止播放, 因为循环播放的超时为 10 次\*10 秒=100 秒, 在播放 50 秒后系统首先满足播放 50 秒超时的条件后就停止。

**例如 (Example)**

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_SETPLAYTIMEOUT,
              i32PlayFileHandle,
              3000,
              0);
```

- 把通道 1 播放的 i32PlayFileHandle 播放 3000 毫秒 (3 秒) 后超时



## 2. 8. 14 CPC\_PLAY\_FILE\_GETPLAYTIMEOUT 获取播放超时时长

### 功能描述 (Description)

获取播放超时时长

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败, 其它返回播放超时时长

### 备注 (Remarks)

- 默认该值为 0, 表示没有播放时长超时限制

### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_GETPLAYTIMEOUT,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 15 CPC\_PLAY\_FILE\_TOTALLEN 播放的语音文件总共时长

### 功能描述 (Description)

播放的语音文件总共时长

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败, 其它返回语音文件的总共时长.(单位: 毫秒)

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_TOTALLEN,
              i32PlayFileHandle,
              0,
              0);
```



## 2. 8. 16 CPC\_PLAY\_FILE\_CURSEEK 当前播放时间的离文件开始偏移时长

### 功能描述 (Description)

当前播放时间的离文件开始偏移时长

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，当前播放时间的离文件开始偏移时长.(单位：毫秒)

### 备注 (Remarks)

- 播放一个 10000 毫秒长的语音文件，播放到中间时，偏移时长为 5000 毫秒。
- 该值不考虑循环播放的时长，也就是说即使第 10 次开始循环播放，播放到语音文件中间时，返回的偏移时长也是 5000 毫秒。

### 例如 (Example)

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_CURSEEK,  
              i32PlayFileHandle,  
              0,  
              0);
```

## 2. 8. 17 CPC\_PLAY\_FILE\_ELAPSE 已经总共播放语音文件的时长

### 功能描述 (Description)

已经总共播放语音文件的时长

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，已经总共播放语音文件的时长.(单位：毫秒)

### 备注 (Remarks)

- 该值返回该语音文件总共播放语音的时长，包括循环播放的时长，但不包括暂停/跳转的时长。

如：播放一个 10000 毫秒的语音文件，进行循环播放，在播放完 3 次后暂



停播放 1000 毫秒，再继续播放，又播放 4 次，那返回的时长为：  
 $3 \times 10000 + 4 \times 10000 = 70000$  毫秒，中间暂停的时长被忽略。

如：播放一个 10000 毫秒的语音文件，播放到中间 5000 毫秒后如果用户执行跳转到第 8000 毫秒开始播放，再播放 1000 毫秒后返回的值为  $5000 + 1000 = 6000$  毫秒，忽略中间跳转的时间。

如：播放一个 10000 毫秒的语音文件，播放到中间 5000 毫秒后如果用户执行跳转到第 2000 毫秒开始播放，再播放 1000 毫秒后返回的值为  $5000 + 1000 = 6000$  毫秒，忽略中间跳转的时间。

#### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_ELAPSE,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 18 CPC\_PLAY\_FILE\_ISPLAY 检查播放句柄是否有效

#### 功能描述 (Description)

检查播放句柄是否有效

#### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 `CPC_PLAY_FILE_START` 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0	—	检测失败
0	—	该播放句柄已经无效
1	—	该播放句柄已经有效

#### 备注 (Remarks)

开始播放后，如果没有播放完毕或者调用停止播放，不管是否暂停了播放，该句柄将一直有效。

#### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_ISPLAY,
              i32PlayFileHandle,
              0,
              0);
```



## 2. 8. 19 CPC\_PLAY\_FILE\_ENABLEAGC 打开/关闭自动增益播放

### 功能描述 (Description)

打开/关闭自动增益播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx*

0 — 关闭自动增益模式播放

1 — 打开自动增益模式播放

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败。

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayFile(1,  
              CPC_PLAY_FILE_ENABLEAGC,  
              i32PlayFileHandle,  
              1,  
              0);
```

- 打开自动增益模式播放

## 2. 8. 20 CPC\_PLAY\_FILE\_ISENABLEAGC 检测是否使用了自动增益播放

### 功能描述 (Description)

检测是否使用了自动增益播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 — 检测失败

0 — 没有使用自动增益播放

1 — 使用了自动增益播放

### 备注 (Remarks)

无

### 例如 (Example)



```
CPC_PlayFile(1,
              CPC_PLAY_FILE_ISENABLEAGC,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 21 CPC\_PLAY\_FILE\_STOP 停止播放指定文件

### 功能描述 (Description)

停止播放指定文件

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue*

使用 CPC\_PLAY\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

停止播放后，该播放句柄将无效，以后不能对该播放句柄进行操作。

### 例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_STOP,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 22 CPC\_PLAY\_FILE\_GETCOUNT 获取有效的播放文件句柄数量

### 功能描述 (Description)

获取有效的播放文件句柄数量

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，其它返回有效的播放文件句柄数量

### 备注 (Remarks)

- 开始播放后，如果没有播放完毕或者调用停止播放，不管是否暂停了播放，该



句柄将一直有效。

- 如果没有播放句柄时, 可以考虑关闭喇叭

例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_GETCOUNT,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 8. 23 CPC\_PLAY\_FILE\_STOPALL 停止播放所有文件

功能描述 (Description)

停止播放所有文件

参数介绍 (Parameters)

[in] *nChannelID*

播放语音文件的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

返回值 (Return Values)

大于 0 表示成功,其它表示失败

备注 (Remarks)

只停止该通道的所有播放的语音文件,如果要停止所有通道, 需要对每个通道进行调用 CPC\_PLAY\_FILE\_STOPALL

例如 (Example)

```
CPC_PlayFile(1,
              CPC_PLAY_FILE_CPC_PLAY_FILE_STOPALL,
              i32PlayFileHandle,
              0,
              0);
```

## 2. 9 播放语音流缓冲(CPC\_PlayBuf)

功能描述 (Description)

播放语音流到设备(建议用户不要在等待接收来电号码前播放语音到线路上,避免影响接收来电号码)

函数原型 (Function)



```

CPCINT32  CPC_PlayBuf (/*[in]*/CPCINT32  nChannelID,
                        /*[in]*/CPCUINT32  uPlayType,
                        /*[in]*/CPCINT32  nValue,
                        /*[in]*/CPCINT32  nValueEx,
                        /*[in]*/CPCCHAR8* pValue);

```

### 返回值 (Return Values)

有符号 32bit 位

### 参数描述 (Parameters description)

## 2.9.1 CPC\_PLAY\_BUF\_START 开始缓冲播放

### 功能描述 (Description)

开始缓冲播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue* — 保留

[in] *nValueEx*

初始化分配的缓冲大小, 如果为 0, 系统默认分配 4000 字节缓冲

[in] *pValue*

语音流的编码格式(LPWAVEFORMATEX)

- 如果该值为 0, 系统就使用默认 8k/16 位/单声道方式播放语音流, 如果编码格式不匹配会使播放声音成为杂音

### 返回值 (Return Values)

小于等于 0 表示失败

其它返回播放的句柄, 用户必须保存该句柄, 以后对该缓冲播放的操作全部需要指定该句柄。

### 备注 (Remarks)

- 关于语音编码格式信息(LPWAVEFORMATEX)在 windows 的 mmsystem.h 中定义, 具体意义请查看 windows 相关文档。
- 设备播放的速度为每秒 16K。也就是说如果初始化分配 16K 缓冲大小, 您一次最多可以写入(CPC\_PLAY\_BUF\_WRITEDATA)1 秒的语音数据。系统默认的 4000 字节的缓冲可以保存 250 毫秒的语音数据。

### 例如 (Example)

```

WAVEFORMATEX waveformat;
waveformat.cbSize=0;
waveformat.wFormatTag=1;//在 mmsystem.h 中定义
waveformat.nChannels=1;
waveformat.nSamplesPerSec=8000;
waveformat.nAvgBytesPerSec=16000;
waveformat.nBlockAlign=2

```





```

waveformat.wBitsPerSample=16;
CPCINT32    i32PlayBufHandle=CPC_PlayBuf(1,
                                           CPC_PLAY_BUF_START,
                                           0,
                                           8000,
                                           (LPWAVEFORMATEX)&waveformat);
- 通道 1 开始缓冲播放, 初始化分配 8000 字节缓冲, 语音流格式为 8K/16
  位 WAV。

```

## 2.9.2 CPC\_PLAY\_BUF\_SETCALLBACK 设置语音流缓冲播放回调函数

### 功能描述 (Description)

设置语音流缓冲播放回调函数

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx*

回调函数的用户自定义参数, 在回调时返回给用户

[in] *pValue*

缓冲播放回调函数地址(PCallBack\_PlayBuf), 查看 CC301 原型定义

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 在 C++ 中用户自定义参数一般使用类指针(this)
- 回调函数的原型请查看 cc301 的回调函数(PCallBack\_PlayBuf)定义
- 设置回调函数系统在播放一段语音流后将立即调用该回调函数, 把缓冲信息通过该回调函数传递给用户, 用户可以根据回调的缓冲信息来进行下一步操作。
- 缓冲信息一般包含当前缓冲剩余的语音数据长度和当前缓冲空闲长度, 语音长度加上空闲长度, 就为用户初始化时分配的缓冲长度。

### 例如 (Example)

```

CPC_PlayBuf(1,
             CPC_PLAY_BUF_SETCALLBACK,
             i32PlayBufHandle,
             (CPCINT32)this,
             (PCallBack_PlayBuf)PlayBufCallbackFunc);
- PlayBufCallbackFunc: 为 PCallBack_PlayBuf 类型回调函数地址。
- this 为用户自定义参数, 在 C++ 一般使用类指针(this)。

```



## 2.9.3 CPC\_PLAY\_BUF\_SETWAVEFORMAT 指定缓冲播放的语音流的编码格式

### 功能描述 (Description)

指定缓冲播放的语音流的编码格式

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 `CPC_PLAY_BUF_START` 返回的句柄

[in] *nValueEx*

CC301 定义的语音编码 ID, 如果您的语音编码格式在该定义的范围里, 可以直接使用指定 ID 方式, 范围 (0-255), 不再需要 LPWAVEFORMATEX。

[in] *pValue*

如果您的语音格式不在 CC301 定义的 ID 里, 使用该参数指定语音编码格式 LPWAVEFORMATEX。

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 初始化播放时(`CPC_PLAY_BUF_START`)如果没有指定语音流编码格式信息, 可以通过该函数来指定语音流编码格式。
- 如果用户对 WAVEFORMATEX 不熟悉, 可以考虑使用指定 CC301 定义的 ID 方式, 该方式的缺点为并不支持所有 wav 格式的语音。

### 例如 (Example)

使用 CC301 定义的编码 ID 方式:

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_SETWAVEFORMAT,
             i32PlayBufHandle,
             BRI_WAV_FORMAT_ALAW8K,
             0);
```

- 指定播放的语音流编码格式为 CC301 定义的 ID 为 `BRI_WAV_FORMAT_ALAW8K`
- 如果该值大于 255 就为无效, 如果该值小于 255, 而 CC301 未定义, 系统就使用默认的 8K/16 位/单声道编码格式进行播放
- 使用指定 CC301 定义的 ID 方式时 `pValue` 必须指定为 0

使用 windows 定义的详细语音编码格式方式:

```
WAVEFORMATEX waveformat;
waveformat.cbSize=0;
waveformat.wFormatTag=1;//在 windows 的 mmsystem.h 定义
waveformat.nChannels=1;
waveformat.nSamplesPerSec=8000;
waveformat.nAvgBytesPerSec=16000;
```



```

waveformat.nBlockAlign=2
waveformat.wBitsPerSample=16;
CPC_PlayBuf(1,
            CPC_PLAY_BUF_SETWAVEFORMAT,
            i32PlayBufHandle,
            0,
            (LPWAVEFORMATEX)& waveformat);

```

- 指定播放的语音流编码格式为 8K/16 位/单声道语音编码

#### 使用自动识别语音流的方式:

```

CPC_PlayBuf (0,
            CPC_PLAY_BUF_SETWAVEFORMAT,
            i32PlayBufHandle,
            256,
            0);

```

- 256 超过有效范围 255, 该设置方式为两个参数都无效, 系统将自动获取语音流的最前面部分自动识别语音编码格式。
- 该播放方式适合在用流方式播放一个语音文件, 因为语音文件的编码信息已经保存在文件头。
- 非特殊需求, 不建议用户使用该方式。

## 2.9.4 CPC\_PLAY\_BUF\_WRITEDATA 写语音流数据到缓冲

### 功能描述 (Description)

写语音流数据到缓冲

### 参数介绍 (Parameters)

[in] *nChannelID*  
播放语音流的通道

[in] *nValue*  
使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx*  
准备写入缓冲的语音数据长度

[in] *pValue*  
准备写入缓冲的语音数据内存地址

### 返回值 (Return Values)

小于 0 表示失败, 其它表示当前缓冲的语音数据长度

### 备注 (Remarks)

- 具体查看错误原因. 如: BCERR\_NOBUFSIZE 表示缓冲不够
- 建议用户在写入语音数据前使用 *CPC\_PLAY\_BUF\_FREESIZE* 判断当前是否有足够的缓冲
- 如果写入失败后用户不要丢弃该语音, 以便下次重新写入。
- 经过压缩的语音写入的长度并不一定跟系统提示的语音长度相同,  
如: 使用 8K/4bit 的 ADPCM 语音流, 写入 100 字节的语音流, 实际产生的



语音可能有 400 字节左右，所以当缓冲剩余 200 空闲时，写如 100 个字节的 ADPCM 语音流，可能会引起缓冲不够而失败，用户就需要保存该 ADPCM 数据，以便下次空闲时重新写入。

**例如 (Example)**

```
char szBuf[128];  
//获取语音  
//-----  
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_WRITEDATA,  
             i32PlayBufHandle,  
             128,  
             szBuf);
```

## 2.9.5 CPC\_PLAY\_BUF\_SETVOLUME 设置播放语音流缓冲的音量

**功能描述 (Description)**

设置播放语音流缓冲的音量

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx*

音量值。范围(0-10000)

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

**例如 (Example)**

```
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_SETVOLUME,  
             i32PlayBufHandle,  
             200,  
             0);
```

## 2.9.6 CPC\_PLAY\_BUF\_GETVOLUME 获取播放语音流缓冲的音量

**功能描述 (Description)**

获取播放语音流缓冲的音量

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

小于 0 表示失败, 其它返回音量值

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍, 如果为 50 就表示缩小一半, 如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

**例如 (Example)**

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_GETVOLUME,
             i32PlayBufHandle,
             200,
             0);
```

## 2.9.7 CPC\_PLAY\_BUF\_SETUSERVALUE 设置本次语音流播放的用户自定义数值

**功能描述 (Description)**

设置本次语音流播放的用户自定义数值

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx*

用户定义的任意数值

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 该值默认为 0
- 该值无任何意义, 只为用户保存
- 用户在停止本次语音流播放前可以使用 CPC\_PLAY\_BUF\_GETUSERVALUE 来获取保存的值
- 建议用户设置该值时不要跟 CC301 定义的错误 ID 相同

**例如 (Example)**

```
CPC_PlayBuf(1,
```



```
CPC_PLAY_BUF_SETUSERVALUE,  
i32PlayBufHandle,  
200,  
0);
```

- 设置本次语音流播放的用户自定义数值为 200。

## 2.9.8 CPC\_PLAY\_BUF\_GETUSERVALUE 获取本次语音流播放的用户自定义数值

### 功能描述 (Description)

获取本次语音流播放的用户自定义数值

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，其它返回用户设置的自定义数值

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayBuf(1,  
CPC_PLAY_BUF_GETUSERVALUE,  
i32PlayBufHandle,  
0,  
0);
```

## 2.9.9 CPC\_PLAY\_BUF\_ENABLEAGC 打开/关闭语音流自动增益处理

### 功能描述 (Description)

打开/关闭语音流自动增益处理

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx*

0 — 打开自动增益处理

1 — 打开自动增益处理

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它返回失败

**备注 (Remarks)**

默认自动增益是关闭状态

**例如 (Example)**

```
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_ENABLEAGC,  
             i32PlayBufHandle,  
             1,  
             0);
```

- 通道 1 的 i32PlayBufHandle 语音流打开自动增益处理

## 2.9.10 CPC\_PLAY\_BUF\_ISENABLEAGC 检测是否使用语音流自动增益处理

**功能描述 (Description)**

检测是否使用语音流自动增益处理

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

小于 0 — 检测失败

0 — 语音流没有使用自动增益处理

1 — 语音流使用了自动增益处理

**备注 (Remarks)**

默认自动增益是关闭状态

**例如 (Example)**

```
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_ISENABLEAGC,  
             i32PlayBufHandle,  
             0,  
             0);
```

## 2.9.11 CPC\_PLAY\_BUF\_PAUSE 暂停播放缓冲

**功能描述 (Description)**

暂停播放缓冲

**参数介绍 (Parameters)**

[in] *nChannelID*



播放语音流的通道

[in] *nValue*

使用 `CPC_PLAY_BUF_START` 返回的句柄

[in] *nValueEx*    — 保留

[in] *pValue*       — 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_PAUSE,
             i32PlayBufHandle,
             0,
             0);
- 暂停通道 1 播放 i32PlayBufHandle 句柄的缓冲
```

## 2.9.12 `CPC_PLAY_BUF_RESUME` 恢复播放缓冲

**功能描述 (Description)**

恢复播放缓冲

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 `CPC_PLAY_BUF_START` 返回的句柄

[in] *nValueEx*    — 保留

[in] *pValue*       — 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_RESUME,
             i32PlayBufHandle,
             0,
             0);
- 恢复通道 1 播放 i32PlayBufHandle 句柄的缓冲
```





## 2.9.13 CPC\_PLAY\_BUF\_ISPAUSE 检测是否已暂停播放

### 功能描述 (Description)

检测是否已暂停播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 — 检测失败

0 — 表示没有暂停播放

1 — 表示已经暂停播放

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_ISPAUSE,
             i32PlayBufHandle,
             0,
             0);
```

## 2.9.14 CPC\_PLAY\_BUF\_STOP 停止播放语音流缓冲

### 功能描述 (Description)

停止播放语音流缓冲

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_STOP,
             i32PlayBufHandle,
```



```
0,  
0);
```

## 2. 9. 15 CPC\_PLAY\_BUF\_FREESIZE 获取当前缓冲空闲字节长度

### 功能描述 (Description)

获取当前缓冲空闲字节长度

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，其它返回空闲缓冲长度

### 备注 (Remarks)

单位为字节

### 例如 (Example)

```
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_FREESIZE,  
             i32PlayBufHandle,  
             0,  
             0);
```

## 2. 9. 16 CPC\_PLAY\_BUF\_DATASIZE 获取当前缓冲语音数据长度

### 功能描述 (Description)

获取当前缓冲语音数据长度

### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，其它返回语音数据长度

### 备注 (Remarks)

单位为字节

### 例如 (Example)

```
CPC_PlayBuf(1,
```



```
CPC_PLAY_BUF_DATASIZE,  
i32PlayBufHandle,  
0,  
0);
```

### 2.9.17 CPC\_PLAY\_BUF\_TOTALSAMPLES 获取本次语音流播放已经总共播放的采样数

#### 功能描述 (Description)

获取本次语音流播放已经总共播放的采样数

#### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0 表示失败，其它返回已经播放的语音采样数

#### 备注 (Remarks)

- 单位为字节
- 正常播放速度为每秒 8000 个采样数

#### 例如 (Example)

```
CPC_PlayBuf(1,  
CPC_PLAY_BUF_TOTALSAMPLES,  
i32PlayBufHandle,  
0,  
0);
```

### 2.9.18 CPC\_PLAY\_BUF\_SETJITTERBUFSIZE 设置动态缓冲播放的长度

#### 功能描述 (Description)

设置动态缓冲播放的长度，当缓冲数据播放为空后下次播放前缓冲内必须大于该长度的语音

#### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx*

动态的采样数

- 0 表示不启用



[in] *pValue* — 保留

#### 返回值 (Return Values)

小于等于 0 表示失败，其它表示成功

#### 备注 (Remarks)

- 该功能为解决用户缓冲流的抖动后引起声音停顿
- 可用在播放网络数据包时用来避免网络抖动
- 非特殊功能一般不需要使用该功能

#### 例如 (Example)

```
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_GETJITTERBUFSIZE,  
             i32PlayBufHandle,  
             1000,  
             0);
```

- 如果缓冲播放空之后在下次开始播放前，缓冲里的数据必须大于 1000 个采样点

## 2. 9. 19 CPC\_PLAY\_BUF\_GETJITTERBUFSIZE 获取动态缓冲播放的长度

#### 功能描述 (Description)

获取动态缓冲播放的长度

#### 参数介绍 (Parameters)

[in] *nChannelID*

播放语音流的通道

[in] *nValue*

使用 CPC\_PLAY\_BUF\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0 表示失败，其它返回设置的采样数

#### 备注 (Remarks)

- 无

#### 例如 (Example)

```
CPC_PlayBuf(1,  
             CPC_PLAY_BUF_SETJITTERBUFSIZE,  
             i32PlayBufHandle,  
             0,  
             0);
```

## 2. 9. 20 CPC\_PLAY\_BUF\_GETCOUNT 获取有效的语音流播放句柄数量

#### 功能描述 (Description)

获取有效的语音流播放句柄数量

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

小于 0 表示失败，其它返回有效的语音流播放句柄数量

**备注 (Remarks)**

- 开始语音流播放后，如果没有调用停止播放，不管是否暂停了播放，该句柄将一直有效。
- 如果已经没有播放句柄时，可以考虑关闭喇叭

**例如 (Example)**

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_GETCOUNT,
             0,
             0,
             0);
```

## 2.9.21 CPC\_PLAY\_BUF\_STOPALL 停止所有语音流播放

**功能描述 (Description)**

停止所有语音流播放

**参数介绍 (Parameters)**

[in] *nChannelID*

播放语音流的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

只停止该通道的所有语音流播放,如果要停止所有通道，需要对每个通道进行调用 CPC\_PLAY\_BUF\_STOPALL

**例如 (Example)**

```
CPC_PlayBuf(1,
             CPC_PLAY_BUF_STOPALL,
             0,
             0,
             0);
```



## 2. 10 多文件连续播放(CPC\_PlayMultiFile)

### 功能描述 (Description)

连续播放多个文件到设备

### 函数原型 (Function)

```
CPCINT32 CPC_PlayMultiFile (/*[in]*/CPCINT16 nChannelID,  
                             /*[in]*/CPCUINT32 uPlayType,  
                             /*[in]*/CPCINT32 nValue,  
                             /*[in]*/CPCINT32 nValueEx,  
                             /*[in]*/CPCCHAR8* pValue);
```

### 返回值 (Return Values)

有符号 32bit 位,大于 0 播放成功,其它表示失败。

### 参数描述 (Parameters description)

### 2. 10. 1 CPC\_PLAY\_MULTIFILE\_START 开始多文件连续播放

#### 功能描述 (Description)

开始多文件连续播放

#### 参数介绍 (Parameters)

[in] *nChannelID*

多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue*

使用`|`分隔的多个文件

- 使用绝对路径。如: "c:\a.wav|c:\b.wav"。
- 使用相对路径。如: "a.wav|b.wav",系统将自动定为该文件为本进程所在目录下的 a.wav 和 b.wav。
- 混合相对路径。如: "a.wav|c:\b.wav",系统将自动定为该文件为本进程所在目录下的 a.wav 和 c:\b.wav。
- 该播放文件路径支持网络共享文件路径:  
如: [\\192.168.0.1\wav\a.wav](http://192.168.0.1/wav/a.wav)

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 一个通道只能进行一次多文件播放, 如果第一次多文件播放没有停止, 第二次



不能进行播放。

- 建议用户使用绝对路径的方式来播放。
- 目前不支持播放远程 HTTP 协议和 FTP 协议语音流，所以如下格式的语音文件路径不能被播放：如：http://192.168.0.1/a.wav

#### 例如 (Example)

绝对路径：

```
CPC_PlayMultiFile(1,
    CPC_PLAY_MULTIFILE_START,
    0,
    0,
    "c:\a.wav|c:\b.wav");
```

- 通道 1 开始多文件播放,先播放 c:\a.wav 后立即播放 c:\b.wav。

相对路径：

```
CPC_PlayMultiFile(1,
    CPC_PLAY_MULTIFILE_START,
    0,
    0,
    "a.wav|b.wav");
```

- 通道 1 开始多文件播放,先播放本程序目录下的 a.wav 后立即播放本程序目录下的 b.wav。

## 2. 10. 2 CPC\_PLAY\_MULTIFILE\_PAUSE 暂停多文件连续播放

### 功能描述 (Description)

暂停多文件连续播放

### 参数介绍 (Parameters)

[in] *nChannelID*  
多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayMultiFile(1,
    CPC_PLAY_MULTIFILE_PAUSE,
    0,
    0,
    0);
```

- 通道 1 暂停多文件连续播放



### 2. 10. 3 CPC\_PLAY\_MULTIFILE\_RESUME 恢复多文件连续播放

#### 功能描述 (Description)

恢复多文件连续播放

#### 参数介绍 (Parameters)

[in] *nChannelID*

多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_PlayMultiFile(1,  
                  CPC_PLAY_MULTIFILE_RESUME,  
                  0,  
                  0,  
                  0);
```

### 2. 10. 4 CPC\_PLAY\_MULTIFILE\_ISPAUSE 检测是否暂停了多文件连续播放

#### 功能描述 (Description)

检测是否暂停了多文件连续播放

#### 参数介绍 (Parameters)

[in] *nChannelID*

多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0 — 检测失败

0 — 没有暂停多文件连续播放

1 — 暂停了多文件连续播放

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_PlayMultiFile(1,  
                  CPC_PLAY_MULTIFILE_ISPAUSE,  
                  0,  
                  0,  
                  0);
```





## 2. 10. 5 CPC\_PLAY\_MULTIFILE\_SETVOLUME 设置多文件连续播放的音量

### 功能描述 (Description)

设置多文件连续播放的音量

### 参数介绍 (Parameters)

[in] *nChannelID*

多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx*

音量值。范围(0-10000)

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

### 例如 (Example)

```
CPC_PlayMultiFile(1,  
                  CPC_PLAY_MULTIFILE_SETVOLUME,  
                  0,  
                  200,  
                  0);
```

## 2. 10. 6 CPC\_PLAY\_MULTIFILE\_GETVOLUME 获取多文件连续播放的音量

### 功能描述 (Description)

获取多文件连续播放的音量

### 参数介绍 (Parameters)

[in] *nChannelID*

多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 表示失败，其它返回多文件连续播放的音量值

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayMultiFile(1,  
                  CPC_PLAY_MULTIFILE_GETVOLUME,  
                  0,
```



```
0,  
0);
```

## 2. 10. 7 CPC\_PLAY\_MULTIFILE\_ISSTART 检测是否启动了多文件连续播放

### 功能描述 (Description)

检测是否启动了多文件连续播放

### 参数介绍 (Parameters)

[in] *nChannelID*  
多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 — 检测失败  
0 — 没有启动多文件连续播放  
1 — 启动了多文件连续播放

### 备注 (Remarks)

调用 CPC\_PLAY\_MULTIFILE\_START 开始播放后, 如果没有播放完毕或强制停止播放, 不管是否暂停播放, 该值都返回 1

### 例如 (Example)

```
CPC_PlayMultiFile(1,  
CPC_PLAY_MULTIFILE_ISSTART,  
0,  
0,  
0);
```

## 2. 10. 8 CPC\_PLAY\_MULTIFILE\_STOP 停止多文件连续播放

### 功能描述 (Description)

停止多文件连续播放

### 参数介绍 (Parameters)

[in] *nChannelID*  
多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)



```
CPC_PlayMultiFile(1,
                  CPC_PLAY_MULTIFILE_STOP,
                  0,
                  0,
                  0);
```

## 2. 10. 9 CPC\_PLAY\_MULTIFILE\_STOPALL 停止全部多文件连续播放

### 功能描述 (Description)

停止全部多文件连续播放

### 参数介绍 (Parameters)

[in] *nChannelID*  
多文件连续播放的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

由于目前一个通道同时只能执行一次多文件连续播放,所以该功能等同与  
CPC\_PLAY\_MULTIFILE\_STOP

### 例如 (Example)

```
CPC_PlayMultiFile(1,
                  CPC_PLAY_MULTIFILE_STOPALL,
                  0,
                  0,
                  0);
```

- 通道 1 停止全部多文件连续播放并且释放所有资源

## 2. 11 字符串播放(CPC\_PlayString)

### 功能描述 (Description)

播放指定的字符串到设备

### 函数原型 (Function)

```
CPCINT32 CPC_PlayString (/*[in]*/CPCINT16 nChannelID,
                        /*[in]*/CPCUINT32 uPlayType,
                        /*[in]*/CPCINT32 nValue,
```



```
/*[in]*/ CPCINT32 nValueEx,  
/*[in]*/ CPCCHAR8* pValue);
```

### 返回值 (Return Values)

有符号 32bit 位。

### 参数描述 (Parameters description)

## 2. 11. 1 CPC\_PLAY\_STRING\_INITLIST 初始化字符播放列表

### 功能描述 (Description)

初始化字符播放列表

### 参数介绍 (Parameters)

[in] nChannelID

播放字符的通道

[in] nValue — 保留

[in] nValueEx — 保留

[in] pValue

保存字符串对应关系的语音文件的列表文件路径

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

按指定格式把字符串和事先录制好的语音文件建立对应关系，如：

你好,voice\A.wav

小王,voice\B.wav

A.wav 为事先录音好的‘你好’的语音,并且存放在本程序的 voice 目录下

B.wav 为事先录音好的‘小王’的语音,并且存放在本程序的 voice 目录下

具体查看演示程序中的列表文件例子

该播放的字符只支持数字和汉字

该播放方式合适与用户播放固定少量的字符，如果用户想播放随机任意字符,建议使用专用的 TTS 库。如：microsoft speech SDK。

### 例如 (Example)

```
CPC_PlayString(1,  
                CPC_PLAY_STRING_INITLIST,  
                0,  
                0,  
                "c:\filelist.txt");
```

- 初始化字符串和语音文件对应关系列表



## 2. 11. 2 CPC\_PLAY\_STRING\_START 开始字符播放

### 功能描述 (Description)

开始字符播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放字符的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue*

需要播放的字符串

该字符串的单元字符必须在初始化列表里有对应的语音文件

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

开始播放前必须先初始化字符播放列表(CPC\_PLAY\_STRING\_INITLIST),而且需要播放字符必须在该列表里存在。

如：播放‘你好小王’，该列表里必须有个‘你好’和‘小王’的两个对应语音文件，或者有个‘你好小王’的对应语音文件。如下格式：

你好,a.wav

小王,b.wav

或者

你好小王,c.wav

### 例如 (Example)

```
CPC_PlayString(1,  
                CPC_PLAY_STRING_START,  
                0,  
                0,  
                “你好小王”);
```

- 系统将自动建立语音文件关联为 a.wav|b.wav 并自动播放

## 2. 11. 3 CPC\_PLAY\_STRING\_PAUSE 暂停字符播放

### 功能描述 (Description)

暂停字符播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放字符的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_PlayString(1,
                CPC_PLAY_STRING_PAUSE,
                0,
                0,
                0);
```

## 2. 11. 4 CPC\_PLAY\_STRING\_RESUME 恢复字符播放

**功能描述 (Description)**

恢复字符播放

**参数介绍 (Parameters)**

[in] *nChannelID*

播放字符的通道

[in] *nValue*        — 保留

[in] *nValueEx*     — 保留

[in] *pValue*        — 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_PlayString(1,
                CPC_PLAY_STRING_RESUME,
                0,
                0,
                0);
```

## 2. 11. 5 CPC\_PLAY\_STRING\_ISPAUSE 检测是否暂停了字符播放

**功能描述 (Description)**

检测是否暂停了字符播放

**参数介绍 (Parameters)**

[in] *nChannelID*

播放字符的通道

[in] *nValue*        — 保留

[in] *nValueEx*     — 保留

[in] *pValue*        — 保留

**返回值 (Return Values)**



- |      |   |          |
|------|---|----------|
| 小于 0 | — | 检测失败     |
| 0    | — | 没有暂停字符播放 |
| 1    | — | 暂停了字符播放  |

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_PlayString(1,
                CPC_PLAY_STRING_ISPAUSE,
                0,
                0,
                0);
```

## 2. 11. 6 CPC\_PLAY\_STRING\_SETVOLUME 设置字符播放音量

**功能描述 (Description)**

设置字符播放音量

**参数介绍 (Parameters)**

- [in] *nChannelID*  
播放字符的通道
- [in] *nValue* — 保留
- [in] *nValueEx* — 保留  
音量值。范围(0-10000)
- [in] *pValue* — 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

**例如 (Example)**

```
CPC_PlayString(1,
                CPC_PLAY_STRING_SETVOLUME,
                0,
                200,
                0);
```

- 设置通道 1 字符播放的音量值为 200。

## 2. 11. 7 CPC\_PLAY\_STRING\_GETVOLUME 获取字符播放音量

**功能描述 (Description)**

获取字符播放音量

**参数介绍 (Parameters)**



[in] *nChannelID*

播放字符的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0 表示失败，其它返回播放字符的音量值

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_PlayString(1,
                CPC_PLAY_STRING_GETVOLUME,
                0,
                0,
                0);
```

## 2. 11. 8 CPC\_PLAY\_STRING\_ISSTART 检测是否启动了字符播放

#### 功能描述 (Description)

检测是否启动了字符播放

#### 参数介绍 (Parameters)

[in] *nChannelID*

播放字符的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0 — 检测失败

0 — 没有启动字符播放

1 — 启动了字符播放

#### 备注 (Remarks)

调用 CPC\_PLAY\_STRING\_START 开始播放后，如果没有播放完毕或强制停止播放，不管是否暂停播放，该值都返回 1

#### 例如 (Example)

```
CPC_PlayString(1,
                CPC_PLAY_STRING_ISSTART,
                0,
                0,
                0);
```





## 2. 11. 9 CPC\_PLAY\_STRING\_STOP 停止字符播放

### 功能描述 (Description)

停止字符播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放字符的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_PlayString(1,  
                CPC_PLAY_STRING_STOP,  
                0,  
                0,  
                0);
```

- 通道 1 停止字符播放。

## 2. 11. 10 CPC\_PLAY\_STRING\_STOPALL 停止全部字符播放

### 功能描述 (Description)

停止全部字符播放

### 参数介绍 (Parameters)

[in] *nChannelID*

播放字符的通道

[in] *nValue* — 保留

[in] *nValueEx* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

由于目前一个通道同时只能执行一次字符播放, 所以该功能等同与  
CPC\_PLAY\_STRING\_STOP

### 例如 (Example)

```
CPC_PlayString(1,  
                CPC_PLAY_STRING_STOPALL,  
                0,  
                0,  
                0);
```



- 通道 1 停止全部字符播放，并释放所有资源。

## 2. 12 文件录音 (CPC\_RecordFile)

### 功能描述 (Description)

对设备上的语音录制到指定文件

### 函数原型 (Function)

```
CPCINT32 CPC_RecordFile (/*[in]*/CPCINT32 nChannelID,  
                          /*[in]*/CPCUINT32 uRecordType,  
                          /*[in]*/CPCINT32 nValue,  
                          /*[in]*/CPCINT32 nValueEx,  
                          /*[in,out]*/CPCCHAR8 *pValue);
```

### 返回值 (Return Values)

有符号 32bit 位,大于 0 表示成功,其它表示失败.

### 参数描述 (Parameters description)

### 2. 12. 1 CPC\_RECORD\_FILE\_START 开始文件录音

#### 功能描述 (Description)

开始文件录音

#### 参数介绍 (Parameters)

[in] nChannelID

文件录音的通道

[in] nValue

系统定义的语音编码格式 ID (BRI\_WAV\_FORMAT\_XXX)

[in] nValueEx

文件录音属性

- RECORD\_MASK\_ECHO 使用回音抵消后的语音
- RECORD\_MASK\_AGC 使用自动增益后的语音

[in] pValue

保存录音的文件路径.录音文件名长度不能少于 1 个字节

- 使用绝对路径。

如: "c:\wave\1.wav".

- 使用相对路径。



如：“a.wav”,系统将自动定为该文件为本进程所在目录下的  
a.wav。

- 该文件路径支持网络共享文件路径:

如: [\\192.168.0.1\wav\ a.wav](http://192.168.0.1/wav/a.wav)

#### 返回值 (Return Values)

小于等于表示失败, 其它返回录音句柄, 该句柄用户必须保存起来, 以后对该录音的操作都需要使用该句柄。

#### 备注 (Remarks)

建议用户使用绝对路径的方式。

如果保存录音的目录不存在, 系统将自动创建目录。

如果保存录音的文件已经存在, 系统将删除原文件。

文件录音的多个属性使用逻辑与(OR)可以一起使用,如:

RECORD\_MASK\_ECHO|RECORD\_MASK\_AGC ,表示回音抵消后并且自动增益处理后的语音数据

#### 例如 (Example)

```
CPC_RecordFile(1,
                CPC_RECORD_FILE_START,
                BRI_WAV_FORMAT_ALAW8K,
                RECORD_MASK_ECHO,
                "c:\wave\ a.wav");
```

通道 1 开始文件录音, 格式为 BRI\_WAV\_FORMAT\_ALAW8K, 录音路径为  
“c:\wave\ a.wav”,使用语音为回音抵消后的数据。

## 2. 12. 2 CPC\_RECORD\_FILE\_PAUSE 暂停文件录音

#### 功能描述 (Description)

暂停文件录音

#### 参数介绍 (Parameters)

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 CPC\_RECORD\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_RecordFile(1,
                CPC_RECORD_FILE_PAUSE,
                i32RecordHandle,
                0,
```



0);

### 2. 12. 3 CPC\_RECORD\_FILE\_RESUME 恢复文件录音

#### 功能描述 (Description)

恢复文件录音

#### 参数介绍 (Parameters)

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 CPC\_RECORD\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_RecordFile(1,  
                CPC_RECORD_FILE_RESUME,  
                i32RecordHandle,  
                0,  
                0);
```

### 2. 12. 4 CPC\_RECORD\_FILE\_ISPAUSE 检测是否暂停文件录音

#### 功能描述 (Description)

检测是否暂停文件录音

#### 参数介绍 (Parameters)

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 CPC\_RECORD\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

#### 返回值 (Return Values)

小于 0 — 检测失败

0 — 没有暂停文件录音

1 — 暂停文件录音

#### 备注 (Remarks)

无

**例如 (Example)**

```
CPC_RecordFile(1,
                CPC_RECORD_FILE_ISPAUSE,
                i32RecordHandle,
                0,
                0);
```

## 2. 12. 5 CPC\_RECORD\_FILE\_ELAPSE 获取已经文件录音的时长

**功能描述 (Description)**

获取已经文件录音的时长

**参数介绍 (Parameters)**

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 CPC\_RECORD\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

小于 0 表示失败，其它返回已经文件录音的时长

**备注 (Remarks)**

返回时间单位为秒

**例如 (Example)**

```
CPC_RecordFile(1,
                CPC_RECORD_FILE_ELAPSE,
                i32RecordHandle,
                0,
                0);
```

## 2. 12. 6 CPC\_RECORD\_FILE\_SETVOLUME 设置文件录音音量

**功能描述 (Description)**

设置文件录音音量

**参数介绍 (Parameters)**

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 CPC\_RECORD\_FILE\_START 返回的句柄

[in] *nValueEx*

需要设置的音量值。范围(0-10000)

[in] *pValue* — 保留

**返回值 (Return Values)**



大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

**例如 (Example)**

```
CPC_RecordFile(1,  
                CPC_RECORD_FILE_SETVOLUME,  
                i32RecordHandle,  
                200,  
                0);
```

## 2. 12. 7 CPC\_RECORD\_FILE\_GETVOLUME 获取文件录音音量

**功能描述 (Description)**

获取文件录音音量

**参数介绍 (Parameters)**

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 CPC\_RECORD\_FILE\_START 返回的句柄

[in] *nValueEx* — 保留

[in] *pValue* — 保留

**返回值 (Return Values)**

小于 0 表示失败, 其它返回文件录音音量值.

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_RecordFile(1,  
                CPC_RECORD_FILE_GETVOLUME,  
                i32RecordHandle,  
                0,  
                0);
```

## 2. 12. 8 CPC\_RECORD\_FILE\_PATH 获取当前录音文件的路径

**功能描述 (Description)**

获取当前录音文件的路径

**参数介绍 (Parameters)**

[in] *nChannelID*

文件录音的通道

[in] *nValue*



使用 `CPC_RECORD_FILE_START` 返回的句柄

[in] *nValueEx*

保存文件路径的内存长度,也就为 *pValue* 分配的内存长度

[in] *pValue*

保存录音文件的内存地址

#### 返回值 (Return Values)

小于等于 0 表示失败, 其它返回的文件路径长度.

#### 备注 (Remarks)

建议给 *pValue* 分配足够大的内存(大于 256 字节), 避免分配的内存长度不够而返回失败。

#### 例如 (Example)

```
char  szBuf[260];//定义一个 260 字节的内存
memset(szBuf,0,260);//初始化为空
CPC_RecordFile(1,
               CPC_RECORD_FILE_PATH,
               i32RecordHandle,
               260,
               szBuf);
```

## 2. 12. 9 CPC\_RECORD\_FILE\_STOP 停止指定文件录音

#### 功能描述 (Description)

停止指定文件录音

#### 参数介绍 (Parameters)

[in] *nChannelID*

文件录音的通道

[in] *nValue*

使用 `CPC_RECORD_FILE_START` 返回的句柄

[in] *nValueEx*     – 保留

[in] *pValue*       – 保留

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_RecordFile(1,
               CPC_RECORD_FILE_STOP,
               i32RecordHandle,
               0,
               0);
```



## 2. 12. 10 CPC\_RECORD\_FILE\_STOPALL 停止全部文件录音

### 功能描述 (Description)

停止全部文件录音

### 参数介绍 (Parameters)

[in] *nChannelID*

文件录音的通道

[in] *nValue* – 保留

[in] *nValueEx* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_RecordFile(1,
                CPC_RECORD_FILE_STOPALL,
                0,
                0,
                0);
```

## 2. 13 缓冲语音流录音 (CPC\_RecordBuf)

### 功能描述 (Description)

获取设备上的语音流到用户应用层

### 函数原型 (Function)

```
CPCINT32 CPC_RecordBuf (/*[in]*/CPCINT16 nChannelID,
                        /*[in]*/CPCUINT32 uRecordType,
                        /*[in]*/CPCINT32 nValue,
                        /*[in]*/CPCINT32 nValueEx,
                        /*[in]*/CPCCHAR8 *pValue);
```

### 返回值 (Return Values)

有符号 32bit 位,大于 0 表示成功,其它表示失败.

### 参数描述 (Parameters description)





## 2. 13. 1 CPC\_RECORD\_BUF\_HWND\_START 开始缓冲录音窗口回调

### 功能描述 (Description)

开始缓冲录音窗口回调

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

窗口句柄值

[in] *nValueEx*

回调语音流的属性

- RECORD\_MASK\_PAUSE      - 暂停回调
- RECORD\_MASK\_ECHO      - 回音抵消后的数据
- RECORD\_MASK\_AGC      - 自动增益处理后的数据

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)

- 开始回调后系统从通道中采集到的语音数据后发送消息到该窗口
- 该窗口将接收到 BRI\_RECBUF\_MESSAGE 消息,消息 ID 为(WM\_USER+2001)
- 消息中的第一个参数(wParam)为数据的长度, 第二个(lParam)为数据地址

### 例如 (Example)

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_HWND_START,
               (CPCINT32)m_hWnd,
               RECORD_MASK_ECHO,
               0);
```

- 系统从通道 1 采集到回音抵消后的语音数据发送消息到该窗口。

## 2. 13. 2 CPC\_RECORD\_BUF\_HWND\_STOP 停止某个缓冲录音窗口回调

### 功能描述 (Description)

停止某个缓冲录音窗口回调

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

窗口句柄值

[in] *nValueEx* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)



无

例如 (Example)

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_HWND_STOP,
               (CPCINT32)m_hWnd,
               0,
               0);
```

### 2. 13. 3 CPC\_RECORD\_BUF\_HWND\_STOPALL 停止全部缓冲录音窗口回调

功能描述 (Description)

停止全部缓冲录音窗口回调

参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

窗口句柄值

[in] *nValueEx* - 保留

[in] *pValue* - 保留

返回值 (Return Values)

大于 0 表示成功,其它表示失败.

备注 (Remarks)

无

例如 (Example)

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_HWND_STOPALL,
               0,
               0,
               0);
```

### 2. 13. 4 CPC\_RECORD\_BUF\_CALLBACK\_START 开始缓冲录音回调函数

功能描述 (Description)

开始缓冲录音回调函数

参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

回调函数的用户自定义数值

[in] *nValueEx*

回调语音流的属性

- RECORD\_MASK\_PAUSE - 暂停回调



- RECORD\_MASK\_ECHO      - 回音抵消后的数据
- RECORD\_MASK\_AGC        - 自动增益处理后的数据

[in] *pValue*

回调函数地址(PCallBack\_RecordBuf)

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)

开始回调后系统从通道中采集到的语音数据后将调用指定的回调函数

#### 例如 (Example)

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_CALLBACK_START,
               this,
               RECORD_MASK_ECHO,
               RecordBufCallback);
```

- this 为回调函数的自定义数据(C++里一般使用 this)
- 系统从通道 1 采集到回音抵消后的语音数据将回调到指定的函数。

## 2. 13. 5 CPC\_RECORD\_BUF\_CALLBACK\_STOP 停止缓冲录音回调函数

#### 功能描述 (Description)

停止缓冲录音回调函数

#### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

回调函数的用户自定义数值

[in] *nValueEx*      - 保留

[in] *pValue*

回调函数地址(PCallBack\_RecordBuf)

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_CALLBACK_STOP,
               this,
               0,
               RecordBufCallback);
```

- this 为回调函数的自定义数据,必须跟  
CPC\_RECORD\_BUF\_CALLBACK\_START 一样
- RecordBufCallback 必须跟  
CPC\_RECORD\_BUF\_CALLBACK\_START 一样。



## 2. 13. 6 CPC\_RECORD\_BUF\_CALLBACK\_STOPALL 停止全部缓冲录音回调函数

### 调函数

#### 功能描述 (Description)

停止全部缓冲录音回调函数

#### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue* – 保留

[in] *nValueEx* – 保留

[in] *pValue* – 保留

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_RecordBuf(1,  
               CPC_RECORD_BUF_CALLBACK_STOPALL,  
               0,  
               0,  
               0);
```

## 2. 13. 7 CPC\_RECORD\_BUF\_SETCBSAMPLES 设置回调采样数

#### 功能描述 (Description)

设置回调采样数

#### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx*

回调采样数

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)



每秒 8K,如果需要 20ms 回调一次就设置为  $20 \times 8 = 160$ ,默认为 20ms 回调一次

#### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_SETCBSAMPLES,
               (CPCINT32)m_hWnd,
               160,
               0);
```

函数回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_SETCBSAMPLES,
               this,
               160,
               RecordBufCallback);
```

## 2. 13. 8 CPC\_RECORD\_BUF\_GETCBSAMPLES 获取设置的回调采样数

### 功能描述 (Description)

获取设置的回调采样数

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx* - 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

### 返回值 (Return Values)

小于 0 失败, 其它返回设置的采样数.

### 备注 (Remarks)

无

### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_GETCBSAMPLES,
               (CPCINT32)m_hWnd,
               0,
               0);
```

函数回调:

```
CPC_RecordBuf(1,
```



```
CPC_RECORD_BUF_GETCBSAMPLES,  
this,  
0,  
RecordBufCallback);
```

## 2. 13. 9 CPC\_RECORD\_BUF\_ENABLEECHO 打开/关闭回音抵消后的语音数据

### 功能描述 (Description)

打开/关闭回音抵消后的语音数据

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx*

0 – 回调的语音不经过回音抵消

1 – 回调的语音经过回音抵消

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败。

### 备注 (Remarks)

无

### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,  
CPC_RECORD_BUF_ENABLEECHO,  
(CPCINT32)m_hWnd,  
1,  
0);
```

函数回调:

```
CPC_RecordBuf(1,  
CPC_RECORD_BUF_ENABLEECHO,  
this,  
1,  
RecordBufCallback);
```

- 设置回调的语音经过回音抵消



## 2. 13. 10 CPC\_RECORD\_BUF\_ISENABLEECHO 检测是否使用了回音抵消后的语音数据

### 功能描述 (Description)

检测是否使用了回音抵消后的语音数据

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx* — 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

### 返回值 (Return Values)

0 — 没有使用回音抵消后的语音数据

1 — 使用了回音抵消后的语音数据

其它 — 失败

### 备注 (Remarks)

无

### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_ISENABLEECHO,
               (CPCINT32)m_hWnd,
               0,
               0);
```

函数回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_ISENABLEECHO,
               this,
               0,
               RecordBufCallback);
```

## 2. 13. 11 CPC\_RECORD\_BUF\_PAUSE 暂停缓冲录音

### 功能描述 (Description)

暂停缓冲录音

### 参数介绍 (Parameters)

[in] *nChannelID*



语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx* — 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败。

#### 备注 (Remarks)

无

#### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_PAUSE,
               (CPCINT32)m_hWnd,
               0,
               0);
```

函数回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_PAUSE,
               this,
               0,
               RecordBufCallback);
```

## 2. 13. 12 CPC\_RECORD\_BUF\_ISPAUSE 检测是否暂停缓冲录音

#### 功能描述 (Description)

检测是否暂停缓冲录音

#### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx* — 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

#### 返回值 (Return Values)





- 小于 0 — 检测失败
- 0 — 没有暂停缓冲录音
- 1 — 暂停了缓冲录音

**备注 (Remarks)**

无

**例如 (Example)**

窗口消息回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_ISPAUSE,
                (CPCINT32)m_hWnd,
                0,
                0);
```

函数回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_ISPAUSE,
                this,
                0,
                RecordBufCallback);
```

## 2. 13. 13 CPC\_RECORD\_BUF\_RESUME 恢复缓冲录音

**功能描述 (Description)**

恢复缓冲录音

**参数介绍 (Parameters)**

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx* — 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败。

**备注 (Remarks)**

无

**例如 (Example)**

窗口消息回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_RESUME,
                (CPCINT32)m_hWnd,
```



```
0,  
0);
```

函数回调:

```
CPC_RecordBuf(1,  
CPC_RECORD_BUF_RESUME,  
this,  
0,  
RecordBufCallback);
```

## 2. 13. 14 CPC\_RECORD\_BUF\_SETVOLUME 设置缓冲录音音量

### 功能描述 (Description)

设置缓冲录音音量

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx*

音量值。范围(0-10000)

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败。

### 备注 (Remarks)

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示回调的都为静音, 如果为 10000 就表示把音量放大 100 倍。

### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,  
CPC_RECORD_BUF_SETVOLUME,  
(CPCINT32)m_hWnd,  
200,  
0);
```

函数回调:

```
CPC_RecordBuf(1,  
CPC_RECORD_BUF_SETVOLUME,  
this,  
200,
```



RecordBufCallback);

## 2. 13. 15 CPC\_RECORD\_BUF\_GETVOLUME 获取缓冲录音音量

### 功能描述 (Description)

获取缓冲录音音量

### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx* — 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

### 返回值 (Return Values)

如果返回小于 0 表示失败, 其它返回音量值

### 备注 (Remarks)

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示回调的都为静音, 如果为 10000 就表示把音量放大 100 倍。

### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_GETVOLUME,
                (CPCINT32)m_hWnd,
                0,
                0);
```

函数回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_GETVOLUME,
                this,
                0,
                RecordBufCallback);
```

## 2. 13. 16 CPC\_RECORD\_BUF\_SETWAVEFORMAT 设置录音回调的语音编码格式

### 功能描述 (Description)



设置录音回调的语音编码格式 ID

#### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值

[in] *nValueEx*

CC301 定义的语音编码 ID

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败。

#### 备注 (Remarks)

无

#### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_SETWAVEFORMAT,
               (CPCINT32)m_hWnd,
               BRI_WAV_FORMAT_ALAW8K,
               0);
```

函数回调:

```
CPC_RecordBuf(1,
               CPC_RECORD_BUF_SETWAVEFORMAT,
               this,
               BRI_WAV_FORMAT_ALAW8K,
               RecordBufCallback);
```

## 2. 13. 17 CPC\_RECORD\_BUF\_GETWAVEFORMAT 获取录音回调的语音编码格式

#### 功能描述 (Description)

获取录音回调的语音编码格式 ID

#### 参数介绍 (Parameters)

[in] *nChannelID*

语音缓冲的通道

[in] *nValue*

如果要设置为窗口消息回调,该值为窗口句柄值。

如果要设置为回调函数,该值就为

CPC\_RECORD\_BUF\_CALLBACK\_START 时的用户自定义值



[in] *nValueEx* — 保留

[in] *pValue*

如果要设置为窗口消息回调,该值为 0。

如果要设置为回调函数,该值就为回调函数地址。

#### 返回值 (Return Values)

小于 0 表示失败, 其它返回语音编码格式 ID

#### 备注 (Remarks)

无

#### 例如 (Example)

窗口消息回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_GETWAVEFORMAT,
                (CPCINT32)m_hWnd,
                0,
                0);
```

函数回调:

```
CPC_RecordBuf(1,
                CPC_RECORD_BUF_GETWAVEFORMAT,
                this,
                0,
                RecordBufCallback);
```

## 2. 14 通道语音会议控制 (CPC\_Conference)

### 功能描述 (Description)

多个通道之间语音会议控制

### 函数原型 (Function)

```
CPCINT32 CPC_Conference (/*[in]*/CPCINT16 nChannelID,
                          /*[in]*/CPCINT32 nConfID,
                          /*[in]*/CPCUINT32 uConfType,
                          /*[in]*/CPCINT32 nValue,
                          /*[in,out]*/CPCCHAR8 pValue);
```

### 返回值 (Return Values)

有符号 32bit 位.



## 2. 14. 1 CPC\_CONFERENCE\_CREATE 创建会议

### 功能描述 (Description)

创建会议

### 参数介绍 (Parameters)

[in] *nChannelID*

创建会议自动加入到会议中的通道

[in] *nConfID* – 保留

[in] *nValue*

自动加入到该会议中的通道属性

*RECORD\_MASK\_PAUSE* – 暂停跟会议中的其它通道语音交互

*CONFERENCE\_MASK\_DISABLEMIC* – 暂停该通道说的功能

*CONFERENCE\_MASK\_DISABLESPK* – 暂停该通道听的功能

[in] *pValue* – 保留

### 返回值 (Return Values)

小于等于 0 表示失败,其它返回创建成功的会议 ID

### 备注 (Remarks)

用户必须保存该 ID, 以后对该会议的控制都要使用该 ID

### 例如 (Example)

```
CPCINT32 i32ConfID=CPC_Conference(1,
                                     0,
                                     CPC_CONFERENCE_CREATE,
                                     CONFERENCE_MASK_DISABLEMIC,
                                     0);
```

- 创建一个会议, 把通道 1 加入到该会议中, 并禁止通道 1 的说功能,
- 也就是以后加入到会议中的通道不能听到通道 1 说话, 但通道 1 可以听到会议中其它通道说话。

## 2. 14. 2 CPC\_CONFERENCE\_ADDTOCONF 增加通道到指定会议中

### 功能描述 (Description)

增加通道到指定会议中

### 参数介绍 (Parameters)

[in] *nChannelID*

加入到会议中的通道

[in] *nConfID*

通道需要加入的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

加入到该会议中的通道属性



*RECORD\_MASK\_PAUSE* – 暂停跟会议中的其它通道语音交互

*CONFERENCE\_MASK\_DISABLEMIC* – 暂停该通道说的功能

*CONFERENCE\_MASK\_DISABLESPK* – 暂停该通道听的功能

[in] *pValue* – 保留

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_ADDTOCONF,  
                CONFERENCE_MASK_DISABLESPK,  
                0);
```

- 把通道 2 加入到指定会议中,并禁止该通道听到会议中其它通道说话。

## 2. 14. 3 CPC\_CONFERENCE\_GETCONFID 获取某个通道所在的会议 ID

#### 功能描述 (Description)

获取某个通道所在的会议 ID

#### 参数介绍 (Parameters)

[in] *nChannelID*

需要获取会议 ID 的通道

[in] *nConfID* – 保留

[in] *nValue* – 保留

[in] *pValue* – 保留

#### 返回值 (Return Values)

返回小于等于 0 表示失败,其它返回通道所在的会议 ID.

#### 备注 (Remarks)

如果该通道被加入了多个会议, 返回第一个加入的会议 ID

#### 例如 (Example)

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_GETCONFID,  
                0,  
                0);
```

- 获取通道 2 所在会议 ID。



## 2. 14. 4 CPC\_CONFERENCE\_SETSPKVOLUME 设置会议中某个通道播放音量

### 功能描述 (Description)

设置会议中某个通道播放音量

### 参数介绍 (Parameters)

[in] *nChannelID*

设置播放音量的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

播放音量值(0-10000)

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

### 例如 (Example)

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_SETSPKVOLUME,  
                200,  
                0);
```

- 设置会议 ID 为 i32ConfID 中的通道 2 的放音音量为 200。

## 2. 14. 5 CPC\_CONFERENCE\_GETSPKVOLUME 获取会议中某个通道播放音量

### 功能描述 (Description)

获取会议中某个通道播放音量

### 参数介绍 (Parameters)

[in] *nChannelID*

获取播放音量的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

小于 0 表示失败, 其它返回播放的音量.



**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_GETSPKVOLUME,  
                0,  
                0);
```

## 2. 14. 6 CPC\_CONFERENCE\_SETMICVOLUME 设置会议中某个通道说话音量

**功能描述 (Description)**

设置会议中某个通道说话音量

**参数介绍 (Parameters)**

[in] *nChannelID*

设置说话音量的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

说话音量值(0-10000)

[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

- 默认音量为 100。
- 如果为 200 表示把音量放大一倍,如果为 50 就表示缩小一半,如果设置为 0 就表示为静音, 如果为 10000 就表示把音量放大 100 倍。

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_SETMICVOLUME,  
                200,  
                0);
```

- 设置会议 ID 为 i32ConfID 中的通道 2 的说话音量为 200。

## 2. 14. 7 CPC\_CONFERENCE\_GETMICVOLUME 获取会议中某个通道说话音量

**功能描述 (Description)**

获取会议中某个通道说话音量



#### 参数介绍 (Parameters)

[in] *nChannelID*

获取说话音量的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

#### 返回值 (Return Values)

小于 0 表示失败, 其它返回说话音量值

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_Conference(2,
                i32ConfID,
                CPC_CONFERENCE_GETSPKVOLUME,
                0,
                0);
```

## 2. 14. 8 CPC\_CONFERENCE\_PAUSE 暂停某个会议

#### 功能描述 (Description)

暂停某个会议

#### 参数介绍 (Parameters)

[in] *nChannelID* – 保留

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败.

#### 备注 (Remarks)

暂停会议后所有在该会议中通道都将不能说也不能听

#### 例如 (Example)

```
CPC_Conference(2,
                i32ConfID,
                CPC_CONFERENCE_PAUSE,
                0,
                0);
```



## 2. 14. 9 CPC\_CONFERENCE\_RESUME 恢复某个会议

### 功能描述 (Description)

恢复某个会议

### 参数介绍 (Parameters)

[in] *nChannelID* – 保留

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_RESUME,  
                0,  
                0);
```

## 2. 14. 10 CPC\_CONFERENCE\_ISPAUSE 检测某个会议是否被暂停了

### 功能描述 (Description)

检测某个会议是否被暂停了

### 参数介绍 (Parameters)

[in] *nChannelID* – 保留

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

小于 0 – 检测失败

0 – 没有暂停会议

1 – 已暂停会议

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_ISPAUSE,
```



```
0,  
0);
```

## 2. 14. 11 CPC\_CONFERENCE\_ENABLESPK 打开/关闭会议中某个通道听功能

### 功能描述 (Description)

打开/关闭会议中某个通道听功能

### 参数介绍 (Parameters)

[in] *nChannelID*

需要打开/关闭听功能的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

0 – 关闭该会议中该通道听功能

1 – 打开该会议中该通道听功能

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)

关闭后该通道将听不到会议中其它通道的说话声。

### 例如 (Example)

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_ENABLESPK,  
                1,  
                0);
```

- 打开会议 i32ConfID 中通道 2 的听功能。

## 2. 14. 12 CPC\_CONFERENCE\_IENABLESPK 检测会议听功能是否打开

### 功能描述 (Description)

检测会议听功能是否打开

### 参数介绍 (Parameters)

[in] *nChannelID*

需要检测的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

- 0 – 没有打开听功能
- 1 – 已经打开听功能
- 其它 – 失败.

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_ISENABLESPK,  
                1,  
                0);
```

## 2. 14. 13 CPC\_CONFERENCE\_ENABLEMIC 打开/关闭会议者说功能

**功能描述 (Description)**

打开/关闭会议者说功能

**参数介绍 (Parameters)**

[in] *nChannelID*

需要打开/关闭说功能的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

0 – 关闭该会议中该通道说功能

1 – 打开该会议中该通道说功能

[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

关闭后会议中其它通道都将听不到该通道的说话声。

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_ENABLEMIC,  
                1,  
                0);
```

- 打开会议 i32ConfID 中通道 2 的说功能。

## 2. 14. 14 CPC\_CONFERENCE\_ISENABLEMIC 检测会议者说功能是否打开

**功能描述 (Description)**



检测会议者说功能是否打开

**参数介绍 (Parameters)**

[in] *nChannelID*

需要检测的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

0 – 没有打开说功能

1 – 已经打开说功能

其它 – 失败

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_IENABLEMIC,  
                0,  
                0);
```

## 2. 14. 15 *CPC\_CONFERENCE\_ENABLEAGC* 打开/关闭会议中的通道自动

### 增益

**功能描述 (Description)**

.打开/关闭会议中的通道自动增益

**参数介绍 (Parameters)**

[in] *nChannelID*

需要打开/关闭自动增益的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

0 – 关闭该会议中该通道自动增益功能。

1 – 打开该会议中该通道自动增益功能

[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

打开后该通道说话的声音被进行自动增益处理后在发送到会议中的其它通道

**例如 (Example)**

```
CPC_Conference(2,
```



```
i32ConfID,  
CPC_CONFERENCE_ENABLEAGC,  
1,  
0);
```

- 打开会议 i32ConfID 中通道 2 的自动增益功能。

## 2. 14. 16 CPC\_CONFERENCE\_ISENABLEAGC 检测是否打开了自动增益

### 功能描述 (Description)

检测是否打开了自动增益。

### 参数介绍 (Parameters)

[in] *nChannelID*

需要检测的通道

[in] *nConfID*

通道所在的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

0 – 没有打开自动增益功能

1 – 已经打开自动增益功能

其它 – 失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_Conference(2,  
i32ConfID,  
CPC_CONFERENCE_ISENABLEAGC,  
0,  
0);
```

## 2. 14. 17 CPC\_CONFERENCE\_DELETECHANNEL 把通道从会议中删除

### 功能描述 (Description)

把通道从会议中删除。

### 参数介绍 (Parameters)

[in] *nChannelID*

需要删除的通道

[in] *nConfID*

通道所在的会议 ID.

使用 *CPC\_CONFERENCE\_CREATE* 返回的会议 ID.

[in] *nValue* – 保留



[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_DELETECHANNEL,  
                0,  
                0);
```

- 把通道 2 从会议 i32ConfID 中删除

## 2. 14. 18 CPC\_CONFERENCE\_DELETECONF 删除一个会议

**功能描述 (Description)**

删除一个会议

**参数介绍 (Parameters)**

[in] *nChannelID* – 保留

[in] *nConfID*

需要删除的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

所有在该会议中的通道将被从该会议中删除

**例如 (Example)**

```
CPC_Conference(2,  
                i32ConfID,  
                CPC_CONFERENCE_DELETECONF,  
                0,  
                0);
```

## 2. 14. 19 CPC\_CONFERENCE\_DELETEALLCONF 删除所有会议

**功能描述 (Description)**

删除所有会议

**参数介绍 (Parameters)**

[in] *nChannelID* – 保留

[in] *nConfID* – 保留





[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(0,  
                i32ConfID,  
                CPC_CONFERENCE_DELETEALLCONF,  
                0,  
                0);
```

## 2. 14. 20 CPC\_CONFERENCE\_GETCONFCOUNT 获取会议数量

**功能描述 (Description)**

获取会议数量

**参数介绍 (Parameters)**

[in] *nChannelID* – 保留

[in] *nConfID* – 保留

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(0,  
                i32ConfID,  
                CPC_CONFERENCE_GETCONFCOUNT,  
                0,  
                0);
```

## 2. 14. 21 CPC\_CONFERENCE\_RECORD\_START 开始会议录音

**功能描述 (Description)**

开始会议录音

**参数介绍 (Parameters)**

[in] *nChannelID* – 保留

[in] *nConfID*

需要录音的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

录音文件的语音编码格式 ID

[in] *pValue*

保存录音的文件路径

- 使用绝对路径。

如: "c:\wave\a.wav"。

- 使用相对路径。

如: "a.wav", 系统将自动定为该文件为本进程所在目录下的 a.wav。

- 该文件路径支持网络共享文件路径:

如: [\\192.168.0.1\wav\a.wav](http://192.168.0.1/wav/a.wav)**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

建议用户使用绝对路径的方式

如果保存录音的目录不存在, 系统将自动创建目录

如果保存录音的文件已经存在, 系统将删除原文件

录音文件的语音编码格式 ID 查看 CC301 的定义"语音文件编码格式 ID"

**例如 (Example)**

```
CPC_Conference(0,
                i32ConfID,
                CPC_CONFERENCE_RECORD_START,
                1,
                "c:\wave\a.wav");
```

- 使用 ALAW 格式开始给会议 i32ConfID 录音.

## 2. 14. 22 CPC\_CONFERENCE\_RECORD\_PAUSE 暂停会议录音

**功能描述 (Description)**

暂停会议录音

**参数介绍 (Parameters)**[in] *nChannelID* – 保留[in] *nConfID*

需要暂停录音的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的[in] *nValue* – 保留[in] *pValue* – 保留**返回值 (Return Values)**

大于 0 表示成功,其它表示失败.

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(0,
```



```
i32ConfID,  
CPC_CONFERENCE_RECORD_PAUSE,  
0,  
0);
```

## 2. 14. 23 CPC\_CONFERENCE\_RECORD\_RESUME 恢复会议录音

### 功能描述 (Description)

恢复会议录音

### 参数介绍 (Parameters)

[in] *nChannelID* – 保留

[in] *nConfID*

需要恢复录音的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_Conference(0,  
i32ConfID,  
CPC_CONFERENCE_RECORD_RESUME,  
0,  
0);
```

## 2. 14. 24 CPC\_CONFERENCE\_RECORD\_ISPAUSE 检测是否暂停会议录音

### 功能描述 (Description)

检测是否暂停会议录音

### 参数介绍 (Parameters)

[in] *nChannelID* – 保留

[in] *nConfID*

检测是否暂停录音的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

小于 0 – 检测失败

0 – 没有暂停会议录音

1 – 已经暂停了会议录音

**备注 (Remarks)**

如果没有启动录音返回 0

**例如 (Example)**

```
CPC_Conference(0,
                i32ConfID,
                CPC_CONFERENCE_RECORD_ISPAUSE,
                0,
                0);
```

## 2. 14. 25 CPC\_CONFERENCE\_RECORD\_ISSTART 检测是否启动了会议录音

**功能描述 (Description)**

检测是否启动了会议录音

**参数介绍 (Parameters)**

[in] *nChannelID* – 保留

[in] *nConfID*

检测是否启动了录音的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

0 – 没有启动会议录音

1 – 已经启动了会议录音

其它 – 失败

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_Conference(0,
                i32ConfID,
                CPC_CONFERENCE_RECORD_ISSTART,
                0,
                0);
```

## 2. 14. 26 CPC\_CONFERENCE\_RECORD\_FILEPATH 获取会议录音文件路径

**功能描述 (Description)**

获取会议录音文件路径

**参数介绍 (Parameters)**

[in] *nChannelID* – 保留

[in] *nConfID*



需要获取录音文件路径的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*

保存文件路径的内存长度,也就为 *pValue* 分配的内存长度。

[out] *pValue*

需要保存文件路径的内存地址

#### 返回值 (Return Values)

大于 0 表示成功, 长度为文件路径的长度,其它表示失败.

#### 备注 (Remarks)

建议给 *pValue* 分配足够大的内存(大于 256 字节),避免分配的内存长度不够而返回失败。

#### 例如 (Example)

```
char  szBuf[260];//定义一个 260 字节的内存
memset(szBuf,0,260);//初始化为空
CPC_Conference(0,
                i32ConfID,
                CPC_CONFERENCE_RECORD_FILEPATH,
                260,
                szBuf);
```

## 2. 14. 27 CPC\_CONFERENCE\_RECORD\_STOP 停止指定会议录音

#### 功能描述 (Description)

停止指定会议录音

#### 参数介绍 (Parameters)

[in] *nChannelID*      - 保留

[in] *nConfID*

需要停止录音的会议 ID

会议 ID 必须是使用 *CPC\_CONFERENCE\_CREATE* 返回的

[in] *nValue*      - 保留

[in] *pValue*      - 保留

#### 返回值 (Return Values)

大于 0 表示成功,其它表示失败.

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_Conference(0,
                i32ConfID,
                CPC_CONFERENCE_RECORD_STOP,
                0,
                0);
```



## 2. 14. 28 CPC\_CONFERENCE\_RECORD\_STOPALL 停止全部会议录音

### 功能描述 (Description)

停止全部会议录音

### 参数介绍 (Parameters)

[in] *nChannelID* – 保留  
[in] *nConfID* – 保留  
[in] *nValue* – 保留  
[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败.

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_Conference(0,  
                i32ConfID,  
                CPC_CONFERENCE_RECORD_STOPALL,  
                0,  
                0);
```

## 2. 15 事件获取 (CPC\_Event)

### 功能描述 (Description)

获取事件

### 函数原型 (Function)

```
CPCINT32 CPC_Event ([in]CPCINT32 nChannelID,  
                    [in]CPCUINT32 uEventType,  
                    [in]CPCINT32 nValue,  
                    [in]CPCCHAR8 pInValue,  
                    [out]CPCCHAR8 pOutValue,  
                    [in]CPCINT32 lBufSize);
```

### 返回值 (Return Values)

有符号 32bit 位.

### 参数描述 (Parameters description)



## 2. 15. 1 CPC\_EVENT\_POP 获取事件缓冲里最早的一个事件

### 功能描述 (Description)

获取事件缓冲里最早的一个事件,获取后系统会自动删除当前事件

### 参数介绍 (Parameters)

[in] *nChannelID*

事件所在通道号

[in] *nValue*      – 保留

[in] *pInValue*    – 保留

[out] *pOutValue*

保存事件内容的内存地址, 该地址为指向 PBRI\_EVENT 结构

[in] *lBufSize*    – 保留

### 返回值 (Return Values)

小于 0      – 表示失败

等于 0      – 表示没有事件

大于 0      – 表示有事件, 具体事件信息放在(PBRI\_EVENT)pOutValue

### 备注 (Remarks)

- 具体事件信息的定义请查看 BRI\_EVENT 结构介绍
- 一般应用层使用定时调用该功能来获取所有事件
- 一般定时间隔为 200ms 左右比较合适
- 该功能跟 CPC\_EVENT\_POPEX 同时只需要使用其中一个就可以

### 例如 (Example)

```
BRI_EVENT event; //保存事件信息的结构地址
memset((void*)&event,0,sizeof(event));
CPC_Event(0, CPC_EVENT_POP, NULL ,(char*)& event,0);
```

## 2. 15. 2 CPC\_EVENT\_POPEX 获取事件缓冲里最早的一个事件

### 功能描述 (Description)

获取事件缓冲里最早的一个事件,获取后系统会自动删除当前事件

### 参数介绍 (Parameters)

[in] *nChannelID*

事件所在通道号

[in] *nValue*      – 保留

[in] *pInValue*    – 保留

[out] *pOutValue*

保存事件内容的内存地址, 该地址为指向字符内存缓冲

[in] *lBufSize*

指定 pOutValue 缓冲分配的内存大小

### 返回值 (Return Values)

小于 0      – 表示失败

等于 0      – 表示没有事件

大于 0      – 表示有事件,事件信息保存在 pOutValue 里

**备注 (Remarks)**

- 保存的事件信息按逗号分隔.格式为:  
通道号, 事件类型, 事件句柄, 事件数值, 事件数据, 事件附加数据
- 一般应用层使用定时调用该功能来获取所有事件
- 一般定时间隔为 200ms 左右比较合适
- 该功能跟 CPC\_EVENT\_POP 同时只需要使用其中一个就可以

**例如 (Example)**

```
char  szEventBuf[1024];//分配足够大的缓冲
memset((void*)&event,0,sizeof(event));
CPC_Event(0, CPC_EVENT_POPEX, 0, NULL,szEventBuf,1024);
```

## 2. 15. 3 CPC\_EVENT\_TYPE 获取事件类型

**功能描述 (Description)**

获取事件类型,如果缓冲有事件,获取最早一个事件, 但系统不会自动删除该事件

**参数介绍 (Parameters)**

[in] *nChannelID*  
事件所在通道号

[in] *nValue*        - 保留

[in] *pInValue*      - 保留

[out] *pOutValue*   - 保留

[in] *lBufSize*      - 保留

**返回值 (Return Values)**

小于 0        - 表示失败

等于 0        - 表示没有事件

大于 0        - 表示有事件, 该值就为事件类型

**备注 (Remarks)**

- 获取足够信息后如果想删除该事件, 请调用 CPC\_EVENT\_REMOVE 来删除该事件, 如果不删除, 该事件一直保存在队列里下次获取时还是该事件值
- 一般应用层使用定时调用该功能来获取所有事件
- 一般定时间隔为 200ms 左右比较合适

**例如 (Example)**

```
CPCINT32 i32EventType=CPC_Event(0, CPC_EVENT_TYPE,0 ,NULL,NULL,0);
if(i32EventType > 0)
{
    CPC_Event(0, CPC_EVENT_REMOVE,0 , NULL,NULL,0);//删除该事件
}
```

## 2. 15. 4 CPC\_EVENT\_HANDLE 获取事件句柄

**功能描述 (Description)**

获取事件句柄,如果缓冲有事件, 返回最早一个事件的句柄值



**参数介绍 (Parameters)**

[in] *nChannelID*  
事件所在通道号

[in] *nValue* – 保留

[in] *pInValue* – 保留

[out] *pOutValue* – 保留

[in] *lBufSize* – 保留

**返回值 (Return Values)**

小于 0 – 表示失败

其它 – 最早一个事件的句柄值

**备注 (Remarks)**

调用该功能前应该先调用 `CPC_EVENT_TYPE` 来获取事件类型,如果没有事件类型就没有必要调用这个来获取参数

**例如 (Example)**

```
CPCINT32 i32EventType=CPC_Event(0, CPC_EVENT_TYPE,0, NULL,NULL,0);
if(i32EventType > 0)
{
CPCINT32 i32EventResult=CPC_Event(0,CPC_EVENT_RESULT,0, NULL,NULL,0);
CPCINT32 i32EventHandle=CPC_Event(0,CPC_EVENT_HANDLE, 0, NULL,NULL,0);
CPC_Event(0,CPC_EVENT_REMOVE, 0, NULL,NULL,0);
//删除该事件
}
```

## 2. 15. 5 CPC\_EVENT\_RESULT 获取事件数值

**功能描述 (Description)**

获取事件数值,如果缓冲有事件, 返回最早一个事件的数值。

**参数介绍 (Parameters)**

[in] *nChannelID*  
事件所在通道号

[in] *nValue* – 保留

[in] *pInValue* – 保留

[out] *pOutValue* – 保留

[in] *lBufSize* – 保留

**返回值 (Return Values)**

小于 0 – 表示失败

其它 – 最早一个事件的数值

**备注 (Remarks)**

调用该功能前应该先调用 `CPC_EVENT_TYPE` 来获取事件类型,如果没有事件类型就没有必要调用这个来获取参数

**例如 (Example)**

```
CPCINT32 i32EventType=CPC_Event(0, CPC_EVENT_TYPE,0, NULL,NULL,0);
if(i32EventType > 0)
```



```

{
    CPCINT32 i32EventResult= CPC_Event(0, CPC_EVENT_RESULT,0 , NULL,NULL,0);
    CPC_Event(0, CPC_EVENT_REMOVE,0 , NULL,NULL,0);
    //删除该事件
}

```

## 2. 15. 6 CPC\_EVENT\_PARAM 获取事件参数

### 功能描述 (Description)

获取事件的参数,如果缓冲有事件, 返回最早一个事件的参数。

### 参数介绍 (Parameters)

[in] *nChannelID*  
事件所在通道号

[in] *nValue*        – 保留

[in] *pInValue*     – 保留

[out] *pOutValue* – 保留

[in] *lBufSize*     – 保留

### 返回值 (Return Values)

小于 0        – 表示失败

其它         – 最早一个事件的数值

### 备注 (Remarks)

调用该功能前应该先调用 CPC\_EVENT\_TYPE 来获取事件类型,如果没有事件类型就没有必要调用这个来获取参数

### 例如 (Example)

```

CPCINT32 i32EventType=CPC_Event(0, CPC_EVENT_TYPE,0 , NULL,NULL,0);
if(i32EventType > 0)
{
    CPCINT32 i32EventParam= CPC_Event(0, CPC_EVENT_PARAM,0 , NULL,NULL,0);
    CPC_Event(0, CPC_EVENT_REMOVE,0 , NULL,NULL,0);
    //删除该事件
}

```

## 2. 15. 7 CPC\_EVENT\_DATA 获取事件数据

### 功能描述 (Description)

获取事件数据,如果缓冲有事件, 返回最早一个事件的数据。

### 参数介绍 (Parameters)

[in] *nChannelID*  
事件所在通道号

[in] *nValue*        – 保留

[in] *pInValue*     – 保留

[out] *pOutValue*  
保存数据的内存地址



[in] *lBufSize*

分配的 pOutValue 内存缓冲的长度

#### 返回值 (Return Values)

小于 0      – 表示失败  
其它        – 返回事件数据的长度

#### 备注 (Remarks)

调用该功能前应该先调用 CPC\_EVENT\_TYPE 来获取事件类型,如果没有事件类型就没有必要调用这个来获取参数

#### 例如 (Example)

```
CPCINT32 i32EventType=CPC_Event(0, CPC_EVENT_TYPE,0 ,NULL,NULL,0,0);
if(i32EventType > 0)
{
CPCINT32 i32EventResult= CPC_Event(0, CPC_EVENT_RESULT,0 , NULL,NULL,0);
char szData[512];//分配足够大的内存
CPCINT32 i32DataLen=CPC_Event(0, CPC_EVENT_DATA, 0, NULL,szData,512);
CPC_Event(0, CPC_EVENT_REMOVE, 0, NULL,NULL ,0);
//删除该事件
}
```

## 2. 15. 8 CPC\_EVENT\_DATAEX 获取事件附加数据

#### 功能描述 (Description)

获取事件附加数据,如果缓冲有事件, 返回最早一个事件的附加数据。

#### 参数介绍 (Parameters)

[in] *nChannelID*

事件所在通道号

[in] *nValue*      – 保留

[in] *pInValue*    – 保留

[out] *pOutValue*

保存数据的内存地址

[in] *lBufSize*

分配的 pOutValue 内存缓冲的长度

#### 返回值 (Return Values)

小于 0      – 表示失败  
其它        – 返回事件数据的长度

#### 备注 (Remarks)

该内容保留,数据将一直为空

#### 例如 (Example)

```
CPCINT32 i32EventType=CPC_Event(0,CPC_EVENT_TYPE,0 , NULL,NULL,0);
if(i32EventType > 0)
{
CPCINT32 i32EventResult= CPC_Event(0,CPC_EVENT_RESULT,0 , NULL,NULL,0);
char szData[32];//分配足够大的内存
```



```
CPCINT32 i32DataLen=CPC_Event(0,CPC_EVENT_DATAEX,0, NULL,szData,32);
CPC_Event(0,CPC_EVENT_REMOVE,0, NULL,NULL,0);
//删除该事件
}
```

## 2. 15. 9 CPC\_EVENT\_REMOVE 删除事件队列里最早的一个事件

### 功能描述 (Description)

删除事件队列里最早的一个事件

### 参数介绍 (Parameters)

[in] *nChannelID*  
事件所在通道号

[in] *nValue* – 保留

[in] *pInValue* – 保留

[out] *pOutValue* – 保留

[in] *lBufSize* – 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

在定时获取事件(CPC\_EVENT\_TYPE)信息后应该立即删除该事件，不然下次还是获取到该事件

### 例如 (Example)

```
CPCINT32 i32EventType=CPC_Event(0,CPC_EVENT_TYPE,0, NULL,NULL,0);
if(i32EventType > 0)
{
    CPC_Event(0,CPC_EVENT_REMOVE,0, NULL,NULL,0);
    //删除该事件
}
```

## 2. 15. 10 CPC\_EVENT\_REMOVEALL 删除事件队列里所有的事件

### 功能描述 (Description)

删除事件队列里所有的事件

### 参数介绍 (Parameters)

[in] *nChannelID*  
事件所在通道号

[in] *nValue* – 保留

[in] *pInValue* – 保留

[out] *pOutValue* – 保留

[in] *lBufSize* – 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 用户在定时获取数据,一般不需要调用该函数

**例如 (Example)**

```
CPC_Event(0,CPC_EVENT_REMOVEALL,0,0,0);
```

## 2. 15. 11 CPC\_EVENT\_REGWND 注册事件接收窗口

**功能描述 (Description)**

注册事件接收窗口,注册成功后当通道有事件时自动发送到该窗口,不再需要定时去读取事件

**参数介绍 (Parameters)**

[in] *nChannelID*

事件所在通道号

[in] *nValue*

接收事件的窗口句柄

[in] *pInValue* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 注册成功后当有事件发生时该窗口将接收到 `BRI_EVENT_MESSAGE` 消息, 该消息 ID 为 (`WM_USER+2000`)
- 消息中的第一个参数(`wParam`)保留为 0, 第二个(`lParam`)为 `PBRI_EVENT` 结构地址
- 如果使用该方式获取事件, 用户就需要在应用层创建一个窗口/Form, 在该窗口里接收事件, 跟定时获取事件方式相比更时时更节省系统资源。
- 如果使用定时方式就不需要该方式

**例如 (Example)**

```
CPC_Event(0,CPC_EVENT_REGWND,m_hWnd, NULL, NULL, 0);
```

## 2. 15. 12 CPC\_EVENT\_UNREGWND 删除事件接收窗口

**功能描述 (Description)**

删除事件接收窗口

**参数介绍 (Parameters)**

[in] *nChannelID*

事件所在通道号

[in] *nValue*

删除接收事件的窗口句柄

[in] *pInValue* - 保留

[out] *pOutValue* - 保留



[in] *lBufSize*     – 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

使用 *CPC\_EVENT\_REGWND* 注册的窗口, 如果不再需要接收事件或者窗口被销毁前调用该功能来删除

**例如 (Example)**

```
CPC_Event(0,CPC_EVENT_UNREGWND,m_hWnd ,NULL,NULL,0);
```

## 2. 15. 13 CPC\_EVENT\_REGCBFUNC 注册接收事件的回调函数

**功能描述 (Description)**

注册接收事件的回调函数,注册成功后当通道有事件时会调用该回调函数来通知具体事件信息

**参数介绍 (Parameters)**

[in] *nChannelID*

事件所在通道号

[in] *nValue*

回调函数的用户自定义参数

[in] *pInValue*

回调函数(PCallBack\_Event)地址

[out] *pOutValue*   – 保留

[in] *lBufSize*     – 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 回调函数(PCallBack\_Event)的原型请查看 CC301 的定义
- 如果使用该方式, 就不需要窗口模式和定时获取事件方式,也就是说获取事件的方式有 3 中, 一种为定时查询模式, 一种为窗口消息模式, 一种为回调函数模式, 这 3 种方式里只要选择其中一种获取事件的方式就可以了。
- 注册后需要删除时使用 *CPC\_EVENT\_UNREGCBFUNC* 进行删除

**例如 (Example)**

```
CPC_Event(0,CPC_EVENT_REGCBFUNC,this,(char*)CallBackEventFunc,NULL,0);
```

## 2. 15. 14 CPC\_EVENT\_REGCBFUNCX 注册接收事件的回调函数

**功能描述 (Description)**

注册接收事件的回调函数,注册成功后当通道有事件时会调用该回调函数来通知具体事件信息

**参数介绍 (Parameters)**

[in] *nChannelID*

事件所在通道号



[in] *nValue*

回调函数的用户自定义参数

[in] *pInValue*

回调函数(PCallBack\_EventEx)地址

[out] *pOutValue* – 保留

[in] *lBufSize* – 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 回调函数(PCallBack\_EventEx)的原型请查看 CC301 的定义
- 如果使用该方式, 就不需要窗口模式和定时获取事件方式, 也就是说获取事件的方式有 3 中, 一种为定时查询模式, 一种为窗口消息模式, 一种为回调函数模式, 这 3 种方式里只要选择其中一种获取事件的方式就可以了。
- 注册后需要删除时使用 CPC\_EVENT\_UNREGCBFUNC 进行删除
- 跟 CPC\_EVENT\_REGCBFUNC 区别就是回调函数的参数类型不一样  
CPC\_EVENT\_REGCBFUNC 的回调方式为多个参数  
CPC\_EVENT\_REGCBFUNCEX 的回调方式使用结构体参数

#### 例如 (Example)

```
CPC_Event(0,CPC_EVENT_REGCBFUNCEX,this,(char*) CallBackEventFuncEx,NULL,0);
```

## 2. 15. 15 CPC\_EVENT\_UNREGCBFUNC 删除接收事件的回调函数

#### 功能描述 (Description)

删除接收事件的回调函数

#### 参数介绍 (Parameters)

[in] *nChannelID*

事件所在通道号

[in] *nValue*

回调函数的用户自定义参数(必须跟注册时一样)

[in] *pInValue*

回调函数(PCallBack\_Event)地址(必须跟注册时一样)

[out] *pOutValue* – 保留

[in] *lBufSize* – 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

在应用层销毁回调函数前(回调函数地址无效前),必须调用该功能删除回调函数。  
如: 在销毁回调函数所在的类前必须调用该功能删除回调函数,不然有可能会导  
致系统调用无效回调函数地址而崩溃。

#### 例如 (Example)

```
CPC_Event(0,CPC_EVENT_UNREGCBFUNC,this,(char*) CallBackEventFunc,NULL,0);
```



## 2. 16 通用功能 (CPC\_General)

### 功能描述 (Description)

设备通用功能控制

### 函数原型 (Function)

```
CPCINT32 CPC_General (/*[in]*/CPCINT16 nChannelID,  
                      /*[in]*/CPCUINT32 uGeneralType,  
                      /*[in]*/CPCINT32 nValue,  
                      /*[in]*/CPCCHAR8 *pValue);
```

### 返回值 (Return Values)

有符号 32bit 位.

### 参数描述 (Parameters description)

### 2. 16. 1 CPC\_GENERAL\_STARTDIAL 开始拨号

#### 功能描述 (Description)

开始拨号

#### 参数介绍 (Parameters)

[in] nChannelID

通用控制通道号

[in] nValue

拨号方式(选择其中一种)

DIALTYPE\_DTMF - DTMF 方式拨号

DIALTYPE\_FSK - FSK 方式拨号

[in] pValue

需要拨的号码.如: (13810000000)

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 使用 DTMF 方式拨号时一个逗号延迟 1 秒,“,01082896766”
- 调用该功能前确认线路是空闲可拨号状态
- 调用该功能后系统将自动按如下步骤进行拨号:
  - 1.如果当前还没有软摘机, 系统自动调用调用软摘机  
CPC\_SetDevCtrl(0, CPC\_CTRL\_DOHOOK,1);
  - 2.开始检测拨号音
  - 3.如果检测到拨号音或者检测拨号音超时开始发送号码到线路





4.发送号码结束后系统调用(BriEvent\_DialEnd)事件给用户,该事件的 Result 为 DTT-DIAL

- 一般使用 DTMF 方式拨号
- 非特殊需求不要使用 FSK 方式拨号

例如 (Example)

```
CPC_General(0,
             CPC_GENERAL_STARTDIAL,
             DIALTYPE_DTMF,
             "01082895766");
```

接通线路后二次拨号

## 2. 16. 2 CPC\_GENERAL\_SENDDNUMBER 接通线路后二次拨号

功能描述 (Description)

接通线路后二次拨号

参数介绍 (Parameters)

[in] nChannelID

通用控制通道号

[in] nValue

拨号方式(选择其中一种)

DIALTYPE\_DTMF - DTMF 方式拨号

DIALTYPE\_FSK - FSK 方式拨号

[in] pValue

需要发送的二次号码.如: (1099)

返回值 (Return Values)

大于 0 表示成功, 其它表示失败

备注 (Remarks)

- 使用 DTMF 方式时发送号码时一个逗号延迟 1 秒, „1099”
- 调用该功能前确认线路已经接通
- 发送号码结束后系统调用(BriEvent\_DialEnd)事件给用户,该事件的 Result 为 DTT-SEND
- 一般使用 DTMF 方式拨号
- 非特殊需求不要使用 FSK 方式拨号

例如 (Example)

```
CPC_General(0,
             CPC_GENERAL_SENDDNUMBER,
             DIALTYPE_DTMF,
             "1099");
```

## 2. 16. 3 CPC\_GENERAL\_REDIAL 重拨最后一次呼叫的号码

功能描述 (Description)



重拨最后一次呼叫的号码,程序退出后该号码被释放

#### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

拨号方式(选择其中一种)

DIALTYPE\_DTMF - DTMF 方式拨号

DIALTYPE\_FSK - FSK 方式拨号

[in] *pValue* - 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 系统将自动读取最后一个拨的号码并自动调用 `CPC_GENERAL_STARTDIAL` 开始拨号
- 一般使用 DTMF 方式拨号
- 非特殊需求不要使用 FSK 方式拨号
- 程序退出后该号码被释放,如果需要保存该号码请在程序退出前调用 `CPC_GENERAL_WRITEPARAM` 把所有系统的参数都写入 CC301 的配置文件里,然后在下次打开设备时调用 `CPC_GENERAL_READPARAM` 从 CC301 的配置文件里读取所有参数

#### 例如 (Example)

```
CPC_General(0,
             CPC_GENERAL_REDIAL,
             DIALTYPE_DTMF,
             0);
```

## 2. 16. 4 CPC\_GENERAL\_STOPDIAL 终止拨号/二次拨号

#### 功能描述 (Description)

终止拨号/二次拨号,如果号码已经拨完, 调用该功能无效

#### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* - 保留

[in] *pValue* - 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 请在开始拨号后, 而且还没有接收到 `BriEvent_DialEnd` 前调用该函数来终止拨号, 如果号码已经拨完, 调用该功能无效
- 如果想同时释放线路再调用软挂机(电话机没拿着才能挂断)

```
CPC_SetDevCtrl(0,CPC_CTRL_DOHOOK,0);
```

#### 例如 (Example)



```
CPC_General(0,CPC_GENERAL_STOPDIAL,0,0);
```

## 2. 16. 5 CPC\_GENERAL\_ISDIALING 检测是否正在拨号/二次拨号

### 功能描述 (Description)

检测是否正在拨号/二次拨号

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

小于 0 – 表示检测失败

0 – 没有在拨号

1 – 正在拨号

### 备注 (Remarks)

- 在开始拨号成功后, 而且还没有接收到 `BriEvent_DialEnd` 前表示正在拨号, 其它表示没有在拨号

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_ISDIALING,0,0);
```

## 2. 16. 6 CPC\_GENERAL\_STARTRING 开始给接在 phone 口设备模拟间隔震铃

### 功能描述 (Description)

开始给接在 phone 口的话机/交换机模拟间隔震铃

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue*

发送给话机的号码(类似来电号码功能), 电话机接收到后会在液晶屏里显示该号码

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 不是所有设备具有该功能, 具有 `DEVMODULE_RING` 模块功能的设备才有效
- 默认使用响 1 秒, 停 4 秒的方式
- 可以使用 `CPC_PARAM_RINGELAPSE` 和 `CPC_PARAM_RINGSILENCEELAPSE` 来调节响铃节奏

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_STARTRING,0,"13800000000");
```



## 2. 16. 7 CPC\_GENERAL\_STOPRING 停止给接在 phone 口设备模拟间隔震铃

### 功能描述 (Description)

停止接在 phone 口的话机/交换机模拟间隔震铃

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

- 开始模拟间隔震铃后,在没有震铃超时前,使用该功能来停止震铃

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_STOPRING,0,0);
```

## 2. 16. 8 CPC\_GENERAL\_ISRINGING 检测是否给接在 phone 口设备模拟震铃

### 功能描述 (Description)

检测是否正在给接在 phone 口的话机/交换机模拟间隔震铃

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

小于 0 – 表示检测失败

0 – 没有在模拟间隔震铃

1 – 正在模拟间隔震铃

### 备注 (Remarks)

- 在开始模拟间隔震铃成功后,而且还没有接收到 BriEvent\_RingTimeOut 前表示正在模拟间隔震铃, 其它表示没有在模拟间隔震铃

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_ISRINGING,0,0);
```

## 2. 16. 9 CPC\_GENERAL\_STARTFLASH 开始拍插簧

### 功能描述 (Description)

开始拍插簧



#### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

拍插簧类型

FT\_TEL - 使用话机拍插簧(如果此时系统软摘机状态会失败)

FT\_PC - 使用盒子拍插簧(如果此时用户拿着话机会失败)

FT\_ALL - 话机和盒子都使用

[in] *pValue*

拍插簧完毕后自动拨分机的号码,如:“,1099”(一个逗号表示延迟 1 秒)

- 使用该指定分机号码后用户在接收到拍插簧结束后就不再需要调用拨号分机的功能

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 拍插簧前保证线路已经在摘机状态, 电话机摘着并接通线路或者软已摘机接通电话
- 如果返回失败, 那可能是当前状态不对
- 建议用户使用 FT\_ALL 进行拍插簧, 这样不需要判断当前话机和线路的摘机状态, 系统自动检测后强制断开所有线路进行拍插簧
- 如果调用拍插簧成功, 但没有转分机成功:
  - 首先考虑交换机是否支持拍插簧
  - 调节拍查簧的间隔时间(*CPC\_PARAM\_FLASHELAPSE*)
  - 延迟拍插簧后的拨号时间
- 拍插簧转分机成功后, 在分机接听前您可以再拍插簧一次还原通话状态(一般交换机支持该方式, 具体根据交换机的性能来定)

#### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_STARTFLASH,0,“1099”);
```

## 2. 16. 10 CPC\_GENERAL\_STOPFLASH 停止拍插簧

#### 功能描述 (Description)

停止拍插簧

#### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* - 保留

[in] *pValue* - 保留

#### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

#### 备注 (Remarks)

- 开始拍插簧后, 在没有拍插完毕时, 使用该功能来停止拍插簧
- 建议用户不要强制停止拍插簧, 在停止时也有可能交换机已经识别拍插簧成



功，:如：您设置的拍插簧间隔为 1 秒，您在 600ms 时停止拍插簧恢复到摘机状态，可能此时交换机已经识别到拍插簧成功，等待拨号。

例如 (Example)

```
CPC_General(0,CPC_GENERAL_STOPFLASH,0,0);
```

## 2. 16. 11 CPC\_GENERAL\_ISFLASHING 检测是否正在拍插簧

**功能描述 (Description)**

检测是否正在拍插簧

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

小于 0 – 表示检测失败

0 – 没有在拍插簧

1 – 正在拍插簧

**备注 (Remarks)**

- 在开始拍插簧成功后，而且还没有接收到 BriEvent\_FlashEnd 前表示正在拍插簧，其它表示没有在拍插簧

例如 (Example)

```
CPC_General(0,CPC_GENERAL_ISFLASHING,0,0);
```

## 2. 16. 12 CPC\_GENERAL\_STARTREFUSE 拒接当前正在呼入的来电

**功能描述 (Description)**

拒接当前正在呼入的来电

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue*

- REFUSE\_ASYN

■ 异步模式,调用后函数立即返回，但并不表示拒接完成，拒接完成后将接收到一个拒接完成的事件

- REFUSE\_SYN

■ 同步模式,调用后该函数被堵塞，等待拒接完成返回，系统不再有拒接完成的事件

- 其它 – 返回失败

[in] *pValue* – 保留

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败



**备注 (Remarks)**

- 拒接前保证该线路正在来电状态
- 拒接时系统将自动摘机,在间隔一定时间内自动挂机
- 该拒接间隔时间可以 `CPC_PARAM_REFUSEELAPSE` 来调节
- 电信一般为 3 秒/移动 6 秒开始计费,所以建议间隔时间控制在 3 秒以内,避免引起计费

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_STARTREFUSE, REFUSE_ASYN,0);
```

## 2. 16. 13 CPC\_GENERAL\_STOPREFUSE 停止拒接

**功能描述 (Description)**

停止拒接

**参数介绍 (Parameters)**

[in] `nChannelID`

通用控制通道号

[in] `nValue` - 保留

[in] `pValue` - 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

—停止后线路将进入摘机接通状态

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_STOPREFUSE,0,0);
```

## 2. 16. 14 CPC\_GENERAL\_ISREFUSEING 检测是否正拒接来电

**功能描述 (Description)**

检测是否正拒接来电

**参数介绍 (Parameters)**

[in] `nChannelID`

通用控制通道号

[in] `nValue` - 保留

[in] `pValue` - 保留

**返回值 (Return Values)**

小于 0 - 表示检测失败

0 - 没有在拒接来电

1 - 正在拒接来电

**备注 (Remarks)**

无

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_ISREFUSEING,0,0);
```



## 2. 16. 15 CPC\_GENERAL\_GETCALLIDTYPE 获取本次来电号码的获取类型

### 功能描述 (Description)

获取本次来电号码的获取类型

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

### 返回值 (Return Values)

小于 0

– 表示检测失败

CALLIDMODE\_DTMF

– DTMF 方式来电号码

CALLIDMODE\_FSK

– FSK 方式来电号码

### 备注 (Remarks)

- 一般直线都采用 FSK 方式发送来电号码, 该方式的来电号码电信一般在 1 声响铃后, 2 声响前发送过来.
- 一般交换机采用 DTMF 方式发送来电号码, 该方式一般是在一声响铃前达到, 也有部分交换机做成一声响铃后发送
- 非特殊需求不需要使用, 用户可以在来电号码事件 `BriEvent_GetCallID` 里获取到该信息

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_GETCALLIDTYPE,0,0);
```

## 2. 16. 16 CPC\_GENERAL\_GETCALLID 获取本次呼入的来电号码

### 功能描述 (Description)

获取本次呼入的来电号码

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

给保存来电号码分配的内存大小

[in] *pValue*

保存来电号码的内存地址

### 返回值 (Return Values)

小于等于 0 表示获取失败, 其它返回号码的内存地址

### 备注 (Remarks)

非特殊需求不需要使用, 用户可以在来电号码事件 `BriEvent_GetCallID` 里获取到该信息

### 例如 (Example)

使用返回的方式获取:

```
char *pCallID=CPC_General(0,CPC_GENERAL_GETCALLID,0,0);
```





使用分配内存地址方式获取:

```
char szBuf[128];//分配保存号码内存足够长度  
CPC_General(0,CPC_GENERAL_GETCALLID,szBuf,128);
```

## 2. 16. 17 CPC\_GENERAL\_GETTELDIALCODE 获取本次话机已经拨出的号码

### 功能描述 (Description)

获取本次话机已经拨出的号码

### 参数介绍 (Parameters)

[in] *nChannelID*  
通用控制通道号  
[in] *nValue*        – 保留  
[in] *pValue*        – 保留

### 返回值 (Return Values)

小于等于 0 表示获取失败, 其它返回号码的内存地址

### 备注 (Remarks)

— 非特殊需求不需要使用, 用户可以在话机拨号事件 BriEvent\_PhoneDial 里获取到该信息

### 例如 (Example)

```
char *pCallID=CPC_General(0,CPC_GENERAL_GETTELDIALCODE,0,0);
```

## 2. 16. 18 CPC\_GENERAL\_GETTELDIALCODEEX 获取本次话机已经拨出的号码

### 功能描述 (Description)

获取本次话机已经拨出的号码

### 参数介绍 (Parameters)

[in] *nChannelID*  
通用控制通道号  
[in] *nValue*  
给保存话机拨的号码分配的内存大小  
[in] *pValue*  
保存话机拨的号码的内存地址

### 返回值 (Return Values)

小于 0 表示获取失败, 其它返回号码的长度

### 备注 (Remarks)

非特殊需求不需要使用, 用户可以在来电号码事件 BriEvent\_GetCallID 里获取到该信息

### 例如 (Example)

```
char szBuf[128];//分配保存号码内存足够长度  
CPC_General(0,CPC_GENERAL_GETTELDIALCODEEX,szBuf,128);
```



## 2. 16. 19 CPC\_GENERAL\_RESETTELDIALBUF 清空话机已经拨出的号码

### 功能描述 (Description)

清空保存的本次话机已经拨出的号码

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*        – 保留

[in] *pValue*        – 保留

### 返回值 (Return Values)

成功返回大于 0, 其它表示失败

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_RESETTELDIALBUF,0,0);
```

## 2. 16. 20 CPC\_GENERAL\_GETTELDIALLEN 获取本次话机已经拨出的号码长度

### 功能描述 (Description)

获取本次话机已经拨出的号码长度

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*        – 保留

[in] *pValue*        – 保留

### 返回值 (Return Values)

小于 0 表示失败, 其它返回话机已经拨出的号码长度

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_GETTELDIALLEN,0,0);
```

## 2. 16. 21 CPC\_GENERAL\_STARTSHARE 启动设备共享服务模块

### 功能描述 (Description)

启动设备共享服务模块

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*        – 保留



[in] *pValue* – 保留

**返回值 (Return Values)**

成功返回大于 0,其它表示失败

**备注 (Remarks)**

该功能保留, 目前用户不能使用

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_STARTSHARE,0,0);
```

## 2. 16. 22 CPC\_GENERAL\_STOPSHARE 停止设备共享服务模块

**功能描述 (Description)**

停止设备共享服务模块

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

成功返回大于 0,其它表示失败

**备注 (Remarks)**

该功能保留, 目前用户不能使用

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_STOPSHARE,0,0);
```

## 2. 16. 23 CPC\_GENERAL\_ISSHARE 检测是否启动了共享服务模块

**功能描述 (Description)**

检测是否启动了共享服务模块

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue* – 保留

[in] *pValue* – 保留

**返回值 (Return Values)**

小于 0 – 其它表示失败

0 – 没有启动共享

1 – 已经启动共享

**备注 (Remarks)**

该功能保留, 目前用户不能使用

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_ISSHARE,0,0);
```



## 2. 16. 24 CPC\_GENERAL\_ENABLECALLIN 禁止/允许外线呼入

### 功能描述 (Description)

禁止/允许外线呼入

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

0 — 禁止外线 PSTN 呼入

1 — 允许外线 PSTN 呼入

[in] *pValue* — 保留

### 返回值 (Return Values)

成功返回大于 0, 其它表示失败

### 备注 (Remarks)

禁止后该线路将一直处于占线状态, 所有呼叫该线路都将收到占线状态

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_ENABLECALLIN,0,0);
```

- 禁止外线被呼入

## 2. 16. 25 CPC\_GENERAL\_ISENABLECALLIN 检测是否允许外线呼入

### 功能描述 (Description)

检测是否允许外线呼入

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* — 保留

[in] *pValue* — 保留

### 返回值 (Return Values)

小于 0 — 其它表示失败

0 — 允许外线呼入状态

1 — 禁止外线呼入状态

### 备注 (Remarks)

无

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_ISENABLECALLIN,0,0);
```

## 2. 16. 26 CPC\_GENERAL\_ISLINEHOOK 检测线路是否处于摘机状态

### 功能描述 (Description)

检测线路是否处于摘机状态

### 参数介绍 (Parameters)



[in] *nChannelID*

通用控制通道号

[in] *nValue*        - 保留

[in] *pValue*        - 保留

**返回值 (Return Values)**

- 小于 0    - 其它表示失败
- 0        - 外线没有摘机状态
- 1        - 外线处于摘机状态

**备注 (Remarks)**

- 电话机摘机并且没有跟 PSTN 线路断开
- 系统调用了软摘机
- 以上两个状态满足其中任意一个就表示线路已经处于摘机状态
- 在摘机状态下, 呼叫该线路将收到占线状态

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_ISLINEHOOK,0,0);
```

## 2. 16. 27 CPC\_GENERAL\_ISLINEFREE 检测线路是否处于空闲状态

**功能描述 (Description)**

检测线路是否处于空闲状态

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue*        - 保留

[in] *pValue*        - 保留

**返回值 (Return Values)**

- 小于 0    - 其它表示失败
- 0        - 外线不在空闲状态
- 1        - 外线处于空闲状态

**备注 (Remarks)**

- 线路没有摘机 (*CPC\_GENERAL\_ISLINEHOOK*) 并且当前没有来电呼入表示线路空闲

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_ISLINEFREE,0,0);
```

## 2. 16. 28 CPC\_GENERAL\_RESETRINGBACK 复位检测到的回铃音,重新启动检测

**功能描述 (Description)**

复位检测到的回铃音,重新启动检测

**参数介绍 (Parameters)**

[in] *nChannelID*



通用控制通道号

[in] *nValue* - 保留

[in] *pValue* - 保留

#### 返回值 (Return Values)

成功返回大于 0, 其它表示失败,

#### 备注 (Remarks)

- 当线路呼叫一个号码时可能会接收到多个回铃时间.

如: 当您设置拨一个号码允许检测回铃, 那么在一般分机线上, 拨一个出局号码后交换机将发送一个回铃信号过来, 这是系统将检测到回铃, 认为拨号已经结束, 等用户再拨完 PSTN 号码后再接收到电信局的回铃时将接收不到该回铃事件, 这样就需要用户在检测第一个号码的回铃时调用该功能, 要求系统重新检测回铃。

- 非特殊情况, 一般用户设置 2 个号码后开始检测回铃就可以满足大部分拨号环境, 不需要使用该功能。

#### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_RESETRINGBACK,0,0);
```

## 2. 16. 29 CPC\_GENERAL\_CHECKCHANNELID 检测通道 ID 是否合法

#### 功能描述 (Description)

检测通道 ID 是否合法

#### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue* - 保留

[in] *pValue* - 保留

#### 返回值 (Return Values)

小于 0 - 其它表示失败

0 - 通道 ID 不合法

1 - 通道 ID 合法

#### 备注 (Remarks)

无

#### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_CHECKCHANNELID,0,0);
```

## 2. 16. 30 CPC\_GENERAL\_CHECKDIALTONE 获取系统是否已经检测到拨号音了

#### 功能描述 (Description)

获取系统是否已经检测到拨号音了

#### 参数介绍 (Parameters)

[in] *nChannelID*



通用控制通道号

[in] *nValue* - 保留

[in] *pValue* - 保留

#### 返回值 (Return Values)

小于 0 - 其它表示失败

0 - 还没有检测到拨号音

1 - 已经检测到了拨号音

#### 备注 (Remarks)

- 一般用户不需要使用该功能，在拨号音事件里(BriEvent\_DialTone)获取到该状态就可以

- 也就是说当用户接收到 BriEvent\_DialTone 事件后调用该功能返回为 1,在摘机后没有检测到该事件前返回的为 0

#### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_CHECKDIALTONE,0,0);
```

## 2. 16. 31 CPC\_GENERAL\_CHECKSILENCE 开始检测静音

#### 功能描述 (Description)

开始检测静音

#### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

静音需要持续的时长.(单位: ms)

0 - 表示停止检测

其它 - 表示开始检测

[in] *pValue* - 保留

#### 返回值 (Return Values)

大于 0 表示启动成功，其它表示失败

#### 备注 (Remarks)

- 启动成功后开始检测线路静音，如果连续静音时间达到指定的时长用户就会接收到 BriEvent\_RecvSilence 事件,并且系统也自动停止检测,想再次检测重新调用该功能

- 启动该功能前确认已经启动了信号音检测功能(CPC\_CTRL\_RECVSIGN)

- 如果启动后在还没有接收到 BriEvent\_RecvSilence 前想停止就重新调用该函数把 nValue 指定为 0

#### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_CHECKSILENCE,2000,0);
```

- 开始检测静音,如果检测到连续 2000ms(2 秒)的静音就回调事件

```
CPC_General(0,CPC_GENERAL_CHECKSILENCE,0,0);
```

- 停止检测静音



## 2. 16. 32 CPC\_GENERAL\_CHECKVOICE 开始检测声音

### 功能描述 (Description)

开始检测声音

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

声音需要持续的时长.(单位: ms)

0 - 表示停止检测

其它 - 表示开始检测

[in] *pValue* - 保留

### 返回值 (Return Values)

大于 0 表示启动成功, 其它表示失败

### 备注 (Remarks)

- 启动成功后开始检测线路声音, 如果连续声音时间达到指定的时长用户就会接收到 BriEvent\_RecvVoice 事件, 并且系统也自动停止检测, 想再次检测重新调用该功能
- 启动该功能前确认已经启动了信号音检测功能(CPC\_CTRL\_RECVSIGN)
- 如果启动后在还没有接收到 BriEvent\_RecvVoice 前想停止就重新调用该函数把 nValue 指定为 0
- 该功能配合 CPC\_GENERAL\_CHECKSILENCE 可以用来实现有声音自动录音, 静音时停止录音从而达到节省录音空间的效果

### 例如 (Example)

```
CPC_General(0,CPC_GENERAL_CHECKVOICE,100,0);
```

- 开始检测声音, 如果检测到连续 100ms(0.1 秒)的声音就回调事件

```
CPC_General(0,CPC_GENERAL_CHECKVOICE,0,0);
```

- 停止检测声音

## 2. 16. 33 CPC\_GENERAL\_CHECKLINESTATE 检测线路状态是否可用/接反

### 功能描述 (Description)

检测线路 LINE 口是否可用, LINE 口/PHONE 口是否接反

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*

超时时间, 默认使用 2000.(单位: ms)

0 - 使用默认值 2000ms

[in] *pValue* - 保留

### 返回值 (Return Values)

大于 0 表示启动成功, 其它表示失败

### 备注 (Remarks)





- 使用该功能的通道必须是支持软摘机的 LINE 口通道
- 检测成功或者超时用户将接收到 BriEvent\_CheckLine 事件
- 检测前确认没有拿着话机
- 检测时系统会自动触发软摘机事件,检测完毕后触发软挂机事件
- 默认使用 2000ms 超时, 如果特殊交换机摘机发送拨号音时间要超过 2000ms 时,可以考虑延长超时时间

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_CHECKLINESTATE,0,0);
```

- 检测线路状态

## 2. 16. 34 CPC\_GENERAL\_SETUSERVALUE 设置用户自定义通道数值

**功能描述 (Description)**

设置用户自定义通道数值,系统退出后自动释放

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue*

用户自定义数值

[in] *pValue*      - 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 该值对系统没有任何意义, 只为用户保存后可以再获取
- 建议用户不要把该值设置成该 CC301 错误值
- 关闭设备后自动被释放

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_SETUSERVALUE,100,0);
```

- 设置用户自定义数值为 100

## 2. 16. 35 CPC\_GENERAL\_GETUSERVALUE 获取保存的用户自定义通道数值

**功能描述 (Description)**

获取保存的用户自定义通道数值,该值为 CPC\_GENERAL\_SETUSERVALUE 设置的数值

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue*      - 保留

[in] *pValue*      - 保留

**返回值 (Return Values)**



小于 0 表示失败，其它返回保存的自定义通道数值

**备注 (Remarks)**

- 该值对系统没有任何意义

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_GETUSERVALUE,0,0);
```

## 2. 16. 36 CPC\_GENERAL\_SETUSERSTRING 设置用户自定义通道字符串

**功能描述 (Description)**

设置用户自定义通道字符串,系统退出后自动释放

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue*      - 保留

[in] *pValue*

用户自定义通道字符串地址

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 该值对系统没有任何意义，只为用户保存后可以再获取
- 关闭设备后自动被释放

**例如 (Example)**

```
CPC_General(0,CPC_GENERAL_SETUSERSTRING,0,"abcd");  
- 设置用户自定义字符串为"abcd"
```

## 2. 16. 37 CPC\_GENERAL\_GETUSERSTRING 获取保存的用户自定义通道字符串

**功能描述 (Description)**

获取保存的用户自定义通道字符串,该值为 CPC\_GENERAL\_SETUSERSTRING 设置的字符串

**参数介绍 (Parameters)**

[in] *nChannelID*

通用控制通道号

[in] *nValue*      - 保留

[in] *pValue*      - 保留

**返回值 (Return Values)**

小于 0 表示失败，其它返回保存的自定义通道字符串的内存地址

**备注 (Remarks)**

- 该值对系统没有任何意义

**例如 (Example)**

```
char *pValue=CPC_General(0,CPC_GENERAL_GETUSERSTRING,0,0);
```



## 2. 16. 38 CPC\_GENERAL\_READPARAM 从配置文件里读取所有参数

### 功能描述 (Description)

从配置文件里读取所有参数把所有参数都重新初始化

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*      - 保留

[in] *pValue*

保存参数配置的 INI 文件路径

- 使用绝对路径。如: "c:\a.ini"。
- 使用相对路径。如: "a.ini", 系统将自动定为该文件为本进程所在目录下的 a.ini
- 如果该路径为空系统将自动使用本进程目录下的默认配置文件 "cc301config.ini" 进行读取

### 返回值 (Return Values)

大于 0 表示成功, 其它表示失败

### 备注 (Remarks)

- 读取后系统的所有参数将被该文件指定的参数覆盖
- 如果文件不存在会读取失败
- 具体 INI 配置文件信息查看例子文件
- 如果有多个通道每个通道都要需要独立调用该功能, 也就是一次调用只对一个通道有效

### 例如 (Example)

使用默认配置文件:

```
CPC_General(0,CPC_GENERAL_READPARAM,0,0);
```

指定配置文件路径:

```
CPC_General(0,CPC_GENERAL_READPARAM,0,"c:\cc301.ini");
```

## 2. 16. 39 CPC\_GENERAL\_WRITEPARAM 把当前的所有参数都写入到配置文件

### 功能描述 (Description)

把当前的所有参数都写入到配置文件

### 参数介绍 (Parameters)

[in] *nChannelID*

通用控制通道号

[in] *nValue*      - 保留

[in] *pValue*

保存参数配置的 INI 文件路径

- 使用绝对路径。如: "c:\a.ini"。



- 使用相对路径。如：“a.ini”,系统将自动定为该文件为本进程所在目录下的 a.wav
- 如果该路径为空系统将自动使用本进程目录下的默认配置文件"cc301config.ini"进行读取

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 如果文件不存在系统将自动创建新文件并把参数写入到该文件
- 如果把多个通道都使用同一个配置文件，保存为最后一次写入的通道参数

**例如 (Example)**

使用默认配置文件：

```
CPC_General(0,CPC_GENERAL_WRITEPARAM,0,0);
```

指定配置文件路径：

```
CPC_General(0,CPC_GENERAL_WRITEPARAM,0,"c:\cc301.ini");
```

## 2. 17 PSTN 线路呼叫日志信息(CPC\_CallLog)

### 功能描述 (Description)

获取 PSTN 线路呼叫日志信息，可以在接收到 BriEvent\_PSTNFree 事件后用该模块来获取具体呼叫的信息

### 函数原型 (Function)

```
CPCINT32  CPC_CallLog (/*[in]*/CPCINT32  nChannelID,
                      /*[in]*/CPCUINT32  uLogType,
                      /*[in]*/CPCCHAR8 *pValue,
                      /*[in]*/CPCINT32  nValue);
```

### 返回值 (Return Values)

有符号 32bit 位.

### 参数描述 (Parameters description)

### 2. 17. 1 CPC\_CALLLOG\_BEGINTIME 呼叫开始时间

**功能描述 (Description)**

获取呼叫开始时间

**参数介绍 (Parameters)**

- [in] *nChannelID*      – 日志所在通道
- [in] *pValue*            – 保留
- [in] *nValue*            – 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 时间使用的格式为 GMT 格式, 1970 年 1 月 1 日 08:00:00 起至今的时间(秒)

**例如 (Example)**

```
CPC_CallLog(0,CPC_CALLLOG_BEGINTIME,"0);
```

## 2. 17. 2 CPC\_CALLLOG\_RINGBACKTIME 呼出时检测到回铃时间

**功能描述 (Description)**

呼出时检测到回铃时间

**参数介绍 (Parameters)**

- [in] *nChannelID*      – 日志所在通道
- [in] *pValue*            – 保留
- [in] *nValue*            – 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 时间使用的格式为 GMT 格式, 1970 年 1 月 1 日 08:00:00 起至今的时间(秒)
- 不是所有的呼出都有回铃时间, 比如彩铃时就没有该时间

**例如 (Example)**

```
CPC_CallLog(0,CPC_CALLLOG_RINGBACKTIME,"0);
```

## 2. 17. 3 CPC\_CALLLOG\_CONNECTEDTIME 呼入/呼出时检测到接通时间

**功能描述 (Description)**

呼入/呼出时接通时间

**参数介绍 (Parameters)**

- [in] *nChannelID*      – 日志所在通道
- [in] *pValue*            – 保留
- [in] *nValue*            – 保留

**返回值 (Return Values)**

大于 0 表示成功, 其它表示失败

**备注 (Remarks)**

- 时间使用的格式为 GMT 格式, 1970 年 1 月 1 日 08:00:00 起至今的时间(秒)
- 不是所有的呼出都有准确检测到接通时间, 比如彩铃时可能就没有该时间, 只有在反级服务时才能准确检测到呼出的接通时间

**例如 (Example)**

```
CPC_CallLog(0,CPC_CALLLOG_CONNECTEDTIME,"0);
```



## 2. 17. 4 CPC\_CALLLOG\_ENDTIME 呼叫结束时间

### 功能描述 (Description)

呼叫结束时间

### 参数介绍 (Parameters)

[in] *nChannelID*      – 日志所在通道  
[in] *pValue*            – 保留  
[in] *nValue*            – 保留

### 返回值 (Return Values)

大于 0 表示成功，其它表示失败

### 备注 (Remarks)

- 时间使用的格式为 GMT 格式, 1970 年 1 月 1 日 08:00:00 起至今的时间(秒)

### 例如 (Example)

```
CPC_CallLog(0,CPC_CALLLOG_ENDTIME,"",0);
```

## 2. 17. 5 CPC\_CALLLOG\_CALLTYPE 呼入/呼出类型

### 功能描述 (Description)

呼入/呼出类型

### 参数介绍 (Parameters)

[in] *nChannelID*      – 日志所在通道  
[in] *pValue*            – 保留  
[in] *nValue*            – 保留

### 返回值 (Return Values)

1            – 来电  
2            – 去电  
其它        – 未知

### 备注 (Remarks)

-

### 例如 (Example)

```
CPC_CallLog(0,CPC_CALLLOG_CALLTYPE,"",0);
```

## 2. 17. 6 CPC\_CALLLOG\_CALLRESULT 呼入/呼出结果

### 功能描述 (Description)

呼入/呼出结果

### 参数介绍 (Parameters)

[in] *nChannelID*      – 日志所在通道  
[in] *pValue*            – 保留  
[in] *nValue*            – 保留

### 返回值 (Return Values)

1            – 呼入未接  
2            – 呼入拒接



- |    |   |         |
|----|---|---------|
| 3  | — | 呼出检测到回铃 |
| 4  | — | 接通      |
| 其它 | — | 未知      |

**备注 (Remarks)**

- 接通结果只有在反级服务下准确，其它状态下该状态并不准确，建议在非反级服下忽略该呼叫结果

**例如 (Example)**

```
CPC_CallLog(0,CPC_CALLLOG_CALLRESULT,"0);
```

## 2. 17. 7 CPC\_CALLLOG\_CALLID 呼入/呼出号码

**功能描述 (Description)**

获取呼入/呼出号码

**参数介绍 (Parameters)**

- |                        |   |           |
|------------------------|---|-----------|
| [in] <i>nChannelID</i> | — | 日志所在通道    |
| [in] <i>pValue</i>     | — | 保存号码的内存地址 |
| [in] <i>nValue</i>     | — | 保存号码的内存长度 |

**返回值 (Return Values)**

- 返回号码长度

**备注 (Remarks)**

- 如果传入的内存地址为空，系统将直接返回号码内存地址,不再返回号码长度

**例如 (Example)**

```
Char szBuf[256]={0};  
CPC_CallLog(0,CPC_CALLLOG_CALLID,szBuf,256);
```

## 2. 17. 8 CPC\_CALLLOG\_CALLRECFILE 获取录音文件路径

**功能描述 (Description)**

获取呼叫过程产生的最后一个录音文件路径

**参数介绍 (Parameters)**

- |                        |   |           |
|------------------------|---|-----------|
| [in] <i>nChannelID</i> | — | 日志所在通道    |
| [in] <i>pValue</i>     | — | 保存路径的内存地址 |
| [in] <i>nValue</i>     | — | 保存路径的内存长度 |

**返回值 (Return Values)**

- 返回路径长度

**备注 (Remarks)**

- 如果传入的内存地址为空，系统将直接返回路径内存地址,不再返回路径长度

**例如 (Example)**

```
Char szBuf[256]={0};  
CPC_CallLog(0,CPC_CALLLOG_CALLRECFILE,szBuf,256);
```

## 2. 17. 9 CPC\_CALLLOG\_DELRECFILE 删除录音文件

**功能描述 (Description)**

删除呼叫过程产生的最后一个录音文件路径

**参数介绍 (Parameters)**

- [in] *nChannelID* – 日志所在通道
- [in] *pValue* – 保留
- [in] *nValue* – 保留

**返回值 (Return Values)**

- 大于 0 表示成功，其它表示失败

**备注 (Remarks)**

- 要删除录音文件前必须先确认已经停止文件录音
- 在接收到 BriEvent\_PSTNFree 事件后使用该模块获取呼叫相关信息，如果不需要保留录音就可以使用该功能删除该录音文件
- 适用用 B/S 系统里不能对文件操作的开发模式中,C/S 模式开发用户也可以自行调用 Windows API 进行处理

**例如 (Example)**

```
CPC_CallLog(0,CPC_CALLLOG_DELRECFILE,"",0);
```

**2. 17. 10 CPC\_CALLLOG\_RESET 复位所有呼叫信息****功能描述 (Description)**

复位所有呼叫信息为空状态

**参数介绍 (Parameters)**

- [in] *nChannelID* – 日志所在通道
- [in] *pValue* – 保留
- [in] *nValue* – 保留

**返回值 (Return Values)**

- 大于 0 成功，其它失败

**备注 (Remarks)**

- 建议在每次获取完信息后立即复位。

**例如 (Example)**

```
CPC_CallLog(0,CPC_CALLLOG_RESET,"",0);
```

**2. 18 设备存储读写功能 (CPC\_Storage)****功能描述 (Description)**

对具有存储功能的芯片进行任意读写操作，该功能只对 I 系列的设备有效，写入的数据将被永久存储在 USB 芯片内,不会随着系统重启或者 USB 掉电而丢失

**函数原型 (Function)**

```
CPCINT32 CPC_Storage (/*[in]*/CPCINT32 nDevID,
```





```

/*[in]*/CPCUINT32 uOPType,
/*[in]*/CPCINT32 nSeek,
/*[in]*/CPCCHAR8 *pPwd,
/*[in,out]*/CPCCHAR8 *pPValue,
/*[in]*/CPCUINT32 nBufSize);

```

### 返回值 (Return Values)

有符号 32bit 位.

### 参数描述 (Parameters description)

#### *nDevID*

[in] 指定芯片设备 ID – 第一个芯片设备 *nDevD* 为 0,依次递增

备注: 1.有些设备型号一个芯片有 2 个通道号,如: IR2(2 路录音),IC2\_LP。这样该两个通道就共用一个存储空间.

2.可以用 CPC\_DEVINFO\_GETDEVID 根据通道号获得设备号

## 2. 18. 1 CPC\_STORAGE\_PUBLIC\_READ 读取共享区域数据

### 功能描述 (Description)

读取共享区域数据

### 参数介绍 (Parameters)

[in] *nDevID*

读取控制的设备 ID

[in] *uSeek* – 共享区域的偏移位置,单位(字节)

[in] *pPwd* – 读取的密码(默认为空密码)

[out] *pValue* – 保存读取的数据内存地址

[in] *nBufSize* – 需要读取的数据长度

### 返回值 (Return Values)

大于 0 表示读取的实际长度,其它表示失败

### 备注 (Remarks)

*uSeek* 偏移位置不能超过实际存储的大小

### 例如 (Example)

```
char szBuf[129]={0};
```

//分配的内存一定要超过实际想读取长度,不然会引起内存错误

```
CPC_Storage(0, CPC_STORAGE_PUBLIC_READ,0,"",szBuf,128);
```

## 2. 18. 2 CPC\_STORAGE\_PUBLIC\_READSTR 读取共享区域的字符串数据

### 功能描述 (Description)



读取共享区域的字符串数据

#### 参数介绍 (Parameters)

[in] *nDevID*

读取控制的设备 ID

[in] *uSeek* – 共享区域的偏移位置,单位(字节)

[in] *pPwd* – 读取的密码(默认为空密码)

[out] *pValue* – 保存读取的数据内存地址

[in] *nBufSize* – 需要读取的数据长度

#### 返回值 (Return Values)

大于 0 表示读取的实际长度,其它表示失败

#### 备注 (Remarks)

- *uSeek* 偏移位置不能超过实际存储的大小
- 读取时如果读取到 0 就自动结束,区别 `CPC_STORAGE_PUBLIC_READ` 是一直读取,直到读到足够的长度或者读到存储的结束地址

#### 例如 (Example)

```
char szBuf[129]={0};
//分配的内存一定要超过实际想读取长度,不然会引起内存错误
CPC_Storage(0, CPC_STORAGE_PUBLIC_READSTR,0,"",szBuf,128);
```

## 2. 18. 3 CPC\_STORAGE\_PUBLIC\_WRITE 写入数据到共享区域

#### 功能描述 (Description)

写入数据到共享区域,写入的数据将被永久存储在 USB 芯片内,不会随着系统重启或者 USB 掉电而丢失

#### 参数介绍 (Parameters)

[in] *nDevID*

写入控制的设备 ID

[in] *uSeek* – 共享区域的偏移位置,单位(字节)

[in] *pPwd* – 写入的密码(默认为空密码)

[in] *pValue* – 写入数据的内存地址

[in] *nBufSize* – 需要写入的数据长度

#### 返回值 (Return Values)

大于 0 表示写入的实际长度,其它表示失败

#### 备注 (Remarks)

- *uSeek* 偏移位置不能超过实际存储的大小

#### 例如 (Example)

```
char *pBuf="abcd";
CPC_Storage(0, CPC_STORAGE_PUBLIC_WRITE,,0,"",pBuf,5);
写入 5 个表示在字符串后面多写入一个 0;
```



## 2. 18. 4 CPC\_STORAGE\_PUBLIC\_SETREADPWD 设置读取共享区域数据的密码

### 功能描述 (Description)

设置读取共享区域数据的密码

### 参数介绍 (Parameters)

[in] *nDevID*

写入控制的设备 ID

[in] *uSeek*      – 保留

[in] *pPwd*       – 读取数据的原密码(默认为空密码)

[in] *pValue*     – 新密码的字符串(密码长度不能超过 8 个字符)

[in] *nBufSize*   – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

- 密码必须是小于等于 8 位的字符长度
- 设置读取共享区域新密码后, 再使用 `CPC_STORAGE_PUBLIC_READSTR` 和 `CPC_STORAGE_PUBLIC_READ` 时, `pPwd` 参数必须指定为设置的新密码, 不然读取时就会返回密码错误 `BCERR_INVALIDPWD` (-41)

### 例如 (Example)

```
char *pBuf="abcdefgx";
CPC_Storage(0, CPC_STORAGE_PUBLIC_SETREADPWD,,0,"",pBuf,0);
设置读取共享区域密码为"abcdefgx".
```

## 2. 18. 5 CPC\_STORAGE\_PUBLIC\_SETWRITEPWD 设置写入共享区域数据的密码

### 功能描述 (Description)

设置写入共享区域数据的密码

### 参数介绍 (Parameters)

[in] *nDevID*

写入控制的设备 ID

[in] *uSeek*      – 保留

[in] *pPwd*       – 写入数据的原密码(默认为空密码)

[in] *pValue*     – 新密码的字符串(密码长度不能超过 8 个字符)

[in] *nBufSize*   – 保留

### 返回值 (Return Values)

大于 0 表示成功,其它表示失败

### 备注 (Remarks)

- 密码必须是小于等于 8 位的字符长度
- 设置写入共享区域新密码后, 再使用 `CPC_STORAGE_PUBLIC_WRITE` 时, `pPwd` 参数必须指定为设置的新密码, 不然写入时就会返回密码错误



BCERR\_INVALIDPWD (-41)

例如 (Example)

```
char *pBuf="abcdefgx";
CPC_Storage(0, CPC_STORAGE_PUBLIC_SETWRITEPWD,0,"",pBuf,0);
设置写入共享区域密码为"abcdefgx".
```

## 2. 18. 6 CPC\_STORAGE\_PUBLIC\_GETSPACESIZE 获取共享区域的总空间长度

功能描述 (Description)

获取共享区域的总空间长度

参数介绍 (Parameters)

[in] nDevID  
共享区域的设备 ID

[in] uSeek        - 保留

[in] pPwd        - 保留

[in] pValue      - 保留

[in] nBufSize    - 保留

返回值 (Return Values)

大于 0 表示成功,其它表示失败

备注 (Remarks)

- 正常 I 系列返回的长度正常为大于 128 字节, 其它系列产品没有存储功能

例如 (Example)

```
CPCINT32 iBufSize=0;
iBufSize =CPC_Storage(0, CPC_STORAGE_PUBLIC_GETSPACESIZE,0,"",0,0);
```

## 2. 19 通用工具类 (CPC\_Tool)

功能描述 (Description)

通用工具函数,打开 CC301 设备前该工具函数不能使用

函数原型 (Function)

```
CPCINT32    CPC_Tool (/*[in]*/CPCUINT32 uToolType,
                      /*[in]*/CPCINT32    nValue,
                      /*[in]*/CPCCHAR8 *pInValue,
                      /*[in]*/CPCCHAR8 *pInValueEx,
                      /*[out]*/CPCCHAR8 *pOutValue,
                      /*[in]*/CPCINT32 lBufSize);
```

返回值 (Return Values)

有符号 32bit 位.



## 参数描述 (Parameters description)

### 2. 19. 1 CPC\_TOOL\_PSTNEND 检测中国大陆的号码长度是否结束

#### 功能描述 (Description)

检测中国大陆的号码长度是否结束

#### 参数介绍 (Parameters)

[in] *nValue* - 保留

[in] *pInValue*

需要检测的号码

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

#### 返回值 (Return Values)

小于 0 - 表示检测失败

0 - 号码没有结束

1 - 号码已结束

#### 备注 (Remarks)

- 该功能返回结果仅做参考不保存完全正确

- 如：随着中国大陆区号的位数改动，可能就识别该区号的号码会错误

#### 例如 (Example)

```
CPC_Tool(CPC_TOOL_PSTNEND,0,"01088888888",0,0,0);
```

- 010 的后面应该是 8 位，已经结束返回 1

```
CPC_Tool(CPC_TOOL_PSTNEND,0,"0107777777",0,0,0);
```

- 010 的后面应该是 8 位，7 位表示未结束,返回 0

### 2. 19. 2 CPC\_TOOL\_CODETYPE 检测中国大陆的号码类型

#### 功能描述 (Description)

检测中国大陆的号码类型

#### 参数介绍 (Parameters)

[in] *nValue* - 保留

[in] *pInValue*

需要检测的号码

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

#### 返回值 (Return Values)

小于 0 - 表示检测失败

CTT\_NULL - 未知

CTT\_MOBILE - 手机

CTT\_PSTN - 固话

#### 备注 (Remarks)



- 仅作参考
- 只适合中国大陆的号码

**例如 (Example)**

```
CPC_Tool(CPC_TOOL_PSTNEND,0,"01088888888",0,0,0);
```

- 返回 CTT\_PSTN

```
CPC_Tool(CPC_TOOL_PSTNEND,0,"13800000000",0,0,0);
```

- 返回 CTT\_MOBILE

## 2. 19. 3 CPC\_TOOL\_LOCATION 获取中国大陆的号码所在地信息

**功能描述 (Description)**

获取中国大陆的号码所在地信息(区号/城市名)

**参数介绍 (Parameters)**

[in] *nValue* - 保留

[in] *pInValue*

需要检测的号码

[in] *pInValueEx* - 保留

[out] *pOutValue*

保存所在地信息的内存地址

[in] *lBufSize*

保存所在地信息的内存长度

**返回值 (Return Values)**

大于 0 其它表示成功，其它表示失败

**备注 (Remarks)**

- 仅作参考
- 由于随着中国手机号码归属地的升级，该数据信息不一定完全准确

**例如 (Example)**

```
char szBuf[128]; //分配足够大的空间
```

```
CPC_Tool(CPC_TOOL_LOCATION,0,"01088888888",0,szBuf,128);
```

## 2. 19. 4 CPC\_TOOL\_DISKFREESPACE 获取该硬盘剩余空间

**功能描述 (Description)**

获取该硬盘剩余空间.单位: 兆

**参数介绍 (Parameters)**

[in] *nValue* - 保留

[in] *pInValue*

需要获取的盘符.如: "c:\\"

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

**返回值 (Return Values)**



小于 0 其它表示失败，其它返回该盘符的剩余空间大小。单位：兆

**备注 (Remarks)**

- 该功能建议在开发 B/S 模式时使用，C/S 模式开发时用户可以使用 Windows 的 API 自行读取

**例如 (Example)**

```
CPC_Tool(CPC_TOOL_DISKFREESPACE,0,"C:\\",0,0,0);
```

## 2. 19. 5 CPC\_TOOL\_DISKTOTALSPACE 获取该硬盘总共空间

**功能描述 (Description)**

获取该硬盘总共空间.单位：兆

**参数介绍 (Parameters)**

[in] *nValue* - 保留

[in] *pInValue*

需要获取的盘符.如: "c:\""

[in] *pInValueEx* - 保留

[out]*pOutValue* - 保留

[in] *lBufSize* - 保留

**返回值 (Return Values)**

小于 0 其它表示失败，其它返回该盘符的剩余空间大小。单位：兆

**备注 (Remarks)**

- 打开 CC301 设备后才能使用

- 该功能建议在开发 B/S 模式时使用，C/S 模式开发时用户可以使用 Windows 的 API 自行读取

**例如 (Example)**

```
CPC_Tool(CPC_TOOL_DISKTOTALSPACE,0,"C:\\",0,0,0);
```

## 2. 19. 6 CPC\_TOOL\_DISKLIST 获取该硬盘所有的盘符列表

**功能描述 (Description)**

获取该硬盘所有的盘符列表

**参数介绍 (Parameters)**

[in] *nValue* - 保留

[in] *pInValue* - 保留

[in] *pInValueEx* - 保留

[out]*pOutValue*

保存盘符列表的内存地址

[in] *lBufSize*

分配给保存列表的内存大小

**返回值 (Return Values)**

大于 0 表示成功，其它表示失败

**备注 (Remarks)**



- 多个盘符号之间使用','分隔.如: "c:\d\,:e:\,
- 该功能建议在开发 B/S 模式时使用, C/S 模式开发时用户可以使用 Windows 的 API 自行读取

**例如 (Example)**

```
char szBuf[128];  
CPC_Tool(CPC_TOOL_DISKLIST,0,0,0,szBuf,128);
```

## 2. 19. 7 CPC\_TOOL\_CONVERTFMT 转换 wav 语音文件格式

**功能描述 (Description)**

转换 wav 语音文件格式

**参数介绍 (Parameters)**

[in] *nValue*

转换到目的语音格式 ID

- 该格式只能为 CC301 定义语音格式 ID 值

[in] *pInValue*

需要转换的已经存在的语音文件

[in] *pInValueEx*

保存转换后语音文件的路径

[out]*pOutValue* - 保留

[in] *lBufSize* - 保留

**返回值 (Return Values)**

成功返回大于 0, 其它表示失败

**备注 (Remarks)**

- 无

**例如 (Example)**

```
CPC_Tool(CPC_TOOL_CONVERTFMT,  
BRI_WAV_FORMAT_GSM6108K,"c:\a.wav","c:\b.wav",0,0);  
- 把 c:\a.wav 转换成 GSM610 格式后保存到 c:\b.wav
```

## 2. 19. 8 CPC\_TOOL\_SELECTDIRECTORY 弹出一个选择目录的对话框

**功能描述 (Description)**

弹出一个选择目录的对话框

**参数介绍 (Parameters)**

[in] *nValue* - 保留

[in] *pInValue*

显示在对话框的标题名,不能为空

[in] *pInValueEx* - 保留

[out]*pOutValue*

保存选择的完整目录路径的内存地址

[in] *lBufSize*





给保存目录分配的内存长度

**返回值 (Return Values)**

成功返回大于 0，其它表示失败

**备注 (Remarks)**

- 该功能建议在开发 B/S 模式时使用，C/S 模式开发时用户可以使用 Windows 的 API 获取

**例如 (Example)**

```
char szBuf[260]; // 分配足够长的内存
CPC_Tool(CPC_TOOL_SELECTDIRECTORY,
0,
"选择保存录音的目录",
0,
szBuf,
260);
```

## 2. 19. 9 CPC\_TOOL\_SELECTFILE 弹出一个选择文件的对话框

**功能描述 (Description)**

弹出一个选择文件的对话框

**参数介绍 (Parameters)**

[in] *nValue*

选择类型

- |   |   |            |
|---|---|------------|
| 0 | — | 打开一个已存在的文件 |
| 1 | — | 选择保存的目标文件  |

[in] *pInValue*

显示的文件后缀名列表

[in] *pInValueEx*

默认选择的文件名

[out] *pOutValue*

保存选择的文件完整路径的内存地址

[in] *lBufSize*

给保存文件完整路径分配的内存长度

**返回值 (Return Values)**

成功返回大于 0，其它表示失败

**备注 (Remarks)**

- 该功能建议在开发 B/S 模式时使用，C/S 模式开发时用户可以使用 Windows 的 API 获取

**例如 (Example)**

```
char szBuf[260]; // 分配足够长的内存
CPC_Tool(CPC_TOOL_SELECTFILE,
0,
"wav files|.wav|all files|..*|",
"a.wav",
```



```
szBuf,  
260);
```

## 2. 19. 10 CPC\_TOOL\_APMQUERYSUSPEND 是否允许 PC 系统进入待机/休眠状态

### 功能描述 (Description)

打开/关闭允许 PC 系统进入待机/休眠状态

### 参数介绍 (Parameters)

[in] *nValue*

- 0 - 不允许 PC 系统进入待机/休眠状态
- 1 - 允许 PC 系统进入待机/休眠状态

[in] *pInValue* - 保留

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

### 返回值 (Return Values)

成功返回大于 0，其它表示失败

### 备注 (Remarks)

- 默认允许 PC 系统进入待机状态
- CC301 设备被成功打开后才能使用
- PC 系统进入待机/休眠状态后设备将不能正常使用
- 为避免用户在使用时长时间没有动作后 PC 系统进入如待机状态后，设备就不能正常工作而丢失来电号码，建议用户不要允许 PC 系统自动进入如待机/休眠状态

### 例如 (Example)

```
CPC_Tool(CPC_TOOL_APMQUERYSUSPEND,0,0,0,0,0);
```

- 禁止 PC 系统进入待机/休眠状态

```
CPC_Tool(CPC_TOOL_APMQUERYSUSPEND,1,0,0,0,0);
```

- 允许 PC 系统进入待机/休眠状态

## 2. 19. 11 CPC\_TOOL\_SLEEP 让调用该方法的线程堵塞一定时间

### 功能描述 (Description)

让调用该方法的线程堵塞一定时间

### 参数介绍 (Parameters)

[in] *nValue*

堵塞的时长.(单位: 毫秒)

[in] *pInValue* - 保留

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

### 返回值 (Return Values)



成功返回大于 0，其它表示失败

#### 备注 (Remarks)

- 该功能建议在开发 B/S 模式时使用，C/S 模式开发时用户可以使用 Windows 的 API 实现
- 非特殊需求，不需要使用该功能

#### 例如 (Example)

```
CPC_Tool(CPC_TOOL_SLEEP,1000,0,0,0,0);
```

- 调用该功能的线程被堵塞 1 秒，也就是该功能将在 1 秒后才能返回

## 2. 20 WEB 服务器操作(CPC\_Remote)

### 功能描述 (Description)

该模块实现跟 WEB 服务器进行数据交换操作

### 函数原型 (Function)

```
CPCINT32 CPC_Remote (/*[in]*/CPCUINT32 uRemoteType,
/*[in]*/CPCINT32 nValue,
/*[in]*/CPCCHAR8 *pInValue,
/*[in]*/CPCCHAR8 *pInValueEx,
/*[out]*/CPCCHAR8 *pOutValue,
/*[in]*/CPCINT32 lBufSize);
```

### 返回值 (Return Values)

有符号 32bit 位。

### 参数描述 (Parameters description)

*uRemoteType*

[in] 指定交换控制类型

### 2. 20. 1 CPC\_REMOTE\_UPLOADFILE 使用 http 协议上传文件到 WEB 服务器

#### 功能描述 (Description)

使用 http 协议上传文件到 WEB 服务器

#### 参数介绍 (Parameters)

[in] *nValue*

操作类型

OPTYPE\_REMOVE - 上传完成后删除原文件



其它 - 无效

[in] *pInValue*

远程 WEB 服务器的接收上传文件的 URL

[in] *pInValueEx*

需要上传的原文件路径

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

#### 返回值 (Return Values)

- 小于等于 0 表示失败, 其它返回本次操作的唯一句柄,用户必须保存该句柄,以便在事件回调中能知道是哪次上传成功或失败
- 该操作成功不意味的上传成功,再接收到 `BriEvent_UploadSuccess` 事件后才表示上传成功,(服务器返回 `Result:1`)

#### 备注 (Remarks)

- 该功能建议在开发 B/S 模式时使用, C/S 模式开发时用户可以使用 Windows 的 API 实现
- 接收到 `BriEvent_UploadSuccess` 表示上传成功,服务器返回 `Result:1` 表示成功
- 接收到 `BriEvent_UploadFailed` 表示上传失败,服务器返回 `Result:0` 表示失败

#### 例如 (Example)

```
CPC_Remote(CPC_REMOTE_UPLOADFILE,  
0,  
"http://192.168.0.1/fileupload.asp"  
"c:a.wav",  
0,  
0);
```

- 把 c:a.wav 上传到 192.168.0.1 服务器,fileupload.asp 为接收保存文件的页面

## 2. 20. 2 CPC\_REMOTE\_DOWNLOADFILE 使用 http 协议下载文件

#### 功能描述 (Description)

使用 http 协议下载文件(该功能保留)

#### 参数介绍 (Parameters)

[in] *nValue* - 保留

[in] *pInValue* - 保留

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

#### 返回值 (Return Values)

- 小于等于 0 表示失败, 其它成功

#### 备注 (Remarks)

- 保留功能, 未实现

#### 例如 (Example)

无



## 2. 20. 3 CPC\_REMOTE\_UPLOADDATA 使用 http 上传数据

### 功能描述 (Description)

使用 http 上传数据(SEND/POST)到 WEB 服务器 (该功能保留)

### 参数介绍 (Parameters)

[in] *nValue* - 保留

[in] *pInValue* - 保留

[in] *pInValueEx* - 保留

[out] *pOutValue* - 保留

[in] *lBufSize* - 保留

### 返回值 (Return Values)

- 小于等于 0 表示失败, 其它成功

### 备注 (Remarks)

- 保留功能, 未实现

### 例如 (Example)

无

<完>

<End>