

线上开发者更新V3.12.00 iOS SDK注意事项

客户端	代码类	V3.12.00版SDK对比旧版的改动	说明
iOS SDK	LCOpenSDK_PlayWindow	该播放类标记为废弃。建议开发者跟自己的播放业务需求切换至新的API类：实时播放API类（LCOpenSDK_PlayRealWindow）、回放播放API类（LCOpenSDK_PlayBackWindow）、本地文件播放API类（LCOpenSDK_PlayFileWindow）	
	LCOpenSDK_PlayRealWindow	<pre>@property(nonatomic, assign)LCOpenSDK_PlayStatus playStatus; //当前播放状态 /// 是否支持电子放大(默认支持) @property(nonatomic, assign)BOOL isZoomEnabled; /** * 初始化播放窗口 * * @param frame 窗口 * @param index 播放窗口索引 * * @return LCOpenSDK_PlayWindow指针 */ - (id)initPlayWindow:(CGRect)frame Index:(NSInteger)index; /** * 反初始化播放窗口 */ - (void)uninitPlayWindow; /** * 设置播放窗口 * * @param rect 播放窗口 */ - (void)setWindowFrame:(CGRect)rect; /** * 设置播放窗口背景色 * * @param normalColor 背景色 */ - (void)setSurfaceBGColor:(UIColor*)normalColor; /** * 获取播放窗口 * * @return 播放窗口指针 */ - (UIView*)getWindowView; /** * 设置播放窗口监听对象 * * @param lis 监听对象指针 */ - (void)setPlayRealListener:(id<LCOpenSDK_PlayRealListener>)listener; /** * 设置播放窗口手势事件监听 * * @param lis 监听对象指针 */ - (void)setTouchListener:(id<LCOpenSDK_TouchListener>)listener; /** * 设置播放窗口手势操作事件是否被上层窗口捕获 * * @param flag 布尔值 */ - (void)openTouchListener:(BOOL)flag; /** * 播放实时视频 * * @param paramReal 实时视频参数模型 * * @return 0, 接口调用成功 * -1, 接口调用失败 */ - (NSInteger)playRtspReal:(LCOpenSDK_ParamReal *)paramReal;</pre>	新实时播放API类

```

/**
 * 停止实时视频播放
 *
 * @param isKeepLastFrame 是否保留最后一帧画面
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)stopRtspReal:(BOOL)isKeepLastFrame;

/**
 * 播放音频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)playAudio;

/**
 * 停止音频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)stopAudio;

/**
 * 截图
 *
 * @param filePath 图片保存路径
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)snapShot:(NSString*)filePath;

/**
 * 开始录制视频
 *
 * @param filePath 录制视频保存路径
 * @param nRecordType 录制视频格式: 0, dav
 *                   1, mp4
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)startRecord:(NSString*)filePath recordType:(NSInteger)nRecordType;

/**
 * 停止录制视频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */

```

LCOpenSDK_EventListenerProxy

	<pre>*/ - (NSInteger)stopRecord; /** * 设置PS/TS标准流导出 * * @param streamFormat [in] 标准流类型 */ - (void)setStreamCallback:(LC_OUTPUT_STREAM_FORMAT)streamFormat; /** * EPTZ缩放操作 * * @param scale 缩放比例 */ - (void)doScale:(CGFloat)scale; /** * 获取EPTZ缩放比例 * * @return - 1, 失败 * 其他, 成功 */ - (CGFloat)getScale; /** * EPTZ滑动操作 * * @param x 播放窗口X坐标值 * @param y 播放窗口Y坐标值 */ - (void)doTranslateX:(CGFloat)x Y:(CGFloat)y; //MARK: - 扩展功能 /** * 设置去噪模式等级 */ - (void)setSEnhanceMode:(LCOpenSDK_EnhanceMode)mode;</pre>	
LCOpenSDK_PlayBackWindow	<pre>@property(nonatomic, assign)LCOpenSDK_PlayStatus playStatus; //当前播放状态 /// 是否支持电子放大(默认支持) @property(nonatomic, assign)BOOL isZoomEnabled;</pre>	新回放PAI类

```

/**
 * 初始化播放窗口
 *
 * @param frame 窗口
 * @param index 播放窗口索引
 *
 * @return LCOpenSDK_PlayWindow指针
 */
- (id)initPlayWindow:(CGRect)frame Index:(NSInteger)index;
/** * 反初始化播放窗口
 */
- (void)uninitPlayWindow;
/**
 * 设置播放窗口
 *
 * @param rect 播放窗口
 */
- (void)setWindowFrame:(CGRect)rect;
/**
 * 设置播放窗口背景色
 *
 * @param normalColor 背景色
 */
- (void)setSurfaceBGColor:(UIColor*)normalColor;
/**
 * 获取播放窗口
 *
 * @return 播放窗口指针
 */
- (UIView*)getWindowView;
/**
 * 设置播放窗口监听对象
 *
 * @param lis 监听对象指针
 */
- (void)setPlayBackListener:(id<LCOpenSDK_PlayBackListener>)listener;
/**
 * 设置播放窗口手势事件监听
 *
 * @param lis 监听对象指针
 */
- (void)setTouchListener:(id<LCOpenSDK_TouchListener>)listener;
/**
 * 设置播放窗口手势操作事件是否被上层窗口捕获
 *
 * @param flag 布尔值
 */
- (void)openTouchListener:(BOOL)flag;
/**
 * 开始录像播放
 *
 * @param paramRecord 录像参数模型(该方法为通用方法,LCOpenSDK_ParamRecord为参数模型父类,请根据播
放类型(云录像/设备录像),传入对应子类参数模型),
云录像播放参数子类模型:LCOpenSDK_ParamCloudRecord
设备录像播放参数子类模型:LCOpenSDK_ParamDeviceRecord
禁止直接使用LCOpenSDK_ParamRecord参数对象传入
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)playRecordStream:(LCOpenSDK_ParamRecord *)paramRecord;
/**
 * 停止云录像播放
 *
 * @param isKeepLastFrame 是否保留最后一帧画面
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)stopRecordStream:(BOOL)isKeepLastFrame;

- (NSInteger)setPlaySpeed:(float)speed;

/**
 * 播放音频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */

```

```

- (NSInteger)playAudio;
/**
 * 停止音频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)stopAudio;
/**
 * 录像拖动
 *
 * @param timeInfo 相对开始时间偏移的秒数
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)seek:(NSInteger)timeInfo;
/**
 * 暂停播放
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)pause;
/**
 * 恢复播放
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)resume;
/**
 * 截图
 *
 * @param filePath 图片保存路径
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)snapShot:(NSString*)filePath;
/**
 * 开始录制视频
 *
 * @param filePath 录制视频保存路径
 * @param nRecordType 录制视频格式: 0, dav
 *                   1, mp4
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)startRecord:(NSString*)filePath recordType:(NSInteger)nRecordType;
/**
 * 停止录制视频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)stopRecord;
/**
 * 设置PS/TS标准流导出
 *
 * @param streamFormat [in] 标准流类型
 */
- (void)setStreamCallback:(LC_OUTPUT_STREAM_FORMAT)streamFormat;
/**
 * EPTZ缩放操作
 *
 * @param scale 缩放比例
 */
- (void)doScale:(CGFloat)scale;
/**
 * 获取EPTZ缩放比例
 *
 * @return -1, 失败
 *         其他, 成功
 */
- (CGFloat)getScale;
/**
 * EPTZ滑动操作

```

	<pre> * * @param x 播放窗口X坐标值 * @param y 播放窗口Y坐标值 */ - (void)doTranslateX:(CGFloat)x Y:(CGFloat)y; //MARK: - 扩展功能 /** * 设置去噪模式等级 */ - (void)setSEnhanceMode:(LCOpenSDK_EnhanceMode)mode; </pre>	
LCOpenSDK_PlayFileWindow	<pre> @property(nonatomic, assign)LCOpenSDK_PlayStatus playStatus; //当前播放状态 /// 是否支持电子放大(默认支持) @property(nonatomic, assign)BOOL isZoomEnabled; /** * 初始化播放窗口 * * @param frame 窗口 * @param index 播放窗口索引 * * @return LCOpenSDK_PlayWindow指针 */ - (id)initPlayWindow:(CGRect)frame Index:(NSInteger)index; /** * 反初始化播放窗口 */ - (void)uninitPlayWindow; /** * 设置播放窗口 * * @param rect 播放窗口 */ - (void)setWindowFrame:(CGRect)rect; /** * 设置播放窗口背景色 * * @param normalColor 背景色 */ - (void)setSurfaceBGColor:(UIColor*)normalColor; /** * 获取播放窗口 * * @return 播放窗口指针 */ - (UIView*)getWindowView; /** * 设置播放窗口监听对象 * * @param lis 监听对象指针 */ - (void)setPlayFileListener:(id<LCOpenSDK_PlayFileListener>)listener; /** * 设置播放窗口手势事件监听 * * @param lis 监听对象指针 */ - (void)setTouchListener:(id<LCOpenSDK_TouchListener>)listener; /** * 设置播放窗口手势操作事件是否被上层窗口捕获 * * @param flag 布尔值 */ - (void)openTouchListener:(BOOL)flag; - (NSInteger)setPlaySpeed:(float)speed; - (NSInteger)playFile:(NSString*)fileName; - (NSInteger)stopFile:(BOOL)isKeepLastFrame; </pre>	新本地文件播放API类

```

/**
 * 播放音频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)playAudio;
/**
 * 停止音频
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)stopAudio;
/**
 * 录像拖动
 *
 * @param timeInfo 相对开始时间偏移的秒数
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)seek:(NSInteger)timeInfo;
/**
 * 暂停播放
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)pause;
/**
 * 恢复播放
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)resume;
/**
 * 截图
 *
 * @param filePath 图片保存路径
 *
 * @return 0, 接口调用成功
 *         -1, 接口调用失败
 */
- (NSInteger)snapShot:(NSString*)filePath;
/**
 * 设置PS/TS标准流导出
 *
 * @param streamFormat [in] 标准流类型
 */
- (void)setStreamCallback:(LC_OUTPUT_STREAM_FORMAT)streamFormat;
/**
 * EPTZ缩放操作
 *
 * @param scale 缩放比例
 */
- (void)doScale:(CGFloat)scale;
/**
 * 获取EPTZ缩放比例
 *
 * @return -1, 失败
 *         其他, 成功
 */
- (CGFloat)getScale;
/**
 * EPTZ滑动操作
 *
 * @param x 播放窗口X坐标值
 * @param y 播放窗口Y坐标值
 */
- (void)doTranslateX:(CGFloat)x Y:(CGFloat)y;

//MARK: - 扩展功能
/**
 * 设置去噪模式等级
 */
- (void)setSEnhanceMode:(LCOpenSDK_EnhanceMode)mode;

```

	LCOpenSDK_ParamReal	去除priorityP2P属性 /** 默认值 0 海外优先尝试P2P，国内不尝试P2P, 1 优先尝试P2P, 2 不尝试P2P*/	
	LCOpenSDK_ParamDeviceRecordFileName	去除priorityP2P属性 /** 默认值 0 海外优先尝试P2P，国内不尝试P2P, 1 优先尝试P2P, 2 不尝试P2P*/	
	LCOpenSDK_ParamDeviceRecordUTCTime	去除priorityP2P属性 /** 默认值 0 海外优先尝试P2P，国内不尝试P2P, 1 优先尝试P2P, 2 不尝试P2P*/	

LCOpenSDK_PlayRealListerner	<pre> /** * 视频开始加载回调 * * @param index 播放窗口索引值 */ - (void)onPlayLoading:(NSInteger)index; /** * 视频播放开始回调 * * @param index 播放窗口索引值 */ - (void)onPlayBegan:(NSInteger)index; /** * 视频播放停止回调 * * @param index 播放窗口索引值 */ - (void)onPlayStop:(NSInteger)index; /** * 视频播放状态失败回调 * * @param code 错误码(根据type而定) * @param type 0, RTSP * 1, HLS * 99, OPENAPI * @param index 播放窗口索引值 */ - (void)onPlayFail:(NSString*)code Type:(NSInteger)type Index:(NSInteger)index; /** * 分辨率改变时回调 * * @param width 新分辨率长度 * @param height 新分辨率高度 */ - (void)onResolutionChanged:(NSInteger)width Height:(NSInteger)height Index:(NSInteger)index; /** * 视频播放数据回调 * * @param len 数据长度 * @param index 播放窗口索引值 */ - (void)onReceiveData:(NSInteger)len Index:(NSInteger)index; /** * TS/PS标准流导出数据回调 * * @param data [out] 标准流导出数据 */ - (void)onStreamCallback:(NSData*)data Index:(NSInteger)index; /** * 云台到达限位点 * * @param direction 到达限位的方向 */ - (void)onPtzLimit:(LCOpenSDK_Direction)direction; /** * 停止录制 * * @param error 录制停止错误类型 * @param index 播放窗口索引值 */ - (void)onRecordStop:(NSInteger)error Index:(NSInteger)index; </pre>	新实时播放生命周期回调协议，配合LCOpenSDK_PlayRealWindow使用
LCOpenSDK_PlayBackListerner	<pre> /** * 视频开始加载回调 * * @param index 播放窗口索引值 */ - (void)onPlayLoading:(NSInteger)index; </pre>	新回放生命周期回调协议，配合LCOpenSDK_PlayBackWindow使用

	<pre> /** * 视频播放开始回调 * * @param index 播放窗口索引值 */ - (void)onPlayBegan:(NSInteger)index; /** * 视频播放停止回调 * * @param index 播放窗口索引值 */ - (void)onPlayStop:(NSInteger)index; /** * 视频播放暂停回调 * * @param index 播放窗口索引值 */ - (void)onPlayPause:(NSInteger)index; /** * 视频播放完成回调 * * @param index 播放窗口索引值 */ - (void)onPlayFinished:(NSInteger)index; /** * 视频播放状态失败回调 * * @param code 错误码(根据type而定) * @param type 0, RTSP * 1, HLS * 99, OPENAPI * @param index 播放窗口索引值 */ - (void)onPlayFail:(NSString*)code Type:(NSInteger)type Index:(NSInteger)index; /** * 视频播放帧时间回调 * * @param time 视频帧时间回调 * @param index 播放窗口索引值 */ - (void)onPlayerTime:(long)time Index:(NSInteger)index; /** * 分辨率改变时回调 * * @param width 新分辨率长度 * @param height 新分辨率高度 */ - (void)onResolutionChanged:(NSInteger)width Height:(NSInteger)height Index:(NSInteger)index; /** * 视频播放数据回调 * * @param len 数据长度 * @param index 播放窗口索引值 */ - (void)onReceiveData:(NSInteger)len Index:(NSInteger)index; /** * TS/PS标准流导出数据回调 * * @param data [out] 标准流导出数据 */ - (void)onStreamCallback:(NSData*)data Index:(NSInteger)index; /** * 停止录制 * * @param error 录制停止错误类型 * @param index 播放窗口索引值 */ - (void)onRecordStop:(NSInteger)error Index:(NSInteger)index; </pre>	
LCOpenSDK_ PlayFileListen		本地文件播放生命周期回调协议类，配合

er

```
/**
 * 视频开始加载回调
 *
 * @param index 播放窗口索引值
 */
- (void)onPlayLoading:(NSInteger)index;

/**
 * 视频播放开始回调
 *
 * @param index 播放窗口索引值
 */
- (void)onPlayBegan:(NSInteger)index;

/**
 * 视频播放停止回调
 *
 * @param index 播放窗口索引值
 */
- (void)onPlayStop:(NSInteger)index;

/**
 * 视频播放暂停回调
 *
 * @param index 播放窗口索引值
 */
- (void)onPlayPause:(NSInteger)index;

/**
 * 视频播放完成回调
 *
 * @param index 播放窗口索引值
 */
- (void)onPlayFinished:(NSInteger)index;

/**
 * 视频播放状态失败回调
 *
 * @param code 错误码(根据type而定)
 * @param type 0, RTSP
 *              1, HLS
 *              99, OPENAPI
 * @param index 播放窗口索引值
 */
- (void)onPlayFail:(NSString*)code Type:(NSInteger)type Index:(NSInteger)index;

/**
 * 视频播放帧时间回调
 *
 * @param time 视频帧时间回调
 * @param index 播放窗口索引值
 */
- (void)onPlayerTime:(long)time Index:(NSInteger)index;

/**
 * 视频文件开始和结束时间回调
 *
 * @param beginTime 开始时间
 * @param EndTime 结束时间
 * @param index 播放窗口索引值
 */
- (void) onFileTime:(long)beginTime EndTime:(long)endTime Index:(NSInteger)index;

/**
 * 分辨率改变时回调
 *
 * @param width 新分辨率长度
 * @param height 新分辨率高度
 */
- (void)onResolutionChanged:(NSInteger)width Height:(NSInteger)height Index:(NSInteger)index;

/**
 * 视频播放数据回调
 *
 * @param len 数据长度
 * @param index 播放窗口索引值
 */
- (void)onReceiveData:(NSInteger)len Index:(NSInteger)index;
```

LCOpenSDK_PlayFileWi
ndow使用

	<pre> /** * TS/PS标准流导出数据回调 * * @param data [out] 标准流导出数据 */ - (void)onStreamCallback:(NSData*)data Index:(NSInteger)index; </pre>	
LCOpenSDK_TouchListener	<pre> /// Click Callback callback /// @param dx window X coordinate /// @param dy window Y coordinate /// @param index window index value /// /// 单击回调 /// @param dx 窗口X坐标 /// @param dy 窗口Y坐标 /// @param index 窗口索引值 - (void)onControlClick:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; ///Double click callback /// @param dx window X coordinate /// @param dy window Y coordinate /// @param index window index value /// /// 双击回调 /// @param dx 窗口X坐标 /// @param dy 窗口Y坐标 /// @param index 窗口索引值 - (void)onWindowDBClick:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; ///Long press to start callback /// @param dir Long press the direction in the relative window /// @param dx window X coordinate value /// @param dy window Y coordinate value /// @param index window index value /// /// 长按开始回调 /// @param dir 长按相对窗口中的方向 /// @param dx 窗口X坐标值 /// @param dy 窗口Y坐标值 /// @param index 窗口索引值 - (void)onWindowLongPressBegin:(Direction)dir dx:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; ///Long press to end callback /// @param index window index value /// /// 长按结束回调 /// @param index 窗口索引值 - (void)onWindowLongPressEnd:(NSInteger)index; </pre>	播放窗口手势回调协议

LCOpenSDK_Netsdk	<p>新增 LCOpenSDK_Netsdk , demo 直接引用 netsdk 的方法下沉至 sdk</p> <pre> /// 初始化 netsdk /// init netsdk /// @param callback netsdk连接断开回调 /// @param callback netsdk connection disconnection callback + (void)initSDKWithCallback:(LCNetSDKDisconnetCallback)callback; /// 开启日志 /// open log /// @param path 日志存放路径 /// @param path log path /// @param logCallback 日志回调 /// @param logCallback log callback + (void)logOpen:(NSString *)path callback:(LCNetSDKLogOpenCallback)logCallback; /// 关闭日志 /// close log + (void)logClose; /// 初始化设备 /// @param deviceNetInfo 设备信息 /// @param isUseIp 是否使用ip的方式 /// init device /// @param deviceNetInfo device info /// @param isUseIp Whether to use IP mode + (BOOL)initDevAccountWithDevInfo:(LCNetsdkDevInfo *)deviceNetInfo useIp:(BOOL)isUseIp; /// 登出设备 /// @param loginHandle 句柄 /// logout device /// @param loginHandle login handle + (void)logout:(long)loginHandle; //MARK: Device Password Reset /// 重置设备密码 /// reset device password /// @param password 密码 /// @param password password /// @param device 设备信息 /// @param device device info /// @param securityCode 安全码 /// @param securityCode security Code /// @param contact 接收验证码的手机/邮箱 /// @param contact Mobile phone/email receiving verification code /// @param useAsPreset 是否设置为预留联系方式 /// @param useAsPreset Set as Reserved Contact Method /// @param phoneIp 手机ip /// @param phoneIp phone ip /// @return 重置后的密码信息 /// @return Password information after reset + (LCDevicePWDResetInfo *)resetPassword:(NSString *)password device:(LCNetsdkDevInfo *)device securityCode:(NSString *)securityCode contact:(NSString *)contact useAsPreset:(BOOL)useAsPreset byPhoneIp:(NSString *)phoneIp; /// 异步查询设备支持的密码重置方式 /// Password reset mode supported by asynchronous query device /// @param device 设备 /// @param device /// @param phoneIp 手机ip , 可以为空 /// @param phoneIp Mobile IP, can be empty /// @return result + (LCDeviceResetPWDInfo *)queryPasswordResetType:(LCNetsdkDevInfo *)device byPhoneIp:(NSString *)phoneIp; #pragma mark - Production /// 获取设备信息 /// get device definition info /// @param loginHandle 句柄 + (LCNetsdkProductDefinitionInfo *)queryProductDefinitionInfo:(long)loginHandle; /// 获取是否具有云台权限 /// Get whether you have PTZ permission /// @param loginHandle 句柄 + (BOOL)queryDevicePtzAuth:(long)loginHandle; </pre>	
------------------	---	--

<div>LCOpenSDK_SearchDevices</div>	<div>新增，直接将原有 LCOpenSDK_SearchDevice 修改为 LCOpenSDK_SearchDevices；统一设备搜索API；废弃历史方法；</div> <div>/// 初始化单例 + (instancetype)shareSearchDevices;</div> <div>/// 搜索设备 search devices /// @param deviceId 非必传，搜索单个设备时，传入此参数 /// @param timeout 必传，搜索设备超时时间 /// @param searchDeviceCallback 搜索设备回调 - (void)startSearchDevicesWithDeviceId:(NSString *)deviceId timeout:(int)timeout callback:(void (^)(LCOpenSDK_SearchDeviceInfo *deviceInfo))searchDeviceCallback;</div> <div>/// 停止搜索设备 stop search devices /// @param handle handle description - (void)stopSearchDevices;</div> <div>/// 错误信息 + (unsigned int)getLastError;</div> <div>//MARK: - DEPRECATED METHOD</div> <div>/// 开始搜索设备（方法废弃） /// /// @param searchDeviceCallback 回调信息，id类型 /// @param localIp ip信息 /// /// @return 搜索句柄，用于停止设备搜索 - (long)startSearchDevices:(void (^)(LCOpenSDK_SearchDeviceInfo *deviceInfo))searchDeviceCallback byLocalIp:(NSString *)localIp DEPRECATED_MSG_ATTRIBUTE("use startSearchDevicesWithDeviceId:timeout:callback: instead");</div> <div>/// 停止搜索设备（方法废弃） /// @param handle handle description - (void)stopSearchDevices:(long)handle DEPRECATED_MSG_ATTRIBUTE("use stopSearchDevices instead");</div>	
------------------------------------	--	--

<p>LCOpenSDK_DeviceInit</p>	<p>更改点：1、设备初始化方法参数合并；2、设备登录方法合并；3、废弃老的API；</p> <pre> /// 初始化设备 /// @param mac device mac address /// @param password device password /// @param ip device ip - (int)initDevice:(nonnull NSString *)mac password:(nonnull NSString *)password ip:(nullable NSString *)ip; /// Check if device passcode is available /// @param deviceId device id /// @param ip IP address /// @param port port number /// @param password password /// 检查设备密码是否可用 /// @param deviceId 设备id /// @param ip IP地址 /// @param port 端口号 /// @param password 密码 - (int)checkPwdValidity:(nonnull NSString *)deviceId ip:(nonnull NSString *)ip port:(NSInteger)port password:(nonnull NSString *)password; /// Login to the device /// @param devIp IP address /// @param port port number /// @param username username /// @param password password /// @param security YES: safe mode login NO: compatibility mode login /// @param errorCode error code /// /// 登录设备 /// @param devIp IP地址 /// @param port 端口号 /// @param username 用户名 /// @param password 密码 /// @param security YES : 安全模式登录 NO : 兼容模式登录 /// @param errorCode 错误码 + (long)loginDeviceByIP:(NSString *)devIp port:(NSInteger)port username:(NSString *)username password:(NSString *)password highLevelSecurity:(BOOL)security errorCode:(unsigned int *)errorCode; //MARK: - DEPRECATED METHOD /// 初始化设备（废弃方法） /// @param mac device mac address /// @param password device password - (int)initDevice:(NSString *)mac password:(NSString *)password DEPRECATED_MSG_ATTRIBUTE("use initDevice:password:ip: instead"); /// 兼容模式登陆，已废弃，后续删除 /// @param devIp IP地址 /// @param port 端口号 /// @param username 用户名 /// @param password 密码 /// @param success /// @param failure + (long)loginDeviceByIP:(NSString *)devIp port:(NSInteger)port username:(NSString *)username password:(NSString *)password errorCode:(unsigned int *)errorCode DEPRECATED_MSG_ATTRIBUTE("use loginDeviceByIP:port:username:password:highLevelSecurity:errorCode: instead"); /// 安全模式登陆，已废弃，后续删除 /// @param devIp IP地址 /// @param port 端口号 /// @param username 用户名 /// @param password 密码 /// @param errorCode + (long)loginWithHighLevelSecurityByIP:(NSString *)devIp port:(NSInteger)port username:(NSString *)username password:(NSString *)password errorCode:(unsigned int *)errorCode DEPRECATED_MSG_ATTRIBUTE("use loginDeviceByIP:port:username:password:highLevelSecurity:errorCode: instead"); /** * 搜索设备初始化信息，已废弃，后续删除 * @param deviceId 设备ID * @param timeOut 超时时间 * @param info 搜索到的设备初始化信息 */ - (void)searchDeviceInitInfo:(NSString *)deviceId timeOut:(int)timeOut success:(void (^)(LCOpenSDK_DEVICE_INIT_INFO info))success DEPRECATED_MSG_ATTRIBUTE("use LCOpenSDK_SearchDevices startSearchDevicesWithDeviceId:timeOut:callback: instead"); </pre>	
-----------------------------	---	--

LCOpenSDK_ConfigWifi	<p>修改点：1、统一wifi设置的API参数；2、废弃老的API；</p> <pre> /// Start smartConfig distribution network /// @param devid device ID /// @param ssid Wifi SSID /// @param pwd Wifi password /// @param security (optional, default @ "") /// @param voiceFreq (acoustic frequency, which needs to be differentiated according to the type of equipment. For example: 11000 for general use, 17000 for TP7C) /// @return 0,interface call succeeded, -1,interface call failed /// /// 开始smartConfig配网 /// @param devid 设备ID /// @param ssid Wifi的SSID /// @param pwd Wifi密码 /// @param security (选填，默认@ "") /// @param voiceFreq (声波频率，需要根据设备类型区分。例如：普通11000，TP7C 17000) /// @return 0, 接口调用成功 /// -1, 接口调用失败 - (NSString *)configWifiStart:(NSString*)devid ssid:(NSString*)ssid password:(NSString*)pwd secure:(NSString*)security voiceFreq:(NSInteger)voiceFreq txMode:(NSInteger)txMode; ///Stop the Wifi distribution network ///@return 0 succeeded /// -1 failed /// /// 停止Wifi配网 /// @return 0, 接口调用成功 /// -1, 接口调用失败 - (NSInteger)configWifiStop; //MARK: - DEPRECATED METHOD /// 设置波形发送方式，已弃用，后续删除 /// @param fskMode 波形发送方式(0--新的fsk发送方式，1--老的fsk发送方式，2--新的和老的fsk波形发送方式) - (void)setFskMode:(NSInteger)fskMode DEPRECATED_MSG_ATTRIBUTE("use configWifiStart:ssid:password: secure:voiceFreq:txMode: instead"); /// 开始smartConfig配网，已弃用，后续删除 /// @param devid 设备ID /// @param ssid Wifi的SSID /// @param pwd Wifi密码 /// @param security (选填，默认@ "") /// @param voiceFreq (声波频率，需要根据设备类型区分。例如：普通11000，TP7C 17000) /// @return 0, 接口调用成功 /// -1, 接口调用失败 - (NSInteger)configWifiStart:(NSString*)devid ssid:(NSString*)ssid password:(NSString*)pwd secure: (NSString*)security voiceFreq:(NSInteger)voiceFreq DEPRECATED_MSG_ATTRIBUTE("use configWifiStart: ssid:password:secure:voiceFreq:txMode: instead"); </pre>	
LCOpenSDK_LoginManager	<p>修改点：新增 p2p port 获取的接口</p> <pre> + (LCOpenSDK_LoginManager*) sharedInstance; /// p2p预打通接口 /// @param token account token /// @param devices - (NSInteger)addDevices:(NSString * _Nonnull)token devices:(NSArray <LCOpenSDK_P2PDeviceInfo * > * _Nonnull)devices /// 获取P2P端口 /// @param p2pInfo LCOpenSDK_P2PInfo /// @param time 超时时间 - (NSInteger)getP2pPort:(LCOpenSDK_P2PInfo *)p2pInfo timeout:(NSInteger)time; </pre>	
LCOpenSDK_SearchWifi	<p>该类已弃用，相关方法已迁移至其他类</p>	

LCOpenSDK_ SoftApConfig	<p>修改点：1、统一softAP配网接口；2、弃用老接口；</p> <pre> /// 软AP配网 SoftAP distribution network /// @param wifiName wifi名字 /// @param wifiPwd wifi密码 /// @param wifiEncry wifi加密方式 0 : off, 2 : WEP64bit, 3 : WEP128bit, 4:WPA-PSK-TKIP, 5: WPA-PSK-CCMP /// @param netcardName wifi网卡名字 /// @param deviceIp 设备IP /// @param devicePwd 设备密码 /// @param isSC 设备是否有安全 /// @param handler 回调 /// @param timeout 超时时间 + (void)startSoftAPConfig:(NSString *)wifiName wifiPwd:(NSString *)wifiPwd wifiEncry:(int)wifiEncry netcardName:(NSString *)netcardName deviceIp:(NSString *)deviceIp devicePwd:(NSString *)devicePwd isSC:(BOOL)isSC handler:(void (^)(NSInteger result))handler timeout:(int)timeout; /// 软AP配网获取设备wifi列表 Wifi list of soft AP distribution network acquisition equipment /// @param deviceIP deviceIP IP地址 /// @param port port 端口号 /// @param password 设备密码 /// @param errorCode 错误码 /// @return 成功返回wifi列表 + (void)getSoftApWifiList:(NSString *)deviceIP port:(NSInteger)port devicePassword:(NSString *)password isSC:(BOOL)isSC success:(void (^)(NSArray <LCOpenSDK_WifiInfo *> * _Nullable list))success failure:(void (^)(NSInteger code, NSString* _Nullable describe))failure; /// 用句柄获取wifi信息 Get wifi info with handler /// @param loginHandle 登录句柄 login handler /// @param mssid 网络ssid network ssid /// @param errorCode 错误码 error code + (LCOpenSDK_WifiInfo *)queryWifiByLoginHandle:(long)loginHandle mssId:(NSString *)mssid errorCode:(unsigned int *)errorCode; /// 查询设备是否支持第三代扫描能力 /// Query whether the device supports the third generation scanning capability + (BOOL)querySupportWlanConfigV3:(long)loginHandle; //MARK: - DEPRECATED METHOD /// 老的配网API，已弃用，后续删除 /// The old distribution network API has been discarded and will be deleted later - (NSInteger)startSoftAPConfig:(NSString * _Nonnull)wifiName wifiPwd:(NSString * _Nonnull)wifiPwd deviceId:(NSString * _Nonnull)deviceId devicePwd:(NSString * _Nullable)devicePwd isSC:(BOOL)isSC DEPRECATED_MSG_ATTRIBUTE("use startSoftAPConfig:wifiPwd:wifiEncry:(int)wifiEncrydeviceId:devicePwd:isSC:handler:timeout: instead"); </pre>	
----------------------------	---	--