Android Solution

FAQs

Issue	**10**
Date	**2016-03-31**

HiSilicon Technologies Co., Ltd.

Address:	Huawei Industrial Base Bantian, Longgang Shenzhen 518129 People's Republic of China
Website:	http://www.hisilicon.com
Email:	support@hisilicon.com

About This Document

Purpose

This document describes the usage and debugging methods of some functions of the Android solution as well as precautions to be taken.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3716C	V2*XX*
Hi3719C	V1*XX*
Hi3718C	V1*XX*
Hi3719M	V1*XX*
Hi3718M	V1*XX*
Hi3798C	V1*XX*
Hi3796C	V1*XX*
Hi3798M	V1*XX*
Hi3796M	V1*XX*
HiSTBAndroid	V500R001
HiSTBAndroid	V600R001

Intended Audience

This document is intended for:

- Technical support personnel
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 10 (2016-03-31)

This issue is the tenth official release, which incorporates the following change:

Chapter 8 System

Section 8.4 is added.

Issue 09 (2015-01-26)

This issue is the ninth official release, which incorporates the following changes:

Chapter 6 Media Processing

Section 6.2.4 is added.

Chapter 7 Applications

Section 7.4 is added.

Issue 08 (2014-11-18)

This issue is the eighth official release, which incorporates the following changes:

Chapter 4 Peripherals

Section 4.2 is deleted.

Chapter 8 System

In section 8.3, descriptions of how to create the startup logo, fastplay, and Android boot animation images that support portrait orientation are added.

Issue 07 (2014-11-05)

This issue is the seventh official release, which incorporates the following change:

Chapter 6 Media Processing

Section 6.2.3 is added.

Issue 06 (2014-09-30)

This issue is the sixth official release, which incorporates the following changes:

Chapter 5 Browser

Section 5.1.4 is added.

Chapter 8 System

Section 8.3 is added.

Issue 05 (2014-08-14)

This issue is the fifth official release, which incorporates the following changes:

Chapter 3 Storage

Section 3.1.1 is added.

Chapter 5 Browser

Section 5.1.2 is added.

Chapter 7 Applications

Section 7.3 is updated and section 7.2 is deleted.

Modifications are made to change the supported chips from the Hi371*X* series to Hi379*X* series.

Issue 04 (2014-04-10)

This issue is the fourth official release, which incorporates the following change:

Chapter 6 Media Processing

Section 6.2.2 is added.

Issue 03 (2014-03-13)

This issue is the third official release, which incorporates the following changes:

Chapter 7 Applications

Section 7.3 and section 7.4 are added.

Some modifications are made to support Android 4.4 and Hi3716M V400.

Issue 02 (2014-02-20)

This issue is the second official release, which incorporates the following changes:

Chapters 1, 6, 7, and 8 are added.

Chapter 2 Network

Section 2.2 is added.

Some modifications are made to support Android 4.4 and Hi3716M V400.

Issue 01 (2013-11-25)

This issue is the first official release, which incorporates the following changes:

Chapter 3 and chapter 4 are added.

Issue 00B01 (2013-09-30)

This issue is the first draft release.

Contents

Figures

Tables

1 Android Development Environment

1.1 Android NDK

1.1.1 How Do I Select the Android Version Based on the Current Source Code Environment?

Problem Description

How do I select the Android version based on the current source code environment?

Solution

Table 1-1 describes the mapping between the Android version (until Android 4.4) and the native development kit (NDK).

Table 1-1 Mapping between the Android version and the NDK

NDK	Android	API	NDK EABI
n/a	1.0	1	arm-eabi-4.2.1
n/a	1.1	2	arm-eabi-4.2.1
1.5_r1	1.5	3	arm-eabi-4.2.1
1.6_r1	1.6	4	arm-eabi-4.2.1
r3	2.0	5	arm-eabi-4.4.0
	2.0.1	6	
	2.1	7	
r4	2.2	8	arm-eabi-4.4.0
r4b			
r5	2.3 2.3.3	9 10	arm-linux-androideabi-4.4.3
r5b			

NDK	Android	API	NDK EABI
r5c			
r6	3.0 3.1 3.2	11 12 13	arm-linux-androideabi-4.4.3
r6b			
r7	4.0 4.0.3	14 15	arm-linux-androideabi-4.4.3
r7b			
r7c			
r8	4.1 4.2	16 17	arm-linux-androideabi-4.4.3
r8b			arm-linux-androideabi-4.6
r8c			arm-linux-androideabi-4.6
r8d			arm-linux-androideabi-4.6
r8e			arm-linux-androideabi-4.6
r9	4.3	18	arm-linux-androideabi-4.6
r9b	4.4 4.4.x	19	arm-linux-androideabi-4.6
r9c			arm-linux-androideabi-4.6

Download the corresponding NDK from the following website according to the mapping in the preceding table:

http://dl.google.com/android/ndk/android-ndk-XX-linux-x86.tar.bz2

XX indicates the NDK version.

For example, to download NDK r5, visit the following link:

http://dl.google.com/android/ndk/android-ndk-r5-linux-x86.tar.bz2

1.1.2 How Do I Generate the EABI Tool Chain by Using the NDK?

Problem Description

Third-party developers need to use C/C++ code to generate dynamic linked libraries. What compiler can be used to compile dynamic linked libraries that can be used on the current Android version?

Solution

To generate the embedded application binary interface (EABI) tool chain by using the NDK, perform the following steps:

Step 1 Download the NDK of the current Android version based on section 1.1.1 "How Do I Select the Android Version Based on the Current Source Code Environment?"

Step 2 Untar the NDK by running the following command:

tar jxvf android-ndk-XX-linux-x86.tar.bz2

XX indicates the NDK version.

Step 3 Generate the EABI compilation tool chain.

```
build/tools/make-standalone-toolchain.sh --toolchain=AAA --
platform=android-BBB --install-dir=/tmp/my-android-toolchain
```

- *AAA* indicates the EABI compilation tool chain version. You can obtain the correct version based on Table 1-1.
- *BBB* indicates the Android API ID. You can obtain the correct ID based on Table 1-1.

----End

For example, to generate the EABI tool chain that can be used on Android 4.0, download NDK r7c from the following website:

http://dl.google.com/android/ndk/android-ndk-r7c-linux-x86.tar.bz2

Untar the NDK.

```
tar jxvf android-ndk-r7c-linux-x86.tar.bz2
cd android-ndk-r7c
./build/tools/make-standalone-toolchain.sh --toolchain=arm-linux-
androideabi-4.4.3 --platform=android-14 --install-dir=/tmp/my-android-
toolchain
```

2 Network

2.1 Wi-Fi

2.1.1 How Do I Set the Compilation Options of the Wi-Fi Module?

Problem Description

How do I set the compilation options of the Wi-Fi module during compilation?

Cause Analysis

The Android platform supports various Wi-Fi modules. You need to specify the type of the Wi-Fi module to be used before compilation. Otherwise, the Wi-Fi driver is not compiled into the image and the Wi-Fi function cannot be enabled.

Solution

Take Hi379*XX* V*XXX* as an example. Modify **BoardConfig.mk** in **device/hisilicon/Hi379XXVXXX/** before compilation to enable the compilation of the corresponding Wi-Fi driver based on the Wi-Fi module to be used.

For example:

- To enable the compilation of the driver corresponding to RTL8188EUS, modify the file as follows:
  ```
  BOARD_WLAN_DEVICE_RTL8188EUS = y
  ```
- To enable the compilation of the driver corresponding to RTL8188ETV, modify the file as follows:
  ```
  BOARD_WLAN_DEVICE_RTL8188ETV = y
  ```
- To enable the compilation of the driver corresponding to AR9374, modify the file as follows:
  ```
  BOARD_WLAN_DEVICE_AR9374 = y
  ```

You can enable the compilation of multiple Wi-Fi modules at the same time.

2.1.2 How Do I Set the Country or Region for Using the Wi-Fi?

Problem Description

In regions outside China, the Wi-Fi module sometimes cannot connect to the access point (AP).

Cause Analysis

The Wi-Fi channels vary according to countries or regions. For example, channels 1 to 13 are used in China, and channels 1 to 11 are used in the USA. The country for using the Wi-Fi is set to China by default. If the products are delivered to countries or regions other than China, you need to set the country or region for using the products before compilation. Otherwise, the products may fail to connect to APs.

Solution

Set the country or region to which products are delivered before compilation. To be specific, change the country or region parameter value in the Wi-Fi driver parameters of **wifi.c** in **hardware/ libhardware_legacy/wifi/**.

For example:

- To set the country to the USA for RTL8188EUS/RTL8188ETV, change the parameter value as follows:

```
#define DRIVER_MODULE_RTL8188EUS    2, \
{ \
    {"cfg80211","/system/lib/modules/cfg80211.ko","","cfg80211 "}, \
    {"rtl8188eu","/system/lib/modules/rtl8188eu.ko","ifname=wlan0
if2name=p2p0 rtw_channel_plan=0x22","rtl8188eu "} \
}
```

- To set the country to the USA for AR9374, change the parameter value as follows:

```
#define DRIVER_MODULE_AR9374    2, \
{ \

{"cfg80211_ath6k","/system/lib/modules/cfg80211_ath6k.ko","","cfg8021
1_ath6k "}, \
    {"ath6kl_usb","/system/lib/modules/ath6kl_usb.ko","
reg_domain=0x8348 ath6kl_p2p=0x13","ath6kl_usb "} \
}
```

Contact the Wi-Fi vendor to obtain the parameter value corresponding to the specific country or region.

2.1.3 What Do I Do If the Wi-Fi Module Is Unavailable After Being Woken Up from the Standby Mode?

Problem Description

After the Wi-Fi module is woken up from the standby mode, it cannot connect to any AP or detect APs. In addition, the board cannot be woken up.

Cause Analysis

The standby modes include cut-power mode and deep-sleep mode. In cut-power mode, the Wi-Fi module is power off; in deep-sleep mode, the power supply is required. The supported standby mode varies according to Wi-Fi devices. Some Wi-Fi devices select the deep-sleep mode by default. When the demo board enters the standby mode, the USB port is power off. Therefore, if the Wi-Fi module is woken up from the deep-sleep standby mode, it cannot work properly.

Solutions

Disable the Wi-Fi module before standby and enable it again after wakeup. This solution is used in the current version.

2.2 Ethernet

2.2.1 How Do I Disable the Wi-Fi Disguise Function?

Problem Description

After the system boots, it needs to connect to the network using the wired network or PPPoE. When the API in the Android inherent Connectivity Manager is called to query the current available network type, TYPE_WIFI is found but not TPYE_ETHERNET or TYPE_PPPOE.

Analysis

This issue occurs when the new Wi-Fi disguise function is enabled.

Solution

The Wi-Fi disguise function is added to improve application compatibility so that the applications that run only in the Wi-Fi environment can run in the wired network or PPPoE.

The solutions are provided based on two circumstances:

- The real network status is required for self-developed applications.

  In this case, you can call the API for disabling the Wi-Fi disguise function in the Ethernet Manager before querying the network status, and call the API for enabling the Wi-Fi disguise function in the Ethernet Manager after querying the network status. The reference code is as follows (note that only the logic part is listed):

  ```
  mEthernetManager.setWifiDisguise(false);
  NetworkInfo tempInfo = mConnectivityManager.getActiveNetworkInfo();
  ```

```
mEthernetManager.setWifiDisguise(true);
```

- The Wi-Fi disguise function is not required.

  In this case, you need to comment out the implementation of APIs for setting and querying the Wi-Fi disguise function by modifying the related code in the EthernetService at the frameworks layer of the network. The reference code is as follows:

  Code directory:

  `frameworks/base/services/java/com/android/server/EthernetService.java`

  Original code:

```
    public void setWifiDisguise(boolean setEnable) {
        if(DEBUG) Log.i(TAG, "setWifiDisguise: " + setEnable);
        final ContentResolver cr = mContext.getContentResolver();
        Settings.Secure.putInt(cr, Settings.Secure.WIFI_DISGUISE,
                setEnable ? EthernetManager.WIFI_DISGUISE_ENABLED :
EthernetManager.WIFI_DISGUISE_DISABLED);
    }

    public int getWifiDisguiseState() {
        final ContentResolver cr = mContext.getContentResolver();
        try {
            if(DEBUG) Log.i(TAG, "getWifiDisguiseState: "
                    + Settings.Secure.getInt(cr,
Settings.Secure.WIFI_DISGUISE));
            return Settings.Secure.getInt(cr,
Settings.Secure.WIFI_DISGUISE);
        } catch (Settings.SettingNotFoundException e) {
            if(DEBUG) Log.i(TAG, "getWifiDisguiseState: STATE
UNKNOWN");
            return EthernetManager.WIFI_DISGUISE_STATE_UNKNOWN;
        }
    }
```

  Code after modification:

```
    public void setWifiDisguise(boolean setEnable) {
/*
//comment out these codes to disable WifiDisguise
        if(DEBUG) Log.i(TAG, "setWifiDisguise: " + setEnable);
        final ContentResolver cr = mContext.getContentResolver();
        Settings.Secure.putInt(cr, Settings.Secure.WIFI_DISGUISE,
                setEnable ? EthernetManager.WIFI_DISGUISE_ENABLED :
EthernetManager.WIFI_DISGUISE_DISABLED);
//comment out these codes to disable WifiDisguise
*/
    }
    public int getWifiDisguiseState() {
/*
//comment out these codes to disable WifiDisguise
        final ContentResolver cr = mContext.getContentResolver();
        try {
            if(DEBUG) Log.i(TAG, "getWifiDisguiseState: "
                    + Settings.Secure.getInt(cr,
Settings.Secure.WIFI_DISGUISE));
            return Settings.Secure.getInt(cr,
Settings.Secure.WIFI_DISGUISE);
        } catch (Settings.SettingNotFoundException e) {
            if(DEBUG) Log.i(TAG, "getWifiDisguiseState: STATE
UNKNOWN");
            return EthernetManager.WIFI_DISGUISE_STATE_UNKNOWN;
        }
//comment out these codes to disable WifiDisguise
*/
//always return EthernetManager.WIFI_DISGUISE_DISABLED;
        return EthernetManager.WIFI_DISGUISE_DISABLED;
    }
```

3 Storage

3.1 Vold Modification

3.1.1 How Do I Modify the Vold Mounting?

Problem Description

How do I modify the software when I use other external storage media (for example, USB flash drive and SD card) as a default storage device?

Cause Analysis

The Android system uses the built-in flash drive as the default storage device. However, the flash capacity is insufficient in some hardware solutions, and external storage media must be used as a default storage device. In this case, you need to modify the code.

Solution

Identify the bus where the external storage media is located and modify **vold.fstab**. For example, to mount the device in **/sys/devices/platform/hiusb-ehci.0/usb1/1-0:1.0** as a default storage device, perform the following steps:

Step 1 Mask the device mounted to **/mnt/sdcard** in the [*XXX*]**.fstab** file.

Step 2 Mount the new device as a default storage device.

```
dev_mount  block  /mnt/sdcard  auto  /devices/platform/hiusb-
ehci.0/usb1/1-0:1.0
```

----End

3.1.2 How Do I Modify Vold Mounting When the Virtual SD Card Partition in the UBI File System Changes?

Problem Description

The default flash partition is the fourth partition in the UBI file system. How do I modify the default storage device when several UBI partitions are added in front of the default flash partition.

Solution

Check whether the partitions are added in front of or behind the SD card partition.

- If the partitions are added behind the SD card partition, no modification is required.
- If the partitions are added in front of the SD card partition, check the number of the SD card partition in the UBI file system. Provided that the SD card partition is the fourth partition in the UBI file system and one partition is added in front of the SD card partition, modification is required. (Only the value of **Ubifs=** is changed.)

  Before modification:

  ```
  dev_mount  block  /mnt/sdcard  auto
  /devices/virtual/mtd/mtd13/mtdblock13  Ubifs=ubi3_0
  ```

  After modification:

  ```
  dev_mount  block  /mnt/sdcard  auto
  /devices/virtual/mtd/mtd13/mtdblock13  Ubifs=ubi4_0
  ```

3.1.3 How Do I Virtualize a Peripheral Mounted Over a Fixed Bus As a SATA Device?

Problem Description

Some boards do not have the SATA bus. In this case, how do I display the SATA device on the UI?

Solution

Display the device mounted over a fixed USB bus as a SATA device. For example, display the USB device mounted over the USB bus 1-1:1.0 as a SATA disk.

Modify the functions in **DirectVolume.cpp** and **PartVolume.cpp** of the **system/vold** directory as follows:

Before modification:

```
int PartVolume::addPath(const char *path) {
    mPaths->push_back(strdup(path));
     if ( strstr( path, "mmc") ) {
         mDevType = DEV_TYPE_SDCARD;
     } else if ( strstr( path, "ehci") || strstr( path, "ohci")) {
            mDevType = DEV_TYPE_USB2;
```

```
        } else if ( strstr( path, "usb3.0")) {
            mDevType = DEV_TYPE_USB3;
        } else if ( strstr( path, "ahci") ) {
            mDevType = DEV_TYPE_SATA;
        } else if ( strstr( path, "mtd") ) {
            mDevType = DEV_TYPE_FLASH;
        } else {
            mDevType = DEV_TYPE_UNKOWN;
        }
...
...
...
```

After modification:

```
int PartVolume::addPath(const char *path) {
    mPaths->push_back(strdup(path));
     if ( strstr( path, "mmc") ) {
         mDevType = DEV_TYPE_SDCARD;
     } else if ( strstr( path, "ehci") || strstr( path, "ohci")) {
        if (strstr( path, "1-1:1.0")) {
             mDevType = DEV_TYPE_SATA;
         } else {
             mDevType = DEV_TYPE_USB2;
         }
    } else if ( strstr( path, "usb3.0")) {
         mDevType = DEV_TYPE_USB3;
    } else if ( strstr( path, "ahci") ) {
        mDevType = DEV_TYPE_SATA;
    } else if ( strstr( path, "mtd") ) {
        mDevType = DEV_TYPE_FLASH;
    } else {
        mDevType = DEV_TYPE_UNKOWN;
}
...
```

3.2 eMMC Usage

3.2.1 How Do I Compile the System Image for the eMMC?

Problem Description

How do I compile the system image when the eMMC is used as the board-level storage media?

Solution

To compile the system image for the eMMC (for example, on the Hi379XX VXXX), perform the following steps:

Step 1 Set the size of the system, userdata, cache, and sdcard partitions for the eMMC in **BoardConfig.mk**.

```
BOARD_SYSTEMIMAGE_PARTITION_SIZE := 524288000
BOARD_USERDATAIMAGE_PARTITION_SIZE := 1073741824
BOARD_CACHEIMAGE_PARTITION_SIZE := 104857600
BOARD_CACHEIMAGE_FILE_SYSTEM_TYPE := ext4
BOARD_SDCARDIMAGE_PARTITION_SIZE := 1572864000
```

NOTE

The size of the ext4 partition must be entered when you create the ext4 image by using the tool inherent in the Android system. Therefore, this step is mandatory. The unit is byte. For example, provided that the size of the system partition is 500 MB, the configuration is as follows:

BOARD_SYSTEMIMAGE_PARTITION_SIZE:=500*1024*1024 = 524288000

Step 2 Configure bootargs.

```
bootcmd=mmc read 0 0x1FFFFC0 0x4B000 0x5000; bootm 0x1FFFFC0
bootargs=mem=1G console=ttyAMA0,115200
blkdevparts=mmcblk0:1M(fastboot),1M(bootargs),10M(recovery),2M(deviceinfo
),8M(baseparam),8M(pqparam),20M(logo),20M(logobak),40M(fastplay),40M(fast
playbak),40M(kernel),20M(misc),500M(system),1024M(userdata),100M(cache),-
(sdcard)
```

- The **blkdevparts=mmcblk0** field is mandatory.
- bootcmd=mmc read 0 0x1FFFFC0 0x4B000 0x5000; bootm 0x1FFFFC0

  The preceding command is used to load the kernel.

  - **0x1FFFFC0** indicates the memory loading address of the kernel and does not need to be modified.
  - The start address and read length of the kernel image are 0x4B000 multiplied by 0x200 and 0x5000 multiplied by 0x200 respectively, for example:

    0x **4B00**: 0x4B000 * 0x200 = 157286400/1024/1024 MB = 150 MB

    0x**5000**: 0x5000 * 0x200 = 10485760 = 10 MB

  Provided that the kernel is located in the 80 MB position, and the read partition length is 10 MB, the start address of the kernel is 0x28000 (80 * 1024 * 1024/512), and the read partition length is 0x5000 (10 * 1024 * 1024/512). The bootcm is as follows:

  ```
  bootcmd=mmc read 0 0x1FFFFC0 0x28000 0x5000; bootm 0x1FFFFC0
  ```

Step 3 Run the **make bigfish** or **make ext4fs** command in the **Project** root directory. The ext4 image is generated in the eMMC folder under the **/out** directory.

----End

4 Peripherals

4.1 Front Panel

4.1.1 How Do I Modify the Front Panel Key Value Mapping?

Problem Description

How do I change the function of a key into another function? For example, how do I change the KEY_SETUP key into the POWER key?

Solution

Modify the file **device/hisilicon/bigfish/system/frontpanel/key_pars/frontPanel.xml**.

```
<frontPanel_xml>
   <hisi-key>
      <key value="0x3"  name="KEY_UP"        />     <!--key up-->
      <key value="0x4"  name="KEY_DOWN"      />    <!--key down-->
      <key value="0x2"  name="KEY_LEFT"      />     <!--key left -->
      <key value="0x5"  name="KEY_RIGHT"     />     <!--key right-->
      <key value="0x1"  name="KEY_ENTER"     />     <!--key ok -->
      <key value="0x6"  name="KEY_BACK"      />     <!--Back -->
      <key value="0x0"  name="KEY_SETUP"     />     <!--Setup-->
      <key value="0x8"  name="KEY_HOME"      />     <!--Home-->
   </hisi-key>
</frontPanel_xml>
```

where:

- The values in red are the key values obtained by underlying Linux drivers.
- The values in blue are the mapped key values.

You can modify the values as required. If you want to define other key values, see the Linux_KeyCode_Ary data structure in **device/hisilicon/bigfish/system/frontpanel/key_pars/linux_key.h**.

For example, to change the KEY_SETUP key to the POWER key, do as follows:

```
<frontPanel_xml>
    <hisi-key>
        <key value="0x3"  name="KEY_UP"       />     <!--key up-->
        <key value="0x4"  name="KEY_DOWN"     />    <!--key down-->
        <key value="0x2"  name="KEY_LEFT"     />     <!--key left -->
        <key value="0x5"  name="KEY_RIGHT"    />     <!--key right-->
        <key value="0x1"  name="KEY_ENTER"    />     <!--key ok -->
        <key value="0x6"  name="KEY_BACK"     />     <!--Back -->
        <key value="0x0"  name="KEY_POWER"    />     <!--Power-->
        <key value="0x8"  name="KEY_HOME"     />     <!--Home-->
    </hisi-key>
</frontPanel_xml>
```

4.1.2 How Do I Modify the Corresponding Software If the Front Panel Hardware Is Changed?

Problem Description

If the customer uses a front panel from other vendors, or uses some random GPIO pins as keys, how do I modify the corresponding software?

Solution

As modification is customized based on scenarios to resolve this kind of issues, details are not provided here. You can add the chip driver in the **device/hisilicon/bigfish/system/frontpanel/driver_interface/** directory, and re-implement the key_get() and key_init() functions.

4.1.3 How Do I Modify the Corresponding Software If the Front Panel Chip Is Not CT1642?

Problem Description

Currently HiSilicon provides the following five front panel chips:

- 74HC164
- PT6961
- CT1642
- PT6964
- FD650

CT1642 is supported by default. If one of the other four chips is used, how do I modify the corresponding software?

Solution

There is a macro definition in **device/hisilicon/bigfish/system/frontpanel/driver_interface/keyled.h**.

```
/********************************************************
 *
 *   KEYLED_TYPE_DEF = 0:  74HC164
 *   KEYLED_TYPE_DEF = 1:  PT6961
 *   KEYLED_TYPE_DEF = 2:  CT1642
 *   KEYLED_TYPE_DEF = 3:  PT6964
 *   KEYLED_TYPE_DEF = 4:  FD650
 *
 ********************************************************/
#define  KEYLED_TYPE_DEF   2
```

If the front panel chip other than CT1642 is used, you only need to redefine **KEYLED_TYPE_DEF**.

For example, if 74HC164 is used, change **#define KEYLED_TYPE_DEF 2** to **#define KEYLED_TYPE_DEF 0**.

5 Browser

5.1 GUI Display

5.1.1 What Do I Do If Garbled Characters Are Displayed on Some Web Pages?

Problem Description

What do I do if garbled characters are displayed on some web pages?

Analysis

The Android platform supports various web page encoding modes, such as UTF8, GBK, and Latin. If the encoding mode of the web page source file matches the specified encoding mode of the web page and is correct, no error occurs.

However, if the encoding mode of the web page source file does not match the specified encoding mode of the web page or is incorrect, the browser displays texts based on the specified encoding type or default encoding type, and garbled characters are displayed.

Solution

A configuration option is provided in the browser dedicated for manually setting the encoding type of the web page. When the browser is used in a specific language area, you can manually set the encoding type to the corresponding encoding type to improve compatibility with characters in the specific language.

For example, if the browser is used in an area that uses Simplified Chinese, you can set the option to GBK.

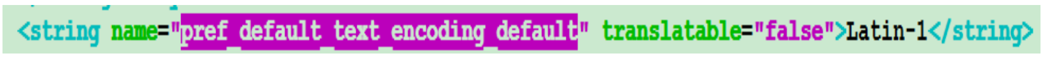
- For developers, you can modify the option in the specific configuration file as follows:

Step 1 Go to the source code directory **packages/apps/Browser** of the browser.

Step 2 Open the file **res/values/strings.xml**.

Step 3 Search for the option "pref_default_text_encoding_default".

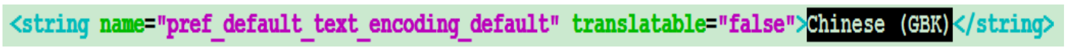
Figure 5-1 Searching for the option



```
<string name="pref_default_text_encoding_default" translatable="false">Latin-1</string>
```

Step 4 Modify the option as required.

Figure 5-2 Modifying the option



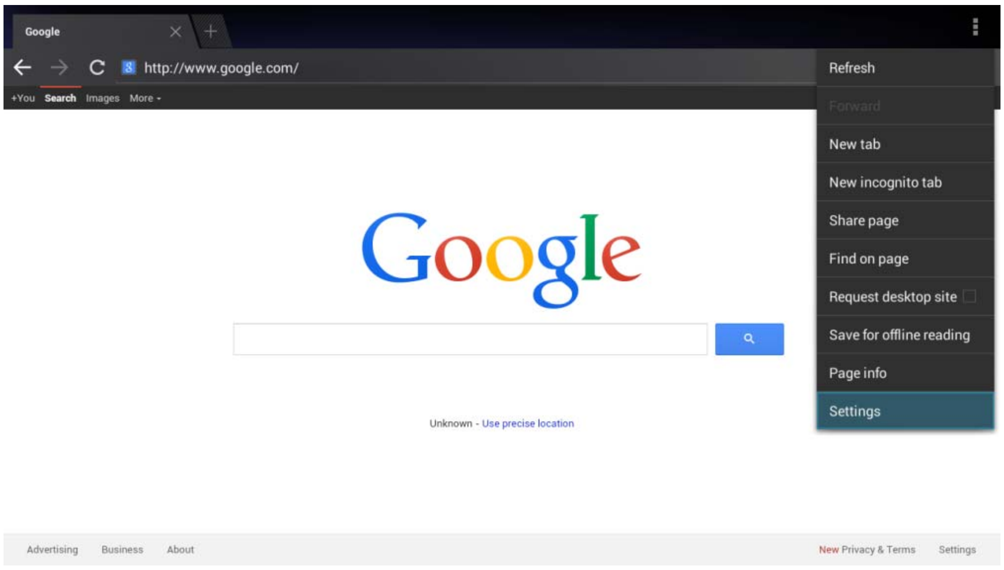
```
<string name="pref_default_text_encoding_default" translatable="false">Chinese (GBK)</string>
```

----End

- For users, you can modify the option in the specific setting GUI as follows:

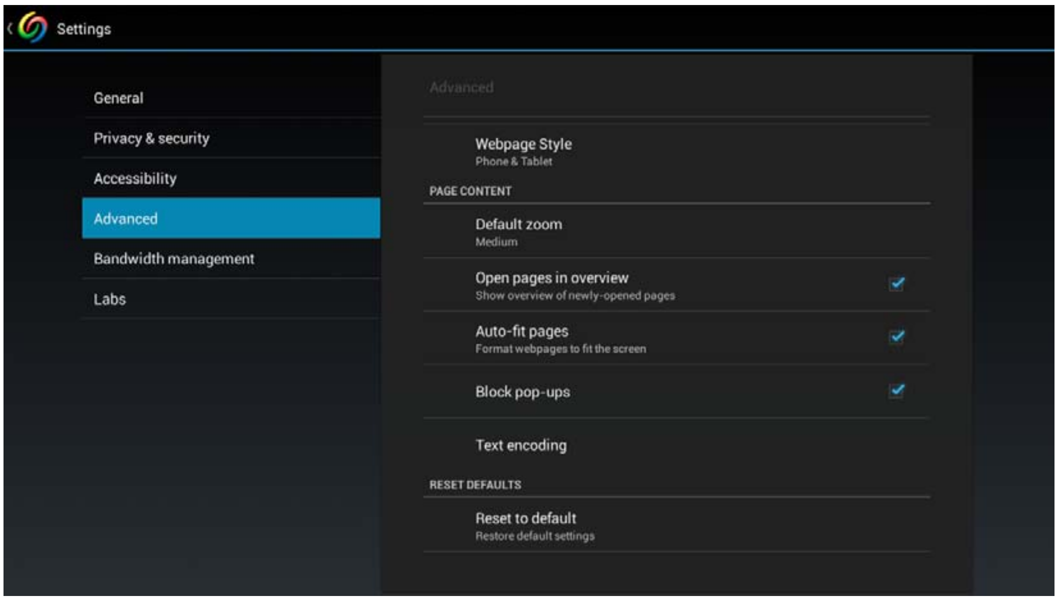
Step 1 Choose **Settings**.

Figure 5-3 Choosing Settings



Step 2 Click **Advanced**.

Figure 5-4 Advanced settings



Settings

General
Privacy & security
Accessibility
Advanced
Bandwidth management
Labs

Advanced

Webpage Style
Phone & Tablet

PAGE CONTENT

Default zoom
Medium

Open pages in overview ✓
Show overview of newly-opened pages

Auto-fit pages ✓
Format webpages to fit the screen

Block pop-ups ✓

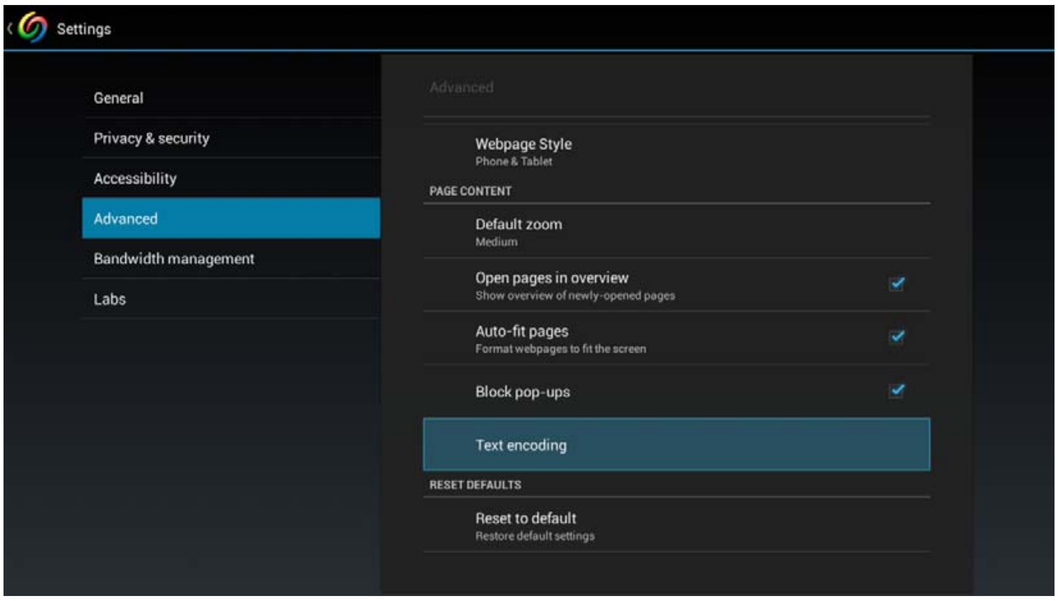
Text encoding

RESET DEFAULTS

Reset to default
Restore default settings

Step 3 Click **Text encoding**.

Figure 5-5 Clicking Text encoding



Settings

General
Privacy & security
Accessibility
Advanced
Bandwidth management
Labs

Advanced

Webpage Style
Phone & Tablet

PAGE CONTENT

Default zoom
Medium

Open pages in overview ✓
Show overview of newly-opened pages

Auto-fit pages ✓
Format webpages to fit the screen

Block pop-ups ✓

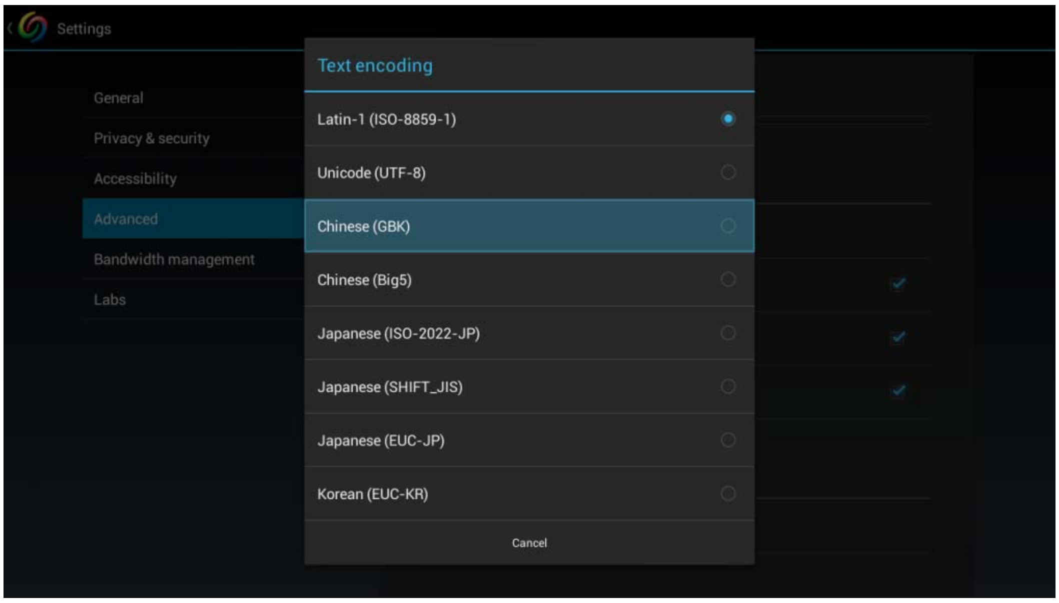
Text encoding

RESET DEFAULTS

Reset to default
Restore default settings

Step 4 Select the encoding format as required.

Figure 5-6 Selecting the encoding format



Latin-1 (ISO-8859-1) ●
Unicode (UTF-8)
Chinese (GBK)
Chinese (Big5)
Japanese (ISO-2022-JP)
Japanese (SHIFT_JIS)
Japanese (EUC-JP)
Korean (EUC-KR)
Cancel

----End

5.1.2 How Do I Enter the URL When the Address Bar Is Not Displayed?

NOTE

This section is dedicated for Android 4.4 and later versions.

Problem Description

The browser displays web pages in full-screen mode, and the address bar is not displayed by default. In this case, how do I enter the URL?

Cause Analysis

The GUI is optimized to bring better experience of using the TV remote control. That is, the web pages are displayed in full-screen mode, and the address bar is not displayed by default.

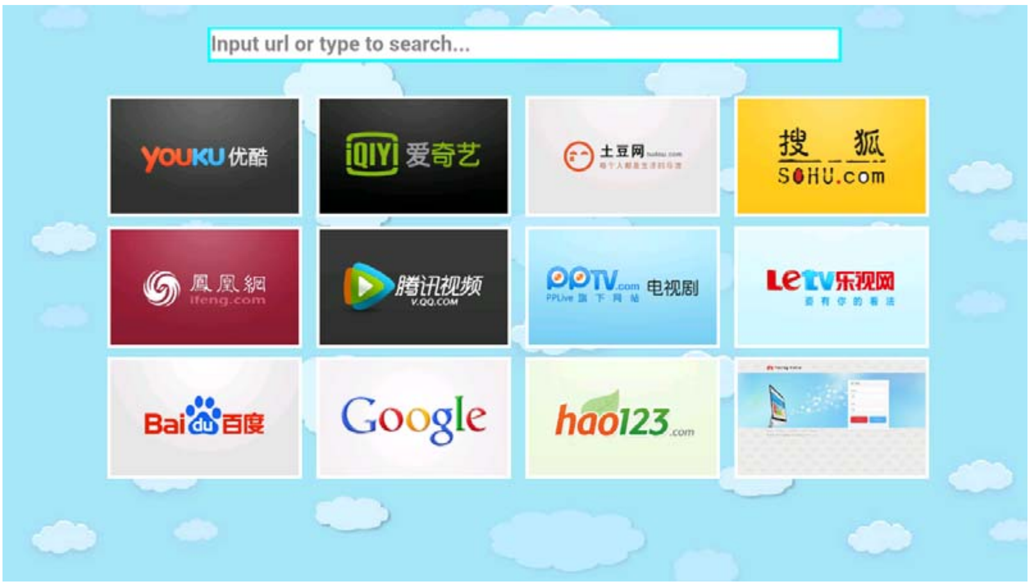
Solution

The browser provides an easy-to-use control panel for users to perform frequently used operations, including entering the URL, redirecting to the home page, setting entries, adding bookmarks, going forward/backward, and viewing history records. There are two methods for entering the URL.

Method 1:

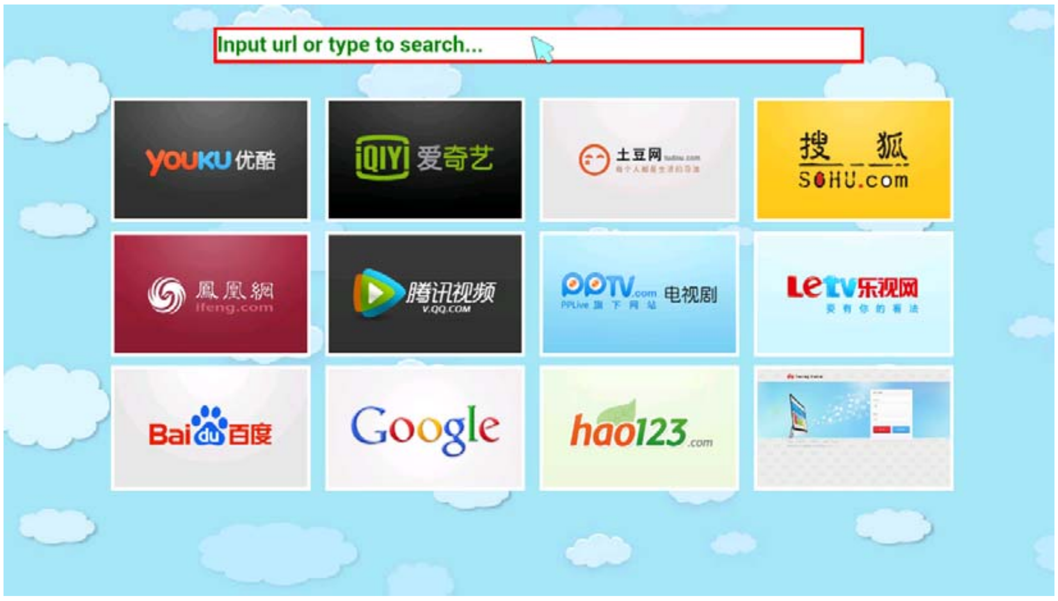
Step 1 Open the browser. The home page is displayed by default, as shown in Figure 5-7.

Figure 5-7 Home page



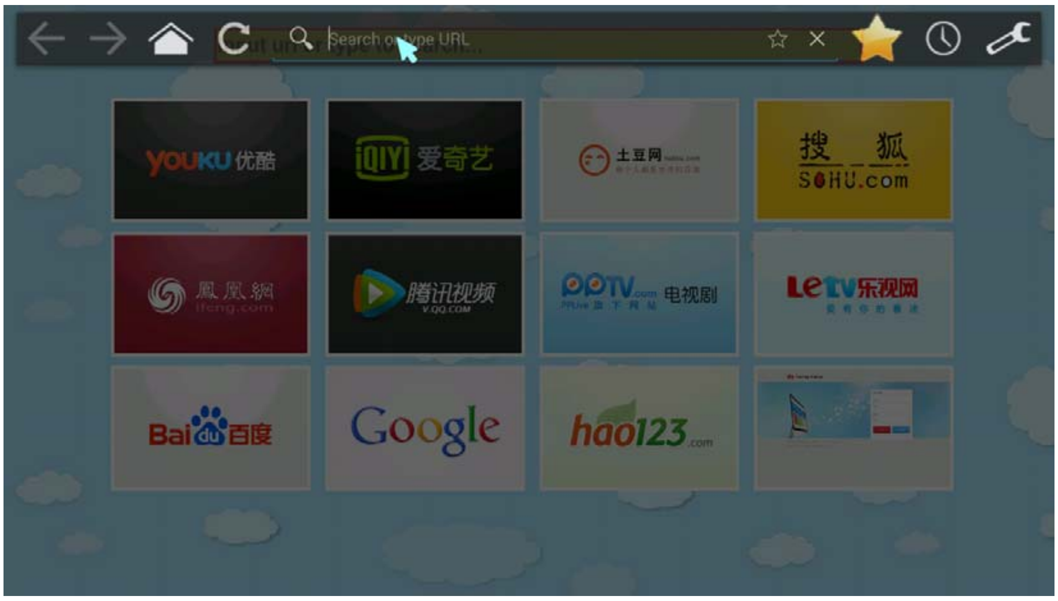
Step 2 Move the pointer over the red rectangle area shown in Figure 5-8.

Figure 5-8 Moving the pointer over the text box



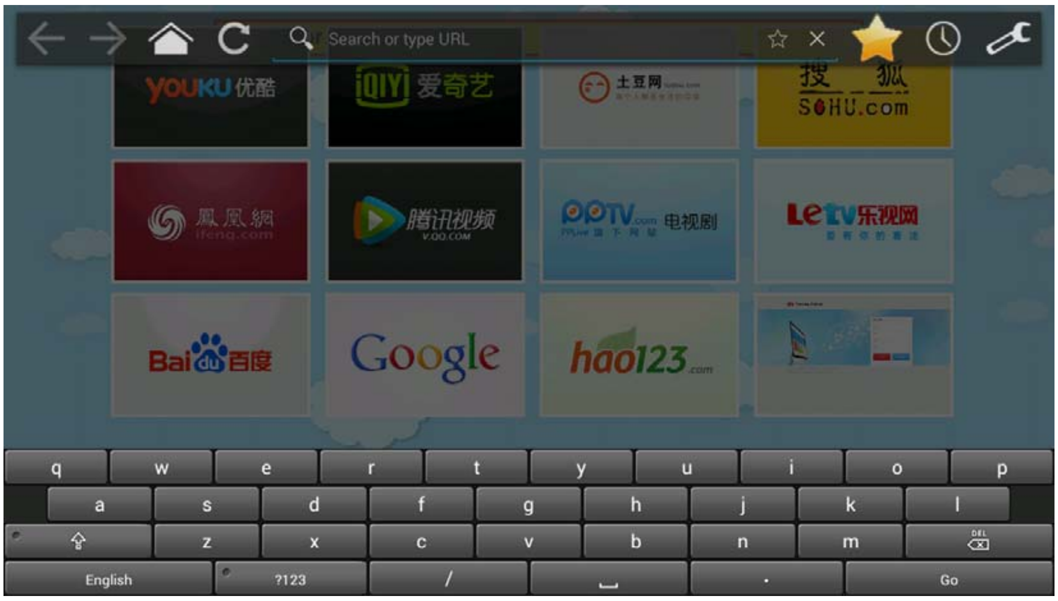
Step 3 Press **OK**. The control panel is displayed, as shown in Figure 5-9.

Figure 5-9 Displaying the control panel by pressing the OK button



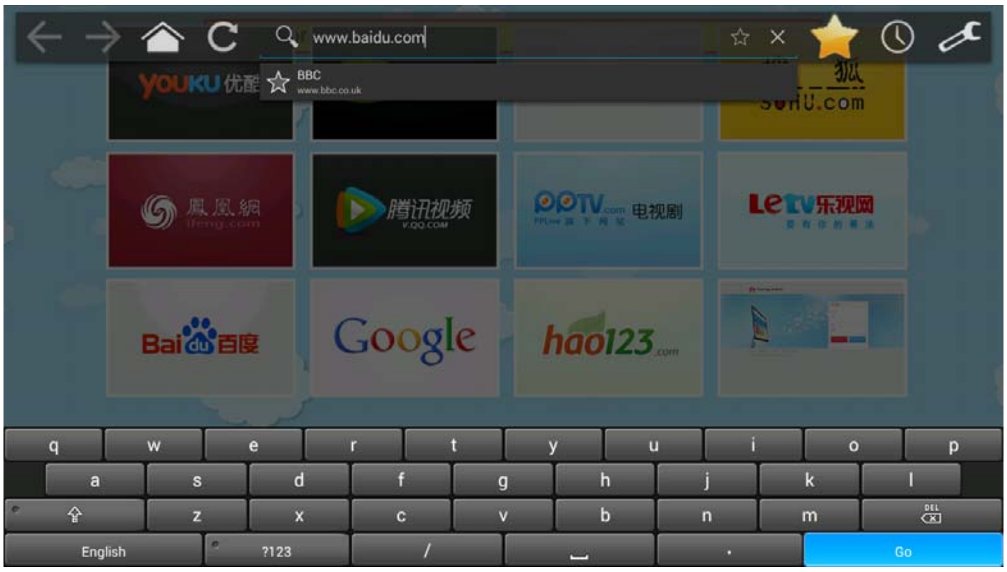
Step 4 Move the cursor over the address bar, and press **OK**. The virtual keyboard is displayed, as shown in Figure 5-10.

Figure 5-10 Calling the input method



Step 5 Enter the URL, move the cursor over **Go**, and press **OK**, as shown in Figure 5-11.

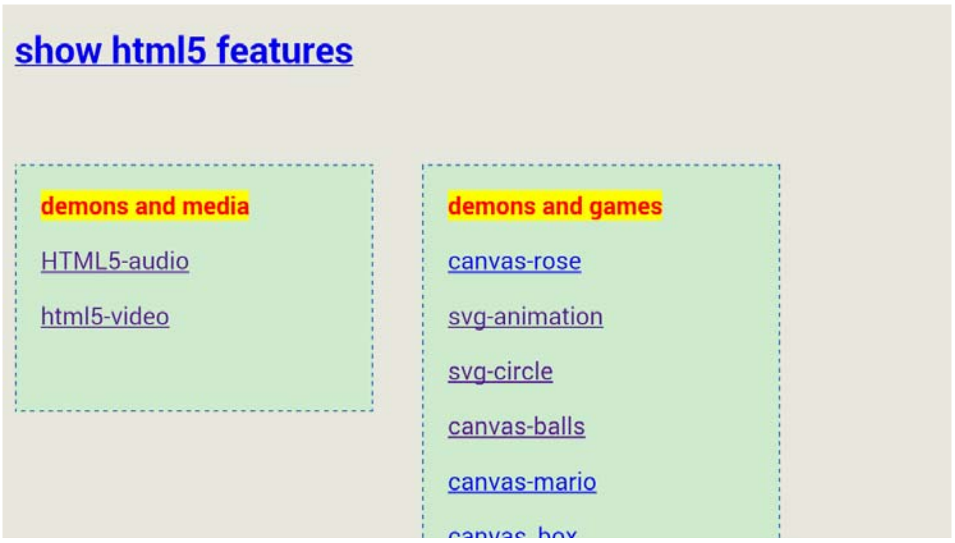
Figure 5-11 Entering the URL



----End

Method 2 (when a website is opened, as shown in Figure 5-12):

Figure 5-12 Website

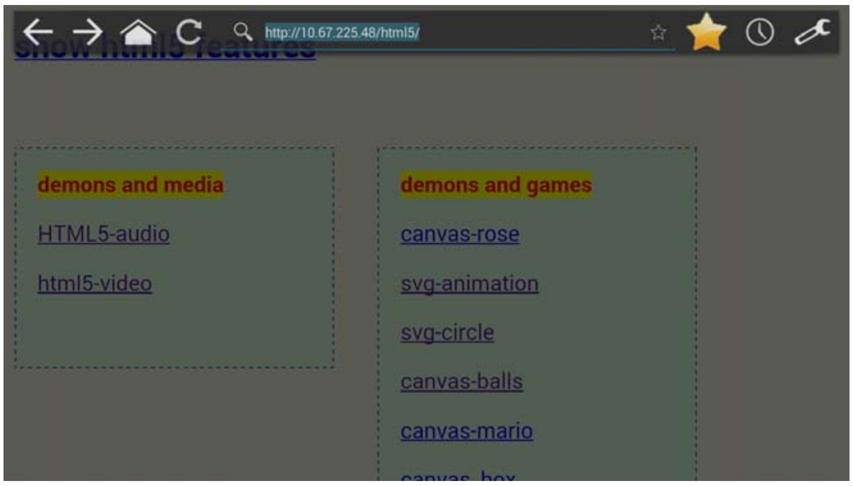

show html5 features

demons and media
HTML5-audio
html5-video

demons and games
canvas-rose
svg-animation
svg-circle
canvas-balls
canvas-mario

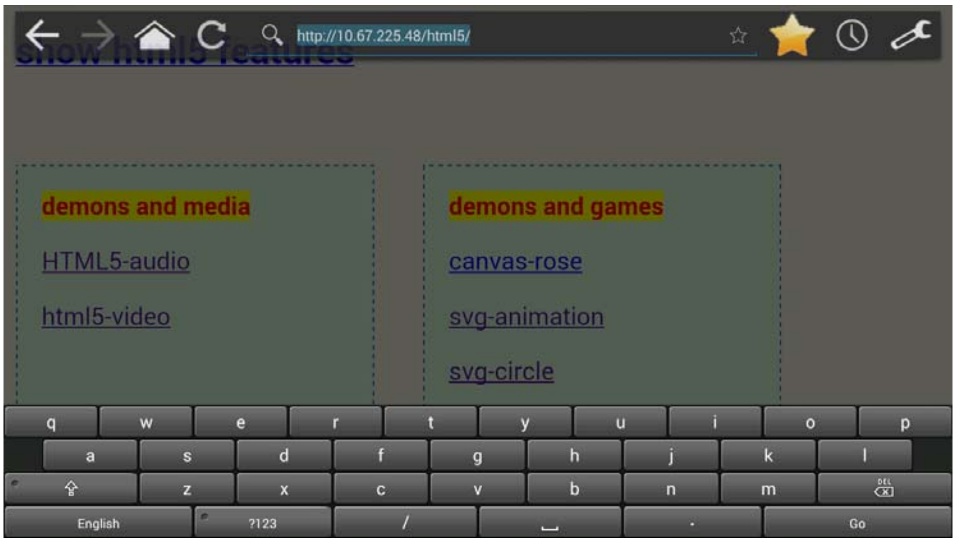
Step 1 Press **MENU**. The control panel is displayed, as shown in Figure 5-13.

Figure 5-13 Displaying the control panel by pressing the menu button



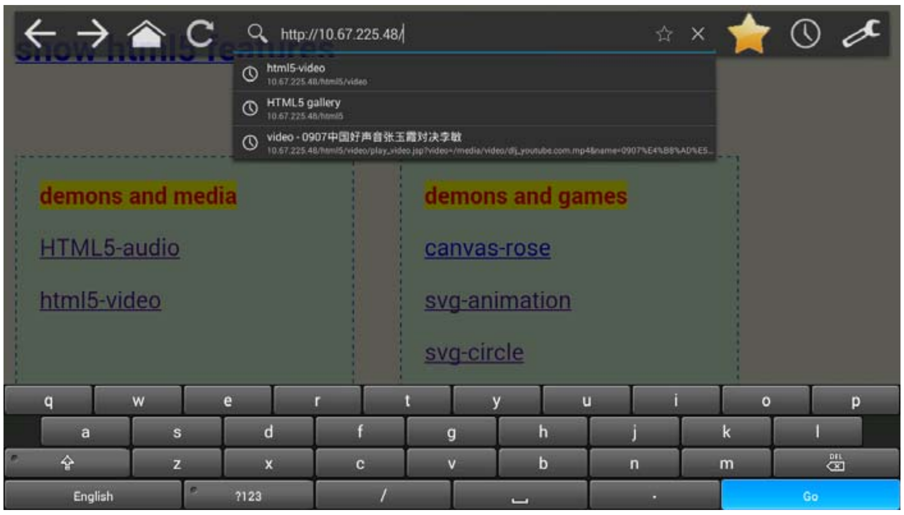
Step 2 Move the pointer over the address bar, and press **OK**. The virtual keyboard is displayed, as shown in Figure 5-14.

Figure 5-14 Calling the input method



Step 3 Enter the URL, move the pointer over **Go**, and press **OK**, as shown in Figure 5-15.

Figure 5-15 Entering the URL



----End

5.1.3 How Do I Replace the Navigation Websites on the Home Page?

NOTE

This section is dedicated for Android 4.4 and later versions.

Problem Description

There are 12 navigation links on the home page of the browser. How do I replace those navigation websites?

Cause Analysis

Currently you need to modify the corresponding resource file in the code to replace the navigation websites.

Solution

Do as follows:

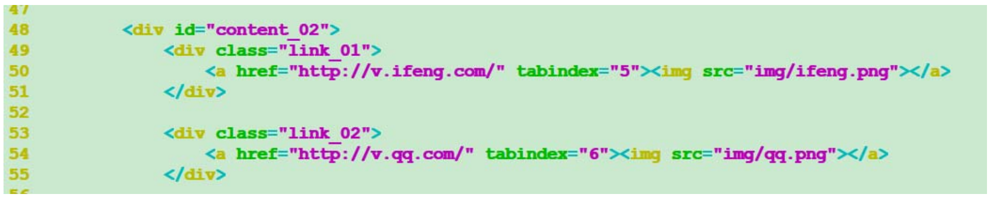
Step 1 Go to the directory for storing the browser code.

```
packages/apps/Browser
```

Step 2 Find the code and resources to be replaced (for example, Tencent Video).

```
assets/HiBrow/LocalNavigation/navigation.html
```

Figure 5-16 Code before modification



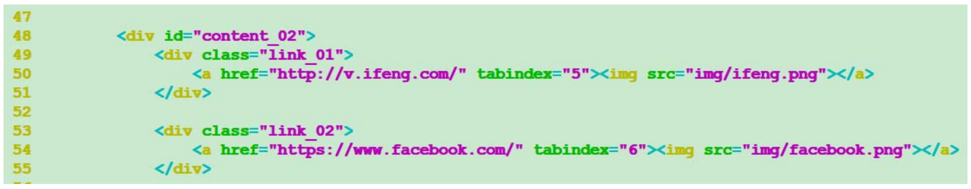
```
        <div id="content_02">
            <div class="link_01">
                <a href="http://v.ifeng.com/" tabindex="5"><img src="img/ifeng.png"></a>
            </div>

            <div class="link_02">
                <a href="http://v.qq.com/" tabindex="6"><img src="img/qq.png"></a>
            </div>
```

`assets/HiBrow/LocalNavigation/img/qq.png`

Step 3 Replace the corresponding code and resources (for example, replace Tencent Videos with Facebook).

`assets/HiBrow/LocalNavigation/navigation.html`

Figure 5-17 Code after modification



```
        <div id="content_02">
            <div class="link_01">
                <a href="http://v.ifeng.com/" tabindex="5"><img src="img/ifeng.png"></a>
            </div>

            <div class="link_02">
                <a href="https://www.facebook.com/" tabindex="6"><img src="img/facebook.png"></a>
            </div>
```

Add the following file:

`assets/HiBrow/LocalNavigation/img/facebook.png`

Delete the following file:

`assets/HiBrow/LocalNavigation/img/qq.png`

Step 4 Recompile the code to generate the application.

----End

5.1.4 How Do I Play HTML5 Videos in Small-Window Mode?

NOTE

This section is dedicated for the Android 4.4 version.

Problem Description

The full-screen mode is used by default when HTML5 videos are played in the browser. How do I change the default mode to small-window mode?

Cause Analysis

When you play HTML5 videos in the browser, the full-screen mode is used by default because most videos on TVs need to be played in full-screen mode. If you want to set the default mode for playing HTML5 videos to the small-window mode, you need to modify the attribute and recompile the system.

Solution

Do as follows:

Step 1 Go to **/device/hisilicon/Hi3798MV100**.

Step 2 Modify the following attribute in **device.mk**.

```
hibrowser.default.fullscreen=false
```

The default value is **true**, indicating full-screen mode. Change it to **false** to use the small-window mode.

Step 3 Recompile and generate the image.

----End

5.2 Multimedia

5.2.1 What Do I Do to Display Flash Resources on the Web Page?

NOTE

This section is dedicated for versions earlier than Android 4.4 because Android 4.4 and later do not support the flash.

Problem Description

When some websites are visited, some content cannot be displayed, and the system displays a message indicating that the Flash Player is not installed. How do I solve this problem?

Analysis

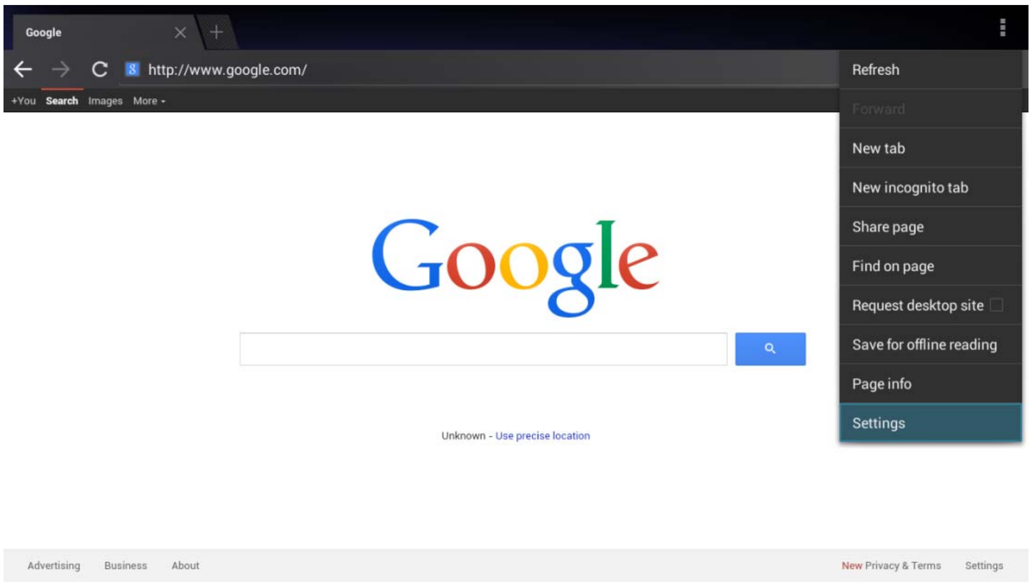
By default, the browser obtains the HTML5 media resources in priority. However, some websites do not have HTML5 resources. Instead, flash resources are transmitted. The browser cannot display some multimedia content on the web page, such as flash advertisements, flash games, and flash videos.

Solution

A configuration option is provided in the browser dedicated for setting the browser mode. The **Phone&Tablet** mode is used by default. To display the flash videos and games on the web page, you need to switch the browser mode to PC mode as follows:

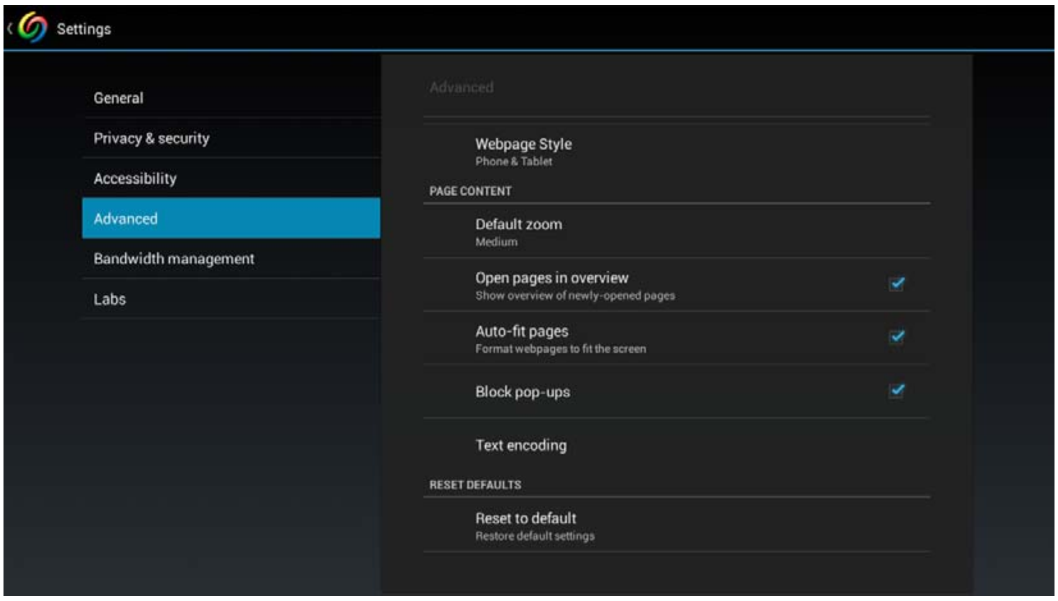
Step 1 Choose **Settings**.

Figure 5-18 Choosing Settings



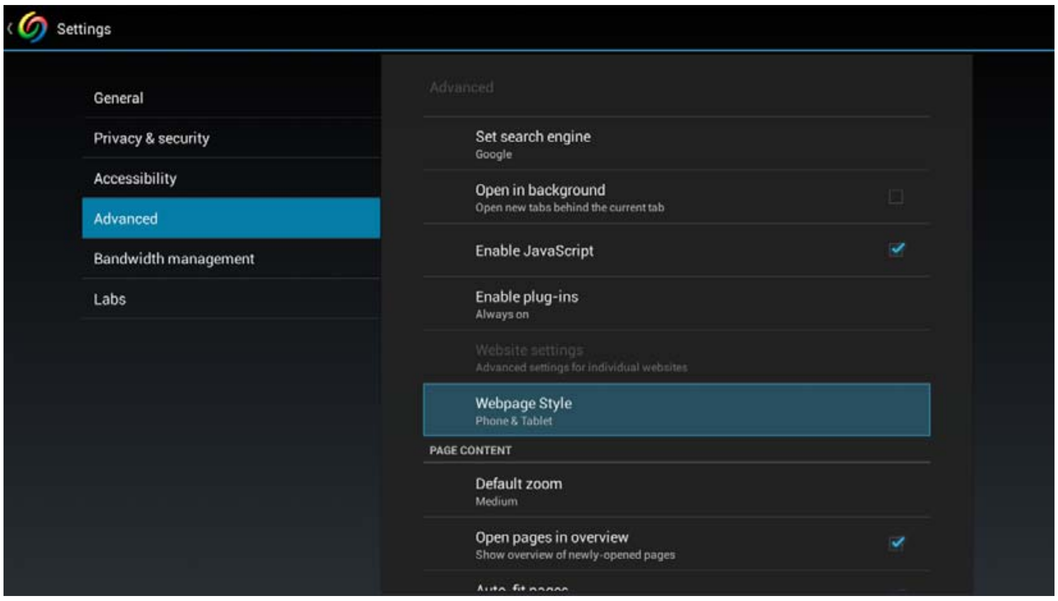
Step 2 Click **Advanced**.

Figure 5-19 Advanced settings



Step 3 Click **Webpage Style**.

Figure 5-20 Clicking Webpage Style



Settings

General
Privacy & security
Accessibility
Advanced
Bandwidth management
Labs

Advanced

Set search engine
Google

Open in background ☐
Open new tabs behind the current tab

Enable JavaScript ✓

Enable plug-ins
Always on

Website settings
Advanced settings for individual websites

Webpage Style
Phone & Tablet

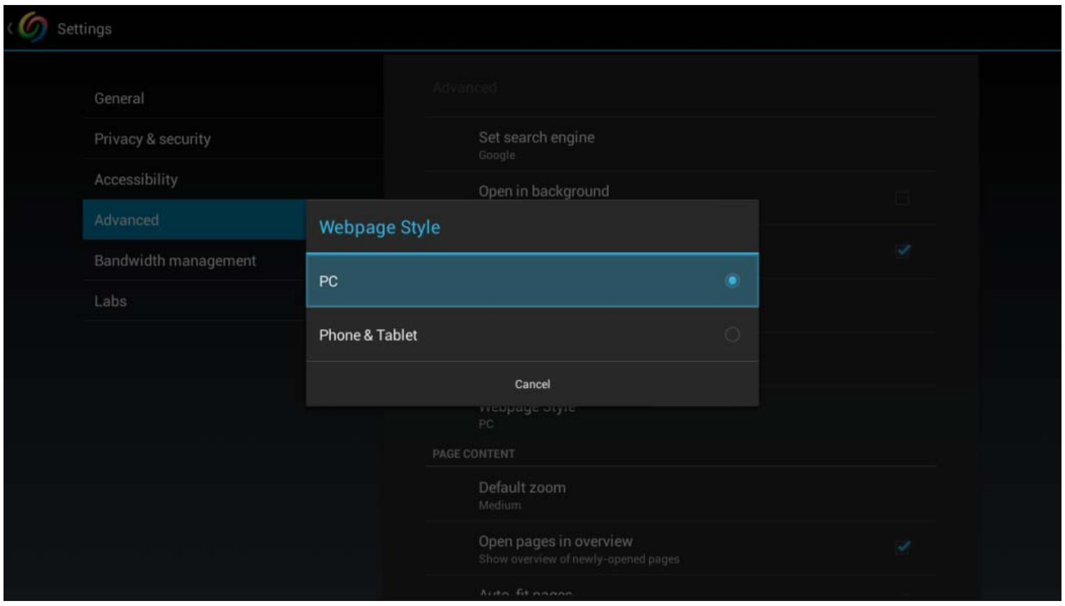
PAGE CONTENT

Default zoom
Medium

Open pages in overview ✓
Show overview of newly-opened pages

Step 4 Select **PC**.

Figure 5-21 Selecting PC



Webpage Style

PC ●

Phone & Tablet ○

Cancel

Step 5 Open the web page again or refresh the web page.

----End

6 Media Processing

6.1 Audio

6.1.1 How Do I Develop the Volume Control Function?

Problem Description

The volume control APIs of Android and the SDK UNF APIs coexist. During the development of DVB, IPTV, and media playback applications, I do not know what APIs I should use.

Analysis

Both the Android inherent code and the SDK UNF provide APIs for adjusting the volume. If the APIs are not called properly, errors occur.

Solution

The volume adjustment API of the HiAO is called at the Alsa HAL layer. Therefore, the volume of the HiAO is also adjusted when the Android inherent API is called. The usage of this solution is described as follows in typical scenarios:

The volume adjustment entry is the **Setting** page or the remote control, both of which adjust the global volume. When the APK directly calls the UNF API, the volume of a single track is adjusted, but not the global volume.

- The volume of applications such as the DVB, IPTV, and HiPlayer is adjusted by adjusting the global volume. Applications do not respond to key events of the remote control. The events are automatically processed by the Android framework.
- The DVB needs to save the volume of a single channel (mainly the current channel) in real time. In this case, the DVB must listen to the VOLUME_CHANGED_ACTION event, which is broadcast after the volume is adjusted using the remote control, and then save the current volume and channel after receiving the event. Before switching to another channel, the DVB reads the saved value and then sets the initial volume of the channel by calling the Audio Manager volume adjustment API.
- Saving the volume
  - Global volume: After adjusting the volume, the Audio Service calls persistvolume to write the volume to the setting provider. When the Audio Service starts, it reads the saved value from the setting provider to initialize the **mStreamStates** variable.

– Single-track volume: It is saved by the application.

6.1.2 How Do I Enable the HDMI Pass-through Output?

Problem Description

How do I enable the HDMI pass-through output?

Analysis

The Setting page has many options.

Solution

The HDMI and SPDIF source code output function is provided to obtain better sound effect.

The Sound UI layout of Setting provides several options for controlling the disable/decoding/pass-through/next-generation audio of the HDMI output, which meets requirements of various scenarios.

The sound options are described as follows:

- HDMI disable: The HDMI interface does not output data when it is disabled.
- HDMI auto: The AO obtains the EDID and automatically matches the output mode based on the EDID information.

  Pass-through output is used if it is supported by the output terminal; otherwise, data is decoded and then output.
- HDMI decoding: This mode is used if no external decoding device is connected. The system decodes all audio data into PCM data and then output.
- HDMI pass-through: If this option is selected, the system always outputs the obtained Dolby Dts Truehd audio data as source code. However, some MP3/AAC data is decoded and then output.
- Blu-ray next-generation down-specifications output: The default value is **Auto**, which indicates disabled and has no effect on the HDMI and SPDIF output. If **RAW5.1/7.1** is selected and the HDMI is not disabled, the HDMI outputs data based on the specifications forcibly. If the played streams do not support the next-generation audio specifications, data is output based on the maximum specifications supported. For the truehd composite stream scenario, if the RAW7.1 output is forcibly specified, the truehd decoding library is used; if the RAW5.1 or decoding output is selected, the Dolby decoding library is used.

6.2 Video

6.2.1 What Do I Do If the Video Played by Using the UNF API Cannot Be Displayed Properly?

Problem Description

When the applications written by customers implement multimedia playback by calling the UNF APIs in Java Native Interface (JNI) mode or standard APIs of the media player, the following issues occur:

- The audio is properly played but there is no video.
- A video is played on the edge of the screen, and most of the video window is covered.

Analysis

When multimedia files are played by calling the UNF APIs, the video displays at the video layer. The video layer is behind the graphics layer by default. Therefore, the region on the graphics layer which corresponds to the application must be set to transparent so that the video at the video layer can be displayed.

You can hide the graphics layer to check whether the video cannot be displayed because it is overlaid by the graphics layer.

To hide the graphics layer, run the following command over the serial port:

echo hide > **/proc/msp/hifb0**

To display the graphics layer, run the following command over the serial port:

echo show > **/proc/msp/hifb0**

If the video displays properly after the graphics layer is hidden, the preceding issue occurs because the video is overlaid by the graphics layer. In this case, you need to set the region on the graphics layer that corresponds to the video to transparent so that the video can be displayed.

Solution

The application creates a surface view on the graphics layer (which corresponds to the position of the played video) and sets the surface view to transparent. On the HiSilicon platform, a transparency flag is required to set the corresponding surface view to transparent. The implementation is as follows:

```
msurfaceholder.setType(SurfaceHolder.SURFACE_TYPE_HISI_TRANSPARENT);
//msurfaceholder is the surface holder corresponding to the surface view
```

If the video still cannot be displayed, you can capture the logs in the scenario in which the issue occurs for further analysis.

- Snapshot the Android screen and capture the screen by using the Dalvik Debug Monitor Server (DDMS) on the PC, or running the following command over the serial port:
  ```
  screencap -p /data/screen.png
  ```
- Capture the SurfaceFligner state log or run the following command over the serial port:

```
dumpsys SurfaceFlinger
```

Send the snapshots and SurfaceFlinger state logs to the HiSilicon FAE and R&D engineers for analysis.

6.2.2 What Do I Do If a Video with Multiple Reference Frames Cannot Be Played?

Problem Description

A video with multiple reference frames cannot be played in the SDK that is compiled by using the default configurations.

Analysis

The number of reference frames in the video exceeds that supported by the current SDK.

Solution

Perform the following steps:

Step 1 Check the SDK configuration file (***XXX*_cfg.mak**) used in the current version.

Take Hi3798M V100 as an example. Go to the corresponding product directory **device/hisilicon/Hi3798MV100**.

NOTE

The product directories for other chips can be found in the **device/hisilicon/** directory.

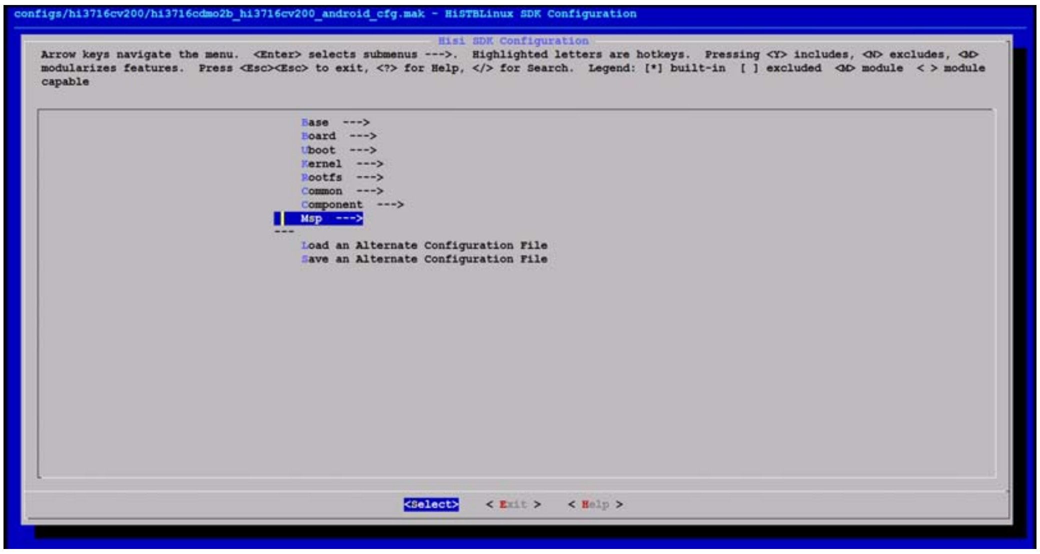
Search for the environment variable **HISI_SDK_ANDROID_CFG** in **BoardConfig.mk**. The value of **HISI_SDK_ANDROID_CFG** is the SDK configuration file. The SDK configuration file for Hi3798M V100 is **hi3798mdmo1b_hi3798mv100_android_cfg.mak.**

Step 2 Go to the **device/hisilicon/bigfish/sdk** directory and run the following command:

```
make menuconfig
SDK_CFGFILE=configs/hi3798mv100/hi3798mdmo1b_hi3798mv100_android_cfg.mak
```

Step 3 Choose **Msp**.

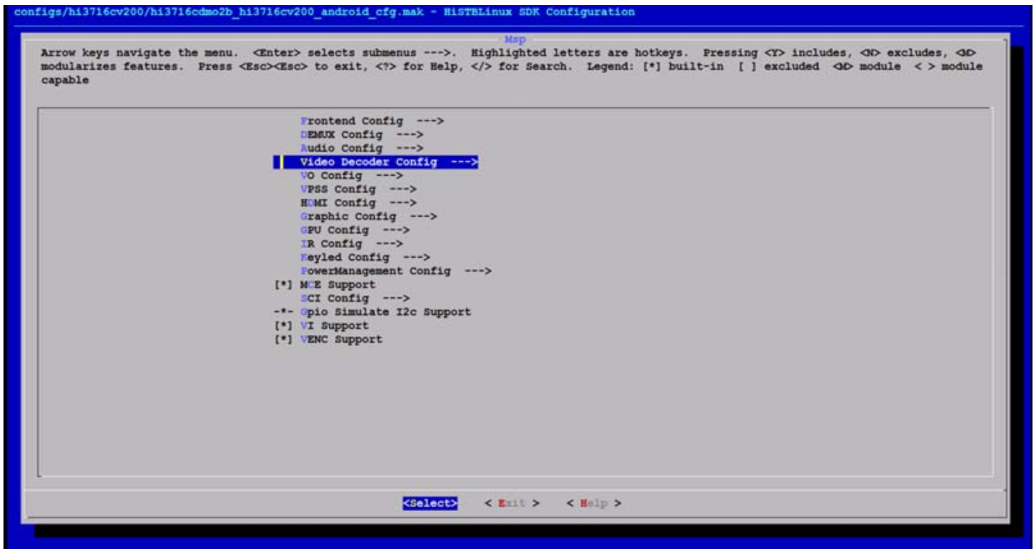
Figure 6-1 Choosing Msp



```
configs/hi3716cv200/hi3716cdmo2b_hi3716cv200_android_cfg.mak - HiSTBLinux SDK Configuration
                                    Hisi SDK Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
  modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                                       Base  --->
                                       Board  --->
                                       Uboot  --->
                                       Kernel  --->
                                       Rootfs  --->
                                       Common  --->
                                       Component  --->
                                       Msp  --->
                                   ---
                                       Load an Alternate Configuration File
                                       Save an Alternate Configuration File

                              <Select>    < Exit >    < Help >
```

Step 4 Choose **Video Decoder Config**.

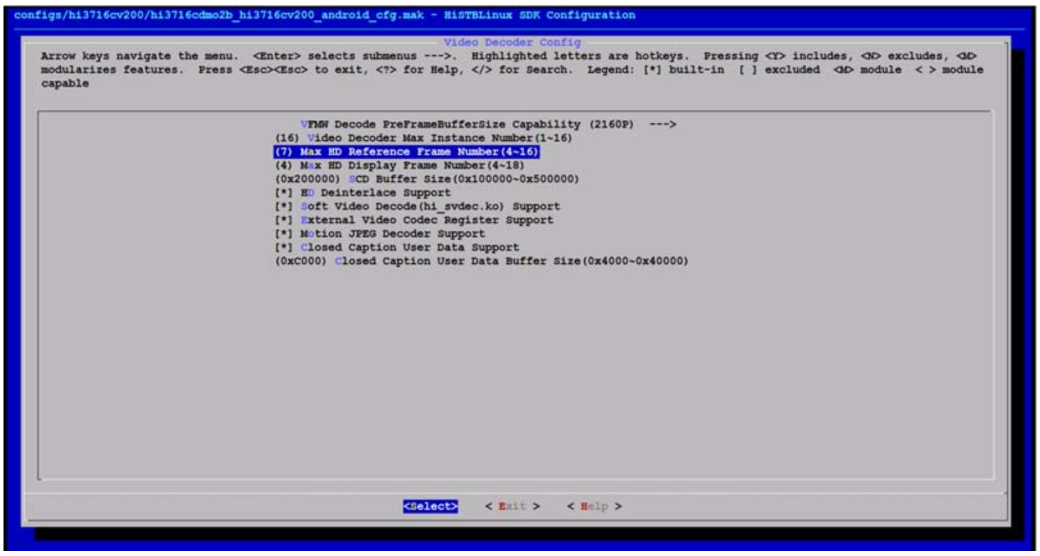
Figure 6-2 Choosing Video Decoder Config



```
configs/hi3716cv200/hi3716cdmo2b_hi3716cv200_android_cfg.mak - HiSTBLinux SDK Configuration
                                              Msp
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
  modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                                       Frontend Config  --->
                                       DEMUX Config  --->
                                       Audio Config  --->
                                       Video Decoder Config  --->
                                       VO Config  --->
                                       VPSS Config  --->
                                       HDMI Config  --->
                                       Graphic Config  --->
                                       GPU Config  --->
                                       IR Config  --->
                                       Keyled Config  --->
                                       PowerManagement Config  --->
                                   [*] MCE Support
                                       SCI Config  --->
                                   -*- Gpio Simulate I2c Support
                                   [*] VI Support
                                   [*] VENC Support

                              <Select>    < Exit >    < Help >
```

Step 5 Choose **Max HD Reference Frame Number(4~16)**.

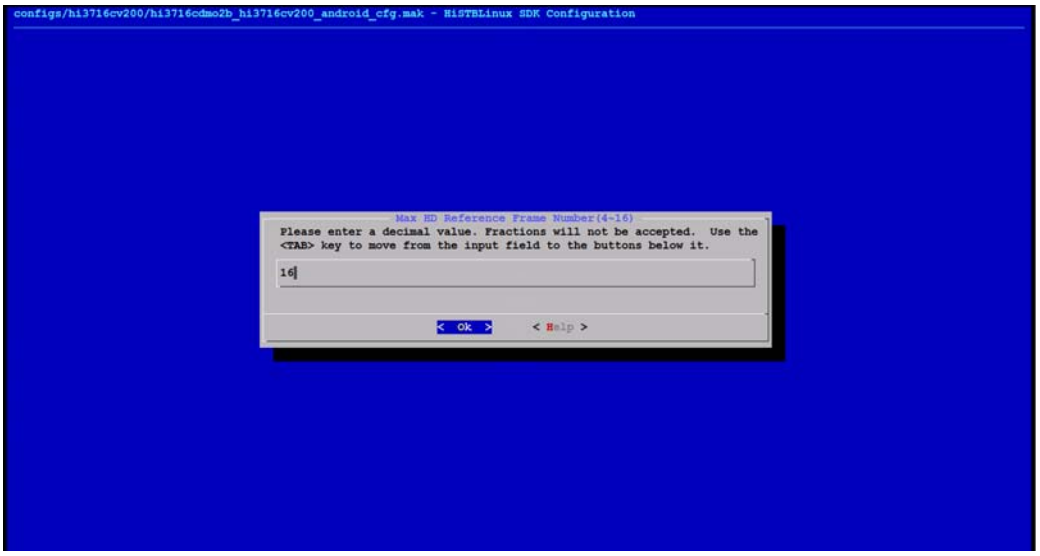
Figure 6-3 Choosing Max HD Reference Frame Number(4~16)



```
configs/hi3716cv200/hi3716cdmo2b_hi3716cv200_android_cfg.mak - HiSTBLinux SDK Configuration
                              Video Decoder Config
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
 modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
 capable

                    VFMW Decode PreFrameBufferSize Capability (2160P)  --->
                (16) Video Decoder Max Instance Number(1~16)
                (7) Max HD Reference Frame Number(4~16)
                (4) Max HD Display Frame Number(4~18)
                (0x200000) SCD Buffer Size(0x100000~0x500000)
                [*] HD Deinterlace Support
                [*] Soft Video Decode(hi_svdec.ko) Support
                [*] External Video Codec Register Support
                [*] Motion JPEG Decoder Support
                [*] Closed Caption User Data Support
                (0xC000) Closed Caption User Data Buffer Size(0x4000~0x40000)

                         <Select>    < Exit >    < Help >
```

Step 6 Change the default value to the required value, for example, **16** (which is the maximum value allowed).

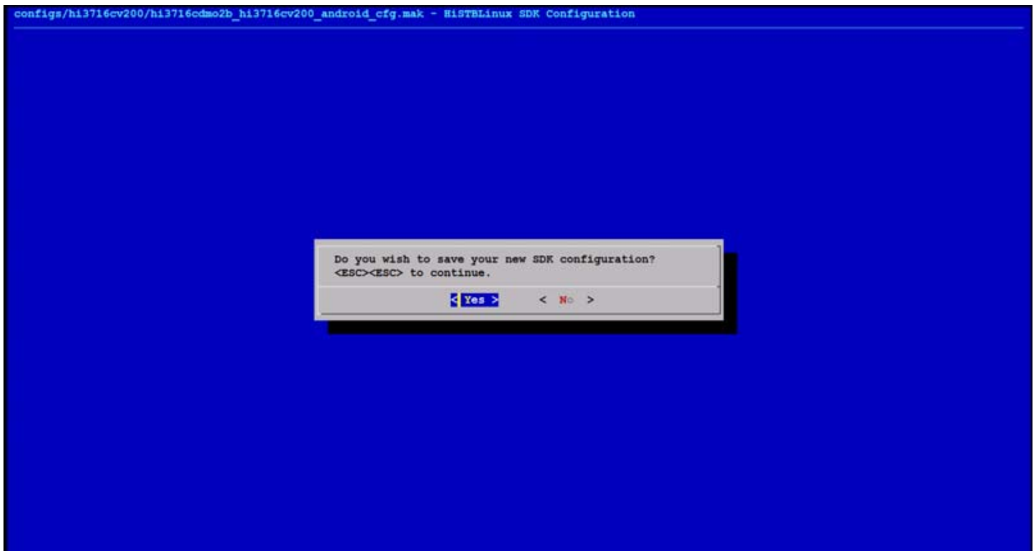
Figure 6-4 Modifying the value



```
configs/hi3716cv200/hi3716cdmo2b_hi3716cv200_android_cfg.mak - HiSTBLinux SDK Configuration

            Max HD Reference Frame Number(4~16)
 Please enter a decimal value. Fractions will not be accepted.  Use the
 <TAB> key to move from the input field to the buttons below it.
 16

                 <  Ok  >     < Help >
```

Step 7 Save the configuration and exit.

Figure 6-5 Save and exit



```
configs/hi3716cv200/hi3716cdmo2b_hi3716cv200_android_cfg.mak - HiSTBLinux SDK Configuration

Do you wish to save your new SDK configuration?
<ESC><ESC> to continue.
```

The SDK configuration file is modified. After the kernel is recompiled, videos with multiple reference frames can be played. For details about how to recompile the kernel, see **install_notes** in the root directory of the code.

----End

6.2.3 How Do I Enable Simultaneous Output of the HDMI and CVBS?

CAUTION

- Currently the mutual exclusion policy of the HDMI and CVBS applies only to the Hi3798M V100.
- During Dolby certification, the HDMI and CVBS output data simultaneously by default. Therefore, **persist.sys.cvbs.and.hdmi** does not need to be modified.

Problem Description

The HDMI and CVBS of Hi3798M V100 are mutually exclusive by default, that is, the HDMI and CVBS data cannot be output at the same time. If an HDMI cable is inserted, CVBS data is not output; after the HDMI cable is removed, CVBS data is output. Introduction of this mutual exclusion policy aims to reduce the power consumption, which is 150 mW higher if this policy is not used.

However, the HDMI and CVBS interfaces must output data at the same time if the simultaneous display of SD and HD is required. How do I enable simultaneous output of the HDMI and CVBS?

Solution

To enable or disable simultaneous output of the HDMI and CVBS, a system prop attribute **persist.sys.cvbs.and.hdmi** is added on Android. The mutual exclusion policy of the HDMI and CVBS can be enabled or disabled before compiling the Android. To be specific, you can change the value of **persist.sys.cvbs.and.hdmi** in **device/hisilicon/Hi3798MV100/customer.mk**.

- To enable simultaneous output of the HDMI and CVBS, set **persist.sys.cvbs.and.hdmi** as follows:
  ```
  persist.sys.cvbs.and.hdmi=true
  ```
- To disable simultaneous output of the HDMI and CVBS (default), set **persist.sys.cvbs.and.hdmi** as follows:
  ```
  persist.sys.cvbs.and.hdmi=false
  ```

6.2.4 How Do I Set the Display Regions of the Video Layer and Graphics Layer to Be the Same?

NOTE

The display regions of the video layer and graphics layer are the same for Android 4.2 but different for Android 4.4 by default. Therefore, this issue is dedicated to Android 4.4.

Problem Description

For Android 4.4, after the screen output region is configured on the Settings UI, the OSD changes accordingly, but local videos are still played in full-screen mode.

Solution

For Android 4.4, you can configure the display parameter on the **Settings** screen to adjust the graphics display region on the TV.

By default, the configuration takes effect only for the graphics layer. Videos are played in full-screen mode by default, which is not restricted by the configuration. This ensures that videos are always played in full-screen mode.

If you want the display regions for the video layer and graphics layer to be same, modify the SDK configuration file as follows:

Step 1 Find the correct configuration file.

The name of the SDK configuration file is defined by the **HISI_SDK_ANDROID_CFG** variable in **BoardConfig.mk**. For example, **HISI_SDK_ANDROID_CFG** in **device/hisilicon/Hi3798MV100/BoardConfig.mk** is as follows:

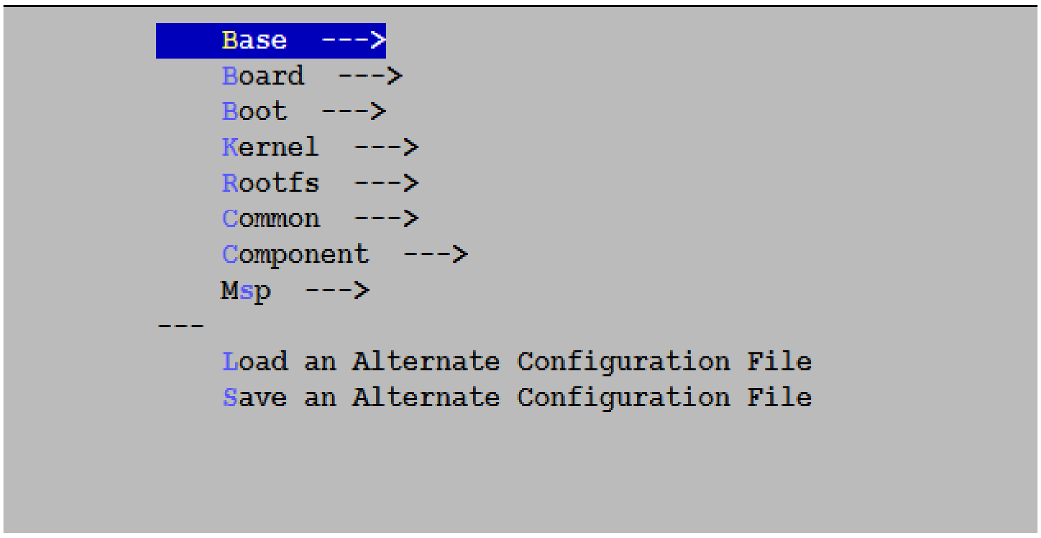
```
HISI_SDK_ANDROID_CFG := hi3798mdmo1a_hi3798mv100_android_cfg.mak
```

Step 2 Configure system parameters.

1. Run **make menuconfig SDK_CFGFILE=configs/***configuration file to be modified* in the **device/hisilicon/bigfish/sdk** directory, for example:
   ```
   make menuconfig
   SDK_CFGFILE=configs/hi3798mv100/hi3798mdmo1b_hi3798mv100_android_cfg.
   mak
   ```

The configuration UI shown in Figure 6-6 is displayed.

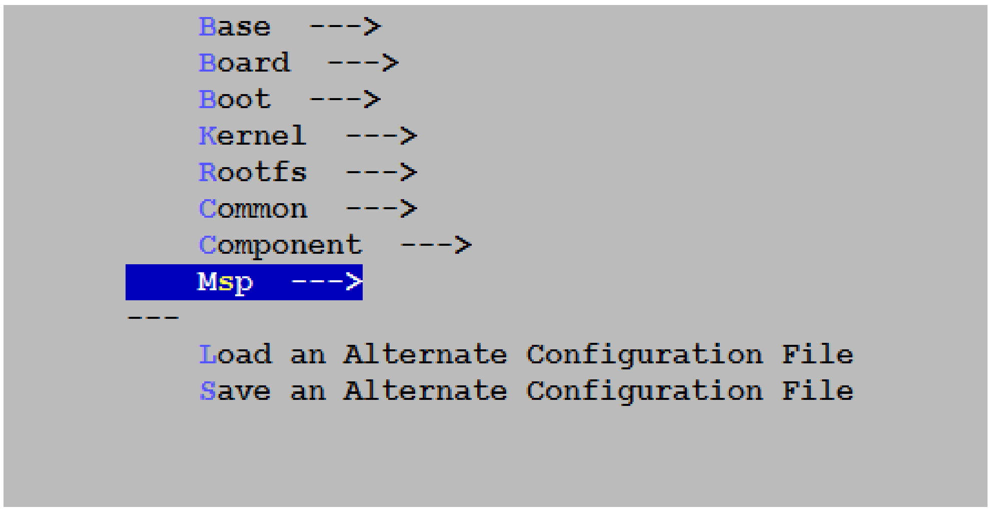
Figure 6-6 SDK configuration UI



```
    Base  --->
    Board  --->
    Boot  --->
    Kernel  --->
    Rootfs  --->
    Common  --->
    Component  --->
    Msp  --->
---
    Load an Alternate Configuration File
    Save an Alternate Configuration File
```

2. Select **Msp**, and press **Enter**.

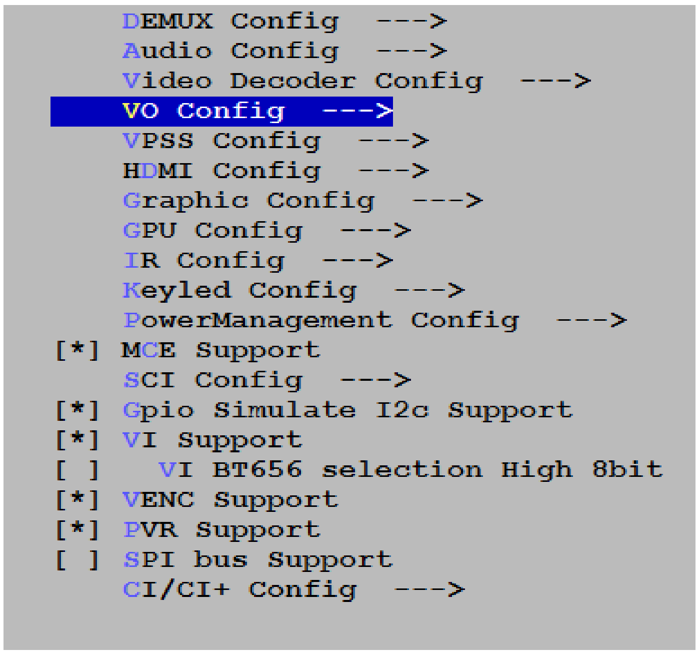
Figure 6-7 Selecting Msp



```
    Base  --->
    Board  --->
    Boot  --->
    Kernel  --->
    Rootfs  --->
    Common  --->
    Component  --->
    Msp  --->
---
    Load an Alternate Configuration File
    Save an Alternate Configuration File
```

3. Select **VO Config**, and press **Enter**.

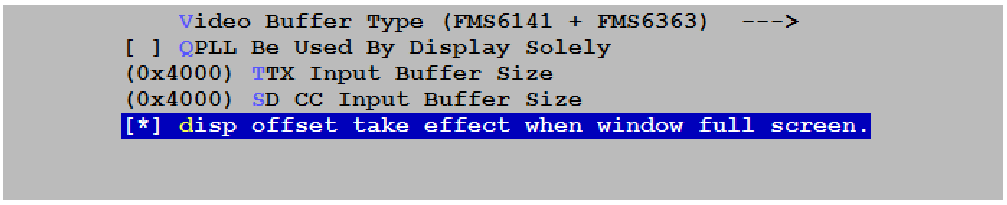
Figure 6-8 Selecting VO Config



```
        DEMUX Config  --->
        Audio Config  --->
        Video Decoder Config  --->
        VO Config  --->
        VPSS Config  --->
        HDMI Config  --->
        Graphic Config  --->
        GPU Config  --->
        IR Config  --->
        Keyled Config  --->
        PowerManagement Config  --->
    [*] MCE Support
        SCI Config  --->
    [*] Gpio Simulate I2c Support
    [*] VI Support
    [ ]   VI BT656 selection High 8bit
    [*] VENC Support
    [*] PVR Support
    [ ] SPI bus Support
        CI/CI+ Config  --->
```

4. Select **disp offset take effect when window full screen**, and press **Y**.

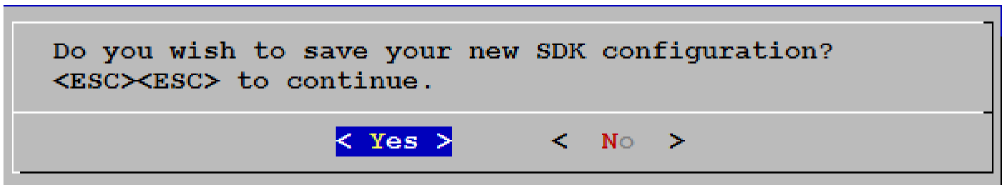
Figure 6-9 Selecting disp offset take effect when window full screen



```
        Video Buffer Type (FMS6141 + FMS6363)  --->
    [ ] QPLL Be Used By Display Solely
    (0x4000) TTX Input Buffer Size
    (0x4000) SD CC Input Buffer Size
    [*] disp offset take effect when window full screen.
```

5. Press **ESC** to exit the configuration UI, and select **Yes** to save the configuration when the dialog box shown in Figure 6-10 is displayed.

Figure 6-10 Saving the configuration



```
 Do you wish to save your new SDK configuration?
 <ESC><ESC> to continue.

                 < Yes >      <  No  >
```

Step 3 Recompile and burn the images.

Go to the root directory of the source code, and run **make clean** to clean all compiled images (if the code was compiled before).

Recompile the code by running the following commands in sequence:

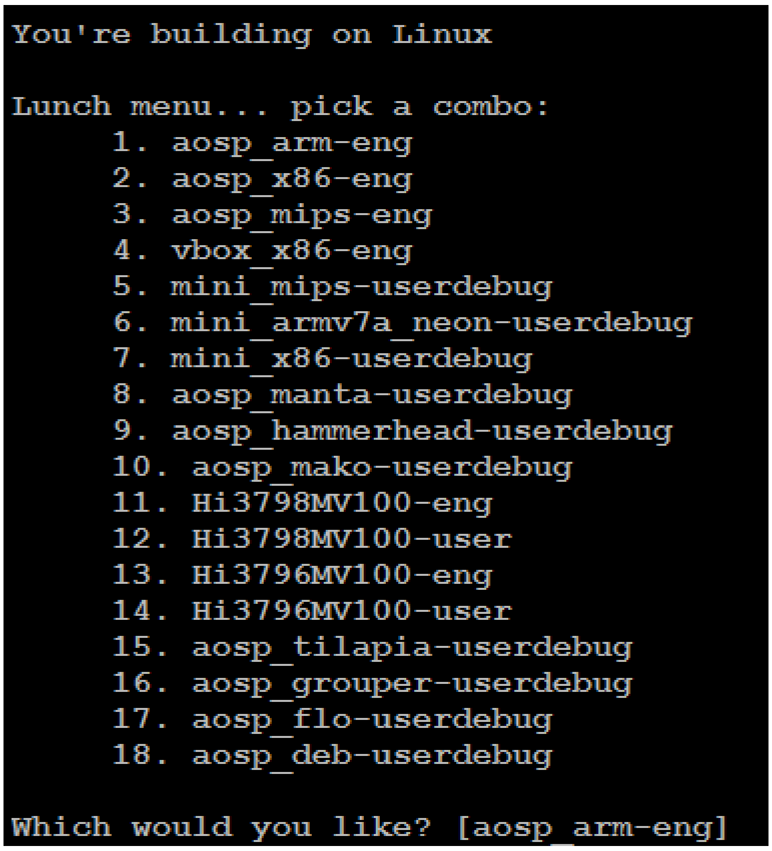
source build/envsetup.sh

lunch Hi3798MV100-eng

make bigfish –j32

In the second command (**lunch Hi3798MV100-eng**), Hi3798M V100 is the chip name, followed by the version to be compiled (eng version in the preceding command). If you want to compile another chip (for example, Hi3796M V100) or another version (for example, user version), run the **lunch** command directly. Then the information shown in Figure 6-11 is displayed.

Figure 6-11 Lunch option list



```
You're building on Linux

Lunch menu... pick a combo:
     1. aosp_arm-eng
     2. aosp_x86-eng
     3. aosp_mips-eng
     4. vbox_x86-eng
     5. mini_mips-userdebug
     6. mini_armv7a_neon-userdebug
     7. mini_x86-userdebug
     8. aosp_manta-userdebug
     9. aosp_hammerhead-userdebug
     10. aosp_mako-userdebug
     11. Hi3798MV100-eng
     12. Hi3798MV100-user
     13. Hi3796MV100-eng
     14. Hi3796MV100-user
     15. aosp_tilapia-userdebug
     16. aosp_grouper-userdebug
     17. aosp_flo-userdebug
     18. aosp_deb-userdebug

Which would you like? [aosp_arm-eng]
```

Enter the number of the version to be compiled, and press **Enter**. For example, to compile Hi3798M V100-eng, enter **11** and press **Enter**.

After compilation is complete, burn the images for each partition.

Then the display regions of the graphics window and video window are the same.

----End

7 Applications

7.1 What Do I Do to Display the System Status Bar?

Problem Description

The system default status bar is hidden. What do I do to display it?

Analysis

During product development, if the style of the system inherent status bar is not consistent with that of the product, customers typically customize a new status bar and hide the inherent one. Therefore, the inherent status bar is hidden by default to facilitate modification.

Solution

To display the inherent status bar, modify **frameworks/base/core/res/res/values/dimens.xml**.

Change the following code to the Android original code:

```
<!-- Height of the status bar
       <dimen name="status_bar_height">25dip</dimen>
       -->
       <dimen name="status_bar_height">0dip</dimen>
    <!-- Height of the bottom navigation / system bar.
    <dimen name="navigation_bar_height">48dp</dimen>
       -->
       <dimen name="navigation_bar_height">0dp</dimen>
    <!-- Height of the bottom navigation bar in portrait; often the same
as @dimen/navigation_bar_height
    <dimen name="navigation_bar_height_landscape">48dp</dimen>
       -->
       <dimen
name="navigation_bar_height_landscape">@dimen/navigation_bar_height</dime
n>
```

The following are the original code:

```
<dimen name="status_bar_height">25dip</dimen>
<dimen name="navigation_bar_height">48dp</dimen>
<dimen name="navigation_bar_height_landscape">48dp</dimen>
```

7.2 How Do I Enable the Media Scanner?

Problem Description

Is the media scanner supported in the system version? How do I enable the media scanner to scan the USB flash drive or hard disk?

Cause Analysis

The media scanner based on the system version does not respond to hot plugging of the USB flash drive or hard disk. Only the **/mnt/sdcard/** and **/system/media/** directories are scanned by default.

Solution

To enable the media scanner, modify **/device/hisilicon/Hi3798MV100/device.mk**.

Change the following code:

```
# MediaScanner
PRODUCT_PROPERTY_OVERRIDES += \
    ro.mediaScanner.enable=false
```

to:

```
# MediaScanner
PRODUCT_PROPERTY_OVERRIDES += \
    ro.mediaScanner.enable=true
```

7.3 How Do I Enable the Android Inherent Animation?

Problem Description

No animation effect is used during the switchover of the Android GUI. How do I enable the animation function during the switchover?

Cause Analysis

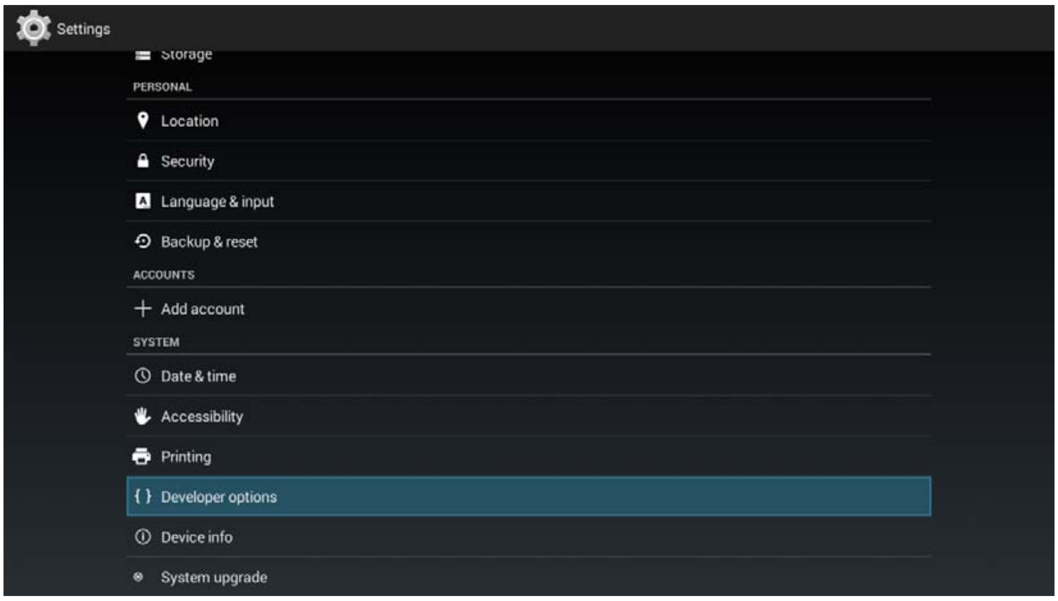
If the animation function is enabled, residues can be perceived during the Android GUI switchover, which results in poor user experiences.

Solution

To enable the animation function, do as follows:

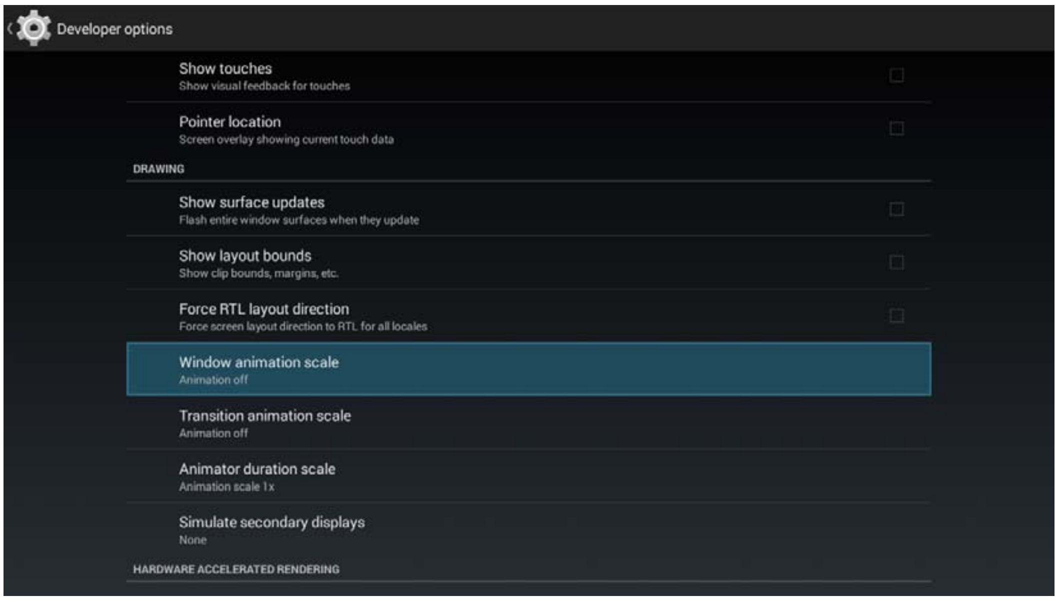
Step 1 Choose **Settings** > **Developer options**, as shown in Figure 7-1.

Figure 7-1 Opening Developer options



Step 2 Choose **Window animation scale**, as shown in Figure 7-2.

Figure 7-2 Choosing Window animation scale



Show touches
Show visual feedback for touches

Pointer location
Screen overlay showing current touch data

Show surface updates
Flash entire window surfaces when they update

Show layout bounds
Show clip bounds, margins, etc.

Force RTL layout direction
Force screen layout direction to RTL for all locales

Window animation scale
Animation off

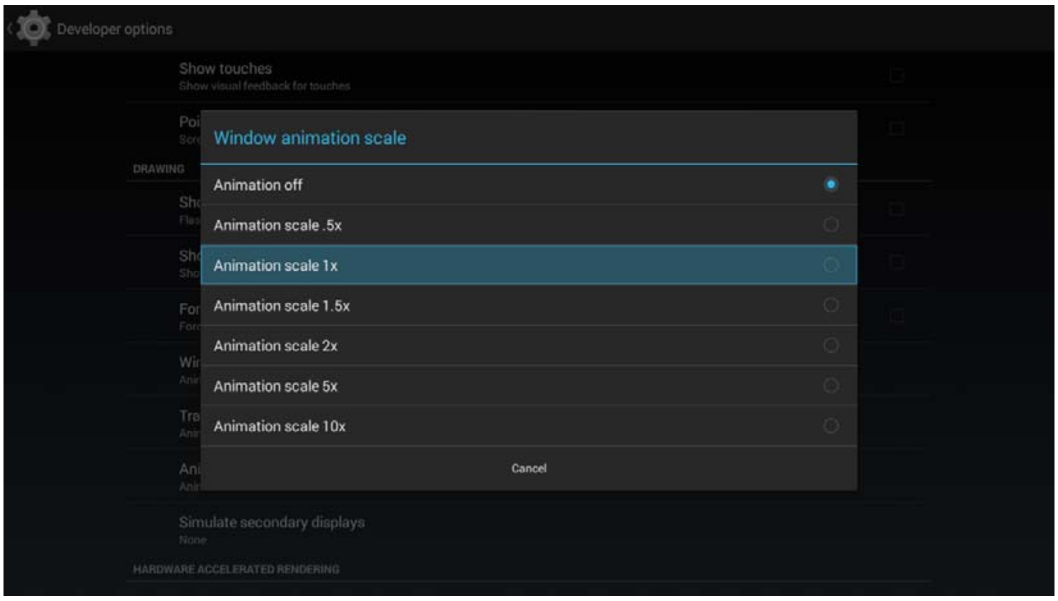
Transition animation scale
Animation off

Animator duration scale
Animation scale 1x

Simulate secondary displays
None

Step 3 Select a scale factor for window animation as required, as shown in Figure 7-3. Selecting **Animation off** disables the animation function.

Figure 7-3 Selecting the scale factor for window animation



Developer options

Show touches
Show visual feedback for touches

Poi
Scr

DRAWING

Sh
Fla

Sh
Sho

For
For

Wi
Ani

Tra
Ani

Ani
Ani

Window animation scale

- [✓ Animation off]
- Animation scale .5x
- Animation scale 1x
- Animation scale 1.5x
- Animation scale 2x
- Animation scale 5x
- Animation scale 10x

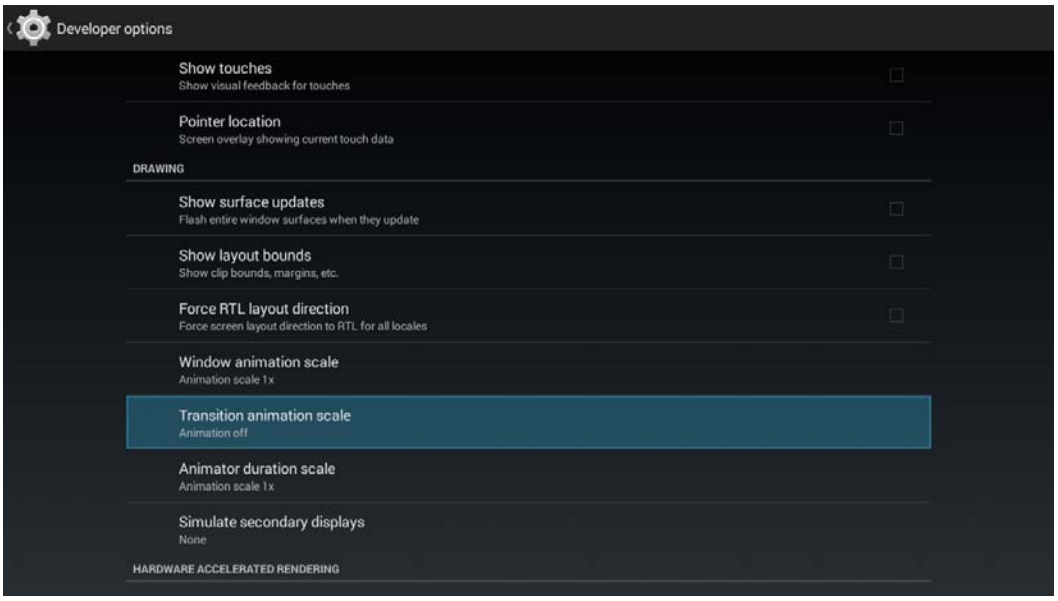
Cancel

Simulate secondary displays
None

HARDWARE ACCELERATED RENDERING

Step 4 Choose **Transition animation scale**, as shown in Figure 7-4.

Figure 7-4 Choosing Transition animation scale



Developer options

Show touches
Show visual feedback for touches

Pointer location
Screen overlay showing current touch data

DRAWING

Show surface updates
Flash entire window surfaces when they update

Show layout bounds
Show clip bounds, margins, etc.

Force RTL layout direction
Force screen layout direction to RTL for all locales

Window animation scale
Animation scale 1x

Transition animation scale
Animation off

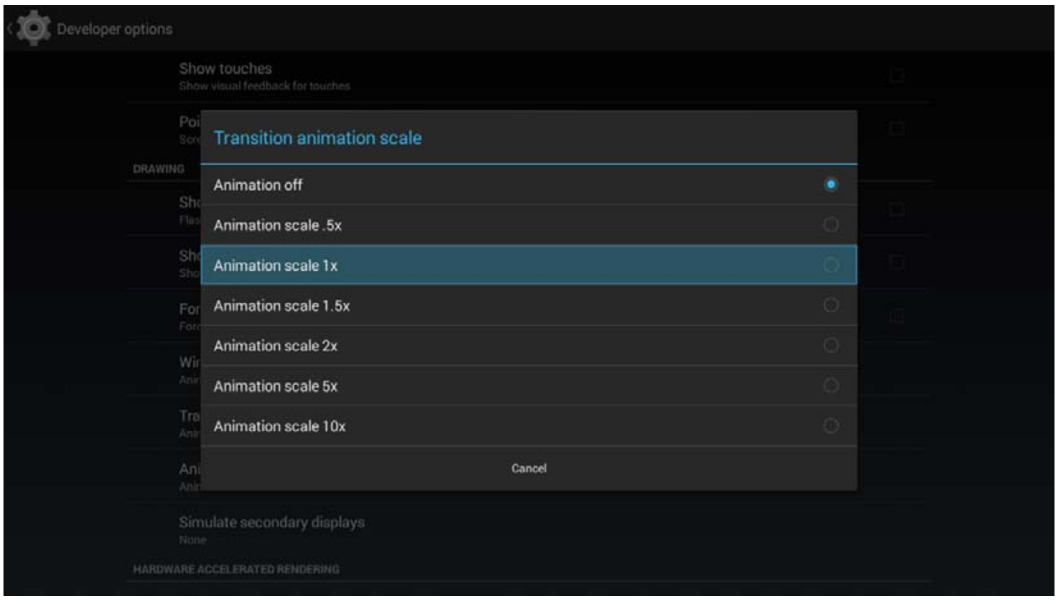
Animator duration scale
Animation scale 1x

Simulate secondary displays
None

HARDWARE ACCELERATED RENDERING

Step 5 Select a scale factor for transition animation as required, as shown in Figure 7-5. Selecting **Animation off** disables the animation function.

Figure 7-5 Selecting the scale factor for transition animation


Developer options
Show touches
Show visual feedback for touches
DRAWING
Simulate secondary displays
None
HARDWARE ACCELERATED RENDERING
Cancel

Transition animation scale

- [✓ Animation off]
- Animation scale .5x
- Animation scale 1x
- Animation scale 1.5x
- Animation scale 2x
- Animation scale 5x
- Animation scale 10x

The GUI switchover animation is configured.

----End

7.4 How Do I Change the Priority of the Bound Network Adapter for the DLNA and MultiScreen?

Problem Description

The Ethernet and Wi-Fi network adapters can coexist on an Android system. However, the DLNA and MultiScreen work only with one specific network adapter. Currently they work with the Ethernet network adapter when the Ethernet network and Wi-Fi network coexist. How do I enable the DLNA and MultiScreen to work with the Wi-Fi adapter?

Solution

Modify the following files for the DLNA and MultiScreen:

- DMR

  device/hisilicon/bigfish/hidolphin/component/hidlna/android/packages/apps/HiMediaRender/src/com/hisilicon/dlna/dmr/UpnpBootBroadcastServiceDMR.java
- DMS

  device/hisilicon/bigfish/hidolphin/component/hidlna/android/packages/apps/HiMediaShare/src/com/hisilicon/dlna/dms/MediaService.java
- MultiScreen

  device/hisilicon/bigfish/hidolphin/component/himultiscreen/android/packages/apps/MultiScreenServer/src/com/hisilicon/multiscreen/server/MultiScreenService.java

Find the following variable in the files:

```
private final static String DEFAULT_USE_ADAPTER_WIFI_ETHERNET =
"eth";//"wlan";
```

The variable is set to **eth** by default, indicating that the DLNA and MultiScreen work with the Ethernet network adapter when the Ethernet network and Wi-Fi network coexist.

Change the value to **wlan**. Then the DLNA and MultiScreen work with the Wi-Fi network adapter.

8 System

8.1 Support for the 512 MB Memory

8.1.1 How Do I Compile an Android Version That Can Run on the 512 MB Memory?

Problem Description

The memory is only 512 MB. How do I compile an Android version that can run on the memory?

Solution

See section 2.15 in **install_notes_cn.txt**.

NOTE

Only Android 4.4 and later versions support the 512 MB memory.

8.2 1080p UI

8.2.1 How Do I Switch to the 1080p UI on the HiSilicon Platform?

Problem Description

The GUI is not clear and aesthetic enough when the Android STB uses the 720p resolution. The UI with a high resolution is required.

Cause Analysis

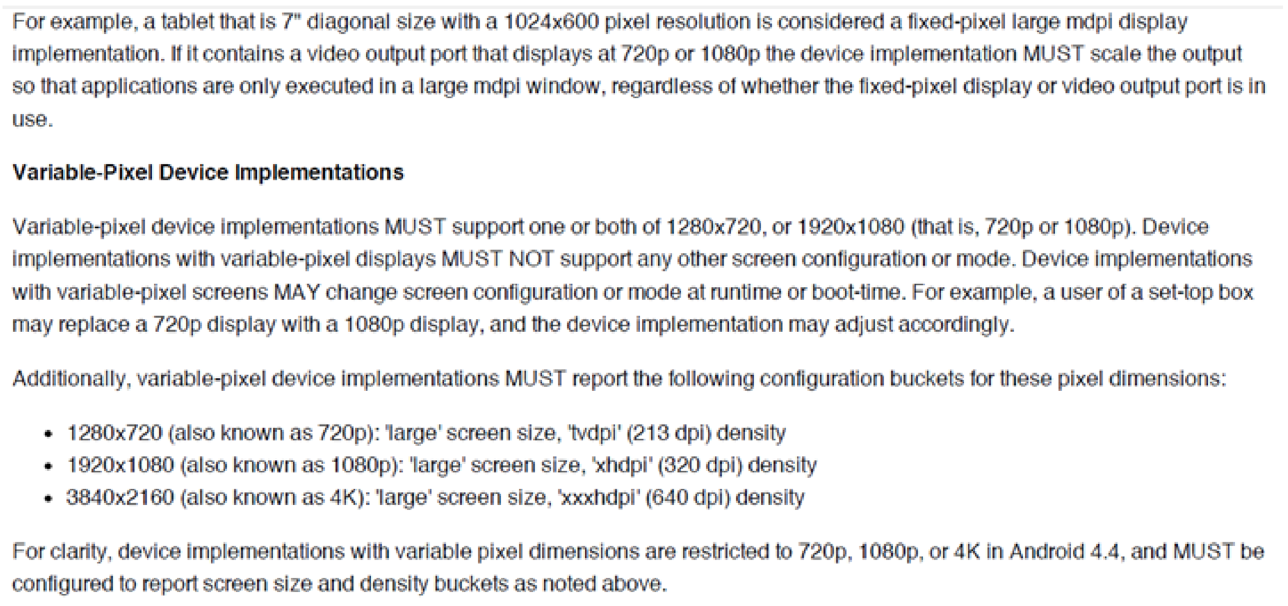
- The 1080p resolution is required in the Android 4.4 CTS certification.

  The over-the-top (OTT) televisions are variable-pixel devices. The CTS certification requires the 720p (213DPI) or 1080p (320DPI) resolution (160DPI is excluded). However, it is difficult to adapt the STB UI layout to the 720p (213DPI) resolution. Therefore, 1080p (320DPI) is usually used for OTT STBs in CTS certification.

NOTE

Figure 8-1 shows the descriptions about requirements on the resolution in **Android4.4-cdd.pdf**.

Figure 8-1 Resolution requirements



For example, a tablet that is 7" diagonal size with a 1024x600 pixel resolution is considered a fixed-pixel large mdpi display implementation. If it contains a video output port that displays at 720p or 1080p the device implementation MUST scale the output so that applications are only executed in a large mdpi window, regardless of whether the fixed-pixel display or video output port is in use.

Variable-Pixel Device Implementations

Variable-pixel device implementations MUST support one or both of 1280x720, or 1920x1080 (that is, 720p or 1080p). Device implementations with variable-pixel displays MUST NOT support any other screen configuration or mode. Device implementations with variable-pixel screens MAY change screen configuration or mode at runtime or boot-time. For example, a user of a set-top box may replace a 720p display with a 1080p display, and the device implementation may adjust accordingly.

Additionally, variable-pixel device implementations MUST report the following configuration buckets for these pixel dimensions:

- 1280x720 (also known as 720p): 'large' screen size, 'tvdpi' (213 dpi) density
- 1920x1080 (also known as 1080p): 'large' screen size, 'xhdpi' (320 dpi) density
- 3840x2160 (also known as 4K): 'large' screen size, 'xxxhdpi' (640 dpi) density

For clarity, device implementations with variable pixel dimensions are restricted to 720p, 1080p, or 4K in Android 4.4, and MUST be configured to report screen size and density buckets as noted above.

- The development of most third-party applications on Android is dedicated for small-screen devices such as mobile phones and pads, and compatibility with 1080p (320DPI) televisions and 4K televisions is not considered. Therefore, application compatibility is poor on the 1080p (320DPI) system, whereas the 240DPI system is compatible with most applications developed for mobile phones and pads.
- Currently the system cannot use multiple DPI values at the same time.

Solution

To switch to the 1080p UI on the HiSilicon platform, do as follows:

Step 1 Set the size of the virtual screen of the base partition to 1920 x 1080 (1080p UI).

Step 2 Set **ro.sf.lcd_density** to 240DPI.

Step 3 Modify the layout and resources of applications to support the 320DPI and 240DPI.

----End

8.2.2 What Do I Do to Use the 1080p UI?

Problem Description

To use the 1080p UI in the Android STB, note the following:

- The 1080p resolution must be supported in the system inherent attributes.
- The system memory must be increased.
- The application layout must be modified to support the 320DPI and 240DPI.

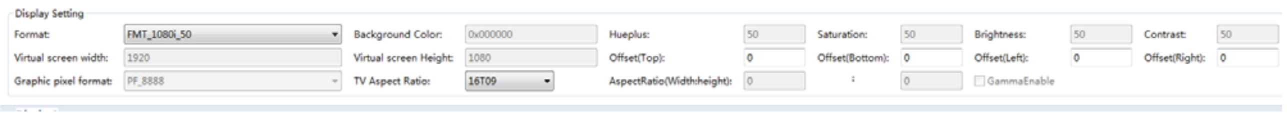
In this case, how do I use the 1080p UI?

Solution

To use the 1080p UI, do as follows:

Step 1 Create a baseparam partition image with the 1080p resolution by using the HiTool.

Figure 8-2 Baseparam partition parameters



Display Setting

Format:	FMT_1080i_50	Background Color:	0x000000	Hueplus:	50	Saturation:	50	Brightness:	50	Contrast:	50
Virtual screen width:	1920	Virtual screen Height:	1080	Offset(Top):	0	Offset(Bottom):	0	Offset(Left):	0	Offset(Right):	0
Graphic pixel format:	PF_8888	TV Aspect Ratio:	16T09	AspectRatio(Width:height):	0	:	0	☐ GammaEnable			

Step 2 Set the system default DPI to 240DPI or 320DPI. Take the Hi3798C as an example. Add the following code to **device\hisilicon\Hi3798CV100\device.mk**.

```
#setup 1080p UI default
PRODUCT_PROPERTY_OVERRIDES += \
    ro.sf.lcd_density=240
```

Step 3 Ensure that the system memory is 1 GB or larger.

Step 4 Modify the application layout and resources.

----End

8.2.3 How Do I Check or Enable the Dynamic UI Switchover Function?

Problem Description

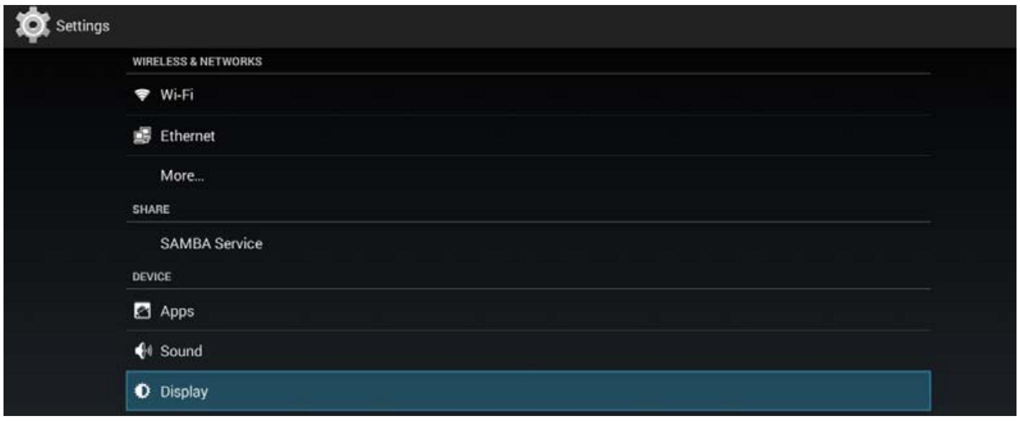
How do I check or enable the dynamic UI switchover function?

Solution

To check or enable the dynamic UI switchover function, do as follows:

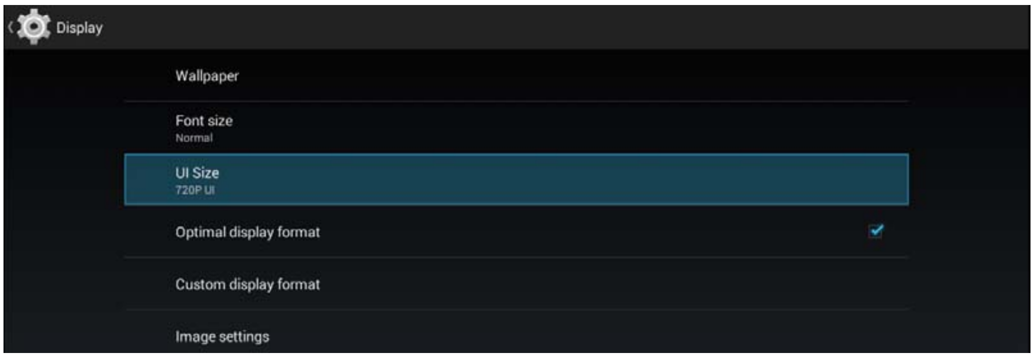
Step 1 Choose **Settings** > **Display**, as shown in Figure 8-3.

Figure 8-3 Choosing Display



Step 2 Check whether the **UI Size** option exists. If yes, the switchover function is supported.

Figure 8-4 Checking the UI Size option



Step 3 Check whether the functional control (**android:key="virtual_screen"**) is enabled in **packages\apps\Settings\res\xml\display_settings.xml**.

Check whether the parameter for dynamically switching the DPI is added in settings. If not, add the corresponding parameter. Take the Hi3798C as an example. Add the following code in red:

```
Setting/src/com/android/settings/DisplaySettings.java
public void onCreate(Bundle savedInstanceState) {
          ---
static final String CHIP_98C =
"type=HI_CHIP_TYPE_HI3798C;version=HI_CHIP_VERSION_V100";
if(!(CHIP_98C.equals(HiSdkinvoke.getChipVersion()))
&& !(CHIP_98M.equals(HiSdkinvoke.getChipVersion()))){
getPreferenceScreen().removePreference(mVirtualScreenPref);
}
---
}
```

Step 4 Recompile the code to replace the original settings code.

----End

8.3 Portrait Orientation

CAUTION

Currently the portrait orientation feature is supported in the following versions:

- Android 4.4.2 dual-core HiSTBAndroid V500R001C01CP0003 and later versions
- Android 4.2.2 dual-core HiSTBAndroid V500R001C00CP0010 and later versions
- Android 4.4.2 quad-core HiSTBAndroid V600R001C00SPC060 and later versions

8.3.1 What Do I Do to Enable the Portrait Orientation?

Problem Description

The Android UI and videos can be displayed in the portrait orientation. What do I do to enable the portrait orientation?

Solution

Note the following:

- Enable the portrait orientation before compiling the Android version.

  A system prop attribute **persist.sys.screenorientation** is added to allow you to choose from the landscape and portrait orientation during compilation. However, the display direction cannot be dynamically switched when the system is running.

  The file to be modified during compilation configuration varies according to the Android version.

  - For Android 4.2.2, change the value of **persist.sys.screenorientation** in **device/hisilicon/***{CHIPNAME}***/device.mk**.

    {CHIPNAME} indicates the chip name. For example, for Hi3716C V200, modify **device/hisilicon/Hi3716CV200/device.mk**.

  - For Android 4.4.2, change the value of **persist.sys.screenorientation** in **device/hisilicon/Hi3798MV100/device.mk**.

    Change the value of **persist.sys.screenorientation** as follows:

    - Enable the portrait orientation

      persist.sys.screenorientation=portrait

    - Disable the portrait orientation (enable the landscape orientation)

      persist.sys.screenorientation=landscape

- To display the startup logo, fastplay, and Android boot animation in the portrait orientation, add support for the portrait orientation when creating **logo.img**, **fastplay.img**, and **bootanimation.zip**.

- Creating **logo.img** that supports portrait orientation

  First, create the image for portrait orientation. Compared with the image for landscape orientation, the image for portrait orientation is rotated by 90 degrees clockwise. The image format can be JPEG, PNG, BMP, or GIF.

  Second, start the HiTool, choose HiFastplay, click **Logo Setting**, add the created image, and then save the image to generate **logo.img** for portrait orientation.
- Creating **fastplay.img** that supports portrait orientation

  First, create the TS for portrait orientation. Compared with the TS for landscape orientation, the TS for portrait orientation is rotated by 90 degrees clockwise. The format of the video in the TS can be MPEG2, MPEG4, or H.264, and the format of the audio in the TS can be MP3 or MP2. The TS size can be 50 MB at the maximum.

  Second, start the HiTool, choose HiFastplay, click **Local Play Setting**, add the created TS, and set the video PID, audio PID, video type, and audio type based on the TS information. Then save the TS to generate **fastplay.img** for portrait orientation.
- Creating **bootanimation.zip** that supports portrait orientation

  Compared with **bootanimation.zip** for landscape orientation, all images in **bootanimation.zip** for portrait orientation are rotated by 90 degrees clockwise. Create all the images for portrait orientation, and then generate **bootanimation.zip** for portrait orientation by following the standard procedures for creating **bootanimation.zip**.

8.4 Support for Fast Boot

8.4.1 What Do I Do to Enable the Fast Boot Feature?

Problem Description

The fast boot feature reduces the startup time for the Android version by 3s. What do I do to enable the fast boot feature?

Solution

When the fast boot feature is enabled, note the following:

- Before the Android version is compiled, configuration should be performed to enable the fast boot feature.
- Two system attributes persist.sys.zygote.optimize and persist.sys.boot.optimize are added. Before compiling the Android version, configure the following attributes in **/device/hisilicon/{CHIPNAME}/customer.mk**:
  - persist.sys.zygote.optimize
  - persist.sys.boot.optimize

  true indicates that two system attributes are enabled and **false** indicates that two system attributes are disabled. The default value is **true**.

NOTE

{CHIPNAME} indicates the chip name. For example, for Hi3798M V100, **/device/hisilicon/{CHIPNAME}/customer.mk** needs to be changed to **device/hisilicon/Hi3798MV100/customer.mk**.

After the version is burnt, to enable the fast boot feature, configure the serial ports by running the following commands:

```
setprop persist.sys.zygote.optimize true and setprop
persist.sys.boot.optimize true
```

Run the following commands to disable the fast boot feature:

```
setprop persist.sys.zygote.optimize false and setprop
persist.sys.boot.optimize false
```

Run the following commands to obtain the values of these two system attributes:

```
getprop persist.sys.zygote.optimize
getprop persist.sys.boot.optimize
```

where

- **true** indicates that two system attributes are enabled.
- **false** indicates that two system attributes are disabled.