

Hisense

Platform Name	Product Name	System Version
MTK9666	32/43/50/55/65/75/86DM66D	Android 11.0
MTK9666	43/50/55/65/75/86GM50D	Android 11.0
MTK9666	46/55VW25E	Android 11.0
T963	43/50/55/65/75/85DP30FE	Android 9

Hisense Commercial Display

Welcome

A seamless Digital Signage(DS) API integration makes it simpler and easier for our end users to interact with our product.

Hisense developed an infrastructure to support Digital Signage flows through Android Studio . Android Studio is an Android platform that allows the customers to interact with our API easily and is generic. We have exposed the DS API with partners. This enables our partners to use the DS API in a more convenient way in order to extend their product and build their own applications.

Disclaimer

This document includes the full list of services, which support the flows used in Hisense Digital Signage.

This Hisense API Reference Guide is still in work and subject to change.

Hisense Copyright

The product described in this publication is a licensed product of HISENSE Group Corporation.

Hisense is a registered trademark of Hisense Group. Other product names mentioned in this publication may be trademarks or registered trademarks of their respective companies and are hereby acknowledged. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Java is a registered trademark of Oracle and/or its affiliates.

Where creation of derivative works, modifications or copies of this Hisense copyrighted documentation is permitted under the terms and conditions of an agreement you have with Hisense, Hisense's copyright notice must be included.

It is the policy of HISENSE Group (HISENSE) to improve products as new technology, components, software, and firmware become available. HISENSE, therefore, reserves the right to change specifications without prior notice. All features, functions, and operations described herein may not be marketed by HISENSE in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult with your HISENSE representative or HISENSE office for information that is applicable and current.

To maintain the quality of our publications, we need your comments on the accuracy, clarity, organization, and value of this publication. Please use the link below to send your comments.

Internet Address:

<https://global.hisense.com/>

Copyright © 1969 – 2024

By HISENSE Group

Global Headquarters

17 Donghaixi Road,

Qingdao, 266072

China

All Rights Reserved

Overview

This document describes what each function does individually and provides information about using APIs to build your applications; it also includes details on integrating multiple APIs into a single application.

Version & Revision History

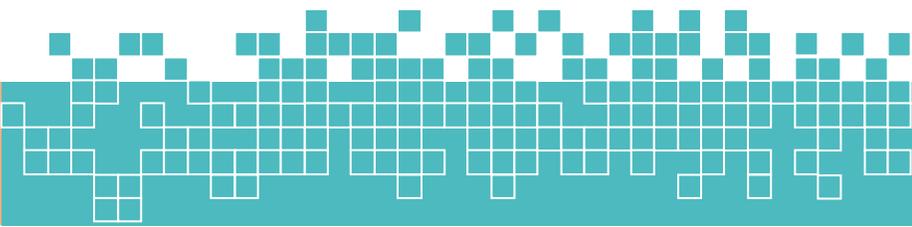
Version	Description	Date
V1.0	The initial release	2023-12-1
V1.1	<ol style="list-style-type: none"> 1.Add “Set Color Temperature Mode” API 2.Add “Get Color Temperature Mode”API 3.Add “Set Screen Rotation”API 4.Add “Get Screen Rotation”API 5.Add “Set Menu Rotation”API 6.Add”Get Menu Rotation”API 	2024-1-4
V1.1.1	<ol style="list-style-type: none"> 1.Add note for “Set Screen Rotation”API 2.Add note for “Get Screen Rotation”API 3.Add note for “Set Menu Rotation”API 4.Add note for “Get Menu Rotation”API 5.Add “setSettingsDBValue”API 	2024-1-23
V1.1.2	<ol style="list-style-type: none"> 1.Add “setTimerSwitchInfo”API 	2024-1-31
V1.1.3	<ol style="list-style-type: none"> 1.Add the example codes for “installSilentApp”API 2.Modify the parameter description of “setSourceWindow” API 	2024-6-26
V1.1.4	<ol style="list-style-type: none"> 1.Add “getUsageTime()” API 2.Add “setPowerOnDelayEnable ()” API 3.Add “getPowerOnDelayEnable ()” API 4.Add “setPowerOnDelay ()” API 5.Add “getPowerOnDelay ()” API 6.Add “setFakePowerOff ()” API 7.Add “getFakePowerOff ()” API 8.Add “setDefaultLauncher ()” API 9.Add “getDefaultLauncher ()” API 10.Add “setMonitorId ()” API 11.Add “getMonitorId ()” API 12.Add “restoreSystemSettings ()” API 13.Add “setCecEnable ()” API 14.Add “getCecEnable ()” API 15.Add “setCecAutoPowerOffEnable ()” API 16.Add “getCecAutoPowerOffEnable ()” API 17.Add “setCecAutoWakeUpEnable ()” API 18.Add “getCecAutoWakeUpEnable ()” API 19.Add “setNoOpreationStandbyTime ()” API 20.Add “getNoOpreationStandbyTime ()” API 	2024-8-15

	<p>21.Add “setFirstTempProtectFlag ()” API</p> <p>22.Add “getFirstTempProtectFlag ()” API</p> <p>23.Add “setSecondTempProtectFlag ()” API</p> <p>24.Add “getSecondTempProtectFlag ()” API</p> <p>25.Add “setThirdTempProtectFlag ()” API</p> <p>26.Add “getThirdTempProtectFlag ()” API</p> <p>27.Add “setSourceBakEnable ()” API</p> <p>28.Add “getSourceBakEnable ()” API</p> <p>29.Add “setSourceBakMasterChannel ()” API</p> <p>30.Add “getSourceBakMasterChannel ()” API</p> <p>31.Add “setSourceBakSlaveChannel ()” API</p> <p>32.Add “getSourceBakSlaveChannel ()” API</p> <p>33.Add “setSourceBakChannelExtra ()” API</p> <p>34.Add “getSourceBakChannelExtra ()” API</p> <p>35.Add “setNoSignalStandbyEnable ()” API</p> <p>36.Add “getNoSignalStandbyEnable ()” API</p> <p>37.Add “setNoSignalStandby ()” API</p> <p>38.Add “getNoSignalStandby ()” API</p>	
V1.1.5	<p>1.” setCurrentInputSource()” add HDMI3</p> <p>2.” getCurrentInputSource()”add HDMI3</p> <p>3.” setStartupInputSource()” add HDMI3</p> <p>4.” getStartupInputSource()” add HDMI3</p>	2025-3-31

Table of contents

Hisense API SDK User Guide.....	9
1. Integrated approach	9
2. Initialization	9
3. Develop	10
4. Test.....	10
System API.....	10

Hisense Commercial Display



1.	Device Version	10
2.	Device ID	10
3.	Serial Number	11
4.	Screenshot	11
5.	Simulate buttons	11
6.	Start Settings	12
7.	Power off	12
8.	Set Power on Mode.....	12
9.	Get Power on Mode	13
10.	Set Boot Logo	13
11.	Set Boot Animation	13
12.	Turn on/off Screen.....	14
13.	Install the third app silently	14
14.	Uninstall the third app silently.....	16
15.	Set Volume.....	16
16.	Get Volume	16
17.	Set Mute	17
18.	Get Mute	17
19.	Set Default Volume.....	17
20.	Get Default Volume	18
21.	Set Max Volume.....	18
22.	Get Max Volume	18
23.	Reboot	18
24.	Set OTA Upgrade Enable.....	19
25.	Get Whether OTA Upgrade Enable.....	19
26.	Install OTA Package.....	19
27.	Set System Time	19
28.	Set White Balance Gain	20
29.	Get White Balance Gain.....	20
30.	Set White Balance Offset	21
31.	Get White Balance Offset.....	21
32.	Set Color Temperature Mode	22
33.	Get Color Temperature Mode.....	22
34.	Set Screen Rotation.....	22
35.	Get Screen Rotation	23
36.	Set Menu Rotation	23
37.	Get Menu Rotation.....	24
38.	Set Settings DB	24
39.	Set Timer On and Off.....	25
40.	Get the runtime	26
41.	Set the power-on delay switch	27
42.	Get the power-on delay switch.....	27
43.	Set the power-on delay time.....	27
44.	Get the power-on delay time	28
45.	Set sleep mode switch	28
46.	Get sleep mode switch.....	28
47.	Set Default Home Page	29
48.	Get Default Home Page.....	29
49.	Set serial port ID	29
50.	Get serial port ID.....	30
51.	Factory reset.....	30

52.	Set HDMI CEC switch.....	30
53.	Get HDMI CEC switch	31
54.	Set the CEC standby control switch	31
55.	Get the CEC standby control switch	31
56.	Set the CEC power-on switch.....	31
57.	Get the CEC power-on switch	32
58.	Set time of No operation standby	32
59.	Get time of No operation standby.....	32
60.	Set the over-temperature protection level 1 switch	33
61.	Get the over-temperature protection level 1 switch status	33
62.	Set the over-temperature protection level 2 switch	33
63.	Get the over-temperature protection level 2 switch status.....	34
64.	Set the over-temperature protection level 3 switch	34
65.	Get the over-temperature protection level 3 switch status.....	35
Source API		35
1.	Switch to the Specific Source	35
2.	Get Current Input Source	35
3.	Set Boot Source.....	36
4.	Get Boot Source	36
5.	Set Brightness.....	37
6.	Get Brightness	37
7.	Set Contrast	37
8.	Get Contrast.....	37
9.	Set Backlight	38
10.	Get Backlight.....	38
11.	Set Saturation.....	38
12.	Get Saturation	39
13.	Set Source Window	39
14.	Set Gamma Mode.....	40
15.	Get Gamma Mode	40
16.	Set Dynamic Contrast.....	40
17.	Get Dynamic Contrast	41
18.	Set Color Correction	41
19.	Get Color Correction.....	41
20.	Set Failover switch	42
21.	Get Failover switch	42
22.	Set master channel of Failover	43
23.	Get master channel of Failover.....	43
24.	Set slave channel of Failover	43
25.	Get slave channel of Failover	44
26.	Set the slave channel package name or id parameter.....	44
27.	Get the slave channel package name or id parameter	44
28.	Set the no-signal standby function switch.....	44
29.	Get the no-signal standby function switch.....	45
30.	Set time of no-signal standby	45
31.	Get time of no-signal standby.....	45
Network API		46

32.	Get Network Type	46
33.	Get IP Address	46
34.	Get Mac Address	47
35.	Wifi status	47
36.	Wired Status	47
37.	Connect Wifi	47
38.	Auto Connect Wired network.....	48
39.	Wired network static connection Settings.....	48
40.	Set Hotspot.....	48
41.	Get Hotspot Status.....	49

Hisense Commercial Display

Resources

The resources are described in three categories: Hisense APISDK User Guide, system API, Source API and Network API.

Hisense API SDK User Guide

1. Integrated approach

a) Download jar library

Download or copy the hisense_api.jar for Hisense API

b) Android Studio Project

Copy the hisense_api.jar file to the libs directory of the module, and add dependencies in the gradle file of the module:

```
dependencies {
    implementation files('libs/hisense_api.jar')
}
```

c) Parameter configuration

To avoid confusion, please add the following configuration to the Proguard obfuscation file:

```
-dontwarn com.hisense.hotel.**
-keep public class com.hisense.hotel.**{*;}

```

2. Initialization

APK need to import com.hisense.hotel.IServicesReadyListener

Listen to whether the interface is registered via addServiceReadyListener

The system will call the SDK interface through allServicesReady notification to the application after the service is ready.

When the application exits, you need to try the onDestroy of HotelSystemManager.

Call the initialization interface:

```
// Register listener
HisenseManager.getInstance().addServiceReadyListener(new
IServicesReadyListener() {
    @Override
    public void allServicesReady() {
        // Service initialization completed
        // TODO Instantiate the HotelSystemManager,

```

HotelSourceManager, and HotelNetworkManager classes to call the SDK interface

```
    }  
});  
//Initialization  
HisenseManager.getInstance().init(context);
```

3. Develop

Instantiate HotelSystemManager and call System API interface;
Instantiate HotelSourceManager and call Source API interface;
Instantiate HotelNetworkManager and call Network API interface;

4. Test

Install HisenseApiTest.apk for interface testing.

System API

Import Package: HotelSystemManager

1. Device Version

Description

Get device software version information.

Function Name

String getSoftwareVersion()

Parameters:

Parameter	Category	Type	Description
	Output	String	version

Example Response

String: V0000.00.01A

2. Device ID

Description

Get the device ID, which is feature code + the last 8 digits of the mac

Function Name

String getDeviceID()

Parameters

Parameter	Category	Type	Description
	Output	String	Device ID

Example Response

86100300900000b00000060a484e847e

3. Serial Number**Description**

Get the serial number of the device

Function Name

String getSerialNumber()

Parameters

Parameter	Category	Type	Description
	Output	String	Serial Number

4. Screenshot**Description**

Get a screenshot of the current display interface. If the parameter "path" is null, a file such as screen_XXXXXX.jpg will be generated in the data/data directory.

Function Name

void shotScreen(String path)

Parameters

Parameter	Category	Type	Description
Path	Input	String	Save the path and name of jpg file Such as /data/data/screen2.jpg

5. Simulate buttons**Description**

Simulate android key value

Function Name

Boolean injectKey(final int keyCode)

Parameters

Parameter	Category	Type	Description
keycode	Input	int	For simulated key values, see the KEYCODE definition in

			android.view.KeyEvent, including: Back: 4 Number0-9: 7-16 Up(Volume up): 24 Down(Volume down): 25 Left(Rewind):89 Right(fast forward):2060 OK:23 Mute:164 Settings:2075 Standby:26 Menu:82 Home:3
--	--	--	--

6. Start Settings

Description

Start Hisense's Settings

Function

Void startSystemSettingMenu()

7. Power off

Description

Power off

Function Name

Boolean shutdownSystem(Context context)

Parameters

Parameter	Category	Type	Description
	Output	boolean	False: setup failed True: setup successful

8. Set Power on Mode

Description

Set the boot mode: direct/standby/last

Function Name

Boolean setBootModel(int model)

Parameters

Parameter	Category	Type	Description
model	Input	int	0: direct 1: last 2:standby
	Output	boolean	False: setup failed True: setup successful

9. Get Power on Mode

Description

get the boot mode

Function Name

int getBootModel()

Parameters

Parameter	Category	Type	Description
model	Output	int	0: direct 1: last 2:standby

10. Set Boot Logo

Description

Set the boot logo. Photos only support jpg and png format, and the size cannot exceed 4MB.

DP30FE only support jpg.

Function Name

boolean setBootLogo(String path)

Parameters

Parameter	Category	Type	Description
path	Input	String	The absolute path of photos
	Output	boolean	False: setup failed True: setup successful

11. Set Boot Animation

Description

Set the boot animation.

The boot animation can be in the form of image compression named "third_party_bootanimation.zip". It can also be in the form of video named "Boot_animation.mp4".

"third_party_bootanimation.zip" and "Boot_animation.mp4" are placed in the "Boot animation" folder in the root directory of the USB disk.

DP30FE : The size cannot exceed 50MB.

Function Name

boolean setBootAnimation(String path)

Parameters

Parameter	Category	Type	Description
path	Input	String	The absolute path of animation
	Output	boolean	False: setup failed True: setup successful

12. Turn on/off Screen

Description

Turn on/off the screen

Function Name

Boolean shutdownScreen(int model)

Parameters

Parameter	Category	Type	Description
model	Input	int	0: Turn on screen 1: Turn off screen 2: Close screen and volume
	Output	boolean	0: setup failed 1: setup successful

13. Install the third app silently

Description

Install the third app silently

Function

Boolean installSilentApp(String packageName, String path)

Parameters

Parameter	Category	Type	Description
-----------	----------	------	-------------

packageName	Input	String	The package name of installed App
path	Input	String	The path of installed App
	Output	boolean	0: setup failed 1: setup successful

Example codes:

1. Create a button to show dialog in MainActivity

```

<Button
    android:id="@+id/installSilentApp"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:text="@string/install_silent_app" />

view.findViewById(R.id.installSilentApp).setOnClickListener(this.buttonListener);
view.findViewById(R.id.uninstallApp).setOnClickListener(this.buttonListener);

return;

case R.id.installSilentApp:
    showInstallAppDialog();
    return;

```

2. In function showInstallAppDialog(), start a DialogFragment

```

public void showInstallAppDialog() {
    InstallAppDialogFragment installAppDialogFragment = new InstallAppDialogFragment();
    installAppDialogFragment.setTargetFragment( fragment this, requestCode: 19);
    installAppDialogFragment.show(getFragmentManager(), tag: "installAppDialogFragment");
}

public class InstallAppDialogFragment extends DialogFragment {
    private EditText etPath;
    private EditText etPackage;

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        LayoutInflater inflater = getActivity().getLayoutInflater();
        View view = inflater.inflate(R.layout.upgradeapppackage_edittext_list_fragment, root: null);
        this.etPath = (EditText) view.findViewById(R.id.et_app_path);
        this.etPackage = (EditText) view.findViewById(R.id.et_packageName);
        etPath.setText("/storage/CEA9-56BC/CusSDK4.0_ui-hisense.apk");
        etPackage.setText("com.xbh.sdk.demo");
        builder.setView(view).setTitle("安装应用(请输入正确的安装包路径和应用包名)").setPositiveButton( text: "确定", new DialogInterface.OnClickListener() { //
            @Override
            public void onClick(DialogInterface dialog, int id) {
                Intent intent = new Intent();
                intent.putExtra( name: "apkPath", InstallAppDialogFragment.this.etPath.getText().toString());
                intent.putExtra( name: "packageName", InstallAppDialogFragment.this.etPackage.getText().toString());
                InstallAppDialogFragment.this.getTargetFragment().onActivityResult( requestCode: 19, resultCode: -1, intent);
            }
        }).setNegativeButton( text: "取消", (DialogInterface.OnClickListener) null);
        return builder.create();
    }
}

```

The DialogFragment putExtra to MainActivity by onActivityResult()

3.MainFragment getStringExtra in onActivityResult(), then call the method installSilentApp() to install app silently.

```

if (requestCode == 19) {
    String apkPath = data.getStringExtra( name: "apkPath");
    String packageName = data.getStringExtra( name: "packageName");
    LogUtils.d(apkPath, packageName);
    boolean bool = mHotelSystemManager.installSilentApp(packageName, apkPath);
    printInfo(bool);
}
    
```

14. Uninstall the third app silently

Description

Uninstall the third app silently

Function

Boolean uninstallApp(String packageName)

Parameters

Parameter	Category	Type	Description
packageName	Input	String	The package name of uninstalled App
	Output	boolean	0: setup failed 1: setup successful

15. Set Volume

Description

Set volume

Function Name

Boolean setVolume(int volume)

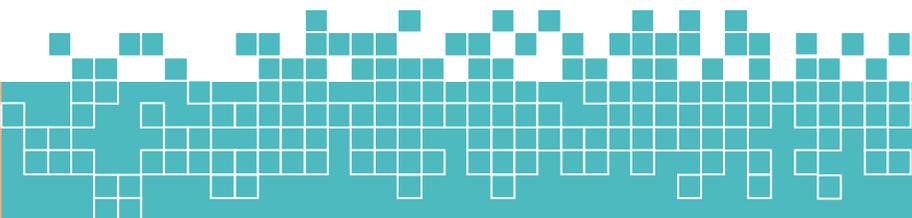
Parameters

Parameter	Category	Type	Description
volume	Input	int	0-100
	Output	boolean	0: setup failed 1: setup successful

16. Get Volume

Description

Get the current volume



Function

int getVolume(int volume)

Parameters

Parameter	Category	Type	Description
	Output	int	0-100

17. Set Mute**Description**

Set Mute/unMute

Function Name

Boolean setMuteFlag(Boolean flag)

Parameters

Parameter	Category	Type	Description
flag	Input	boolean	0: unmute 1: mute
	Output	boolean	0: setup failed 1: setup successful

18. Get Mute**Description**

Get the state of the volume: mute/unmute

Function Name

Boolean getMuteFlag ()

Parameters

Parameter	Category	Type	Description
	Output	boolean	0: Mute on 1: Mute off

19. Set Default Volume**Description**

Set the default volume at startup

Function Name

Boolean setDefaultVolume(int volume)

Parameters

Parameter	Category	Type	Description
-----------	----------	------	-------------

volume	Input	int	0-100
--------	-------	-----	-------

20. Get Default Volume

Description

get the default volume at startup

Function Name

int getDefaultVolume(void)

Parameters

Parameter	Category	Type	Description
volume	Output	int	0-100

21. Set Max Volume

Description

Set max volume

Function Name

Boolean setMaxVolume(int volume)

Parameters

Parameter	Category	Type	Description
volume	Input	int	0-100

22. Get Max Volume

Description

Get max volume

Function Name

int getMaxVolume(void)

Parameters

Parameter	Category	Type	Description
volume	Output	int	0-100

23. Reboot

Description

Reboot device

Function Name

void rebootDevice()

24. Set OTA Upgrade Enable

Description

Set the OTA upgrade switch

Function Name

boolean setOTAUpgradeEnable(boolean enabled)

Parameters

Parameter	Category	Type	Description
enabled	Input	boolean	True: on False: off
	Output	boolean	False: setup failed True: setup successful

25. Get Whether OTA Upgrade Enable

Description

Get whether OTA upgrade enabled

Function Name

boolean getOTAUpgradeEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	False: off True: on

26. Install OTA Package

Description

Install OTA package

Function Name

boolean installOTAPackage(String path)

Parameters

Parameter	Category	Type	Description
path	Input	String	The absolute path of OTA package
	Output	boolean	False: setup failed True: setup successful

27. Set System Time

Description**Set system time****Function Name**

boolean setSystemTime_Year(int year, int month, int day)

boolean setSystemTime_Hour(int hour, int minute, int second)

Parameters

Parameter	Category	Type	Description
year	Input	int	Can't be newer than 2036
month	Input	int	1-12
day	Input	int	1-31
hour	Input	int	0-23
minute	Input	int	0-59
second	Input	int	0-59
	Output	boolean	False: setup failed True: setup successful

28. Set White Balance Gain**Description**

Set white balance gain

Function Name

boolean setWhiteBalanceGain(int[] rgb)

Parameters

Parameter	Category	Type	Description
rgb	Input	int[]	The length is fixed at 3, which are the red gain value, green gain value, and blue gain value, and the range is -30 to 30.
	Output	boolean	False: setup failed True: setup successful

29. Get White Balance Gain**Description**

Get White balance gain

Function Name

int[] getWhiteBalanceGain()

Parameters

Parameter	Category	Type	Description
	Output	int[]	The length is fixed at 3, which are the red gain value, green gain value, and blue gain value, and the range is -30 to 30.

30. Set White Balance Offset

Description

Set white balance offset

Function Name

boolean setWhiteBalanceOffset(int[] rgb)

Parameters

Parameter	Category	Type	Description
rgb	Input	int[]	The length is fixed at 3, which are the red offset value, green offset value, and blue offset value, and the range is -30 to 30.
	Output	boolean	False: setup failed True: setup successful

31. Get White Balance Offset

Description

Set white balance offset

Function Name

int[] getWhiteBalanceOffset()

Parameters

Parameter	Category	Type	Description
	Output	int[]	The length is fixed at 3, which are the red offset value, green offset value, and blue offset value, and the range is -30 to 30.

32. Set Color Temperature Mode

Description

Set color temperature mode

Function Name

boolean setUserColorTempMode(int mode)

Parameters

Parameter	Category	Type	Description
mode	Input	int[]	0: cold 1: coldish 2: standard 3: warmish 4: warm
	Output	boolean	False: setup failed True: setup successful

33. Get Color Temperature Mode

Description

Get color temperature mode

Function Name

int getUserColorTempMode()

Parameters

Parameter	Category	Type	Description
	Output	int	0: cold 1: coldish 2: standard 3: warmish 4: warm

34. Set Screen Rotation

Description

Set the rotation of screen

Function Name

void setScreenRotation(int rotation)

Parameters

Parameter	Category	Type	Description
rotation	Input	int	0: 0°

			1: 90° 2: 180° (DP30FE do not supported) 3: 270° (DP30FE do not supported)
	Output		

Note

If the screen rotation in the settings only has landscape and vertical, then 0 represents landscape screen and 1 represents vertical screen.

35. Get Screen Rotation**Description**

Get the rotation of screen

Function Name

int getScreenRotation()

Parameters

Parameter	Category	Type	Description
	Input		
rotation	Output	int	0: 0° 1: 90° 2: 180° (DP30FE do not supported) 3: 270° (DP30FE do not supported)

Note

If the screen rotation in the settings only has landscape and vertical, then 0 represents landscape screen and 1 represents vertical screen.

36. Set Menu Rotation**Description**

Set the rotation of OSD menu

Function Name

void setMenuRotation(int rotation)

Parameters

Parameter	Category	Type	Description
rotation	Input	int	0: 0° 1: 90°

			2: 180° (DP30FE do not supported) 3: 270° (DP30FE do not supported)
	Output		

Note

If the menu rotation in the settings only has landscape and vertical, then 0 represents landscape menu and 1 represents vertical menu.

37. Get Menu Rotation

Description

Get the rotation of OSD menu

Function Name

int getMenuRotation()

Parameters

Parameter	Category	Type	Description
	Input		
	Output	int	0: 0° 1: 90° 2: 180° (DP30FE do not supported) 3: 270° (DP30FE do not supported)

Note

If the menu rotation in the settings only has landscape and vertical, then 0 represents landscape menu and 1 represents vertical menu.

38. Set Settings DB

Description

Set the value of Settings DB

Function Name

boolean setSettingsDBValue(int type,String key,String value)

Parameters

Parameter	Category	Type	Description
type	Input	int	0: system DB 1: global DB 2: secure DB
key	Input	String	the key of database

	Input	String	the value of the database
	Output	boolean	False: setup failed True: setup succeed

39. Set Timer On and Off

Description

Set the timer on/off (corresponding to the parameters in Settings-On/Off-Timer On/Off). Note that the time interval of the timer on/off sub-option must not be less than 5 minutes.

Function Name

boolean setTimerSwitchInfo(boolean isPowerOff, boolean enable, int index, int type, int hour, int minute, int[] manualWeeks)

Parameters

Parameter	Category	Type	Description
isPowerOff	Input	boolean	True: scheduled power off False: scheduled power on
enable	Input	boolean	True: open the corresponding index scheduled power on/off option; False: turn off the corresponding index scheduled power on and off function
index	Input	int	Settings support 6 groups scheduled power on/off. 0/1/2 : scheduled power-on; 3/4/5: scheduled power-off
type	Input	int	There are 7 types can be set for the timer switch. 0: off 1: once 2: every day 3: Monday to Friday 4: Monday to Saturday 5: Saturday to Sunday 6: manual setting
hour	Input	int	the input hour,

			ranging from 0 to 23
minute	Input	int	The input minute, Ranging from 0 to 59
[]manualWeeks	Input	Int[]	When the type is 6, this parameter should not be null. It is the week array data. When the type is not 6, this parameter can be null. 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday
	Output	boolean	False: setup failed True: setup succeed

Example

```
// Set the first set of scheduled power-on time from Monday to Friday at 9:30
mHotelSystemManager.setTimerSwitchInfo(false,true,0,3,9,30,null);
```

```
// Set the first set of scheduled power-off time from Monday to Friday at 10:30
mHotelSystemManager.setTimerSwitchInfo(true,true,3,3,10,30,null);
```

```
// Set the second set of scheduled power-offs. The type is manual setting. The
time is 11:30 on Sunday and Monday.
```

```
mHotelSystemManager.setTimerSwitchInfo(true,true,4,6,11,30,new
int[]{0,1});
```

```
// Turn off the first set of settings for scheduled power-off
mHotelSystemManager.setTimerSwitchInfo(true,false,3,0,0,0,null);
```

```
// Turn off the second set of settings for scheduled power on
mHotelSystemManager.setTimerSwitchInfo(false,false,1,0,0,0,null);
```

40. Get the runtime**Description**

Get the runtime.

Function Name

Int getUsageTime()

Parameters

Parameter	Category	Type	Description
	Output	int	Run time (in minutes)

41. Set the power-on delay switch**Description**

Set the power-on delay switch status.

Function Name

boolean setPowerOnDelayEnable(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off
	Output	boolean	True: setup succeed False: setup failed

42. Get the power-on delay switch**Description**

Get the power-on delay switch status.

Function Name

boolean getPowerOnDelayEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

43. Set the power-on delay time**Description**

Set the power-on delay time.

Function Name

boolean setPowerOnDelay(int value)

Parameters

Parameter	Category	Type	Description
value	Input	int	Power-on delay time, unit: second
	Output	boolean	True: setup succeed False: setup failed

44. Get the power-on delay time**Description**

Get the power-on delay time.

Function Name

int getPowerOnDelay()

Parameters

Parameter	Category	Type	Description
	Output	int	Power-on delay time, unit: second

45. Set sleep mode switch**Description**

Set the sleep mode switch status.

Function Name

void setFakePowerOff(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off

46. Get sleep mode switch**Description**

Get the sleep mode switch status.

Function Name

boolean getFakePowerOff()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

47. Set Default Home Page

Description

Set Default Home Page.

Function Name

void setDefaultLauncher(String pkgName)

Parameters

Parameter	Category	Type	Description
pkgName	Input	String	The package name of the application to be set as the default home page

48. Get Default Home Page

Description

Get the package name of the application currently set as the default home page.

Function Name

String getDefaultLauncher()

Parameters

Parameter	Category	Type	Description
	Output	String	The package name of the application to be set as the default home page

49. Set serial port ID

Description

Set serial port ID, unique ID.

Function Name

boolean setMonitorId(String monitorId)

Parameters

Parameter	Category	Type	Description
monitorId	Input	String	1~255
	Output	boolean	True: setup succeed False: setup failed

50. Get serial port ID

Description

Get serial port ID.

Function Name

String getMonitorId()

Parameters

Parameter	Category	Type	Description
	Output	String	1~255

51. Factory reset

Description

Reset all data and clear partition data.

Function Name

boolean restoreSystemSettings()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: setup succeed False: setup failed

52. Set HDMI CEC switch

Description

Set HDMI CEC switch status.

Function Name

boolean setCecEnable(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off
	Output	boolean	True: setup succeed False: setup failed

53. Get HDMI CEC switch

Description

Get HDMI CEC switch status.

Function Name

boolean getCecEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

54. Set the CEC standby control switch

Description

Set the CEC standby control switch status.

Function Name

boolean setCecAutoPowerOffEnable(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off
	Output	boolean	True: setup succeed False: setup failed

55. Get the CEC standby control switch

Description

Get the CEC standby control switch status.

Function Name

boolean getCecAutoPowerOffEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

56. Set the CEC power-on switch

Description

Set the CEC power-on switch status.

Function Name

boolean setCecAutoWakeUpEnable(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off
	Output	boolean	True: setup succeed False: setup failed

57. Get the CEC power-on switch

Description

Get the CEC power-on switch status.

Function Name

boolean getCecAutoWakeUpEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

58. Set time of No operation standby

Description

Set time of No Operation standby.

Function Name

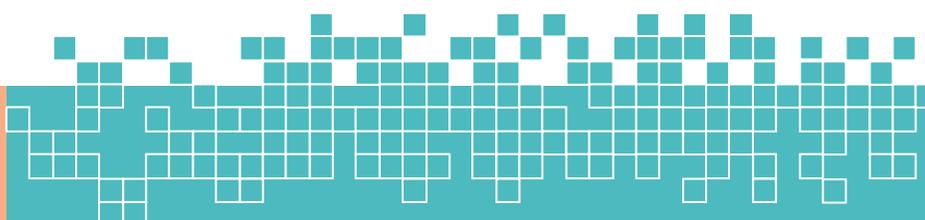
boolean setNoOperationStandbyTime(int time)

Parameters

Parameter	Category	Type	Description
time	Input	int	No operation standby time, unit: seconds
	Output	boolean	True: setup succeed False: setup failed

59. Get time of No operation standby

Description



Get time of No operation standby.

Function Name

int getNoOpreationStandbyTime()

Parameters

Parameter	Category	Type	Description
	Output	int	No operation standby time, unit: seconds

60. Set the over-temperature protection level 1 switch

Description

Set the over-temperature protection level 1 switch.

Function Name

void setFirstTempProtectFlag(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off When the temperature reaches 70 degrees Celsius, the brightness is automatically reduced by half

61. Get the over-temperature protection level 1 switch status

Description

Get the over-temperature protection level 1 switch status.

Function Name

boolean getFirstTempProtectFlag()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: swtich off

62. Set the over-temperature protection level 2 switch

Description

Set the over-temperature protection level 2 switch.

Function Name

void setSecondTempProtectFlag(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off When the temperature reaches 90 degrees Celsius, it will shut down.

63. Get the over-temperature protection level 2 switch status

Description

Get the over-temperature protection level 2 switch status.

Function Name

boolean getSecondTempProtectFlag()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

64. Set the over-temperature protection level 3 switch

Description

Set the over-temperature protection level 3 switch.

Function Name

void setThirdTempProtectFlag(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off Automatically shuts down when the

			temperature reaches 100 degrees Celsius
--	--	--	--

65. Get the over-temperature protection level 3 switch status

Description

Get the over-temperature protection level 3 switch status.

Function Name

boolean getThirdTempProtectFlag()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

Source API

Import Package: HotelSourceManager

1. Switch to the Specific Source

Description

Switch to the specific source

Function Name

Boolean setCurrentInputSource(int source,String packagename)

Parameters

Parameter	Category	Type	Description
packageName	Input	String	The packageName of the calling app
source	Input	int	HDMI1:0x501 HDMI2:0x502 AndroidSource:0xA02 DP:0x503 HDMI3:0x504
	Output	boolean	0: setup failed 1: setup successful

2. Get Current Input Source

Description

Get the current input source

Function Name

int getCurrentInputSource()

Parameters

Parameter	Category	Type	Description
	Output	int	HDMI1:0x501 HDMI2:0x502 AndroidSource:0xA02 DP:0x503 HDMI3:0x504

3. Set Boot Source

Description

Set startup input source

Function Name

Boolean setStartupInputSource(int source)

Parameters

Parameter	Category	Type	Description
source	Input	int	HDMI1:0x501 HDMI2:0x502 AndroidSource:0xA02 DP:0x503 HDMI3:0x504
	Output	boolean	0: setup failed 1: setup successful

4. Get Boot Source

Description

Get the boot source

Function Name

int getStartupInputSource ()

Parameters

Parameter	Category	Type	Description
	Output	int	HDMI1:0x501 HDMI2:0x502 AndroidSource:0xA02 DP:0x503

			HDMI3:0x504
--	--	--	-------------

5. Set Brightness

Description

Set the brightness, only available for the current source.

Function Name

Boolean setBrightness(int bright)

Parameters

Parameter	Category	Type	Description
bright	Input	int	0-100
	Output	boolean	0: setup failed 1: setup successful

6. Get Brightness

Description

Get the current brightness

Function Name

int getBrightness()

Parameters

Parameter	Category	Type	Description
	Output	int	0-100

7. Set Contrast

Description

Set the contrast, only available for the current source.

Function Name

Boolean setContrast(int contrast)

Parameters

Parameter	Category	Type	Description
contrast	Input	int	0-100
	Output	boolean	0: setup failed 1: setup successful

8. Get Contrast

Description

Get the current contrast**Function Name****int getContrast()****Parameters**

Parameter	Category	Type	Description
	Output	int	0-100

9. Set Backlight**Description**

Set backlight value

Function Name

Boolean setBackLightValue (int value)

Parameters

Parameter	Category	Type	Description
value	Input	int	0-100
	Output	boolean	0: setup failed 1: setup successful

10. Get Backlight**Description**

Get backlight value

Function Name

int getBackLightValue ()

Parameters

Parameter	Category	Type	Description
	Output	int	0-100

11. Set Saturation**Description**

Set saturation

Function Name

boolean setSaturation(int value)

Parameters

Parameter	Category	Type	Description
value	Input	int	0-100
	Output	boolean	0: setup failed

			1: setup successful
--	--	--	---------------------

12. Get Saturation

Description

Get saturation value

Function Name

int getSaturation()

Parameters

Parameter	Category	Type	Description
	Output	int	0-100

13. Set Source Window

Description

Set source window

Function Name

Boolean setSourceWindow(Rect rect)

Parameters

Parameter	Category	Type	Description
value	Input	Rect	4 values need to be entered: int left, int right, int top, int bottom Non-4K full screen is (0,0,1920,1080) 4K full screen is (0,0,3840,2160) 4 values need to be entered: int the distance to left of device, int the distance to top of device, int the distance to right of device, int the distance to bottom of device full screen is (0,0,0,0)
	Output	boolean	false: setup failed true: setup successful

14. Set Gamma Mode

Description

Set gamma mode

Function Name

boolean setGammaMode(int mode)

Parameters

Parameter	Category	Type	Description
mode	Input	int	0: standard 1: bias 2: darker
	Output	boolean	false: setup failed true: setup successful

15. Get Gamma Mode

Description

Get gamma mode

Function Name

int getGammaMode ()

Parameters

Parameter	Category	Type	Description
	Output	int	0: standard 1: bias 2: darker

16. Set Dynamic Contrast

Description

Switch on/off dynamic contrast

Function Name

boolean setDynamicContrast(boolean isEnabled)

Parameters

Parameter	Category	Type	Description
isEnabled	Input	boolean	false: off true: on
	Output	boolean	0: setup failed 1: setup successful

17. Get Dynamic Contrast

Description

Get the dynamic contrast state

Function Name

boolean getDynamicContrast ()

Parameters

Parameter	Category	Type	Description
	Output	boolean	false: off true: on

18. Set Color Correction

Description

Set the color correction parameters

Function Name

boolean setColorCorrection(int mode, int[] params)

Parameters

Parameter	Category	Type	Description
mode	Input	int	0: red 1: green 2: blue 3: cyan 4: yellow 5: purple 6: complexion
params	Input	int[]	The length is fixed at 3, which are hue, saturation and brightness respectively, and the range is -15 to 15
	Output	boolean	0: setup failed 1: setup successful

19. Get Color Correction

Description

Get the color correction parameters

Function Name

int[] getColorCorrection (int mode)

Parameters

Parameter	Category	Type	Description
mode	Input	int	0: red 1: green 2: blue 3: cyan 4: yellow 5: purple 6: complexion
	Output	int[]	The length is fixed at 3, which are hue, saturation and brightness respectively, and the range is -15 to 15

20. Set Failover switch

Description

Set Failover switch.

DP30FE do not support this feature.

Function Name

void setSourceBakEnable(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off

21. Get Failover switch

Description

Get Failover switch status.

DP30FE do not support this feature.

Function Name

boolean getSourceBakEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on

			False: switch off
--	--	--	-------------------

22. Set master channel of Failover

Description

Set master channel of Failover.

Function Name

```
void setSourceBakMasterChannel(int channel)
```

Parameters

Parameter	Category	Type	Description
channel	Input	int	0: HDMI 1 1: HDMI 2 3: DP

23. Get master channel of Failover

Description

Get master channel of Failover.

Function Name

```
int getSourceBakMasterChannel()
```

Parameters

Parameter	Category	Type	Description
	Output	int	0: HDMI 1 1: HDMI 2 3: DP

24. Set slave channel of Failover

Description

Set slave channel of Failover.

DP3oFE do not support this feature.

Function Name

```
void setSourceBakSlaveChannel(int channel)
```

Parameters

Parameter	Category	Type	Description
channel	Input	int	0: HDMI 1 1: HDMI 2

			3: DP
--	--	--	-------

25. Get slave channel of Failover

Description

Get slave channel of Failover.
DP30FE do not support this feature.

Function Name

int getSourceBakSlaveChannel()

Parameters

Parameter	Category	Type	Description
	Output	int	0: HDMI 1 1: HDMI 2 3: DP

26. Set the slave channel package name or id parameter

Description

Set the slave channel package name or id parameter.

Function Name

void setSourceBakChannelExtra(String extra)

Parameters

Parameter	Category	Type	Description
extra	Input	String	The APP package name that needs to be set

27. Get the slave channel package name or id parameter

Description

Get the slave channel package name or id parameter.

Function Name

String getSourceBakChannelExtra()

Parameters

Parameter	Category	Type	Description
	Output	String	The APP package name

28. Set the no-signal standby function switch

Description

Set the no-signal standby function switch.

Function Name

boolean setNoSignalStandbyEnable(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: switch on False: switch off
	Output	boolean	False: setup failed True: setup successful

29. Get the no-signal standby function switch**Description**

Set the no-signal standby function switch.

Function Name

boolean getNoSignalStandbyEnable()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: switch on False: switch off

30. Set time of no-signal standby**Description**

Set time of no-signal standby.

Function Name

boolean setNoSignalStandby(int value)

Parameters

Parameter	Category	Type	Description
value	Input	int	No signal standby time, unit: seconds
	Output	boolean	False: setup failed True: setup successful

31. Get time of no-signal standby**Description**

Get time of no-signal standby.

Function Name

int getNoSignalStandby()

Parameters

Parameter	Category	Type	Description
	Output	int	No signal standby time, unit: seconds

Network API

Import Package: HotelNetworkManager

32. Get Network Type

Description

Get current network type

Function Name

int getNetType()

Parameters

Parameter	Category	Type	Description
	Output	int	0: null 1: DHCP Ethernet 2: Static Ethernet 3: WiFi

33. Get IP Address

Description

Get the current IP

Function Name

String getIpAddress()

Parameters

Parameter	Category	Type	Description
	Output	String	Get the current IP address. If now is wired connection, get the wired IP. If now is wireless connection, get the wireless IP

34. Get Mac Address

Description

Get the current mac. When wifi is connected, get the wifi mac first. When wifi is not connected but is connected by wire, get the wired mac.

Function Name

String getMacAddress()

Parameters

Parameter	Category	Type	Description
	Output	String	mac

35. Wifi status

Description

Get wifi connection status

Function Name

boolean isWiFiConnected()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: connected False: not connected

36. Wired Status

Description

Get ethernet connection status

Function Name

boolean isEthernetNetworkConnected()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: connected False: not connected

37. Connect Wifi

Description

Connect wifi via: SSID&password

Function Name

```
void connectWifi(String ssid,String password,String,security)
```

Parameters

Parameter	Category	Type	Description
security	Input	String	Can only be the following specified String: “None” “WEP” “WPA_PSK” “WPA_EAP”

38. Auto Connect Wired network**Description**

DHCP obtain IP

Function Name

```
void autoConnectEthernet()
```

39. Wired network static connection Settings**Description**

set IP address、 subnet mask、 gateway、 DNS parameters manually

Function Name

```
Boolean manualConnectEthernet(String ip,String subnetMask,String gateway,String dns)
```

Parameters

Parameter	Category	Type	Description
ip	Input	String	Ip address
SubnetMask	Input	String	Subnet mask
gateway	Input	String	gateway
dns	Input	String	dns
	Output	boolean	False: The parameter is incorrect True: setup successful

40. Set Hotspot**Description**

Set hotspot

Function Name

boolean setWifiApState(boolean enable)

Parameters

Parameter	Category	Type	Description
enable	Input	boolean	True: turn on hotspot False: turn off hotspot
	Output	boolean	False: setup failed True: setup successful

41. Get Hotspot Status

Description

Get hotspot status

Function Name

boolean getWifiApState()

Parameters

Parameter	Category	Type	Description
	Output	boolean	True: turn on hotspot False: turn off hotspot