

# PC多媒体API文档

## 概览

播放器负责音视频解码、播放功能，底层依赖于p2p库的推流及音视频同步功能。

## 类型定义

- 呼叫类型

```
public class CallTypeEnum {
    public final static int CONN_TYPE_VIDEO_CALL = 0; //视频呼叫，双向音视频
    public final static int CONN_TYPE_MONITOR = 1; //监控，单向视频，双向音频（对讲）
    public final static int CONN_TYPE_PLAY_REC_FILE = 2; //录像回放
}
```

- 播放器状态类型

```
public class PlayerStateEnum {
    public final static int STATE_IDLE = 0;           //播放器初始状态
    public final static int STATE_INITIALIZED = 1;    //已经初始化
    public final static int STATE_PREPARING = 2;      //准备中
    public final static int STATE_READY = 3;          //准备完毕
    public final static int STATE_LOADING = 4;        //加载
    public final static int STATE_PLAY = 5;           //播放状态
    public final static int STATE_PAUSE = 6;          //暂停
    public final static int STATE_STOP = 7;           //停止
}
```

- 编码参数类型

```
public class AVConstants {
    /**
     * 音频类型
     */
    public static final int AUDIO_TYPE_PCM = 0;
    public static final int AUDIO_TYPE_G711A = 1;
    public static final int AUDIO_TYPE_G711U = 2;
    public static final int AUDIO_TYPE_PT_G726 = 3;
    public static final int AUDIO_TYPE_PT_AAC = 4;
    public static final int AUDIO_TYPE_PT_AMR = 5;
    public static final int AUDIO_TYPE_PT_ADPCM = 6;
    public static final int AUDIO_TYPE_MAX = 7;

    /**
     * 音频采样率
     */
    public static final int AUDIO_SAMPLE_RATE_8000 = 8000;    //!< 8K samplerate
    public static final int AUDIO_SAMPLE_RATE_11025 = 11025;  //!< 11.025K
    samplerate
}
```

```

    public static final int AUDIO_SAMPLE_RATE_12000 = 12000;    //!< 12K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_16000 = 16000;    //!< 16K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_22050 = 22050;    //!< 22.050K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_24000 = 24000;    //!< 24K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_32000 = 32000;    //!< 32K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_44100 = 44100;    //!< 44.1K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_48000 = 48000;    //!< 48K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_64000 = 64000;    //!< 64K
    samplerate
    public static final int AUDIO_SAMPLE_RATE_96000 = 96000;    //!< 96K
    samplerate

    /**
     * 音频声音模式
     */
    public static final int AUDIO_SOUND_MODE_MONO = 0;          /*mono*/
    public static final int AUDIO_SOUND_MODE_STEREO = 1;        /*stereo*/
    public static final int AUDIO_SOUND_MODE_NONE = 2;

    /**
     * 音频位宽
     */
    public static final int AUDIO_BIT_WIDTH_8 = 0;              /* 8bit width */
    public static final int AUDIO_BIT_WIDTH_16 = 1;              /* 16bit width*/
    public static final int AUDIO_BIT_WIDTH_32 = 2;              /* 32bit width*/
    public static final int AUDIO_BIT_WIDTH_NONE = 3;

    /**
     * 音频每帧采样数
     */
    public static final int AUDIO_SAMPLE_NUM_80 = 80;           //!<
    public static final int AUDIO_SAMPLE_NUM_160 = 160;         //!<
    public static final int AUDIO_SAMPLE_NUM_320 = 320;         //!<
    public static final int AUDIO_SAMPLE_NUM_480 = 480;         //!<
    public static final int AUDIO_SAMPLE_NUM_1024 = 1024;       //!<
    public static final int AUDIO_SAMPLE_NUM_2048 = 2048;       //!<

    /**
     * 视频类型
     */
    public static final int VIDEO_TYPE_NONE = 0;                //!<
    public static final int VIDEO_TYPE_H264 = 1;                //!<
    public static final int VIDEO_TYPE_MPEG4 = 2;                //!<
    public static final int VIDEO_TYPE_JPEG = 3;                //!<
    public static final int VIDEO_TYPE_MJPEG = 4;                //!<
    public static final int VIDEO_TYPE_H265 = 5;                //!<
}

```

# 播发器功能接口

## 1. 初始化播放器数据源

```
/**
 * 初始化播放器数据源
 *
 * @param deviceId 设备id
 * @param callType 类型参考CallTypeEnum
 * @param definition 类型参考枚举DEFINITION
 */
void setDataResource(uint64_t deviceId, int callType, int definition);
```

## 2. 设置音视频渲染组件

```
/**
 * 设置音视频渲染组件
 *
 * @param view 视频渲染组件
 */
void setVideoView(IotVideoView view);
```

## 3. 设置视频编码器

```
/**
 * 设置视频编码器
 *
 * 说明：音视频语音对讲时可配置，
 *       如果设置，播放器视频编码将采用用户自定义的视频编码器。
 *       如果不设置，将采用内置的编码器。
 * @param encoder 视频编码器
 */
void setVideoEncoder(IVideoEncoder encoder)
```

## 4. 设置视频解码器

```
/**
 * 设置视频解码器
 *
 * 说明：如果设置，播放器视频解码将采用用户自定义的视频解码器。
 *       如果不设置，将采用内置的解码器。
 * @param decoder 视频解码器
 */
void setVideoDecoder(IVideoDecoder decoder);
```

## 5. 设置音频编码器

```

/**
 * 设置视频编码器
 *
 * 说明：音视频语音对讲时可配置，
 *       如果设置，播放器视频编码将采用用户自定义的视频编码器。
 *       如果不设置，将采用内置的编码器。
 * @param encoder 视频编码器
 */
void setVideoEncoder(IVideoEncoder encoder)

```

## 6. 设置音频解码器

```

/**
 * 设置视频解码器
 *
 * 说明：如果设置，播放器视频解码将采用用户自定义的视频解码器。
 *       如果不设置，将采用内置的解码器。
 * @param decoder 视频解码器
 */
void setVideoDecoder(IVideoDecoder decoder);

```

## 7. 设置视频渲染

```

/**
 * 设置视频渲染
 *
 * 说明：如果用户自定义视频帧渲染可以设置该接口
 *       解码的视频帧（yuv420格式）已回调的形式返回
 */
void setVideoRender(IVideoRender listener);

```

## 8. 设置音频渲染

```

/**
 * 设置音频渲染
 *
 * 说明：如果用户自定义音频渲染可以设置该接口
 *       解码的音频帧（PCM格式）已回调的形式返回
 */
void setAudioRender(IAudioRender listener);

```

## 9. 准备播放

```

/**
 * 准备播放
 *
 * 说明：prepare回调可以通过设置setPreparedListener进行监听
 */
void prepare();

```

## 10. 开始播放

```
/**
 * 开始播放
 *
 */
void play();
```

#### 11. 停止播放

```
/**
 * 停止播放
 *
 */
void stop();
```

#### 12. 释放资源

```
/**
 * 释放
 *
 * 说明：播放器使用完毕需要释放
 */
void release();
```

#### 13. 静音状态判断

```
/**
 * 静音
 * @return true==静音, false==非静音
 */
boolean isMute();
```

#### 14. 静音状态设置

```
/**
 * 设置静音
 * @param on true==静音, false==非静音
 */
void mute(boolean on);
```

#### 15. 开始录像

```
/**
 * 录像
 *
 * @param path          录像文件路径
 * @param listener      录像回调
 */
void startRecord(String path, final IResultListener listener);
```

#### 16. 停止录像

```
/**
 * 停止录像
 *
 */
void stopRecord();
```

## 17. 截图

```
/**
 * 视频画面截图
 *
 * @param path          截图文件路径
 * @param listener      截图回调
 */
void snapShot(String path, final IResultListener listener);
```

## 18. 获取播放状态类型

```
/**
 * 获取播放状态
 *
 * @return 状态枚举，参考：
 * {@link PlayerStateEnum#STATE_IDLE} 播放器初始状态
 */
int getPlayState()
```

## 19. 获取播放的视频宽度

```
/**
 * 获取视频宽
 *
 * 说明：prepare 后才能获取到该数据，否则返回0
 * @return 视频宽度
 */
int getVideowidth()
```

## 20. 获取播放的视频高度

```
/**
 * 获取视频宽
 *
 * 说明：prepare 后才能获取到该数据，否则返回0
 * @return 视频高度
 */
int getVideoHeight()
```

## 21. 是否正在播放标识

```
/**
 * 是否在播放
 *
 * @return true==播放, false==没有播放
 */
bool isPlaying()
```

## 22. 设置状态回调

```
/**
 * 设置准备回调
 *
 */
void setPreparedListener(IPreparedListener listener);
```

## 23. 设置时间回调

```
/**
 * 设置时间回调
 *
 */
void setTimeListener(ITimeListener listener);
```

## 24. 设置错误回调

```
/**
 * 设置错误回调
 *
 */
void setErrorListener(IErrorListener listener);
```