

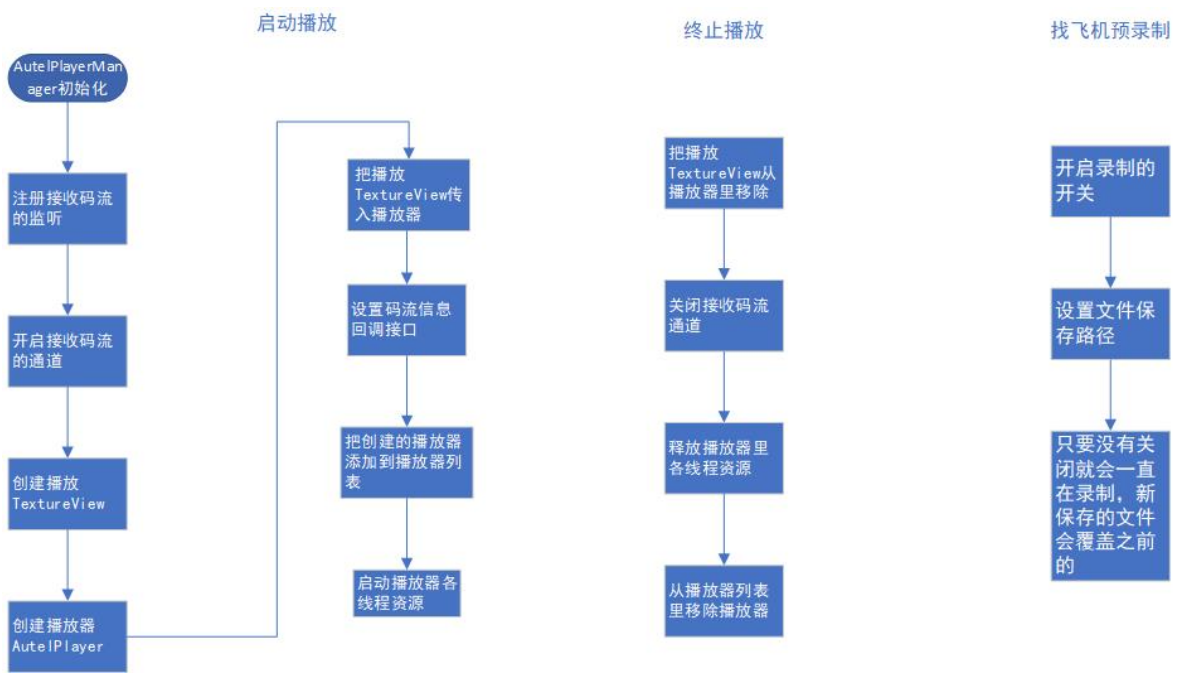
媒体中心教程文档

1. 概述

新媒体中心，支持多码流播放，启动播放、暂停播放、恢复播放、终止播放，找飞机预录制，获取实时码流信息。

2. 流程图

启动播放、终止播放、找飞机预录制的流程图



3. 新媒体中心API说明

3.1 多码流播放器

3.1.1 多码流播放器说明

支持多码流同时播放（红外、可见光等），支持启动播放、暂停播放、恢复播放、终止播放。

3.1.2 多码流播放器API说明

使用AutelPlayerManager这个类的单实例及AutelPlayer、AutelPlayerView类的实例来调用接口。

AutelPlayerManager.java相关接口说明

```
/**
 * singleton class object of AutelPlayerManager
 */
public synchronized static AutelPlayerManager getInstance()

/**
```

```

    * initialises
    * @param cxt Context
    * @param bDroneCenter the boolean value
    */
    public void init(Context cxt,boolean bDroneCenter)

    /**
     * Registers stream data listener
     */
    public void registerStreamDataListener()

    /**
     * Unregisters stream data listeners
     */
    public void unregisterStreamDataListener()

    /**
     * starts streaming video
     * @param port stream channel Id
     */
    public void startStreamChannel(int port)

    /**
     * ends streaming video
     * @param port stream channel Id
     */
    public void endStreamChannel(int port)

    /**
     * Adds item to mAutelPlayerList
     *
     * @param player AutelPlayer instance to be added
     */
    public void addAutelPlayer(AutelPlayer player)

    /**
     * Releases/removes all players
     */
    public void release()

    /**
     * add mediacodec listeners
     * @param key
     * @param callbackWith
     */
    public void addCodecListeners(String key, final OnRenderFrameInfoListener
    callbackWith)

```

AutelPlayer.java相关接口说明

```

    /**
     * construction method
     * @param playerId the player id
     */
    public AutelPlayer(int playerId)

    /**

```

```

* adds view for video
* @param view the AutelPlayerView
*/
public void addVideoView(AutelPlayerView view)

/**
* removing the view for video
*/
public void removeVideoView()

/**
* sets listener for video stream listener
* @param ls the IVideoStreamListener
*/
public void setVideoInfoListener(IVideoStreamListener ls)

/**
* 暂停解码、显示灰色界面
*/
public void pauseVideo()

/**
* 恢复解码、显示彩色界面
*/
public void resumeVideo()

/**
* 释放各线程资源
*/
public void releasePlayer()

```

AutelPlayerView.java相关接口说明

```

/**
* Create a new AutelCodecView.
*
* @param context The context to associate this view with.
*/
public AutelPlayerView(Context context)

/**
* Create a new AutelCodecView.
*
* @param context The context to associate this view with.
* @param attrs The attributes of the XML tag that is inflating the view.
*/
public AutelPlayerView(Context context, AttributeSet attrs)

```

3.1.3 多码流播放器接口调用实例

```

//播放器管理单实例初始化
AutelPlayerManager.getInstance().init(ctx, false)
//注册码流接收监听
override fun onResume() {

```

```

        .....
        AutelPlayerManager.getInstance().registerStreamDataListener()
        .....
    }
    //开启接收码流的通道
    AutelPlayerManager.getInstance().startStreamChannel(getPlayerId())
    //创建播放TextureView
    val codecView = AutelPlayerView(activity)
    uiBinding.root.addView(codecView, 0)
    //创建播放器AutelPlayer
    mAutelPlayer = AutelPlayer(getPlayerId())
    //把播放TextureView传入播放器
    mAutelPlayer?.addVideoView(codecView)
    //设置码流信息回调接口
    mAutelPlayer?.setVideoInfoListener(mVideoStreamListener)
    //把创建的播放器添加到播放器列表
    AutelPlayerManager.getInstance().addAutelPlayer(mAutelPlayer)
    //启动播放器各线程资源
    mAutelPlayer?.startPlayer()

    //暂停播放
    override fun onPause() {
        .....
        mAutelPlayer?.pauseVideo()
        .....
    }
    //恢复播放
    override fun onResume() {
        .....
        mAutelPlayer?.resuemeVideo()
        .....
    }

    //把播放TextureView从播放器里移除
    mAutelPlayer?.removeVideoView()
    //关闭接收码流通道
    AutelPlayerManager.getInstance().endStreamChannel(getPlayerId())
    //释放播放器里各线程资源
    mAutelPlayer?.releasePlayer()
    //从播放器列表里移除播放器
    AutelPlayerManager.getInstance().removeAutelPlayer(mAutelPlayer)

    //取消码流接收监听，移除播放器列表里所有的播放器
    override fun onDestroy() {
        .....
        AutelPlayerManager.getInstance().unregisterStreamDataListener()
        AutelPlayerManager.getInstance().release()
        .....
    }
}

```

3.2 找飞机预录制

3.2.1 找飞机预录制说明

当开启了找飞机预录制，会每次录制大约30秒的内容，就会保存为一个固定名字文件并覆盖之前的保存的文件，这样总能保留飞机最近30秒的视频内容，如果飞机失联，则可以利用这个视频内容来协助找到飞机。

3.2.2 找飞机预录制API说明

找飞机预录制AutelPlayerManager这个类的单实例的API如下所示。

```
/**
 * 开启对应视频类型（可以是可见光、夜视、红外）的录制，最长约30秒,返回true：启动成功，false：
 * 启动失败（传的参数错误）
 * @param videoType: the video type
 * @param cacheDuration: use ms*/
public boolean openVideoCache(VideoType videoType, long cacheDuration)

/**
 * 关闭对应视频类型（可见光、夜视）的录制， 返回true：关闭成功，false:关闭失败（传的参数错
 * 误）
 * @param videoType the video type
 */
public boolean closeVideoCache(VideoType videoType)

/**
 * 设置保存录制视频的路径，如果不设置就默认保存在根目录
 * @param path the video path
 */
public void setCachePath(String path)

/**
 * 是否开启对应视频类型（可见光、夜视）的录制
 */
public boolean isOpenVideoCache(VideoType videoType)
```

3.2.3 找飞机预录接口调用示例代码

```
//开启30s时光回溯功能
AutelPlayerManager.getInstance().openVideoCache(VideoType.VISIBLE_LIGHT, 30_000)
//关闭30s时光回溯功能
AutelPlayerManager.getInstance().closeVideoCache(VideoType.VISIBLE_LIGHT)
//设置保存录制视频的路径，如果不设置就默认保存在根目录
AutelPlayerManager.getInstance().setCachePath(AutelDirPathUtils.getLookFlightVideoCachePath())
//判断是否开启对应视频类型30s时光回溯功能
AutelPlayerManager.getInstance().isOpenVideoCache(VideoType.VISIBLE_LIGHT)
```

3.3 获取码流的实时信息

3.3.1 获取码流实时信息说明

可以获取到码流的实时信息，方便调试定位问题。

3.3.2 获取码流实时信息API说明

AutelPlayer类的实例调用接口

```
//解码帧率
public int getVideoFps()
//接收帧率
```

```
public int getRecvFps()  
//刷新帧率  
public int getRenderFps()  
//接收的码率  
public int getRecvBitrate()  
//60秒内接收到的关键帧数量  
public int getKeyFrameNum()  
//60秒内请求关键帧次数  
public int getKeyFrameReqNum()  
//1秒内丢包数  
public int getPktLossNum()  
//视频的宽度  
public int getvideowidth()  
//视频的高度  
public int getvideoHeigh()
```

3.3.3 获取码流实时信息示例代码

略