

# Rockchip RkDisplayOutputManager 开发指南

文件标识: RK-KF-YF-583

发布版本: V1.0.0

日期: 2023-02-28

文件密级: 绝密 秘密 内部资料 公开

## 免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司 (“本公司”, 下同) 不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

## 版权所有 © 2023 瑞芯微电子股份有限公司

超越合理使用范畴, 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。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: [www.rock-chips.com](http://www.rock-chips.com)

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## 前言

### 概述

RkDisplayOutputManager 是RK提供给应用层获取外接屏幕信息、切换分辨率、切换色彩空间等操作的系统服务, 提供了应用层和HWC、DRM显示框架之间的交互。

本文对该接口的API进行了相关说明和举例。

### 产品版本

芯片名称	内核版本
RK所有平台	Linux 4.19/Linux 5.10

## 读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

## 修订记录

版本号	作者	修改日期	修改说明
V1.0.0	Tao.Weng	2023-02-28	初始版本

## 目录

### Rockchip RkDisplayOutputManager 开发指南

说明

API说明

- RkDisplayOutputManager
- getCurrentDpyConnState
- getDisplayNumber
- getCurrentInterface
- getConnectorInfo
- updateDisplayInfos
- getModeList
- setMode
- getCurrentMode
- getSupportColorList
- setColorMode
- getCurrentColorMode
- getOverScan
- setOverScan
- getBcsh
- getBrightness
- setBrightness
- getContrast
- setContrast
- getSaturation
- setSaturation
- getHue
- setHue
- saveConfig
- getResolutionSupported
- isAiImageQuality
- setAiImageQuality
- isAiImageQualityLabMode
- setAiImageQualityLabMode

## 说明

RkDisplayOutputManager 是RK提供给应用层获取外接屏幕信息、切换分辨率、切换色彩空间等操作的系统服务，提供了应用层和HWC、DRM显示框架之间的交互。

详细使用代码可参考

packages/apps/TvSettings/Settings/src/com/android/tv/settings/display/DrmDisplaySetting.java。

## API说明

该系统服务提供了以下API。

- **RkDisplayOutputManager**: 初始化系统服务的Client端对象。
- **getCurrentDpyConnState**: 获取显示设备的连接状态。
- **getDisplayNumber**: 获取显示设备数量。
- **getCurrentInterface**: 获取显示设备的设备类型。
- **getConnectorInfo**: 获取所有显示设备的连接信息。
- **updateDisplayInfos**: 更新显示设备的状态。
- **getModeList**: 获取显示设备支持的分辨率列表。
- **setMode**: 设置显示设备的分辨率。
- **getCurrentMode**: 获取指定显示设备和接口的当前分辨率。
- **getSupportColorList**: 获取显示设备支持的色彩空间。
- **setColorMode**: 设置显示设备的色彩空间。
- **getCurrentColorMode**: 获取显示设备的当前色彩空间。
- **getOverScan**: 获取显示设备的缩放比例。
- **setOverScan**: 设置显示设备的缩放比例。
- **getBcsh**: 获取显示设备的BCSH值。
- **getBrightness**: 获取显示设备的亮度。
- **setBrightness**: 设置显示设备的亮度。
- **getContrast**: 获取显示设备的对比度。
- **setContrast**: 设置显示设备的对比度。
- **getSaturation**: 获取显示设备的饱和度。
- **setSaturation**: 设置显示设备的饱和度。
- **getHue**: 获取显示设备的色调。
- **setHue**: 设置显示设备的色调。
- **saveConfig**: 保存显示设备信息至Baseparameter。
- **getResolutionSupported**: 获取显示设备此分辨率的支持的HDR格式。
- **isDolbyVisionStatus**: 获取当前产品杜比视界状态（未实现）。
- **setDolbyVisionEnabled**: 设置当前产品杜比视界状态（未实现）。
- **isHDR10Status**: 获取当前产品是否允许播放HDR10（已弃用）。
- **setHDR10Enabled**: 设置当前产品是否允许播放HDR10（已弃用）。
- **isAiImageQuality**: 获取当前产品是否开启AI图像增强。
- **setAiImageQuality**: 设置当前产品开启AI图像增强。
- **isAiImageQualityLabMode**: 获取当前产品是否处于AI图像增强实验室模式。
- **setAiImageQualityLabMode**: 设置当前产品是否处于AI图像增强实验室模式。

## RkDisplayOutputManager

### 【描述】

初始化系统服务的Client端对象。

### 【语法】

```
public RkDisplayOutputManager();
```

### 【返回值】

返回值	描述
RkDisplayOutputManager	RkDisplayOutputManager对象

#### 【举例】

初始化RkDisplayOutputManager，调用方式如下：

```
//导包调用，客户需自行导入相关代码包
import android.os.RkDisplayOutputManager;
RkDisplayOutputManager mRkDisplayOutputManager = new RkDisplayOutputManager();
```

## getCurrentDpyConnState

#### 【描述】

获取显示设备的连接状态。

#### 【语法】

```
public int getCurrentDpyConnState(int display);
```

#### 【参数】

参数名称	描述
display	显示设备的Index

#### 【返回值】

返回值	描述
DRM_MODE_CONNECTED	1, 已连接
DRM_MODE_DISCONNECTED	2, 未连接
DRM_MODE_UNKNOWNCONNECTION	3, 未知状态

#### 【举例】

```
mRkDisplayOutputManager.getCurrentDpyConnState(display);
```

## getDisplayNumber

#### 【描述】

获取显示设备数量。

#### 【语法】

```
public int getDisplayNumber();
```

#### 【返回值】

返回值	描述
int	显示设备数量

【举例】

```
int mDisplayNumber = mRkDisplayOutputManager.getDisplayNumber();
```

## getCurrentInterface

【描述】

获取显示设备当前绑定的显示接口类型。

【语法】

```
public int getCurrentInterface(int display);
```

【参数】

参数名称	描述
display	显示设备的Index

【返回值】

返回值	描述
DRM_MODE_CONNECTOR_Unknown	未知设备
DRM_MODE_CONNECTOR_VGA	VGA
DRM_MODE_CONNECTOR_DVII	DVII
DRM_MODE_CONNECTOR_DVID	DVID
DRM_MODE_CONNECTOR_DVIA	DVIA
DRM_MODE_CONNECTOR_Composite	Composite
DRM_MODE_CONNECTOR_SVIDEO	SVideo
DRM_MODE_CONNECTOR_LVDS	LVDS
DRM_MODE_CONNECTOR_Component	Component
DRM_MODE_CONNECTOR_9PinDIN	9PinDIN
DRM_MODE_CONNECTOR_DisplayPort	DP
DRM_MODE_CONNECTOR_HDMIA	HDMIA
DRM_MODE_CONNECTOR_HDMIB	HDMIB
DRM_MODE_CONNECTOR_TV	TV

返回值	描述
DRM_MODE_CONNECTOR_eDP	EDP
DRM_MODE_CONNECTOR_VIRTUAL	VIRTUAL
DRM_MODE_CONNECTOR_DSI	DSI

#### 【举例】

```
int mDisplayType = mRkDisplayOutputManager.getCurrentInterface(display);
switch(mDisplayType) {
    case DRM_MODE_CONNECTOR_VGA:
        return "VGA";
    case DRM_MODE_CONNECTOR_DVII:
        return "DVII";
    ...
    default:
        return "UNKNOWN";
}
```

## getConnectorInfo

#### 【描述】

获取所有显示设备的连接信息。

#### 【语法】

```
public String[] getConnectorInfo();
```

#### 【返回值】

返回值	描述
String[]	显示设备的连接信息字符串，格式为： <code>type:HDMI A,id:0,state:1</code>

#### 【举例】

```
String[] mConnectorInfos = mRkDisplayOutputManager.getConnectorInfo();
for (int i = 0; i < mDisplayNumber; i++) {
    String typeName = "";
    String id = "";
    String type = "";
    int state = DRM_MODE_UNKNOWNCONNECTION;
    if (null != mConnectorInfos && mConnectorInfos.length == mDisplayNumber) {
        String[] rets = mConnectorInfos[i].split(",");
        if (rets != null && rets.length > 2) {
            type = rets[0].replaceAll("type:", "");
            typeName = CONNECTOR_DISPLAY_NAME.get(type);
            id = rets[1].replaceAll("id:", "");
            try {
                state = Integer.parseInt(rets[2].replaceAll("state:", ""));
            } catch (Exception e) {
                //TODO: handle exception
            }
        }
    }
}
```

```
        e.printStackTrace();
    }
    logd("getDisplayInfoList->typeName:" + typeName + " id:" + id + "
state:" + state);
    if(state != DRM_MODE_CONNECTED) {
        continue;
    }
}
}
```

## updateDisplayInfos

### 【描述】

更新显示设备的状态。

### 【语法】

```
public int updateDisplayInfos()
```

### 【返回值】

返回值	描述
int	0: 成功 -1: 失败

### 【举例】

```
mRkDisplayOutputManager.updateDisplayInfos();
```

## getModeList

### 【描述】

获取显示设备支持的分辨率列表。

### 【语法】

```
public String[] getModeList(int display, int type);
```

### 【参数】

参数名称	描述
display	显示设备的Index
type	显示接口的类型

### 【返回值】

返回值	描述
String[]	显示设备支持的分辨率字符串，格式为： <code>720x480@59.94-739-801-858-488-494-525-</code>

返回值	101a-13500 描述
-----	------------------

#### 【举例】

```
String[] originModes = mRkDisplayOutputManager.getModelList(display, type);  
//过滤时序等信息  
originModes = filterOriginModes(originModes);
```

## setMode

#### 【描述】

设置显示输出的分辨率。

#### 【语法】

```
public void setMode(int display, int type, String mode);
```

#### 【参数】

参数名称	描述
display	显示设备的Index
type	显示接口的类型
mode	分辨率信息

#### 【举例】

```
//可传入过滤后的分辨率字符串  
mRkDisplayOutputManager.setMode(display, type, mode);
```

## getCurrentMode

#### 【描述】

获取指定显示设备和接口的当前分辨率。

#### 【语法】

```
public String getCurrentMode(int display, int type);
```

#### 【参数】

参数名称	描述
display	显示设备的Index
type	显示接口的类型

#### 【返回值】

返回值	描述
String	显示设备当前的分辨率

#### 【举例】

```
String mCurMode = mRkDisplayOutputManager.getCurrentMode(display, type);
```

## getSupportCorlorList

#### 【描述】

获取显示设备支持的色彩空间。

#### 【语法】

```
public String[] getSupportCorlorList(int display, int type);
```

#### 【参数】

参数名称	描述
display	显示设备的Index
type	显示接口的类型

#### 【返回值】

返回值	描述
String[]	显示设备支持的色彩空间, 如RGB-8bit、RGB-10bit、YCBCR444-8bit、YCBCR444-10bit、YCBCR422、YCBCR420-8bit、YCBCR420-10bit

#### 【举例】

```
String[] mSupportColorList =  
mRkDisplayOutputManager.getSupportCorlorList(display, type);
```

## setColorMode

#### 【描述】

设置显示设备的色彩空间。

#### 【语法】

```
public void setColorMode(int display, int type, String format);
```

#### 【参数】

参数名称	描述
display	显示设备的Index
type	显示接口的类型
format	色彩空间

#### 【举例】

```
mRkDisplayOutputManager.setColorMode(display, type, format);
```

## getCurrentColorMode

#### 【描述】

获取显示设备的当前色彩空间。

#### 【语法】

```
public String getCurrentColorMode(int display, int type);
```

#### 【参数】

参数名称	描述
display	显示设备的Index
type	显示接口的类型

#### 【返回值】

返回值	描述
String	显示设备当前的色彩空间，如RGB-8bit、RGB-10bit、YCBCR444-8bit、YCBCR444-10bit、YCBCR422、YCBCR420-8bit、YCBCR420-10bit。

#### 【举例】

```
String mCurColorMode = mRkDisplayOutputManager.getCurrentColorMode(display, type);
```

## getOverScan

#### 【描述】

获取显示设备的缩放比例。

#### 【语法】

```
public Rect getOverScan(int display);
```

#### 【参数】

参数名称	描述
display	显示设备的Index

【返回值】

返回值	描述
Rect	显示设备当前的缩放比例

【举例】

```
Rect overScanRect = mRkDisplayOutputManager.getOverScan(display);
int top = overScanRect.top;
int bottom = overScanRect.bottom;
int left = overScanRect.left;
int right = overScanRect.right;
```

## setOverScan

【描述】

设置显示设备的缩放比例。

【语法】

```
public void setOverScan(int display, int direction, int value);
```

【参数】

参数名称	描述
display	显示设备的Index
direction	方向 left=0 top=1 right=2 bottom=3
value	值, 范围: 80-100

【举例】

```
mRkDisplayOutputManager.setOverScan(display, direction, value);
```

## getBcsh

【描述】

获取显示设备的BCSH值。

【语法】

```
public int[] getBcsh(int display);
```

#### 【参数】

参数名称	描述
display	显示设备的Index

#### 【返回值】

返回值	描述
int[]	BCSH的值 int[0]: 亮度 int[1]: 对比度 int[2]: 饱和度 int[3]: 色调

#### 【举例】

```
int[] mCurBCSH = mRkDisplayOutputManager.getBcsh(display);  
int brightness = mCurBCSH[0];  
int contrast = mCurBCSH[1];  
int saturation = mCurBCSH[2];  
int hue = mCurBCSH[3];
```

## getBrightness

#### 【描述】

获取显示设备的亮度。

#### 【语法】

```
public int getBrightness(int display);
```

#### 【参数】

参数名称	描述
display	显示设备的Index

#### 【返回值】

返回值	描述
int	亮度值

#### 【举例】

```
int mBrightness = mRkDisplayOutputManager.getBrightness(display);
```

## setBrightness

### 【描述】

设置显示设备的亮度。

### 【语法】

```
public int setBrightness(int display, int bright_percent);
```

### 【参数】

参数名称	描述
display	显示设备的Index
bright_percent	亮度值

### 【返回值】

返回值	描述
int	0: 成功 -1: 失败

### 【举例】

```
int mState = mRkDisplayOutputManager.setBrightness(display, bright_percent);
```

## getContrast

### 【描述】

获取显示设备的对比度。

### 【语法】

```
public int getContrast(int display);
```

### 【参数】

参数名称	描述
display	显示设备的Index

### 【返回值】

返回值	描述
int	对比度值

### 【举例】

```
int mContrast = mRkDisplayOutputManager.getContrast(display);
```

## setContrast

### 【描述】

设置显示设备的对比度。

### 【语法】

```
public int setContrast(int display, int con_percent);
```

### 【参数】

参数名称	描述
display	显示设备的Index
con_percent	对比度值

### 【返回值】

返回值	描述
int	0: 成功 -1: 失败

### 【举例】

```
int mState = mRkDisplayOutputManager.setContrast(display, con_percent);
```

## getSaturation

### 【描述】

获取显示设备的饱和度。

### 【语法】

```
public int getSaturation(int display);
```

### 【参数】

参数名称	描述
display	显示设备的Index。

### 【返回值】

返回值	描述
int	饱和度值

### 【举例】

```
int mSaturation = mRkDisplayOutputManager.getSaturation(display);
```

## setSaturation

### 【描述】

设置显示设备的饱和度。

### 【语法】

```
public int setBrightness(int display, int sat_percent);
```

### 【参数】

参数名称	描述
display	显示设备的Index
sat_percent	饱和度值

### 【返回值】

返回值	描述
int	0: 成功 -1: 失败

### 【举例】

```
int mState = mRkDisplayOutputManager.setBrightness(display, sat_percent);
```

## getHue

### 【描述】

获取显示设备的色调。

### 【语法】

```
public int getHue(int display);
```

### 【参数】

参数名称	描述
display	显示设备的Index

### 【返回值】

返回值	描述
int	色调值

【举例】

```
int mHue = mRkDisplayOutputManager.getHue(display);
```

## setHue

【描述】

设置显示设备的色调。

【语法】

```
public int setHue(int display, int degree_percent);
```

【参数】

参数名称	描述
display	显示设备的Index
degree_percent	色调值

【返回值】

返回值	描述
int	0: 成功 -1: 失败

【举例】

```
int mState = mRkDisplayOutputManager.setHue(display, degree_percent);
```

## saveConfig

【描述】

保存显示设备信息至Baseparameter。

【语法】

```
public int saveConfig();
```

【返回值】

返回值	描述
int	0: 成功 -1: 失败

### 【举例】

```
int mState = mRkDisplayOutputManager.saveConfig();
```

## getResolutionSupported

### 【描述】

获取显示设备此分辨率支持的HDR格式，如是否支持HDR10，杜比视界，HLG等等。

### 【语法】

```
public int getResolutionSupported(int display, String resolution);
```

### 【参数】

参数名称	描述
display	显示设备的Index
resolution	需要查询的分辨率

### 【返回值】

返回值	描述
int	此分辨率支持格式的或集 0: SDR 1: HDR10 2: DOLBY_VISION 4: HLG

### 【举例】

```
int mSupport =  
mRkDisplayOutputManager.getResolutionSupported(mDisplayInfo.getDisplayId(),  
mRkDisplayOutputManager.getCurrentMode(mDisplayInfo.getDisplayId()));  
int result = SDR;  
if ((mSupport & DrmDisplaySetting.HDR10) > 0) {  
    result |= HDR10;  
}  
if ((mSupport & DrmDisplaySetting.HLG) > 0) {  
    result |= HLG;  
}
```

## isAiImageQuality

### 【描述】

获取当前产品是否开启AI图像增强。

### 【语法】

```
public boolean isAiImageQuality();
```

#### 【返回值】

返回值	描述
boolean	true: 开启 false: 关闭

#### 【举例】

```
boolean isAiImageQuality = mRkDisplayOutputManager.isAiImageQuality();
```

## setAiImageQuality

#### 【描述】

设置当前产品开启AI图像增强。

#### 【语法】

```
public boolean setAiImageQuality(boolean enabled);
```

#### 【参数】

参数名称	描述
enabled	开启/关闭

#### 【返回值】

返回值	描述
boolean	true: 成功 false: 失败

#### 【举例】

```
boolean mStatus = mRkDisplayOutputManager.setAiImageQuality(enabled);
```

## isAiImageQualityLabMode

#### 【描述】

获取当前产品是否处于AI图像增强实验室模式。

#### 【语法】

```
public boolean isAiImageQualityLabMode();
```

#### 【返回值】

返回值	描述
boolean	true: 开启 false: 关闭

#### 【举例】

```
boolean isAiImageQualityLabMode =  
mRkDisplayOutputManager.isAiImageQualityLabMode();
```

## setAiImageQualityLabMode

#### 【描述】

设置当前产品是否处于AI图像增强实验室模式。

#### 【语法】

```
public boolean setAiImageQualityLabMode(boolean enable);
```

#### 【参数】

参数名称	描述
enable	开启/关闭

#### 【返回值】

返回值	描述
boolean	true: 成功 false: 失败

#### 【举例】

```
boolean mStatus = mRkDisplayOutputManager.setAiImageQualityLabMode();
```