Rockchip RkDisplayOutputManager

ID: RK-KF-YF-583

Release Version: V1.0.0

Release Date: 2023-02-28

Security Level: □Top-Secret □Secret □Internal ■Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED “AS IS”. ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS,MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip’s registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

All rights reserved. ©2023. Rockchip Electronics Co., Ltd.

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

RkDisplayOutputManager is a system service that provides operations for the application layer to access external screen information, switch resolutions, switch color spaces, and more. It facilitates interaction between the application layer and the HWC and DRM display frameworks. This document provides relevant explanations and examples for this interface's API.

Product Version

Chipset	Kernel Version
All RK chip platforms	Linux 4.19/Linux 5.10

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Tao.Weng	2023-02-28	Initial version

Contents

Rockchip RkDisplayOutputManager

1. Introduction
2. API Reference
   - 2.1 RkDisplayOutputManager
   - 2.2 getCurrentDpyConnState
   - 2.3 getDisplayNumber
   - 2.4 getCurrentInterface
   - 2.5 getConnectorInfo
   - 2.6 updateDisplayInfos
   - 2.7 getModeList
   - 2.8 setMode
   - 2.9 getCurrentMode
   - 2.10 getSupportCorlorList
   - 2.11 setColorMode
   - 2.12 getCurrentColorMode
   - 2.13 getOverScan
   - 2.14 setOverScan
   - 2.15 getBcsh
   - 2.16 getBrightness
   - 2.17 setBrightness
   - 2.18 getContrast
   - 2.19 setContrast
   - 2.20 getSaturation
   - 2.21 setSaturation
   - 2.22 getHue
   - 2.23 setHue
   - 2.24 saveConfig
   - 2.25 getResolutionSupported
   - 2.26 isAiImageQuality
   - 2.27 setAiImageQuality
   - 2.28 isAiImageQualityLabMode
   - 2.29 setAiImageQualityLabMode

1. Introduction

RkDisplayOutputManager is a system service that provides operations for the application layer to access external screen information, switch resolutions, switch color spaces, and more. It facilitates interaction between the application layer and the HWC and DRM display frameworks.

For details codes, refer to:
packages/apps/TvSettings/Settings/src/com/android/tv/settings/display/DrmDisplaySetting.java

2. API Reference

The system service provides the following APIs:

- **RkDisplayOutputManager**: Initializes the Client object of the system service.
- **getCurrentDpyConnState**: Obtains the connection status of the display device.
- **getDisplayNumber**: Obtains the number of display devices.
- **getCurrentInterface**: Obtains the current display interface type bound to the display device.
- **getConnectorInfo**: Obtains connection information for all display devices.
- **updateDisplayInfos**: Updates the status of the display device.
- **getModeList**: Obtains the list of supported resolutions for the display device.
- **setMode**: Sets the resolutions of the display device.
- **getCurrentMode**: Obtains the current resolution of the specified display device and interface.
- **getSupportCorlorList**: Obtains the supported color spaces for the display device.
- **setColorMode**: Sets the color space for the display device.
- **getCurrentColorMode**: Obtains the current color space of the display device.
- **getOverScan**: Obtains the scaling factor of the display device.
- **setOverScan**: Sets the scaling factor of the display device.
- **getBcsh**: Obtains the BCSH value of the display device.
- **getBrightness**: Obtains the brightness of the display device.
- **setBrightness**: Sets the brightness of the display device.
- **getContrast**: Obtains the contrast of the display device.
- **setContrast**: Sets the contrast of the display device.
- **getSaturation**: Obtains the saturation of the display device.
- **setSaturation**: Sets the saturation of the display device.
- **getHue**: Obtains the hue of the display device.
- **setHue**: Sets the hue of the display device.
- **saveConfig**: Saves display device information to Baseparameter.
- **getResolutionSupported**: Obtains supported HDR formats for the display device at the resolution.
- **isDolbyVisionStatus**: Obtains the current Dolby Vision status of the product (reserved).
- **setDolbyVisionEnabled**: Sets the current Dolby Vision status of the product (reserved).
- **isHDR10Status**: Obtains information about if the current product allows HDR10 playback (deprecated).
- **setHDR10Enabled**: Sets whether the current product allows HDR10 playback (deprecated).
- **isAiImageQuality**: Obtains information about whether the current product has AI image enhancement enabled.
- **setAiImageQuality**: Enable AI image enhancement for the current product.

- **isAiImageQualityLabMode**: Obtains information about whether the current product is in AI image enhancement laboratory mode.
- **setAiImageQualityLabMode**: Sets whether the current product is in AI image enhancement laboratory mode.

2.1 RkDisplayOutputManager

[Description]

Initializes the Client object of the system service.

[Syntax]

```
public RkDisplayOutputManager();
```

[Return Value]

Return Value	Description
RkDisplayOutputManager	RkDisplayOutputManager object

[Example]

Initializes the RkDisplayOutputManager. The usage is as follows:

```
//Import the package and make the call. Users are responsible for importing the
revelant code package themselves.
import android.os.RkDisplayOutputManager;
RkDisplayOutputManager mRkDisplayOutputManager = new RkDisplayOutputManager();
```

2.2 getCurrentDpyConnState

[Description]

Obtains the connection status of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
DRM_MODE_CONNECTED	1: connected

Return Value	Description
DRM_MODE_DISCONNECTED	2: not connected
DRM_MODE_UNKNOWNCONNECTION	3: unknown

[Example]

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

2.3 getDisplayNumber

[Description]

Obtains the number of display devices.

[Syntax]

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

[Return Value]

Return Value	Description
int	Number of display devices

[Example]

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

2.4 getCurrentInterface

[Description]

Obtains the current display interface type bound to the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
DRM_MODE_CONNECTOR_Unknown	Unknown device
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
DRM_MODE_CONNECTOR_eDP	EDP
DRM_MODE_CONNECTOR_VIRTUAL	VIRTUAL
DRM_MODE_CONNECTOR_DSI	DSI

[Example]

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

2.5 getConnectorInfo

[Description]

Obtain connection information for all display devices.

[Syntax]

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

[Return Value]

Return Value	Description
String[]	Connection information string of the display device, foramt: `type:HDMIA,id:0,state:1`

[Example]

```
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;
            }
        }
    }
}
```

2.6 updateDisplayInfos

[Description]

Updates the status of the display device.

[Syntax]

```
public int updateDisplayInfos()
```

[Return Value]

Return Value	Description
int	0: success -1: failure

[Example]

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

2.7 getModeList

[Description]

Obtains the list of supported resolutions for the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
type	Type of display interface

[Return Value]

Return Value	Description
String[]	Supported resolution strings of the display device, format: `720x480@59.94-739-801-858-488-494-525-101a-13500`

[Example]

```
String[] originModes = mRkDisplayOutputManager.getModeList(display, type);
//Filter timing and other information
originModes = filterOrginModes(originModes);
```

2.8 setMode

[Description]

Set the display output resolution.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
type	Type of display interface
mode	Resolution information

[Example]

```
//Filtered resolution strings can be passed in
mRkDisplayOutputManager.setMode(display, type, mode);
```

2.9 getCurrentMode

[Description]

Obtains the current resolution of the specified display device and interface.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
type	Type of display interface

[Return Value]

Return Value	Description
String	The current resolution of the display device

[Example]

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

2.10 getSupportCorlorList

[Description]

Obtains the supported color spaces for the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
type	Type of display interface

[Return Value]

Return Value	Description
String[]	The supported color spaces for the display device. Such as RGB-8bit, RGB-10bit, YCBCR444-8bit, YCBCR444-10bit, YCBCR422, YCBCR420-8bit, and YCBCR420-10bit

[Example]

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

2.11 setColorMode

[Description]

Sets the color space for the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
type	Type of display interface
format	Color space

[Example]

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

2.12 getCurrentColorMode

[Description]

Obtains the current color space of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
type	Type of display interface

[Return Value]

Return Value	Description
String	The current color space of the display device. Such as RGB-8bit, RGB-10bit, YCBCR444-8bit, YCBCR444-10bit, YCBCR422, YCBCR420-8bit, YCBCR420-10bit

[Example]

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

2.13 getOverScan

[Description]

Obtains the scaling factor of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
Rect	The current scaling factor of the display device

[Example]

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

2.14 setOverScan

[Description]

Sets the scaling factor of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
direction	Direction: left=0 top=1 right=2 bottom=3
value	Value, range: 80-100

[Example]

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

2.15 getBcsh

[Description]

Obtains the BCSH value of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
int[]	BCSH value int[0]: brightness int[1]: contrast int[2]: saturation int[3]: hue

[Example]

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

2.16 getBrightness

[Description]

Obtains the brightness of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
int	Brightness value

[Example]

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

2.17 setBrightness

[Description]

Sets the brightness of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
bright_percent	Brightness value

[Return Value]

Return Value	Description
int	0: success -1: failure

[Example]

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

2.18 getContrast

[Description]

Obtains the contrast of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
int	Contrast value

[Example]

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

2.19 setContrast

[Description]

Sets the contrast of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
con_percent	Contrast value

[Return Value]

Return Value	Description
int	0: success -1: failure

[Example]

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

2.20 getSaturation

[Description]

Obtains the saturation of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device。

[Return Value]

Return Value	Description
int	Saturation value

[Example]

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

2.21 setSaturation

[Description]

Sets the saturation of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
sat_percent	Saturation value

[Return Value]

Return Value	Description
int	0: success -1: failure

[Example]

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

2.22 getHue

[Description]

Obtains the hue of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device

[Return Value]

Return Value	Description
int	Hue value

[Example]

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

2.23 setHue

[Description]

Sets the hue of the display device.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
degree_percent	Hue value

[Return Value]

Return Value	Description
int	0: success -1: failure

[Example]

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

2.24 saveConfig

[Description]

Saves display device information to Baseparameter.

[Syntax]

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

[Return Value]

Return Value	Description
int	0: success -1: failure

[Example]

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

2.25 getResolutionSupported

[Description]

Obtains the HDR format supported by the display device at this resolution, such as HDR10, Dolby Vision, HLG, and others.

[Syntax]

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

[Parameter]

Parameter	Description
display	Index of the display device
resolution	Resolution to be queried

[Return Value]

Return Value	Description
int	The union set represents the formats supported by the resolution 0: SDR 1: HDR10 2: DOLBY_VISION 4: HLG

[Example]

```
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;
}
```

2.26 isAiImageQuality

[Description]

Obtains information about whether the current product has AI image enhancement enabled.

[Syntax]

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

[Return Value]

Return Value	Description
boolean	true: enabled false: disabled

[Example]

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

2.27 setAiImageQuality

[Description]

Enable AI image enhancement for the current product.

[Syntax]

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

[Parameter]

Parameter	Description
enabled	Enabled or disabled

[Return Value]

Return Value	Description
boolean	true: success false: failure

[Example]

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

2.28 isAiImageQualityLabMode

[Description]

Obtains information about whether the current product is in AI image enhancement laboratory mode.

[Syntax]

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

[Return Value]

Return Value	Description
boolean	true: enabled false: disabled

[Example]

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

2.29 setAiImageQualityLabMode

[Description]

Sets whether the current product is in AI image enhancement laboratory mode.

[Syntax]

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

[Parameter]

Parameter	Description
enable	Enabled or disabled

[Return Value]

Return Value	Description
boolean	true: success false: failure

[Example]

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