

# MSS设计及二次开发手册

---

## 声明

由于产品版本升级或其他原因，本手册内容会 不定期进行更新。

本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。如有任何疑问或争议，请以我司最终解释为准。

本公司对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害不承担责任，包括但不限于商业利润损失、数据或文档丢失产生的损失，因遭受网络攻击、黑客攻击、病毒感染等造成的产品工作异常、信息泄露。

# 修改记录

对应版本	修订内容	修订人	日期
1.0.0	初次提交	路凯	2022-08-09

# 1. 内容简介

## 1.1 前言

本手册可能包含技术上不准确的地方或文字错误。

本手册的内容将做定期的更新，恕不另行通知；更新的内容将会在本手册的新版本中加入。

我们随时会改进或更新本手册中描述的产品或程序。

## 1.2 概述

MSS（媒体源服务）是我们以JBOARD抽象接口为基础，为IPC设计的提供基本媒体相关业务服务的应用。本应用为守护进程，启动之后即可提供相关服务。

服务接口依赖ES通信系统，提供libjes\_mss.so 和 jes\_mss.h作为二次开发的sdk。

提供服务如下：直播流拉取，直播流参数配置，图像参数配置，OSD叠加，原始视频数据流拉取。

## 1.3 阅读对象

本文档的阅读对象为在JES-IPC上进行其他业务应用开发的人员，如存储应用、协议应用、算法应用等。

## 1.4 名词解释

名词	解释
MSS	media source service 媒体源服务的缩写
Stream	直播流，包括音视频帧，本服务提供多种分辨率的直播流服务
MIC	麦克风缩写，音频输入参数配置接口中用到的字段
VPS	video process system 视频处理系统缩写，负责对原始视频数据的缩放、旋转、叠加等处理
VENC	video encode 或video encoder 视频编码或视频编码器缩写
AENC	audio encode 或audio encoder 音频编码或音频编码器缩写
ADEC	audio decode 或audio decoder 音频解码或音频解码器缩写
ROI	感兴趣区域，目前仅支持感兴趣区域编码
RAW	指的是编码前的原始视频数据，如YUV或RGB格式的视频数据，主要用于算法分析

## 1.5 系统要求

目前主要运行在各个IPC芯片上，目前仅支持 linux 操作系统。

# 2. 编程引导

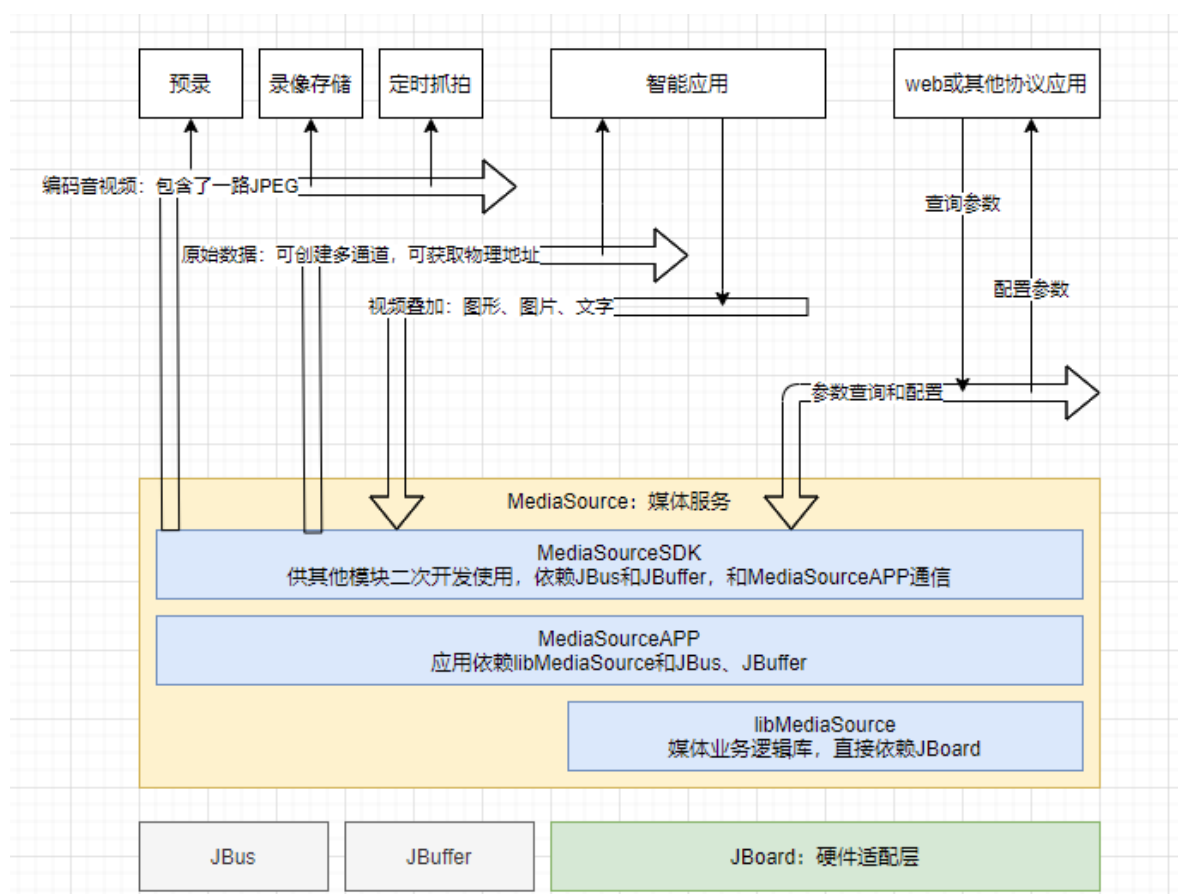
## 2.1 编程引导

- 运行mss服务
- 使用jes\_mss.h 和 jes\_mss.so 开发应用

## 2.1.1 目录结构

└─ build	独立编译目录，单独调试时使用，放到buildroot中统一编译时无用
└─ doc	文档目录
└─ lib	依赖库以及头文件：如jboard、easyrtsp等
└─ release	独立安装目录，单独调试时使用，放到buildroot中统一编译时无用
└─ hi3516dv300	hi3516dv300平台的安装文件
└─ bin	存放mss服务程序
└─ sdk	存放jes_mss.so/jes_mss.a
└─ include	
└─ jes	存放jes_mss.h
└─ src	源码目录
└─ app	mss应用程序源码，依赖media_source和jbus、jbuffer实现服务应用
└─ lib	media_source业务代码，依赖jboard实现具体业务逻辑
└─ sdk	jes_mss二次开发sdk源码和测试demo的源码

## 2.1.2 MSS模块图

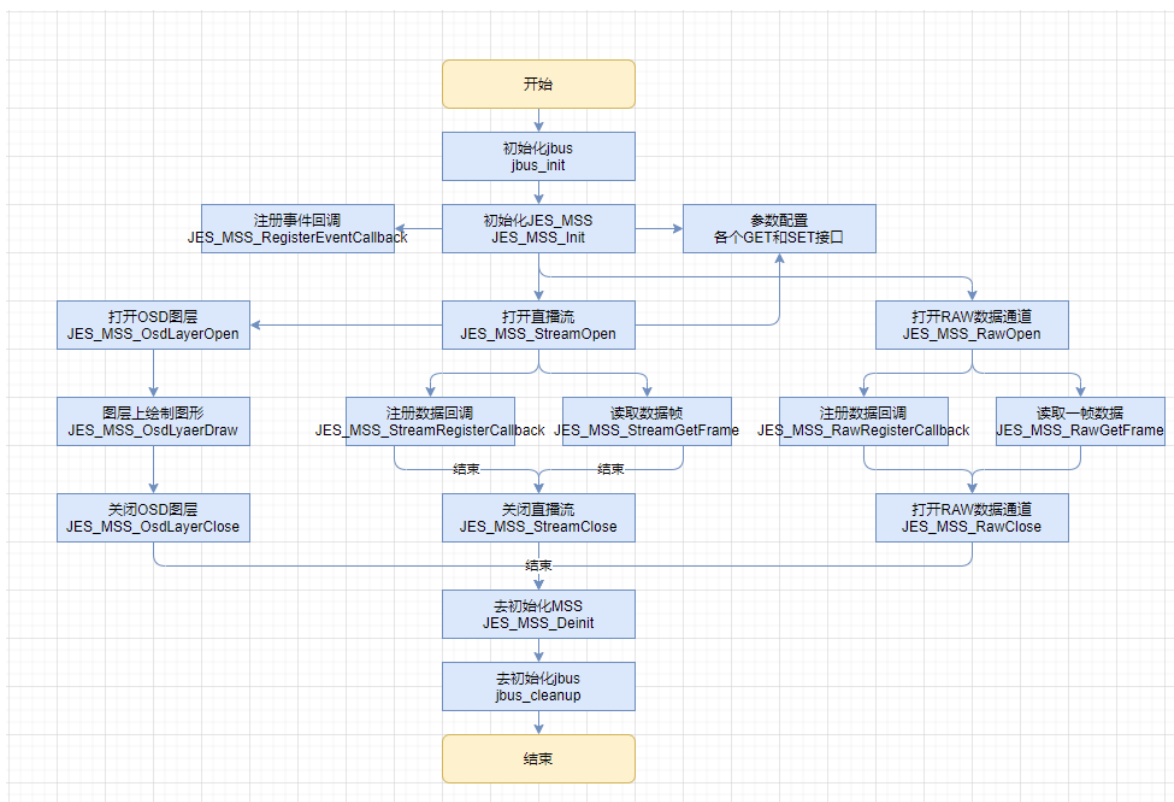


## 2.1.3 SDK开发指南

- 环境准备：烧有文件系统的硬件，内部有jbus、jbuffer运行环境，且相关服务dbus、mss都在正常运行。
- jes\_mss.h头文件和库文件
- jes\_mss\_test.cpp demo源码
- 可以先编译运行一下源码确保demo可以正常运行使用vlc可以拉取到码流
- 参考demo源码和开发手册，进行新应用的开发。

## 2.2 JES\_MSS接口调用流程

### 2.2.1 流程图



## 2.2.2 示例代码

参见demo, jes\_mss\_test.cpp

# 3. 接口定义

## 3.1 初始化

### 3.1.1 MSS初始化

接口声明:

```
jms_hdl_t JES_MSS_Init(jbus_hdl_t jbus);
```

接口描述: MSS初始化

接口参数:

参数名称	参数类型	参数说明
jbus	in	jbus_init 的返回值

返回值: 失败返回 NULL, 成功返回 句柄

专用错误码: 无

Remarks:

MSS为Media Source Service的缩写

jbus\_init 位于 jbus.h 中, 每个进程只可以调用一次 jbus\_init

See Also: 无

### 3.1.2 去初始化

接口声明：

```
void JES_MSS_Deinit(jms_hdl_t hdl);
```

接口描述： 去初始化

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值

返回值： 无

专用错误码： 无

Remarks： 无

See Also： 无

3.1.3 配置事件回调

接口声明：

```
int JES_MSS_RegisterEventCallback(jms_hdl_t hdl, JESMSSEventCB cb, void *arg);
```

接口描述： 配置事件回调

接口参数：

参数名称	参数类型	参数说明
cb	in	事件通知回调
arg	in	用户参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks：

hdl JES\_MSS\_Init 返回值

See Also： 无

3.2 直播业务

3.2.1 打开直播流

接口声明：

```
jms_stm_ctx_t *JES_MSS_StreamOpen(jms_hdl_t hdl, int stream);
```

接口描述： 打开直播流

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
stream	in	流id从0开始

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.2.2 关闭直播流

接口声明：

```
void JES_MSS_StreamClose(jms_stm_ctx_t *ctx);
```

接口描述： 关闭直播流

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值

返回值： 无

专用错误码：无

Remarks：无

See Also：无

### 3.2.3 插入I帧

接口声明：

```
int JES_MSS_StreamInsertIDR(jms_stm_ctx_t *ctx);
```

接口描述： 插入I帧

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.2.4 读取一帧数据



接口声明:

```
int JES_MSS_StreamGetFrame(jms_stm_ctx_t *ctx, JVMediaFrame_t *frame);
```

接口描述: 读取一帧数据

接口参数:

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值
frame	out	帧数据, 其中的buffer无需释放, 其指向的内容缓存满了后会被覆盖

返回值: <0 失败, ==0 成功

专用错误码: 无

Remarks:

此接口和 JES\_MSS\_StreamRegisterCallback 只能使用一种方式, 不支持同时使用

See Also: 无

3.2.5 读取最新一帧数据

接口声明:

```
int JES_MSS_StreamGetLastFrame(jms_stm_ctx_t *ctx, JVMediaFrame_t *frame);
```

接口描述: 读取最新一帧数据

接口参数:

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值
frame	out	帧数据, 其中的buffer无需释放, 其指向的内容缓存满了后会被覆盖

返回值: <0 失败, ==0 成功

专用错误码: 无

Remarks:

本接口和 JES\_MSS\_StreamGetFrame 使用方式一致, 只是本接口每次都获取最新一帧数据, 而不是顺序读取  
可用于从 MJPEG 流中抓拍, 每次都获取最新的图片

See Also: 无

3.2.6 注册数据回调

接口声明:

```
int JES_MSS_StreamRegisterCallback(jms_stm_ctx_t *ctx, JESMSSDataCB cb, void *arg);
```

接口描述： 注册数据回调

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值
cb	in	数据回调
arg	in	用户参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks：

此接口和 JES\_MSS\_StreamGetFrame 只能使用一种方式，不支持同时使用

See Also： 无

### 3.2.7 获取直播流视频编码参数

接口声明：

```
int JES_MSS_StreamGetParam(jms_stm_ctx_t *ctx, JMediaVencParam_t *param, JBOOL bDefault);
```

接口描述： 获取直播流视频编码参数

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值
param	out	输出参数
bDefault	in	是否获取默认参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks： 无

See Also： 无

### 3.2.8 配置直播流视频编码参数

接口声明：

```
int JES_MSS_StreamSetParam(jms_stm_ctx_t *ctx, JMediaVencParam_t *param);
```

接口描述： 配置直播流视频编码参数

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_StreamOpen 返回值
param	in	编码参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.3 通用参数配置

#### 3.3.1 获取音频输入参数

接口声明：

```
int JES_MSS_GetMICParam(jms_hdl_t hdl, JMediaAencParam_t *param, JBOOL
bDefault);
```

接口描述：获取音频输入参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	out	输出参数
bDefault	in	是否获取默认参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

#### 3.3.2 配置音频输入参数

接口声明：

```
int JES_MSS_SetMICParam(jms_hdl_t hdl, const JMediaAencParam_t *param);
```

接口描述：配置音频输入参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	in	编码参数

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks： 无

See Also： 无

### 3.3.3 获取ROI参数

接口声明：

```
int JES_MSS_GetRoiParam(jms_hdl_t hdl, JMediaRoi_t *param, JBOOL bDefault);
```

接口描述： 获取ROI参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	out	输出参数
bDefault	in	是否获取默认参数

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks：

ROI 感兴趣区域

See Also： 无

### 3.3.4 配置ROI参数

接口声明：

```
int JES_MSS_SetRoiParam(jms_hdl_t hdl, JMediaRoi_t *param);
```

接口描述： 配置ROI参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	in	配置参数

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks： 无

See Also： 无

### 3.3.5 获取视频处理参数

接口声明：

```
int JES_MSS_GetVpsParam(jms_hdl_t hdl, JMediaVpsParam_t *param, JBOOL bDefault);
```

接口描述： 获取视频处理参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	out	输出视频处理参数
bDefault	in	是否获取默认参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks：

镜像、翻转、旋转， 参数对所有码流生效

See Also： 无

### 3.3.6 配置视频处理参数

接口声明：

```
int JES_MSS_SetVpsParam(jms_hdl_t hdl, JMediaVpsParam_t *param);
```

接口描述： 配置视频处理参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	in	视频处理参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks：

镜像、翻转、旋转， 参数对所有码流生效

See Also： 无

### 3.3.7 获取OSD参数

接口声明：

```
int JES_MSS_GetOSDParam(jms_hdl_t hdl, JMediaOsdParam_t *param, JBOOL bDefault);
```

接口描述： 获取OSD参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	out	输出参数
bDefault	in	是否获取默认参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks： 无

See Also： 无

### 3.3.8 配置OSD参数

接口声明：

```
int JES_MSS_SetOSDParam(jms_hdl_t hdl, const JMediaOsdParam_t *param);
```

接口描述： 配置OSD参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
param	in	配置参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks： 无

See Also： 无

### 3.3.9 配置OSD闪烁

接口声明：

```
int JES_MSS_SetOSDTwinkle(jms_hdl_t hdl, JBOOL enable);
```

接口描述： 配置OSD闪烁

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
enable	in	是否闪烁

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：

时间和设备名称OSD闪烁，主要用于提示报警

See Also：无

### 3.3.10 获取图像库版本号

接口声明：

```
int JES_MSS_GetISPVersion(jms_hdl_t hdl, char *version, int size);
```

接口描述：获取图像库版本号

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
version	out	输出参数，使用64字节即可
size	in	version大小

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.3.11 获取图像基础参数

接口声明：

```
int JES_MSS_GetISPAttr(jms_hdl_t hdl, JMediaISPAttr_t *attr, JBOOL bDefault);
```

接口描述：获取图像基础参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	out	输出参数
bDefault	in	是否获取默认参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks： 无

See Also： 无

### 3.3.12 配置图像基础参数

接口声明：

```
int JES_MSS_SetISPAAttr(jms_hdl_t hdl, const JMediaISPAAttr_t *attr);
```

接口描述： 配置图像基础参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	in	配置参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks： 无

See Also： 无

### 3.3.13 获取图像曝光参数

接口声明：

```
int JES_MSS_GetISPAEAttr(jms_hdl_t hdl, JMediaISPAE_t *attr, JBOOL bDefault);
```

接口描述： 获取图像曝光参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	out	输出参数
bDefault	in	是否获取默认参数

返回值： <0 失败, ==0 成功

专用错误码： 无

Remarks： 无

See Also： 无

### 3.3.14 配置图像曝光参数



接口声明:

```
int JES_MSS_SetISPAEAttr(jms_hdl_t hdl, const JMediaISPAE_t *attr);
```

接口描述: 配置图像曝光参数

接口参数:

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	in	配置参数

返回值: <0 失败, ==0 成功

专用错误码: 无

Remarks: 无

See Also: 无

3.3.15 获取图像宽动态参数

接口声明:

```
int JES_MSS_GetISPWdrAttr(jms_hdl_t hdl, JMediaISPWdr_t *attr, JB00L bDefault);
```

接口描述: 获取图像宽动态参数

接口参数:

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	out	输出参数
bDefault	in	是否获取默认参数

返回值: <0 失败, ==0 成功

专用错误码: 无

Remarks: 无

See Also: 无

3.3.16 配置图像宽动态参数

接口声明:

```
int JES_MSS_SetISPWdrAttr(jms_hdl_t hdl, const JMediaISPWdr_t *attr);
```

接口描述: 配置图像宽动态参数

接口参数:

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	in	配置参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.3.17 获取图像夜视切换参数

接口声明：

```
int JES_MSS_GetISPDaYNightAttr(jms_hdl_t hdl, JMediaISPDaYNight_t *attr, JBOOL bDefault);
```

接口描述：获取图像夜视切换参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	out	输出参数
bDefault	in	是否获取默认参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.3.18 配置图像夜视切换参数

接口声明：

```
int JES_MSS_SetISPDaYNightAttr(jms_hdl_t hdl, const JMediaISPDaYNight_t *attr);
```

接口描述：配置图像夜视切换参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	in	配置参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.3.19 获取图像高级参数

接口声明：

```
int JES_MSS_GetISPAdvanceAttr(jms_hdl_t hdl, JMediaISPAdvancedAttr_t *attr,
JBOOL bDefault);
```

接口描述：获取图像高级参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	out	输出参数
bDefault	in	是否获取默认参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

### 3.3.20 配置图像高级参数

接口声明：

```
int JES_MSS_SetISPAdvanceAttr(jms_hdl_t hdl, const JMediaISPAdvancedAttr_t
*attr);
```

接口描述：配置图像高级参数

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
attr	in	配置参数

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：无

See Also：无

## 3.4 OSD叠加业务

### 3.4.1 打开OSD图层

接口声明:

```
jms_osd_ctx_t *JES_MSS_OsdLayerOpen(jms_hdl_t hdl);
```

接口描述: 打开OSD图层

接口参数:

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值

返回值: 成功返回图层句柄, 失败返回NULL

专用错误码: 无

Remarks:

每个应用尽量创建一个图层使用, 图层总数有限

See Also: 无

### 3.4.2 关闭OSD图层

接口声明:

```
void JES_MSS_OsdLayerClose(jms_osd_ctx_t *ctx);
```

接口描述: 关闭OSD图层

接口参数:

参数名称	参数类型	参数说明
ctx	in	JES_MSS_OsdLayerOpen 返回值

返回值: 无

专用错误码: 无

Remarks:

不使用的图层必须主动关闭, 否则该图层内容会一直存在

See Also: 无

### 3.4.3 绘制OSD图层

接口声明:

```
void JES_MSS_OsdLayerDraw(jms_osd_ctx_t *ctx, const JMediaOsdRegions_t *region);
```

接口描述: 绘制OSD图层

接口参数:

参数名称	参数类型	参数说明
ctx	in	JES_MSS_OsdLayerOpen 返回值
region	in	绘制的区域内容

返回值：无

专用错误码：无

Remarks：无

See Also：无

### 3.5 原始数据业务

#### 3.5.1 打开原始数据通道

接口声明：

```
jms_raw_ctx_t *JES_MSS_RawOpen(jms_hdl_t hdl, int width, int height,
JMediaRawFrameType_e type);
```

接口描述：打开原始数据通道

接口参数：

参数名称	参数类型	参数说明
hdl	in	JES_MSS_Init 返回值
width	in	分辨率宽
height	in	分辨率高
type	in	数据格式类型：不同芯片可能支持不同请根据具体芯片确定，不支持时返回失败

返回值：成功返回图层句柄，失败返回NULL

专用错误码：无

Remarks：无

See Also：无

#### 3.5.2 关闭原始数据通道

接口声明：

```
void JES_MSS_RawClose(jms_raw_ctx_t *ctx);
```

接口描述：关闭原始数据通道

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_RawOpen 返回值

返回值：无

专用错误码：无

Remarks：

不使用的数据通道需要主动关闭

See Also：无

### 3.5.3 获取一帧原始数据

接口声明：

```
int JES_MSS_RawGetFrame(jms_raw_ctx_t *ctx, JVMediaFrame_t *frame);
```

接口描述：获取一帧原始数据

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_RawOpen 返回值
frame	out	原始数据帧

返回值：<0 失败，==0 成功

专用错误码：无

Remarks：

本接口和 JES\_MSS\_RawRegisterCallback 选择一种方式使用，不支持同时使用  
本接口帧数据内容位于OS内存中，注意和 JES\_MSS\_RawGetPhyAddr 的差异

See Also：无

### 3.5.4 注册原始数据回调

接口声明：

```
int JES_MSS_RawRegisterCallback(jms_raw_ctx_t *ctx, JESMSSDataCB cb, void *arg);
```

接口描述：注册原始数据回调

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_RawOpen 返回值
cb	in	回调
arg	in	用户参数

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks:

本接口和 JES\_MSS\_RawGetFrame 选择一种方式使用，不支持同时使用  
本接口帧数据内容位于OS内存中，注意和 JES\_MSS\_RawGetPhyAddr 的差异

See Also：无

### 3.5.5 获取原始数据帧物理地址

接口声明：

```
int JES_MSS_RawGetPhyAddr(jms_raw_ctx_t *ctx, JVMediaRawFrame_t *frame);
```

接口描述： 获取原始数据帧物理地址

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_RawOpen 返回值
frame	out	帧数据

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks:

使用完毕需要调用 JES\_MSS\_RawFreePhyAddr 接口进行释放  
服务端会缓存10帧数据，缓存满之后会主动释放，应用端使用速度需要快于此速度

See Also：无

### 3.5.6 释放原始数据帧物理地址

接口声明：

```
int JES_MSS_RawFreePhyAddr(jms_raw_ctx_t *ctx, const JVMediaRawFrame_t *frame);
```

接口描述： 释放原始数据帧物理地址

接口参数：

参数名称	参数类型	参数说明
ctx	in	JES_MSS_RawOpen 返回值
frame	in	帧数据

返回值： <0 失败, ==0 成功

专用错误码：无

Remarks：无

See Also: 无

## 4. 数据定义

### 4.1 结构体定义

#### 4.1.1 原始数据帧结构

```
typedef struct{
    void *virAddr;
    void *phyAddr;
    JU32 width;
    JU32 height;
    JU32 len;
    JU64 pts;
    JU64 reserve[10];
    void *handle;
}JVMediaRawFrame_t;
```

参数	说明
*virAddr	虚拟地址
*phyAddr	物理地址，0表示不支持
width	分辨率宽
height	分辨率高
len	虚拟地址数据长度
pts	时间戳
reserve[10]	预留，海思平台reserve[0]存储PoolID
*handle	用于存放底层需要释放的句柄调用者无需关心

#### 4.1.2 媒体解码参数

```
typedef struct{
    int stream;
    JVMediaVDecAttr_t vdec;
    JVMediaAdecAttr_t adec;
}JVMediaDecodeAttr_t;
```

参数	说明
stream	流id
vdec	视频解码参数
adec	音频解码参数

#### 4.1.3 音频编解码参数



```
typedef struct
{
    JVMediaAudioType_e encType;
    int sampleRate;
    int bitwidth;
    int bitrate;
    int channelnum;
    int volume;
}JVMediaAencAttr_t;
```

参数	说明
encType	编码类型
sampleRate	采样率，取值：8000,11025,16000,22050,24000,32000,44100,48000
bitWidth	位宽，取值：8,16,32
bitrate	码率，AAC使用
channelnum	声道数目，1单声道，2双声道
volume	音量0-100范围

#### 4.1.4 音频参数

```
typedef struct{
    JBOOL enable;
    JVMediaAencAttr_t attr;
}JVMediaAencParam_t;
```

参数	说明
enable	开关
attr	编解码参数

#### 4.1.5 视频编码参数

```
typedef struct{
    JVMediaVideotype_e vencType;
    JVMediaRcmode_e rcMode;
    JU32 width;
    JU32 height;
    JU32 framerate;
    JU32 gop;
    JU32 bitrate;
    int quality;
    JBOOL stretchEnable;
    JBOOL smartencEnable;
}JVMediaVencAttr_t;
```

参数	说明
vencType	视频编码协议类型
rcMode	码控模式
width	分辨率宽
height	分辨率高
framerate	帧率
gop	I帧间隔, 帧数
bitrate	码率kbps
quality	图像质量, 1-5, 5个等级: 超清、高清、标清、流畅、极速
stretchEnable	对于16: 9与4: 3之间的差异, 是拉伸, 还是裁剪。为真时表示拉伸
smartencEnable	是否启动智能编码

#### 4.1.6 视频参数

```
typedef struct{
    JBOOL enable;
    JMediaVencAttr_t attr;
}JMediaVencParam_t;
```

参数	说明
enable	开关 (目前强制开启关闭不生效)
attr	视频编码参数

#### 4.1.7 感兴趣区域参数

```
typedef struct{
    int reverse;
    int roiCnt;
    JRect_t roi[4];
    int weight;
} JMediaRoi_t;
```

参数	说明
reverse	是否反选
roiCnt	ROI数量
roi[4]	ROI坐标
weight	权重0-10

#### 4.1.8 视频处理参数

```
typedef struct
{
    JBOOL mirror;
    JBOOL flip;
    JMediaRotate_e rotate;
}JMediaVpsParam_t;
```

参数	说明
mirror	镜像
flip	翻转
rotate	旋转，get时JMEDIA_ROTATE_MAX表示不支持

#### 4.1.9 OSD文字信息

```
typedef struct
{
    char text[1024];
    int fontsize;
    JU32 color;
} JMediaOsdText_t;
```

参数	说明
text[1024]	文字，表示时间时为时间格式
fontsize	字体大小
color	统一使用ARGB8888，内部自行转化为自己支持的

#### 4.1.10 OSD属性信息

```
typedef struct{
    JBOOL enable;
    JMediaOsdText_t text;
    JMediaAreaPos_e pos;
    JPoint_t absPos;
}JMediaOsdAttr_t;
```

参数	说明
enable	是否显示
text	文字
pos	OSD_POS_USER时，xy有效表示自定义位置
absPos	绝对位置

#### 4.1.11 遮挡属性信息

```
typedef struct
{
    JBOOL enable;
    JU32 color;
    JRect_t rect;
} JMediaCover_t;
```

参数	说明
enable	使能
color	统一使用ARGB8888，内部自行转化为自己支持的
rect	遮挡位置

#### 4.1.12 预置OSD参数

```
typedef struct
{
    JMediaOsdAttr_t time;
    JMediaOsdAttr_t name;
    JMediaOsdAttr_t custom[JMEDIA_CUSTOM_OSD_MAX_CNT];
    JMediaCover_t cover[JMEDIA_CUSTOM_OSD_MAX_CNT];
    JBOOL largeOSD;
} JMediaOsdParam_t;
```

参数	说明
time	时间osd，text请配置Format：时间格式如：YYYY-MM-DD hh:mm:ss，DD/MM/YYYY hh:mm:ss
name	设备名称osd
custom[JMEDIA_CUSTOM_OSD_MAX_CNT]	用户自定义多行osd
cover[JMEDIA_CUSTOM_OSD_MAX_CNT]	遮挡
largeOSD	是否用超大OSD

#### 4.1.13 多边形信息

```
typedef struct
{
    int count;
    JPoint_t points[JMEDIA_OSD_REGION_MAX_POINT_NUM];
} JMediaPolygon_t;
```

参数	说明
count	顶点的个数>=2,=2时是绊线
points[JMEDIA_OSD_REGION_MAX_POINT_NUM]	顶点坐标

point[JMEDIA_OSD_REGION_MAX_NUM]	区域坐标
----------------------------------	------

4.1.14 OSD区域内容

```
typedef struct
{
    JMediaOsdRgnType_e type;
    JU32 color;
    union
    {
        JRect_t rect;
        JMediaPolygon_t polygon;
        JMediaOsdAttr_t text;
    };
} JMediaOsdRgn_t;
```

参数	说明
type	区域类型
color	边框颜色，统一使用ARGB8888，内部自行转化为自己支持的
rect	矩形坐标
polygon	多边形坐标
text	文字坐标以及内容：此处仅支持绝对位置

4.1.15 OSD区域参数

```
typedef struct
{
    JU64 pts;
    int count;
    JMediaOsdRgn_t region[JMEDIA_OSD_REGION_MAX_NUM];
} JMediaOsdRegions_t;
```

参数	说明
pts	时间戳，根据它找到区域要绘制的帧，用于帧同步，不关心时可以为0
count	区域数量，最大 JMEDIA_OSD_REGION_MAX_NUM
region[JMEDIA_OSD_REGION_MAX_NUM]	区域内容

4.1.16 图像基础参数

```
typedef struct
{
    JU32 brightness;
    JU32 saturation;
    JU32 contrast;
    JU32 sharpness;
    JMediaStyle_e style;
}JMediaISPAAttr_t;
```

参数	说明
brightness	亮度
saturation	饱和度
contrast	对比度
sharpness	锐度
style	风格

#### 4.1.17 图像曝光参数

```
typedef struct{
    JMediaAEMode_e mode;
    JMediaPowerFreqType_e PFTYPE;
    JU32 exposureMax;
    JU32 exposureMin;
}JMediaISPAE_t;
```

参数	说明
mode	曝光模式
PFTYPE	工频，解决工频干扰问题
exposureMax	最大曝光时间。曝光时间为： 1/exposureMax 秒，取值 3 - 100000
exposureMin	最小曝光时间。曝光时间为： 1/exposureMin 秒，取值 3 - 100000

#### 4.1.18 图像宽动态参数

```
typedef struct
{
    JMediawdrType_e type;
    JMediawdrMode_e mode;
    int strength;
}JMediaISPWdr_t;
```

参数	说明
type	宽动态类型
mode	切换模式，硬件宽动态时有效
strength	wdr强度，硬件宽动态时有效

#### 4.1.19 图像宽动态参数

```
typedef struct
{
    JMediaWdrType_e type;
    JMediaWdrMode_e mode;
    int strength;
    JTime_t openTime;
    JTime_t closeTime;
}JMediaISPWdrParam_t;
```

参数	说明
type	宽动态类型
mode	切换模式，硬件宽动态时有效
strength	wdr强度，硬件宽动态时有效
openTime	开启时间，硬件宽动态定时模式时有效
closeTime	关闭时间，硬件宽动态定时模式时生效

#### 4.1.20 图像夜视切换参数

```
typedef struct{
    JMediaDayNightMode_e mode;
    int sensitivity;
    JMediaFillLightMode_e lightMode;
}JMediaISPDayNight_t;
```

参数	说明
mode	夜视切换模式
sensitivity	灵敏度，1-10，自动模式生效，值越大进入夜视越早离开夜视越晚
lightMode	补光模式，红外补光，白光灯补光，不补光

#### 4.1.21 图像夜视切换参数

```
typedef struct{
    JMediaDayNightMode_e mode;
    int sensitivity;
    JMediaFillLightMode_e lightMode;
    JTime_t dayStart;
    JTime_t dayEnd;
}JMediaISPDayNightParam_t;
```

参数	说明
mode	日夜切换模式
sensitivity	灵敏度，1-10，自动模式生效，值越大进入夜视越早离开夜视越晚
lightMode	补光模式，红外补光，白光灯补光，不补光
dayStart	进入白天模式时间，定时模式生效
dayEnd	进入夜间模式时间，定时模式生效

#### 4.1.22 图像高级参数

```
typedef struct{
    JBOOL nightReduceFrame;
    JBOOL defog;
    JBOOL faceMode;
    JBOOL vehicleMode;
}JMediaISPAdvancedAttr_t;
```

参数	说明
nightReduceFrame	夜视降帧
defog	去雾
faceMode	人脸模式，特定场景下看清人脸
vehicleMode	车辆模式，特定场景下看清车牌

### 4.2 枚举定义

#### 4.2.1 OSD类型

```
typedef enum{
    JMEDIA_OSD_TYPE_TEXT,      //OSD类型文字
    JMEDIA_OSD_TYPE_COVER,     //OSD类型遮挡
    JMEDIA_OSD_TYPE_REGION,    //OSD类型区域
    JMEDIA_OSD_TYPE_MAX
}JMediaOsdType_e; //OSD类型
```

#### 4.2.2 区域类型



```
typedef enum{
    JMEDIA_OSD_RGN_TYPE_COVER,          //OSD区域类型：矩形遮挡
    JMEDIA_OSD_RGN_TYPE_RECT,           //OSD区域类型：矩形
    JMEDIA_OSD_RGN_TYPE_MINIRECT,       //OSD区域类型：矩形只绘制四个角
    JMEDIA_OSD_RGN_TYPE_POLYGON,        //OSD区域类型：多边形
    JMEDIA_OSD_RGN_TYPE_TEXT,           //OSD区域类型：文字
    JMEDIA_OSD_RGN_TYPE_MAX
}JMediaOsdRgnType_e; //区域类型
```

### 4.2.3 旋转

```
typedef enum
{
    JMEDIA_ROTATE_NONE = 0,           //不旋转
    JMEDIA_ROTATE_90 = 1,             //向右旋转90度
    JMEDIA_ROTATE_180 = 2,            //向右旋转180度
    JMEDIA_ROTATE_270 = 3,            //向右旋转270度
    JMEDIA_ROTATE_MAX
} JMediaRotate_e; //旋转
```

### 4.2.4 图形风格

```
typedef enum{
    JMEDIA_STYLE_NORMAL,              //标准
    JMEDIA_STYLE_BRIGHT,              //通透
    JMEDIA_STYLE_SOFT,                //柔和
    JMEDIA_STYLE_MAX
} JMediaStyle_e; //图形风格
```

### 4.2.5 曝光模式

```
typedef enum
{
    JMEDIA_AEMODE_AUTO,               //自动曝光
    JMEDIA_AEMODE_HIGH_LIGHT,         //强光抑制
    JMEDIA_AEMODE_LOW_LIGHT,          //背光补偿
    JMEDIA_AEMODE_MAX
} JMediaAEMode_e; //曝光模式
```

### 4.2.6 工频类型

```
typedef enum
{
    JMEDIA_POWER_FREQ_TYPE_CLOSE,     //关闭
    JMEDIA_POWER_FREQ_TYPE_50HZ,      //50Hz
    JMEDIA_POWER_FREQ_TYPE_60HZ,      //60Hz
    JMEDIA_POWER_FREQ_TYPE_MAX
}JMediaPowerFreqType_e; //工频类型
```

### 4.2.7 宽动态类型

```
typedef enum
{
    JMEDIA_WDR_TYPE_CLOSE,    //关闭
    JMEDIA_WDR_TYPE_DIGITAL,  //数字宽动态
    JMEDIA_WDR_TYPE_HARD,     //硬件宽动态
    JMEDIA_WDR_TYPE_MAX,
} JMediaWdrType_e; //宽动态类型
```

#### 4.2.8 宽动态模式

```
typedef enum
{
    JMEDIA_WDR_MODE_AUTO,    //自动模式
    JMEDIA_WDR_MODE_OPEN,    //一直开启
    JMEDIA_WDR_MODE_CLOSE,   //一直关闭
    JMEDIA_WDR_MODE_TIME,    //定时模式
    JMEDIA_WDR_MODE_MAX,
} JMediaWdrMode_e; //宽动态模式
```

#### 4.2.9 夜视模式

```
typedef enum
{
    JMEDIA_DAYNIGHT_MODE_AUTO,    //自动模式
    JMEDIA_DAYNIGHT_MODE_ALWAYS_DAY,    //一直白天
    JMEDIA_DAYNIGHT_MODE_ALWAYS_NIGHT,  //一直夜间
    JMEDIA_DAYNIGHT_MODE_TIME_DAY,      //定时白天
    JMEDIA_DAYNIGHT_MODE_MAX,
} JMediaDayNightMode_e; //夜视模式
```

#### 4.2.10 补光模式

```
typedef enum
{
    JMEDIA_FILLLIGHT_MODE_AUTO_IR,    //自动红外
    JMEDIA_FILLLIGHT_MODE_AUTO_WHITE, //自动白光
    JMEDIA_FILLLIGHT_MODE_ALWAYS_CLOSE, //常闭
    JMEDIA_FILLLIGHT_MODE_MAX,
} JMediaFillLightMode_e; //补光模式
```

#### 4.2.11 原始数据类型

```
typedef enum
{
    JMEDIA_RAWFRAMETYPE_YV12,    //YUV格式YV12
    JMEDIA_RAWFRAMETYPE_YU12,    //YUV格式YU12
    JMEDIA_RAWFRAMETYPE_NV12,    //YUV格式NV12
    JMEDIA_RAWFRAMETYPE_NV21,    //YUV格式NV21
    JMEDIA_RAWFRAMETYPE_RGB888,  //RGB888
    JMEDIA_RAWFRAMETYPE_MAX,
} JMediaRawFrameType_e; //原始数据类型
```

#### 4.2.12 区域位置类型

```
typedef enum{
    JMEDIA_AREA_POS_LEFT_TOP,
    JMEDIA_AREA_POS_LEFT_BOTTOM,
    JMEDIA_AREA_POS_RIGHT_TOP,
    JMEDIA_AREA_POS_RIGHT_BOTTOM,
    JMEDIA_AREA_POS_CENTER,
    JMEDIA_AREA_POS_CUSTOM,
    JMEDIA_AREA_POS_MAX
}JMediaAreaPos_e; //区域位置类型
```

## 5. 帮助

---

### 5.1 环境准备

- 详见 2.1.3 章节

### 5.2 编译运行demo

- 详见 2.1.3 章节

### 5.3 FAQ