



Android Customization
User Guide

| | |
|-------|------------|
| Issue | 04 |
| Date | 2016-03-02 |

Copyright © HiSilicon Technologies Co., Ltd. 2016. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon Technologies Co., Ltd.

Trademarks and Permissions



HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

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 |
|--------------|----------|
| Hi3798C | V1XX |
| Hi3798C | V2XX |
| Hi3796C | V1XX |
| Hi3798M | V1XX |
| Hi3796M | V1XX |
| HiSTBAndroid | V600R001 |
| HiSTBAndroid | V600R002 |

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 04 (2016-03-02)

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

HiSTBAndroid is supported.

Issue 03 (2016-01-22)

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

Section 1.5 is added.

Issue 02 (2015-03-10)

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

The compilation switch **SUPPORT_OTT_DEMO** is changed to **PRODUCT_TARGET** in section 1.1.

Issue 01 (2014-11-15)

This issue is the first official release.



Contents

| | |
|--|-----------|
| About This Document..... | i |
| 1 Configurations of Environment Variables..... | 2 |
| 1.1 OTT Compilation Switch | 2 |
| 1.2 Media Scan | 3 |
| 1.3 Default Storage Device | 3 |
| 1.4 APK Scan During Startup..... | 4 |
| 1.5 Online Upgrade for BGA and QFP Using a Unified Upgrade Package | 5 |
| 1.5.1 Version Compilation | 6 |
| 1.5.2 Upgrade for the Live Network | 7 |
| 2 System Upgrade..... | 8 |
| 2.1 Generating the OTA Upgrade Package..... | 8 |
| 2.2 Modifying Partition Information When Upgrading the System | 8 |
| 2.3 Changing the GPIO Interface for the Reset Button | 9 |
| 2.4 Enabling the LED to Blink During the Upgrade | 10 |
| 2.5 Upgrading the System Online | 11 |
| 2.5.1 Online Forcible Upgrade..... | 11 |
| 2.5.2 Online Non-Forcible Upgrade | 13 |
| 2.6 Upgrading the Startup Logo, Fastplay, and Startup Animation Online | 17 |
| 2.6.1 Upgrading the Startup Logo..... | 17 |
| 2.6.2 Upgrading the Fastplay | 18 |
| 2.6.3 Upgrading the Boot Animation | 19 |
| 3 Production and Debugging Tools..... | 21 |
| 3.1 FactoryTest | 21 |
| 3.2 HardwareTest | 23 |
| 3.2.1 HDMITest | 23 |
| 3.2.2 CVBSTest | 24 |
| 3.2.3 SataAndUsbTest..... | 25 |
| 3.3 ErrorReport | 26 |
| 3.3.1 Setting the Recipients | 26 |
| 3.3.2 Using the ErrorReport..... | 27 |
| 4 Others | 28 |



| | |
|---|----|
| 4.1 Configuring the Power Indicator | 28 |
| 4.1.1 Configuring the Monochrome LED Indicator..... | 28 |
| 4.1.2 Configuring the Bichrome LED Indicator | 28 |
| 4.2 Displaying Boot Instructions..... | 30 |
| 4.3 Using the USB Handle | 30 |
| 4.4 Changing the Multiscreen Device Name..... | 31 |
| 4.5 Changing the Download Address for the Two-Dimensional Code in the Boot Instructions | 32 |
| 4.6 Setting Default Values for Parameters on the Settings Screen | 32 |



Figures

| | |
|---|----|
| Figure 1-1 Default storage equipment option in Settings | 4 |
| Figure 1-2 Process of online upgrade for BGA and QFP using a unified upgrade package | 6 |
| Figure 2-1 System upgrade information (1) | 12 |
| Figure 2-2 System upgrade information (2) | 13 |
| Figure 2-3 Starting HisiliconSetting | 14 |
| Figure 2-4 Selecting System | 14 |
| Figure 2-5 Selecting Upgrade Online | 15 |
| Figure 2-6 Displayed dialog box | 15 |
| Figure 2-7 Downloading the upgrade package | 16 |
| Figure 2-8 Downloading completion | 16 |
| Figure 2-9 Logo upgrade information | 18 |
| Figure 2-10 Fastplay upgrade information | 19 |
| Figure 2-11 Boot animation upgrade information | 20 |
| Figure 3-1 FactoryTest | 22 |
| Figure 3-2 Specified stream files of the HDMI certification | 23 |
| Figure 3-3 Test item list | 24 |
| Figure 3-4 Selecting a stream and display format | 25 |
| Figure 3-5 Test item screen | 26 |
| Figure 3-6 mail.properties | 27 |
| Figure 3-7 ErrorReport | 27 |
| Figure 4-1 Configuring the monochrome LED indicator | 28 |
| Figure 4-2 Normal settings | 31 |
| Figure 4-3 Changing the value of the character string | 31 |
| Figure 4-4 Modifying the code | 32 |
| Figure 4-5 Dialog box indicating that an application has stopped | 34 |
| Figure 4-6 Dialog box indicating that a process has stopped | 35 |



1 Configurations of Environment Variables

1.1 OTT Compilation Switch

Problem Description

If the **PRODUCT_TARGET:=demo** compilation switch is enabled, the following items are configured:

- The system default launcher (launcher 2) is deleted, ensuring that there is only one launcher in the system.
- The system native development tool is deleted (the tool APK used during development is deleted).
- The over-the-air (OTA) upgrade demo HiOTAUpgrade is deleted (the tool APK used during development is deleted).
- The RunInstructions and HiInstructions are added to introduce the usage of the STB during the first startup.
- The file manager demo HiFileManager is deleted (the media center is retained, but applications with the same functions are deleted).
- The factory test tool demo HiFactoryTest is deleted (the tool APK used during development is deleted).
- The dual-network card configuration demo HiDualNetDemo is deleted (the tool APK used during development is deleted).
- The standby mode selection demo HiSuspendMode is deleted (the tool APK used during development is deleted).
- The debug toolkit HiDebugKit is deleted (the tool APK used during development is deleted).
- The transition animation demo HiAniApp is deleted (the tool APK used during development is deleted).
- The HiErrorReport is added to provide feedback on bugs found during the using process.
- The media recording demo HiMediaRecorderTest is deleted (the tool APK used during development is deleted).
- The aging test demo HiAgingTest is deleted (the tool APK used during development is deleted).



- The recording demo SpeechRecorder is deleted (the tool APK used during development is deleted).

How do I enable/disable the compilation of the OTT version?

Solution

To enable/disable the OTT compilation switch, modify the following code in **device/hisilicon/xxxx/customer.mk**:

```
PRODUCT_TARGET := demo
```

- When **PRODUCT_TARGET** is **demo**, the preceding functions are enabled.
- When **PRODUCT_TARGET** is another value, the preceding functions are disabled.



NOTE

The default value of **PRODUCT_TARGET** is **ott**.

1.2 Media Scan

Problem Description

The environment variable **ro.mediaScanner.enable** specifies whether the Android media scan function is enabled in the current version.

How do I enable/disable the Android media scan function?

Solution

To enable/disable the Android media scan function, modify the following code in **device/hisilicon/Hi3798MV100/device.mk**:

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

- When **ro.mediaScanner.enable** is **true**, the Android media scan function is enabled.
- When **ro.mediaScanner.enable** is **false**, the Android media scan function is disabled.



NOTE

The default value of **ro.mediaScanner.enable** is **false**.

1.3 Default Storage Device

Problem Description

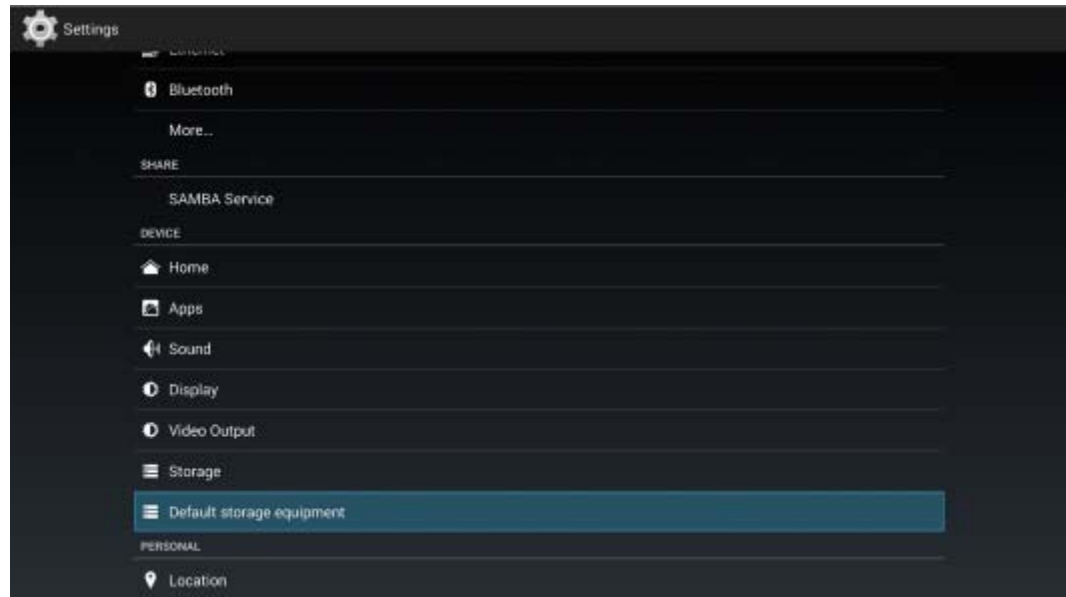
The environment variable **ro.defaultStorage.enable** specifies whether the function of selecting the default storage device is enabled in the current version.

When this function is enabled, the **Default storage equipment** option is added in the Android native **Settings** menu. This function allows you to set any storage device such as an external



USB flash drive or removable hard disk as the default storage device of the Android system. Installed APKs can be ported to the default storage device, and application files in the embedded SD card are also stored in the specified storage device. See [Figure 1-1](#).

Figure 1-1 Default storage equipment option in Settings



How do I enable/disable the function of setting the default storage device?

Solution

To enable/disable the function of setting the default storage device, modify the following code in **device/hisilicon/Hi3798MV100/device.mk**:

```
# DEFAULT STORAGE
PRODUCT_PROPERTY_OVERRIDES += \
ro.defaultStorage.enable=false
```

- When **ro.defaultStorage.enable** is **true**, the preceding function is enabled.
- When **ro.defaultStorage.enable** is **false**, the preceding function is disabled.



NOTE

The default value of **ro.defaultStorage.enable** is **false**.

1.4 APK Scan During Startup

Problem Description

The environment variable **ro.laterscan.enable** specifies whether the function of optimizing the APK scan during startup is enabled.



After this function is enabled, the Android system scans for applications in the **/data/app** directory after startup but not during startup. In this way, the startup speed is the same no matter how many third-party applications are installed.

How do I enable/disable the startup scan optimization function?

Solution

To enable/disable the startup scan optimization function, modify the following code in **device/hisilicon/Hi3798MV100/device.mk**:

```
PRODUCT_PROPERTY_OVERRIDES += \  
ro.laterscan.enable=true
```

- When **ro.laterscan.enable** is **true**, the preceding function is enabled.
- When **ro.laterscan.enable** is **false**, the preceding function is disabled.



NOTE

The default value of **ro.laterscan.enable** is **true**.

1.5 Online Upgrade for BGA and QFP Using a Unified Upgrade Package

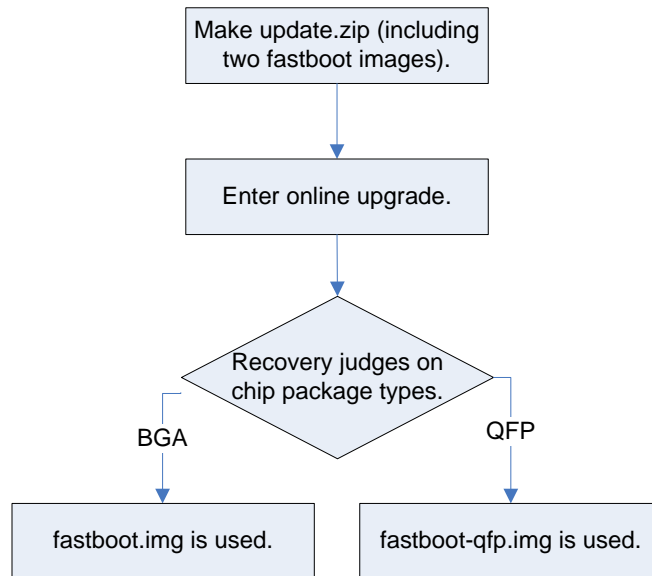
This feature allows the BGA and QFP boards to share the same **update.zip**. In the configuration file, two board tables are specified to the BGA and QFP respectively and two boot images (one fastboot for the BGA and one fastboot for the QFP) are generated during compilation. During upgrade, the recovery will automatically select the corresponding fastboot in **update.zip** based on the chip package type.

The following three aspects are concerned:

- During the compilation of the update package, two fastboot images are generated in **update.zip** (**fastboot.img** is used for the BGA and **fastboot-qfp.img** is used for the QFP).
- The upgrade script of **update.zip** is modified to distinguish the two fastboot images through `getprop`.
- The recovery upgrades the corresponding fastboot image by distinguishing the chip package type.



Figure 1-2 Process of online upgrade for BGA and QFP using a unified upgrade package



CAUTION

This function applies only to Hi3798M V100 and supports only the eMMC flash.

1.5.1 Version Compilation

The compilation control macro `SUPPORT_UNIFIED_UPDATE` is added to control the upgrade enable of the unified upgrade package.

- `SUPPORT_UNIFIED_UPDATE` is defined in `device/hisilicon/Hi3798MV100/customer.mk`.
- When `SUPPORT_UNIFIED_UPDATE` is **true**, the preceding function is enabled.
 - For non-secure version, two fastboot images are generated in `out/target/product/Hi3798MV100/Emmc` during compilation.
 - `fastboot.bin` is used for the BGA.
 - `fastboot-qfp.bin` is used for the QFP.`update.zip` contains two fastboot images:
 - `fastboot.img` is used for the BGA.
 - `fastboot-qfp.img` is used for the QFP.
 - For the level-2 secure solution version, two signed fastboot images are generated in `out/target/product/Hi3798MV100/Security_L2/MAINTAIN` during compilation.
 - `fastboot.bin` is used for the BGA.
 - `fastboot-qfp.bin` is used for the QFP.`update.zip` contains two fastboot images:
 - `fastboot.img` is used for the BGA.
 - `fastboot-qfp.img` is used for the QFP.



Two unsigned fastboot images are generated in **out/target/product/Hi3798MV100/Security_L2/PRODUCTION** during compilation.

➤ **fastboot.bin** is used for the BGA.

➤ **fastboot-qfp.bin** is used for the QFP.

update.zip is same as that under the MAINTAIN directory.

- When **SUPPORT_UNIFIED_UPDATE** is **false**, the preceding functions are disabled and one fastboot image is generated during compilation. **SUPPORT_UNIFIED_UPDATE** is **false** by default.

The QFP eMMC fastboot configuration options **EMMC_BOOT_CFG_NAME_2** and **EMMC_BOOT_REG_NAME_2** are added.

- **EMMC_BOOT_CFG_NAME_2** and **EMMC_BOOT_REG_NAME_2** are defined in **device/hisilicon/Hi3798MV100/BoardConfig.mk**.
- When **SUPPORT_UNIFIED_UPDATE** is **true**, **EMMC_BOOT_CFG_NAME_2** and **EMMC_BOOT_REG_NAME_2** are valid.
 - **EMMC_BOOT_CFG_NAME** and **EMMC_BOOT_REG_NAME** are used to specify the BGA fastboot configuration file. For HiSilicon mass-produced boards, 1a, 1b, 1c, and 1d can be specified.
 - **EMMC_BOOT_CFG_NAME_2** and **EMMC_BOOT_REG_NAME_2** are used to specify the QFP fastboot configuration file. For HiSilicon mass-produced boards, 1f and 1g can be specified.

For example:

```
EMMC_BOOT_CFG_NAME :=
hi3798mdmola_hi3798mv100_ddr3_1gbyte_16bitx2_4layers_emmc.cfg
EMMC_BOOT_REG_NAME :=
hi3798mdmola_hi3798mv100_ddr3_1gbyte_16bitx2_4layers_emmc.reg
EMMC_BOOT_CFG_NAME_2 :=
hi3798mdmolf_hi3798mv100_ddr3_1gbyte_16bitx2_2layers_emmc.cfg
EMMC_BOOT_REG_NAME_2 :=
hi3798mdmolf_hi3798mv100_ddr3_1gbyte_16bitx2_2layers_emmc.reg
```

- The eMMC fastboot configuration file is located in **device/hisilicon/bigfish/sdk/source/boot/sysreg**.

1.5.2 Upgrade for the Live Network

If the software version used by the STB of the live network is not compiled when **SUPPORT_UNIFIED_UPDATE** is **true**, and you want to use the BGA and QFP unified upgrade package for the future online upgrade, do as follows:

- Step 1** Compile **recovery.img** when **SUPPORT_UNIFIED_UPDATE** is **true** to create a **update.zip** that upgrades only **recovery.img** in online mode.

In this way, the recovery image of the STB supports the function of automatically selecting corresponding fastboot image based on the chip package type.

- Step 2** Perform online upgrade by using the BGA and QFP unified upgrade package.

----End



2 System Upgrade

2.1 Generating the OTA Upgrade Package

Problem Description

How do I generate the OTA upgrade package?

Solution

Assume that the package of version A is **update_A.zip**, and that of version B is **update_B.zip**. To upgrade version A to version B, create the OTA upgrade package by running the following command:

```
./device/hisilicon/bigfish/upgrade/ota/ota_from_target_files -i  
update_A.zip update_B.zip ota_A_to_B.zip
```

ota_A_to_B.zip is the OTA package for upgrading version A to version B.

Note the following:

- The OTA package applies to minor versions but not major versions because it features small size.
- Before creating the OTA package, ensure that both versions are the eng version or user version.

2.2 Modifying Partition Information When Upgrading the System

Problem Description

How do I modify the partition information when upgrading the system?

Solution

For details, see section 4.2.5 in the *Android Solution User Guide*.



2.3 Changing the GPIO Interface for the Reset Button

Problem Description

If the **Reset** button is pressed and then the STB is powered on, the system enters the upgrade mode forcibly. In this mode, the system checks whether **update.zip** exists in the root directory of the external USB flash drive. If yes, the system performs upgrade by using this file; if not, the system displays a message indicating the upgrade failure. How do I change the GPIO interface for the reset button?

Solution

Step 1 Change the GPIO interface.

Change the corresponding register value in the code of the static int `check_gpio_recovery(void)` function in **device/hisilicon/bigfish/sdk/source/boot/product/android/recovery.c** based on the system version. For details about the register values (red characters in the following code), see the corresponding data sheet or consult the hardware engineer.

```
static int check_gpio_recovery(void)
{
    int t;
    int count = 2;
    long long chipid;
    chipid = get_chipid();
    printf("get chipid =%llx\n",chipid);
    switch(chipid){
        case _HI3716C_V200:
            printf("get chipType (HI3716C_V200)\n");
#ifdef HI_OTTGPIIO_SUPPORT
            /* get gpio5_1 button */
            while((((REG_VAL(0xF80043FC) & 0x02)>>1) == 0) && (count > 0)){
                udelay(1000*1000);
                count--;
            }
#endif
            break;
#ifdef defined(CHIP_TYPE_hi3798cv100) || defined(CHIP_TYPE_hi3798cv100_a)
        case _HI3798CV100:
        case _HI3798CV100A:
            printf("get chipType (HI3798C_V100)\n");
            // get GPIO_STB1 button
            REG_VAL(0xF8008038) = 0x1 ; // Set the GPIO direction register
            while(((REG_VAL(0xF80093FC) & 0x1) == 0) && (count > 0)){
                udelay(1000*1000);
                count--;
            }
#endif
    }
```



```
        }  
        break;  
#endif  
    }  
    printf("count=%x\n", count);  
    if(count > 0){  
        return HI_FAILURE;  
    }else{  
        return HI_SUCCESS;  
    }  
}
```

Step 2 Add the new version.

Add the version after the switch code in the preceding function. The `get_chipid()` function will obtain the chip ID. Add the code according to the code of other versions. For details about the registers, see the GPIO section in the peripheral chapter of the data sheet.

----End

- If **update.zip** cannot be found in the root directory of the USB flash drive, or the USB flash drive is not inserted, pressing the **Reset** button restores factory settings (erase the cache and data).
- If **update.zip** is found in the root directory of the USB flash drive, pressing the **Reset** button enters the upgrade mode.

2.4 Enabling the LED to Blink During the Upgrade

Problem Description

How do I enable the LED to blink during the upgrade (indicating that the system is being upgraded)?

Solution

Add `open_led()` at the beginning of the upgrade script and `close_led()` at the end.

`open_led()`: The LED starts to blink.

`close_led()`: The LED stops blinking.

See the upgrade script **vim bootable/recovery/etc/META-INF/com/google/android/updater-script-emmc**.

For details about the implementation of `open_led()` and `close_led()`, see `flashled()` in **vim bootable/recovery/updater/gpio.c**.



2.5 Upgrading the System Online

2.5.1 Online Forcible Upgrade

2.5.1.1 Setting the Access Address for the Upgrade List

Problem Description

After the system is started, the background service automatically accesses the list of system upgrade packages on the server and downloads the package for upgrading the system if there is any update. How do I change the access address for the upgrade list?

Solution

Modify the following code in

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/SystemUpgrade.java:

```
private final String INTRA_NET_PATH = "no";
```

If **INTRA_NET_PATH** is **no**, there is no access address for upgrade packages, and upgrade is not performed. If **INTRA_NET_PATH** is set to an actual address, for example:

```
INTRA_NET_PATH=http://10.67.212.148/upgrade.xml )
```

Then the system accesses <http://10.67.212.148/upgrade.xml> and parses the upgrade information.

The default value of **INTRA_NET_PATH** is **no**.

2.5.1.2 Setting Upgrade Information

Problem Description

The system will access and parse the XML file on the server. How do I set the upgrade information in the XML file?

Solution

In the XML file, the **CompleteUpgrade** flag contains information of the upgrade package. This flag consists of the following sub flags:

- **MinVersion**: minimum firmware version
- **MaxVersion**: maximum firmware version
- **TargetVersion**: target firmware version
- **Description**: upgrade description
- **FilePath**: download path of the upgrade package
- **ReleaseTime**: release time
- **FileSize**: size of the upgrade package
- **ChangePartition**: whether to change the partition
- **CompulsoryUpgrade**: whether to perform the upgrade forcibly



The system determines whether to perform the upgrade by checking the time in **TargetVersion** (for example, 20140517.203300). See Figure 2-1 and Figure 2-2.

Figure 2-1 System upgrade information (1)

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <HiSTBAndroidUpgrade>
3      <!--Complete upgrade-->
4      <CompleteUpgrade>
5          <!--Support the minimum version-->
6          <MinVersion>
7              Hi3716CV200-eng 4.2.2 JDQ39 eng.wangyang.20130101.203322 test-keys
8          </MinVersion>
9          <!--Support the highest version-->
10         <MaxVersion>
11             Hi3716CV200-eng 4.4.2 JDQ39 eng.wangyang.20140409.203300 test-keys
12         </MaxVersion>
13         <!--The target version-->
14         <TargetVersion>
15             Hi3716CV200-eng 4.4.2 JDQ39 eng.wangyang.20140409.203448 test-keys
16         </TargetVersion>
17         <!--Upgrade Description-->
18         <Description>
19             Solve the following problem:
20             1.Upgrade to Android4.4 version
21             2.Support online upgrade
22         </Description>
23         <!--Upgrade download url-->
24         <FilePath>
25             http://203.195.181.80/e2372aadad7c2b1a/update0411.zip
26         </FilePath>
27         <!--Release time-->
28         <ReleaseTime>
29             2014-04-11
30         </ReleaseTime>
31         <!--Upgrade file size-->
32         <FileSize>
33             270.4MB
34         </FileSize>
35         <!--Whether to change partitions-->
36         <ChangePartition>
37             false
38         </ChangePartition>
39         <!--Whether to forced to upgrade-->
40         <CompulsoryUpgrade>
41             true
42         </CompulsoryUpgrade>
43     </CompleteUpgrade>
44 </HiSTBAndroidUpgrade>
45
46
```



Figure 2-2 System upgrade information (2)

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--OTA upgrade-->
4   <OTAUpgrade>
5     <!--Support system version-->
6     <CurrentVersion>
7       Hi3716CV200-eng 4.4.2 KOT49H eng.wangyang.20140517.185642 test-keys
8     </CurrentVersion>
9     <!--The target version-->
10    <TargetVersion>
11      Hi3716CV200-eng 4.4.2 KOT49H eng.wangyang.20140529.085406 test-keys
12    </TargetVersion>
13    <!--Upgrade Description-->
14    <Description>
15      Solve the following problem:
16      1.Upgrade to Android4.4 version
17      2.Support online upgrade
18    </Description>
19    <!--Upgrade download url-->
20    <FilePath>
21      http://203.195.181.80/e2372aadad7c2b1a/update0517-0529-ota.zip
22    </FilePath>
23    <!--Release time-->
24    <ReleaseTime>
25      2014-05-29
26    </ReleaseTime>
27    <!--Upgrade file size-->
28    <FileSize>
29      37.1MB
30    </FileSize>
31    <!--Whether to forced to upgrade-->
32    <CompulsoryUpgrade>
33      true
34    </CompulsoryUpgrade>
35  </OTAUpgrade>
36 </HiSTBAndroidUpgrade>
37
```

2.5.2 Online Non-Forcible Upgrade

Problem Description

How do I perform online non-forcible upgrade?

Solution

- Step 1** Set the access address for the upgrade list by modifying **device/hisilicon/bigfish/packages/apps/HiSetting/src/com/android/hisiliconsetting/upgrade/OnlineUpgradeActivity.java**. For details about how to modify the file, see section 2.5.1.1 "Setting the Access Address for the Upgrade List."
- Step 2** Set the upgrade information. For details, see section 2.5.1.2 "Setting Upgrade Information."
- Step 3** Choose **HisiliconSetting**, as shown in Figure 2-3.

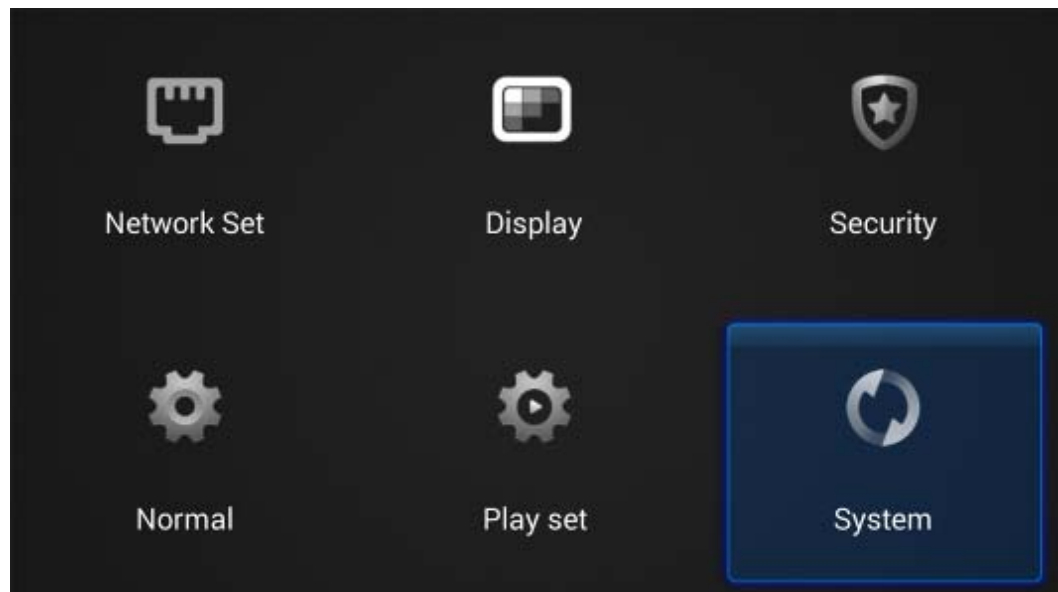


Figure 2-3 Starting HisiliconSetting



Step 4 Select **System**, as shown in [Figure 2-4](#).

Figure 2-4 Selecting System



Step 5 Select **Upgrade Online**, as shown in [Figure 2-5](#).

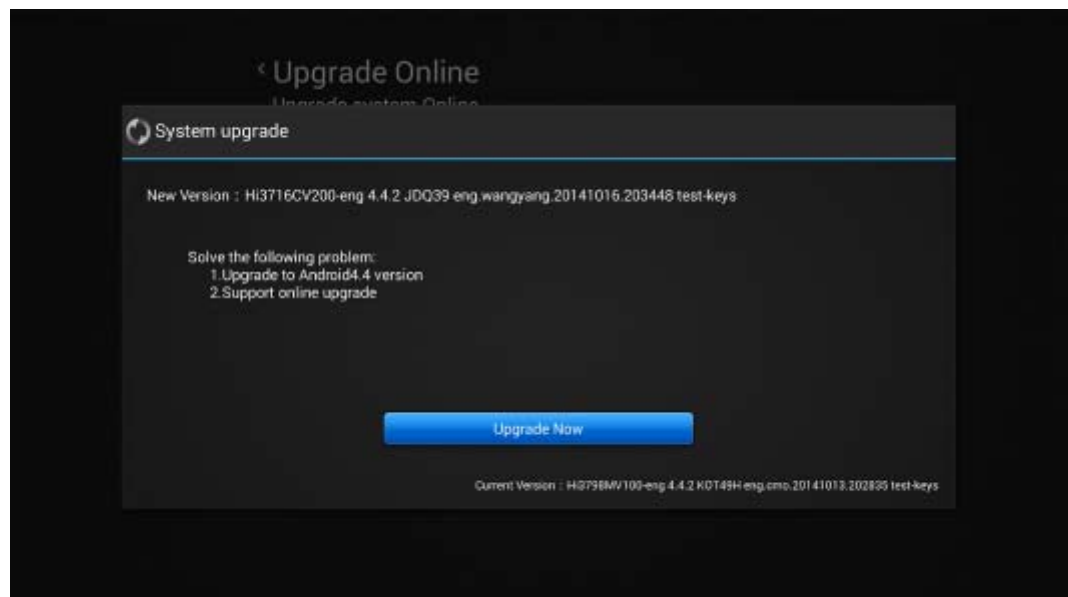


Figure 2-5 Selecting Upgrade Online

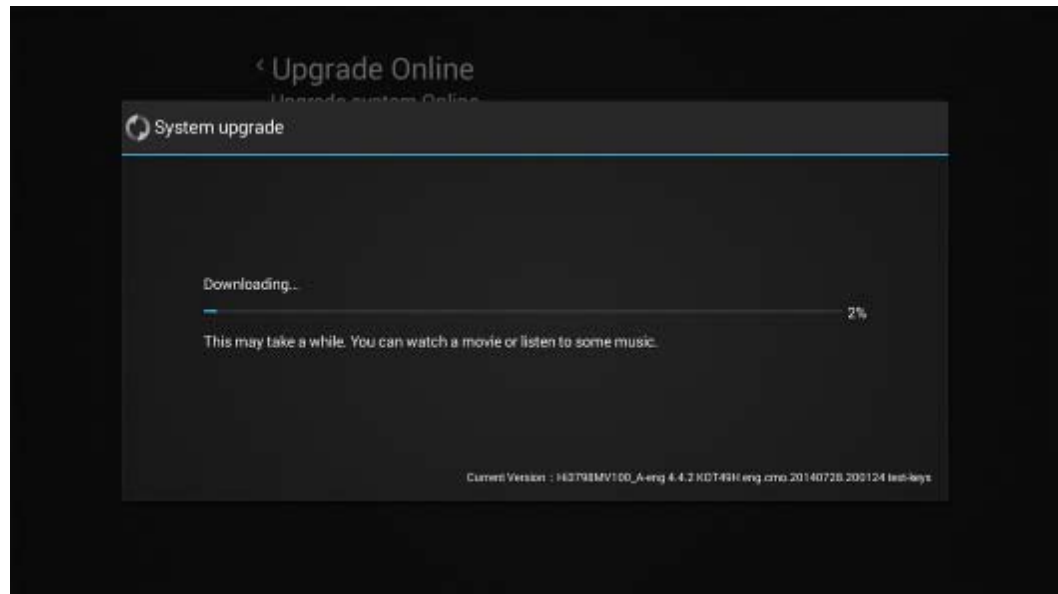


If a new upgrade package is available, a dialog box is displayed, as shown in [Figure 2-6](#).

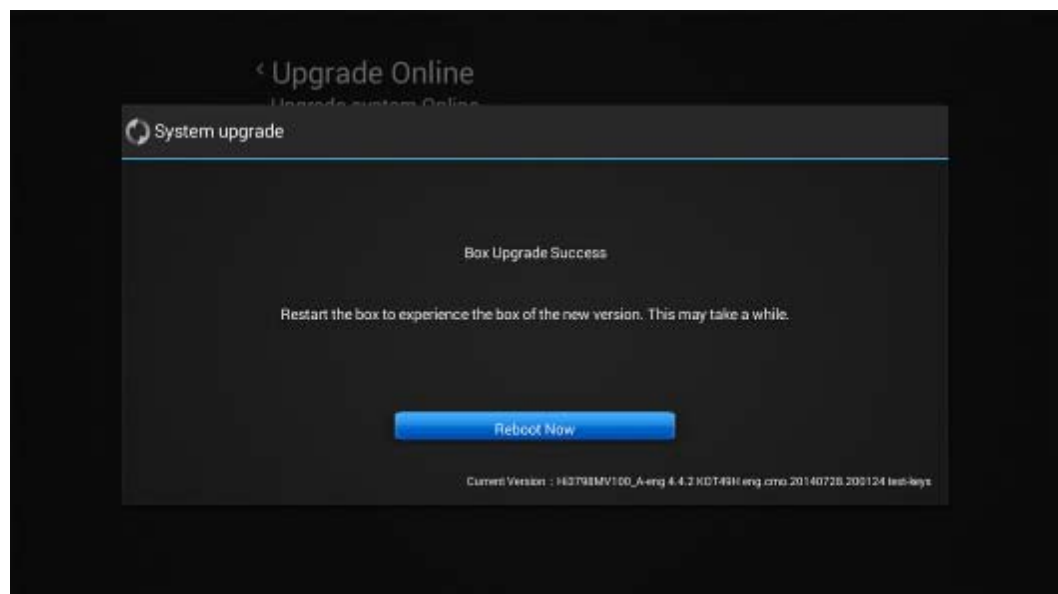
Figure 2-6 Displayed dialog box



Step 6 Select **Upgrade Now** to download the package, as shown in [Figure 2-7](#).

Figure 2-7 Downloading the upgrade package

After the upgrade package is downloaded, a dialog box is displayed, asking you to restart the system, as shown in [Figure 2-8](#).

Figure 2-8 Downloading completion

Step 7 Select **Reboot Now** to restart and upgrade the system.

----End



2.6 Upgrading the Startup Logo, Fastplay, and Startup Animation Online

2.6.1 Upgrading the Startup Logo

2.6.1.1 Setting the Access Address for the Upgrade List

Problem Description

After the system is started, the background service automatically accesses the upgrade list on the server and downloads the upgrade package for the startup logo if there is any update. How do I change the access address for the upgrade list?

Solution

Modify the following code in

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/LogoUpgrade.java:

```
private final String INTRA_NET_PATH = "no";
```

If **INTRA_NET_PATH** is **no**, there is no access address for upgrade packages, and upgrade is not performed.

If **INTRA_NET_PATH** is set to an actual address, for example, <http://10.67.212.148/logo.xml>, the system accesses <http://10.67.212.148/logo.xml> and parses the upgrade information.

The default value of **INTRA_NET_PATH** is **no**.

2.6.1.2 Setting Upgrade Information

Problem Description

The system will access and parse the XML file on the server. How do I set the upgrade information in the XML file?

Solution

In the XML file, the **LogoUpgrade** flag contains the upgrade information. This flag consists of the following sub flags:

- **FilePath**: download path of the upgrade package
- **ReleaseTime**: release time
- **FileSize**: size of the upgrade package

The system determines whether to upgrade the startup logo by checking the release time. See [Figure 2-9](#).



Figure 2-9 Logo upgrade information

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--Logo Upgrade-->
4   <LogoUpgrade>
5     <!--Upgrade download url-->
6     <FilePath>
7       http://10.67.212.148/v5/wy/logo-wanglin.img
8     </FilePath>
9     <!--Release time-->
10    <ReleaseTime>
11      2014-04-01
12    </ReleaseTime>
13    <!--Upgrade file size-->
14    <FileSize>
15      168KB
16    </FileSize>
17  </LogoUpgrade>
18 </HiSTBAndroidUpgrade>
```

2.6.2 Upgrading the Fastplay

2.6.2.1 Setting the Access Address for the Upgrade List

Problem Description

After the system is started, the background service automatically accesses the upgrade list on the server and downloads the upgrade package for the fastplay if there is any update. How do I change the access address for the upgrade list?

Solution

Modify the following code in **device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/FastplayUpgrade.java**:

```
private final String INTRA_NET_PATH = "no";
```

If **INTRA_NET_PATH** is **no**, there is no access address for upgrade packages, and upgrade is not performed.

If **INTRA_NET_PATH** is set to an actual address, for example, `http://10.67.212.148/fastplay.xml`, the system accesses `http://10.67.212.148/fastplay.xml` and parses the upgrade information.

The default value of **INTRA_NET_PATH** is **no**.

2.6.2.2 Setting Upgrade Information

Problem Description

The system will access and parse the XML file on the server. How do I set the upgrade information in the XML file?



Solution

In the XML file, the **FastplayUpgrade** flag contains the upgrade information. This flag consists of the following sub flags:

- **FilePath**: download path of the upgrade package
- **ReleaseTime**: release time
- **FileSize**: size of the upgrade package

The system determines whether to upgrade the fastplay by checking the release time. See [Figure 2-10](#).

Figure 2-10 Fastplay upgrade information

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <HiSTBAndroidUpgrade>
3      <!--Fastplay Upgrade-->
4      <FastplayUpgrade>
5          <!--Upgrade download url-->
6          <FilePath>
7              http://10.67.212.148/v5/hcb/fastplay_old.img
8          </FilePath>
9          <!--Release time-->
10         <ReleaseTime>
11             2014-04-01
12         </ReleaseTime>
13         <!--Upgrade file size-->
14         <FileSize>
15             32MB
16         </FileSize>
17     </FastplayUpgrade>
18 </HiSTBAndroidUpgrade>
19
```

2.6.3 Upgrading the Boot Animation

2.6.3.1 Setting the Access Address for the Upgrade List

Problem Description

After the system is started, the background service automatically accesses the upgrade list on the server and downloads the upgrade package for the boot animation if there is any update. How do I change the access address for the upgrade list?

Solution

Modify the following code in

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/AnimationUpgrade.java:

```
private final String INTRA_NET_PATH = "no";
```

If **INTRA_NET_PATH** is **no**, there is no access address for upgrade packages, and upgrade is not performed.



If **INTRA_NET_PATH** is set to an actual address, for example, `http://10.67.212.148/bootanimation.xml`, the system accesses `http://10.67.212.148/bootanimation.xml` and parses the upgrade information.

The default value of **INTRA_NET_PATH** is **no**.

2.6.3.2 Setting Upgrade Information

Problem Description

The system will access and parse the XML file on the server. How do I set the upgrade information in the XML file?

Solution

In the XML file, the **BootAnimationUpgrade** flag contains the upgrade information. This flag consists of the following sub flags:

- **FilePath**: download path of the upgrade package
- **ReleaseTime**: release time
- **FileSize**: size of the upgrade package

The system determines whether to upgrade the boot animation by checking the release time. See [Figure 2-11](#).

Figure 2-11 Boot animation upgrade information

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <HiSTBAndroidUpgrade>
3      <!--BootAnimation Upgrade-->
4      <BootAnimationUpgrade>
5          <!--Upgrade download url-->
6          <FilePath>
7              http://10.67.212.148/v5/wy/bootanimation-cts.zip
8          </FilePath>
9          <!--Release time-->
10         <ReleaseTime>
11             2014-04-26
12         </ReleaseTime>
13         <!--Upgrade file size-->
14         <FileSize>
15             8MB
16         </FileSize>
17     </BootAnimationUpgrade>
18 </HiSTBAndroidUpgrade>
```



3 Production and Debugging Tools

3.1 FactoryTest

Problem Description

The FactoryTest is used for factory tests. How do I use this tool?

Solution

Make preparations as follows before the test:

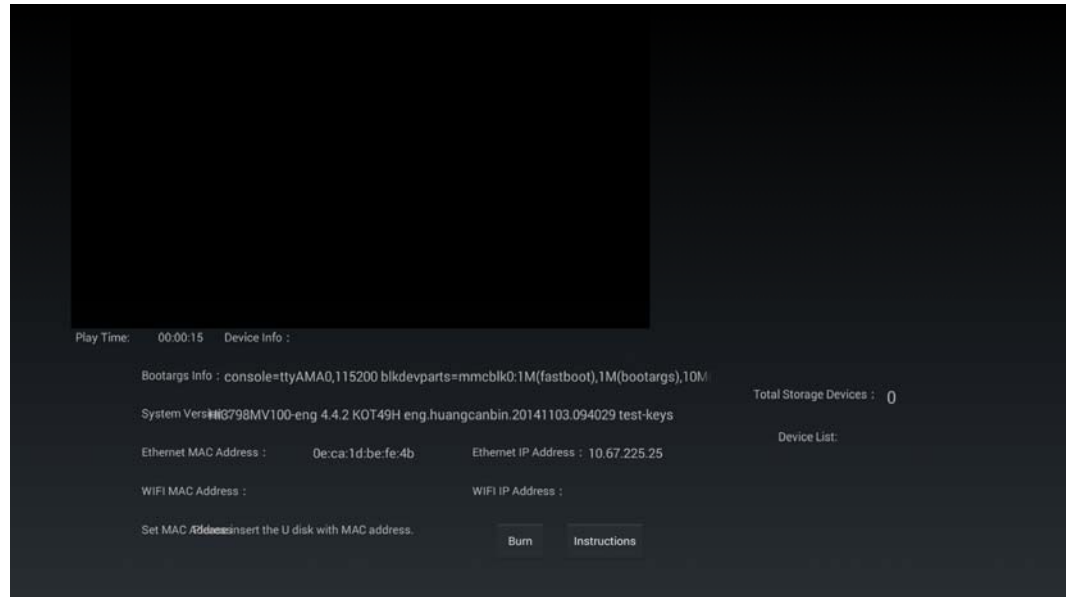
- Router
Enable the DHCP and Wi-Fi, and set the user name to **HiABC**, password to **9988776655**, and encryption mode to **WPA2**.
- Board
 - Insert the network cable. Ensure that the IP address can be obtained by using the DHCP.
 - Insert a USB flash drive to each USB port, and store **usb1.mpg**, **usb2.mpg**, **usb3.mpg**, and **usb4.mpg** in the root directories of the USB flash drives respectively.
 - Insert a hard disk to the SATA port (optional), and store **sata.mpg** in the root directory of the hard disk.
 - Insert an SD card to the SD card port (optional), and store **sdcard.mpg** in the root directory of the SD card.

Pay attention to the following items during tests:

- The video is played repeatedly. No artifact appears and no flicker occurs.
- The wired MAC address and IP address are normal.
- The wireless MAC address and IP address are normal.
- The number and list of connected devices are normal.



Figure 3-1 FactoryTest



The FactoryTest and HiAgingTest icons are hidden by default. To start the FactoryTest APK, press the buttons **5**, **9**, **1**, and **8** in sequence on the remote control in the system native settings UI. To start the HiAgingTest APK, modify the following code in **packages/apps/Settings/src/com/android/settings/Settings.java**:

```
private final String PACKAGE_NAME = "com.android.factorytest";
private final String ACTIVITY_NAME =
    "com.android.factorytest.factorytestActivity";
private int keyTemp = 0;
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (KeyEvent.KEYCODE_5 == keyCode) {
        keyTemp = 1;
    } else if (KeyEvent.KEYCODE_9 == keyCode) {
        if (keyTemp == 1)
            keyTemp++;
        else
            keyTemp = 0;
    } else if (KeyEvent.KEYCODE_1 == keyCode) {
        if (keyTemp == 2)
            keyTemp++;
        else
            keyTemp = 0;
    } else if (KeyEvent.KEYCODE_8 == keyCode) {
        if (keyTemp == 3) {
            try {
                Intent intent = new Intent();
```



```
        ComponentName name = new ComponentName(PACKAGE_NAME,
            ACTIVITY_NAME);
        intent.setComponent(name);
        intent.setAction(Intent.ACTION_VIEW);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        getApplication().startActivity(intent);
    } catch (Exception e) {
        e.printStackTrace();
    }
} else
    keyTemp = 0;
}
return super.onKeyDown(keyCode, event);
}
```

3.2 HardwareTest

3.2.1 HDMITest

Problem Description

The HDMITest is used for HDMI certification tests. How do I use this tool?

Solution








- Step 1** Start the HDMITest.
- Step 2** Select ... to select a stream file.
- Step 3** Select **Start Test** to enter the stream playback screen.



NOTE

The stream file must be a specified stream file of the HDMI certification, as shown in [Figure 3-2](#).

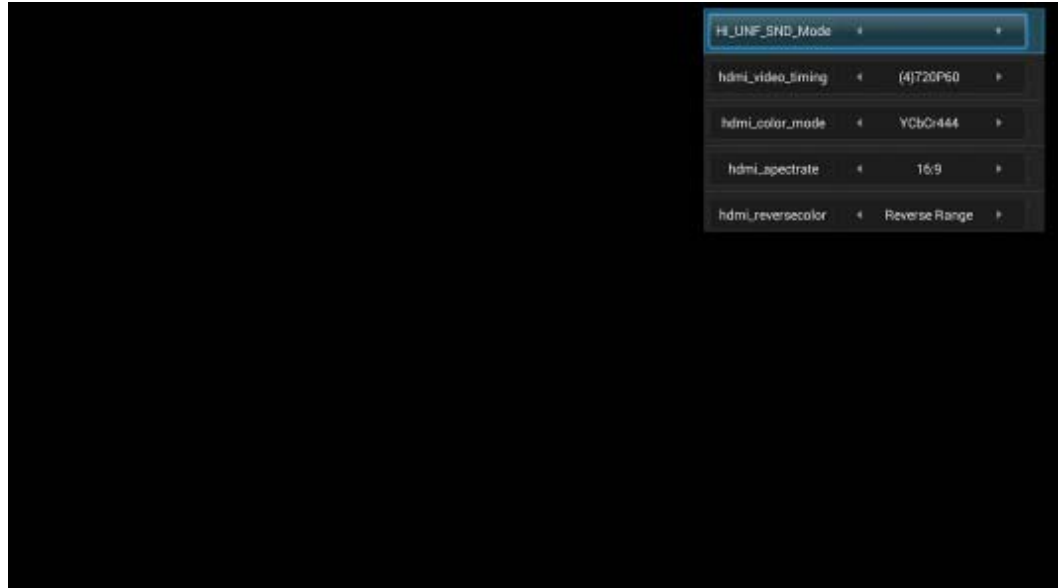
Figure 3-2 Specified stream files of the HDMI certification

| | |
|---|------------------|
|  ccitt_mux_32K.ts | 2010/12/27 15:07 |
|  dada_192KHz_8ch_16bit.wav | 2013/11/16 1:28 |
|  Ghost.MLP | 2010/10/8 15:44 |
|  hd_dts_orchestra_long_lossless_dtshd... | 2010/10/8 15:45 |
|  Matrx625.Trp | 2013/2/20 11:45 |
|  PCM8_96KHz_pcm7.1.m2ts | 2010/7/19 15:48 |
|  serenity_english_5_1_1536.ddp | 2010/10/8 15:44 |



Step 4 Press the **Menu** key on the remote control. The test item list is displayed on the upper right corner of the screen, as shown in [Figure 3-3](#).

Figure 3-3 Test item list



You can press the up and down keys on the remote control to select the test item, press the left and right keys to select the test value, and press **OK** to confirm the selection.

Step 5 Capture and analyze data by using dedicated instruments.

----End

3.2.2 CVBSTest

Problem Description

The CVBSTest is used for CVBS tests. How do I use this tool?

Solution

Step 1 Start the CVBSTest.

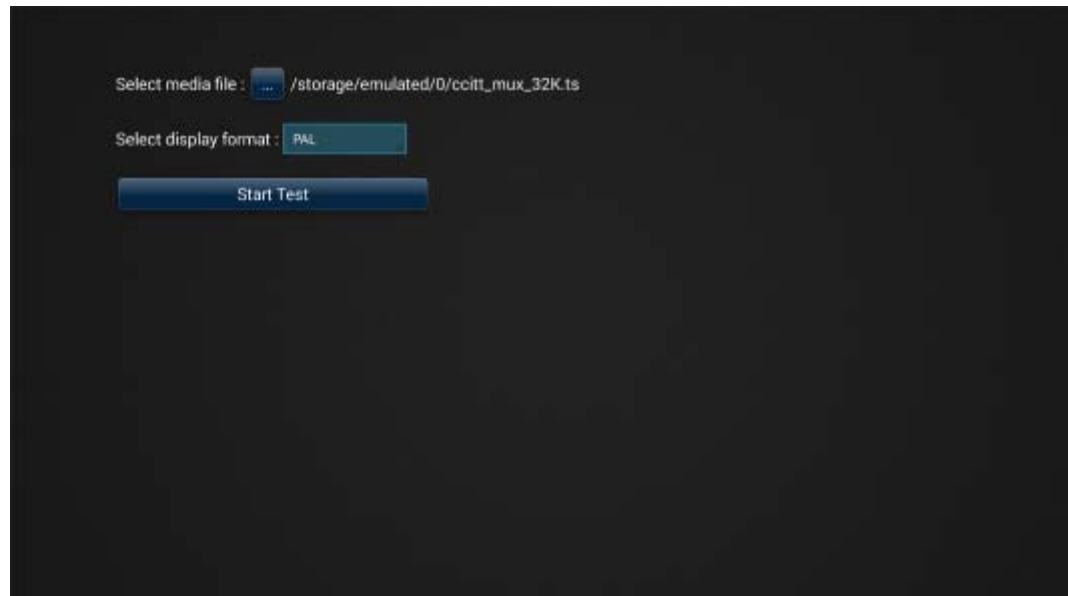
Step 2 Select ... to select a stream file.

The stream file must be a specified stream file for the CVBS test. The specified stream files vary according to the vendor.

Step 3 Select a display format, as shown in [Figure 3-4](#).



Figure 3-4 Selecting a stream and display format



Step 4 Select **Start Test** to enter the stream playback screen.

Step 5 Capture and analyze data by using dedicated instruments.

----End

3.2.3 SataAndUsbTest

Problem Description

The SataAndUsbTest is used for SATA and USB tests. How do I use this tool?

Solution

Step 1 Start the SataAndUsbTest. The screen shown in [Figure 3-5](#) is displayed.



Figure 3-5 Test item screen



Step 2 Select the corresponding test button according to the test requirements.

Step 3 Capture and analyze data by using dedicated instruments.

----End

3.3 ErrorReport

3.3.1 Setting the Recipients

Problem Description

The remote debugging tool ErrorReport can be used to capture device information and logs and send the captured information to specified recipients as an email. How do I set the recipients?

Solution

Modify `/device/hisilicon/bigfish/development/apps/HiErrorReport/assets/mail.properties`. The **to** tag specifies the email addresses of the recipients separated by semicolons (;). The **subject** tag specifies the default subject of the email. The **content** tag specifies the default content of the email. See [Figure 3-6](#).

Figure 3-6 mail.properties

```
f\090AE\u7BB1\u914D\u7F6E
username = debuglog
pwd = 1qaz@WSX
from = debuglog@163.com
to = huawei.wangyang@hisilicon.com;wanting@hisilicon.com;jiangzhonglin@hisilicon.com;wangbaochang@hisilicon.com;
mail_protocol = mail.smtp.protocol
mail_auth = mail.smtp.auth
mail_host = mail.smtp.host
mail_port = mail.smtp.port
protocol = smtp
auth = true
host = smtp.163.com
port = 25
content = \u5B9D\u76D2\u51FA\u9519\u4E86,\u8BB6\u60C5\u8BF7\u770BLOG\u9644\u4EF6.
subject = \u5B9D\u76D2\u51FA\u9519Log\u62A5\u544A.
```

3.3.2 Using the ErrorReport

Problem Description

The ErrorReport is used to capture device information and logs. How do I use this tool?

Solution

Step 1 Start the ErrorReport, as shown in [Figure 3-7](#).

Figure 3-7 ErrorReport

ErrorReport

Error Description

Please Input Content!

QWERTYUIOP
ASDFGHJKL
,ZXCVBNM
Chinese 1123 . ↵

Step 2 Enter the email content. If nothing is entered, the default content is used.

Step 3 Select **SendEmail**.

---End



4 Others

4.1 Configuring the Power Indicator

The power indicator here is the LED indicator driven by the GPIO. It is blue or green during startup and red or colorless during standby. The power indicator can be a monochrome LED indicator or a bichrome LED indicator. The monochrome LED indicator is configured in the **menuconfig** file, and the bichrome LED indicator is configured by modifying the code manually.

4.1.1 Configuring the Monochrome LED Indicator

Problem Description

How do I configure the monochrome LED indicator?

Solution

The monochrome LED indicator driven by the GPIO can be configured in the configuration file **cfg.mak**. Choose **Board > Keyled Config**, select **Gpio Light Support**, and set **Gpio Number**, as shown in [Figure 4-1](#) (**0x2f** indicates GPIO5_7: 5 x 8 + 7).

Figure 4-1 Configuring the monochrome LED indicator

```
Keyled Device Used (Ctl642 Keyled Select) --->
[*] Gpio Light Support
(0x2f) Gpio Number (NEW)
```

Then recompile the boot and kernel images.

4.1.2 Configuring the Bichrome LED Indicator

Problem Description

How do I configure the bichrome LED indicator?



Solution

Step 1 Choose **Board > Keyed Config**, and select **Gpio Light Support**.

Step 2 Modify the code in **source\boot\fastboot\arch\arm\cpu\s5\start.S** of the SDK to turn on the green indicator by configuring registers based on the GPIO port. You need to set the GPIO to output and then specify the GPIO input level. See the following code.

```
#ifndef CONFIG_GPIO_LIGHT_SET
enable_green_light:
    ldr r1, =0xf8004000
    ldr r2, [r1, #0x400]
    orr r2, r2, #0x10
    str r2, [r1, #0x400]
    ldr r2, [r1, #0x40]
    orr r2, r2, #0x10
    str r2, [r1, #0x40]

    ldr r1, =0xf8009000
    ldr r2, [r1, #0x400]
    orr r2, r2, #0x80
    str r2, [r1, #0x400]
    ldr r2, [r1, #0x200]
    and r2, r2, #0x7f
    str r2, [r1, #0x200]
#endif
```

Step 3 Modify the code in the **c51_suspend** function in **source\msp\drv\pm\drv_pmoc_intf.c** of the SDK to turn on the red indicator when the system enters the standby mode.

```
#ifndef HI_GPIO_LIGHT_SUPPORT
    HI_REG_READ32(IO_ADDRESS(0xf8004400), ret);
    ret |= 0x10;
    HI_REG_WRITE32(IO_ADDRESS(0xf8004400), ret);

    HI_REG_READ32(IO_ADDRESS(0xf8004040), ret);
    ret &= 0xef;
    HI_REG_WRITE32(IO_ADDRESS(0xf8004040), ret);

    HI_REG_READ32(IO_ADDRESS(0xf8009400), ret);
    ret |= 0x80;
    HI_REG_WRITE32(IO_ADDRESS(0xf8009400), ret);

    HI_REG_READ32(IO_ADDRESS(0xf8009200), ret);
    ret |= 0x80;
    HI_REG_WRITE32(IO_ADDRESS(0xf8009200), ret);
#endif
```



----End

4.2 Displaying Boot Instructions

Problem Description

The APKs related to boot instructions include the RunInstructions and HiInstructions. What do I do to display the boot instructions?

Analysis

After the system is restored to factory settings, the RunInstructions is started when the system boots for the first time, and the HiInstructions is started by the RunInstructions. After that, the RunInstructions is set to unavailable, ensuring that it is started for only once. The RunInstructions is used to start the HiInstructions, and the actual boot instructions screen is displayed only after the HiInstructions is started. In this way, the HiInstructions is started automatically only during the first startup. The HiInstructions icon is displayed in the application list, allowing you to run the application manually.

Solution

- After the system is restored to factory settings, the RunInstructions and HiInstructions are started when the system boots for the first time.
- After the system is started, you can start the HiInstructions in **Launcher** to display boot instructions.

4.3 Using the USB Handle

Problem Description

What do I do if a USB handle is inserted but cannot be used?

Solution

Typically, the USB handle cannot be used because the corresponding driver is not enabled.

Enable the corresponding driver. Take Hi3798M V100 as an example:

```
cd device/hisilicon/bigfish/sdk/source/kernel/linux-3.10.y/  
cp arch/arm/configs/hi3798cv100_android_defconfig .config  
make ARCH=arm CROSS_COMPILE=arm-hisiv200-linux- menuconfig  
Device Drivers ---> HID support ---> Special HID drivers --->Select the  
driver corresponding to the handle  
cp .config arch/arm/configs/hi3798cv100_android_defconfig  
make distclean
```



Notes

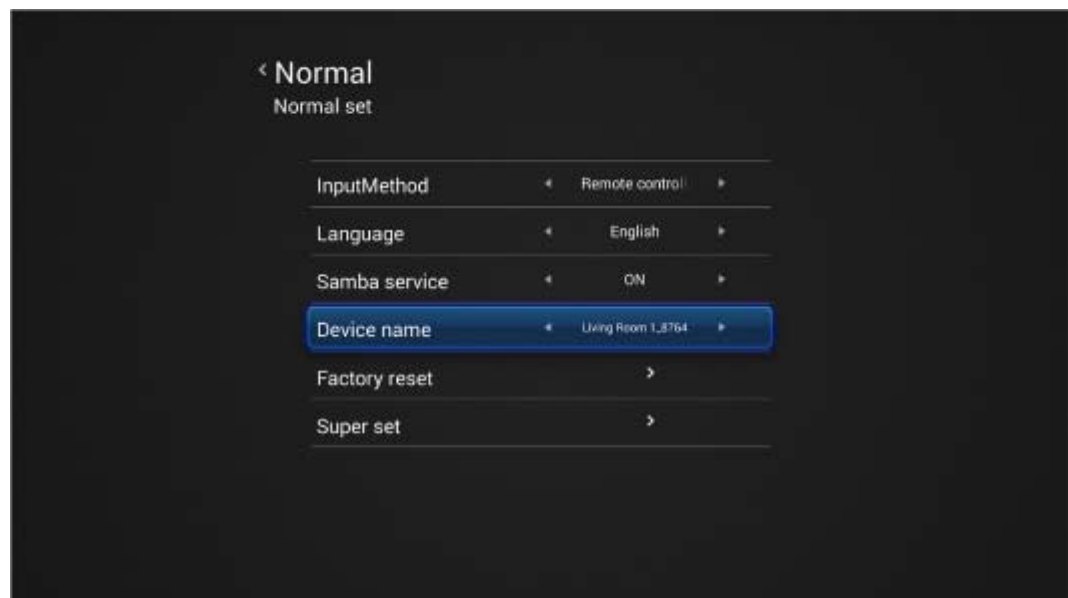
If you do not know which driver corresponds to your handle, enable all drivers under **Special HID drivers**. If the handle still cannot be used, the driver for the handle is not integrated in the kernel. Request the driver from the handle vendor and integrate it into the kernel.

4.4 Changing the Multiscreen Device Name

Problem Description

How do I set the DLNA and Skyplay device name?

Figure 4-2 Normal settings



Solution

Change the value of the character string array **device_name_entries_value** in **device/hisilicon/bigfish/packages/apps/HiSetting/res/values/attrs.xml**. Currently only English is supported. See [Figure 4-3](#).

Figure 4-3 Changing the value of the character string

```
82      <string-array name="device_name_entries_value">
83          <item>Living Room 1</item>
84          <item>Living Room 2</item>
85          <item>Bedroom 1</item>
86          <item>Bedroom 2</item>
```



4.5 Changing the Download Address for the Two-Dimensional Code in the Boot Instructions

Problem Description

The boot instructions contain the download address for the two-dimensional code for multiscreen clients. How do I change this download address to generate a new two-dimensional code?

Solution

Modify row 10 in `device/hisilicon/bigfish/packages/apps/HiInstructions/src/com/hisilicon/instructions/PageActivity3.java`. `mURL` is the URL of the application. See [Figure 4-4](#).

Figure 4-4 Modifying the code

```
8
9 public class PageActivity3 extends PageActivity {
10     private final static String mURL = "http://2.moonlightjewelbox.si
11     private final static int mQRImgWidth = 360;
12     private final static int mQRImgHeight = 288;
13     private boolean isSign;
14
```

4.6 Setting Default Values for Parameters on the Settings Screen

Problem Description

How do I set the default values for parameters on the **Settings** screen?

Solution

- Wired network
 - Wired network switch

Modify `<bool name="def_ethernet_on">true</bool>` in `frameworks\base\packages\SettingsProvider\res/values/defaults.xml`. **true** indicates that the wired network is enabled by default, and **false** indicates that it is disabled by default.
 - DHCP and static IP address

There are no default values.
- Wi-Fi

Modify `<bool name="def_wifi_on">false</bool>` in `frameworks\base\packages\SettingsProvider\res/values/defaults.xml`. **true** indicates that the Wi-Fi is enabled by default, and **false** indicates that it is disabled by default.
- Samba



Modify **private static final String SAMBA_STATUS_DEF_VALUE="true"** in **packages/apps/Settings/src/com/android/settings/MyBroadCastReceiver.java**. **true** indicates that samba is enabled by default when the system is started for the first time or restored to factory settings, and **false** indicates that it is disabled by default.

- Launcher

There is no default value.

- Volume

Modify the variable **int[] DEFAULT_STREAM_VOLUME**, that is, the **STREAM_MUSIC** volume, **frameworks/base/media/java/android/media/AudioManager.java**. The value ranges from 0 to 15, and the default volume is used after the system is started.

- Installation of third-party APKs

Modify **<bool name="def_install_non_market_apps">false</bool>** in **frameworks\base\packages\SettingsProvider\res\values\defaults.xml**. **true** indicates that third-party APKs can be installed by default, and **false** indicates that third-party APKs cannot be installed by default.

- Language

Modify **PRODUCT_LOCALES := en_US** in **build/target/product/full_base.mk**. For example, to set the simplified Chinese as the default language, set **PRODUCT_LOCALES** to **zh_CN**.

- Resolution

Modify the default values of the UI size, customer display format, and display area in the image of the base partition.

To set the default value of the optimal display format, modify the following code in **device/hisilicon/bigfish/frameworks/hidisplaymanager/hal/hi_adp_hdmi.c**:

```
int getOptimalFormat()//0 is disabled; 1 is enabled
{
    char buffer[BUFLLEN];
    int value = 0;
    property_get("persist.sys.optimalfmt.enable", buffer, "1");
    value = atoi(buffer);
    ALOGE("getOptimalFormat, enable = %d", value);
    return value;
}
```

Modify **property_get("persist.sys.optimalfmt.enable", buffer, "1")**. The value **1** indicates that the optimal display format adaptation is enabled by default, and the value **0** indicates that it is disabled by default.

- Display of system error dialog boxes

Modify the following code for processing the **SHOW_ERROR_MSG** event in **/frameworks/base/services/java/com/android/server/am/ActivityManagerService.java**.

```
if (mShowDialogs && !mSleeping && !mShuttingDown&&false) {
    Dialog d = new AppErrorDialog(mContext,
        ActivityManagerService.this, res, proc);
    d.show();
    proc.crashDialog = d;
}
```



```
    } else {  
        // The device is asleep, so just pretend that the  
user  
        // saw a crash dialog and hit "force quit".  
        if (res != null) {  
            res.set(0);  
        }  
    }  
}
```

To display the dialog boxes shown in [Figure 4-5](#) and [Figure 4-6](#) by default, delete **&&false** in the if conditions. Add **&&false** if you do not want to display these dialog boxes by default.

Figure 4-5 Dialog box indicating that an application has stopped

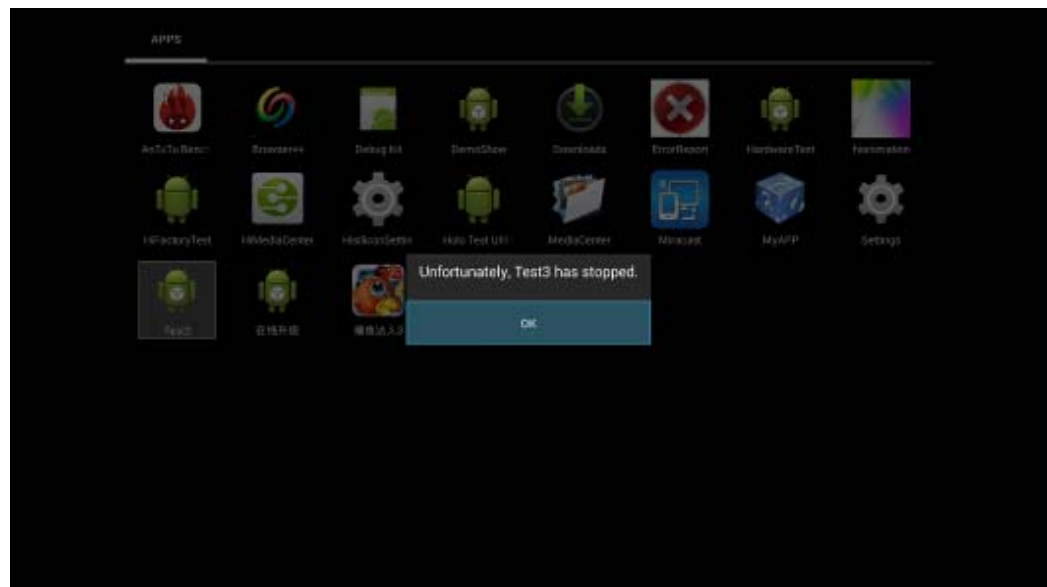


Figure 4-6 Dialog box indicating that a process has stopped

