



Rev 1.0

May.01, 2012

Android SD upgrade application note

© 2010 - 2012 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

Table of Contents

| | | |
|--|---------------------|-----------|
| Rev 1.0 | May.01, 2012 | 1 |
| Table of Contents | | 2 |
| 1 Requirement | | 4 |
| 2 Introduction | | 5 |
| 3 How to build SD upgrade image | | 6 |
| 3.1 Configure for recovery mode | | 6 |
| 3.2 Build full OTA update image step by step | | 6 |
| 3.3 Build differential OTA update image step by step | | 6 |
| 3.3.1 Differential OTA package size | | 6 |
| 3.3.2 Software maintenance | | 6 |
| 3.3.3 Pro and con | | 7 |
| 3.3.4 Step by step to build differential OTA package | | 7 |
| 3.4 Build update image with DSP_BL step by step | | 7 |
| 3.4.1 How to prepare DSP_BL image | | 7 |
| 3.4.2 How to prepare update image with DSP_BL | | 7 |
| 3.5 Note and checking | | 8 |
| 4 How to upgrade firmware by SD card | | 9 |
| 4.1 Upgrade procedure | | 9 |
| 4.2 Error handling | | 10 |
| 4.3 Recovery mode debug | | 10 |
| 5 Other operation | | 11 |
| 5.1 Wipe data/factory reset | | 11 |
| 5.2 Wipe cache partition | | 11 |
| 5.3 Backup user data | | 11 |
| 5.4 Restore user data | | 11 |
| 6 MTK special factory reset | | 12 |
| 6.1 Purpose and How-to | | 12 |
| 6.2 Combinations and Examples | | 12 |

Revision History

| Revision | Author | Date | Description |
|----------|---------|----------------|--|
| 0.1 | WS Chen | 2011 / 02 / 16 | Initial draft |
| 0.2 | WS Chen | 2011 / 04 / 20 | Add wipe data/cache backup and restore user data description |
| 0.3 | WS Chen | 2011 / 05 / 27 | Add introduction |
| 0.4 | WS Chen | 2011 / 06 / 10 | Add how to update DSP_BL |
| 0.5 | WS Chen | 2011 / 06 / 22 | Add how to prepare DSP_BL |
| 0.6 | WS Chen | 2011 / 06 / 30 | Modify DSP_BL session |
| 0.7 | WS Chen | 2011 / 10 / 12 | Add how to prepare differential OTA package |
| 0.8 | WS Chen | 2011 / 10 / 25 | Add sign key option in differential OTA package |
| 0.9 | WS Chen | 2011 / 11 / 21 | Add MTK special factory reset |
| 1.0 | WS Chen | 2012 / 05 / 02 | Add recovery.fstab and debug session |

1 Requirement

This application note is used for Android 2.3 and later firmware upgrade through SD card. The Android 2.1 and 2.2 is not suitable for this document.

Key usage:

Volume Up + Power -> Enter recovery mode

Home -> Enter recovery mode menu

Menu -> Select

Volume Up -> Move up

Volume Down -> Move down

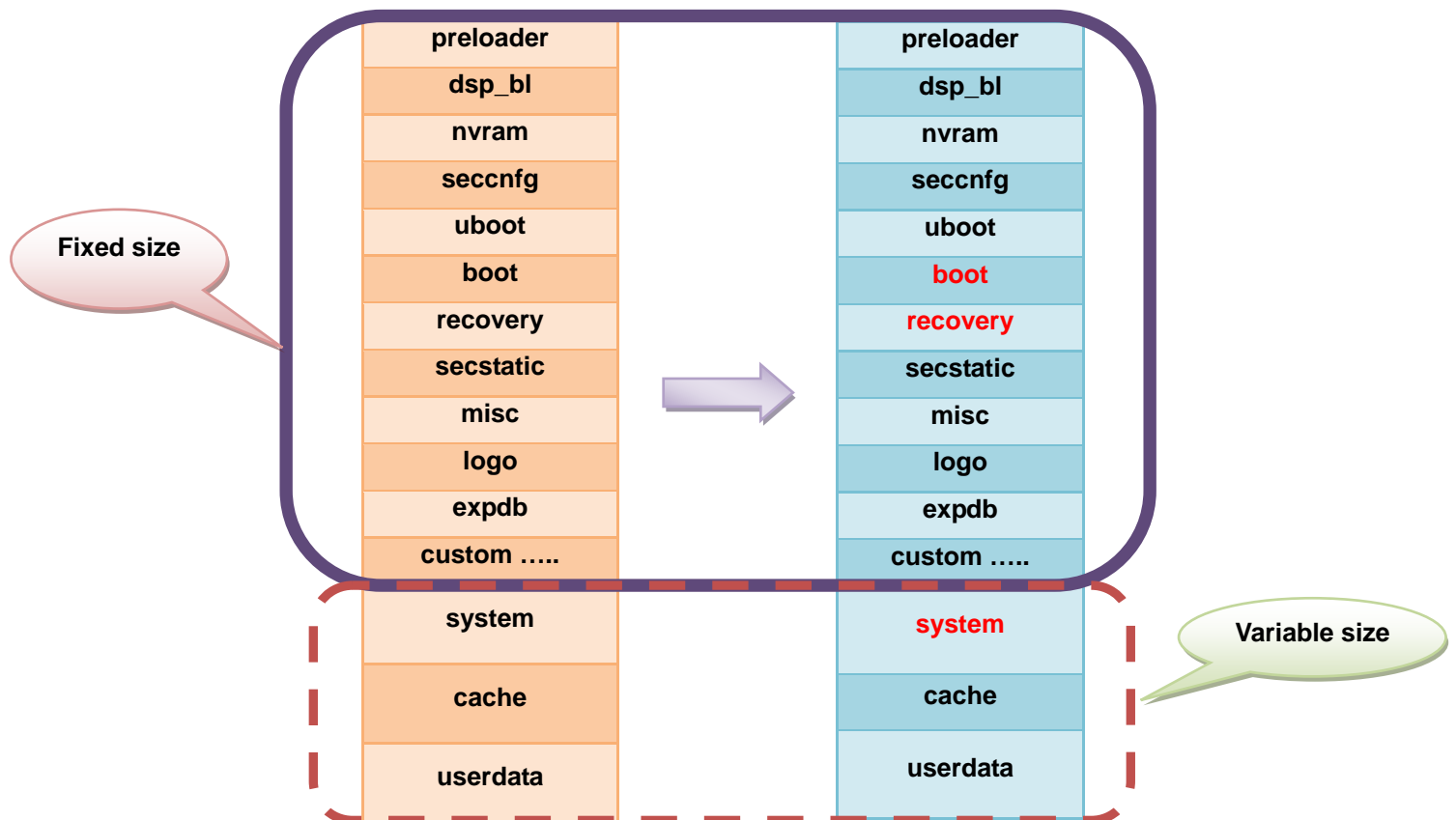
2 Introduction

The SD upgrade 2.0 which supports:

- Changed partition size upgrade (system/cache/userdata). eMMC phone does **NOT** support it.
- Backup and restore user data.

To support these features, there are some common rules must be observed.

- The number of partitions **MUST BE** consistent between the two versions.
- The order of the last three partitions **MUST BE** the order of system/cache/userdata.
- The default partition name of scatter file and format **CANNOT** be altered.
- The custom partition **ONLY** can be added before system partition and after expdb partition.
- **ONLY** three partitions (system/cache/userdata) can have different size between the two versions.
- **ONLY** full update OTA package can do changed partition size upgrade. Differential OTA package **CANNOT** do changed partition size upgrade.
- If process changed partition size upgrade:
 - SD card must not write protected.
 - SD card must have enough free space to store the user data.
 - The size of new userdata partition must be larger than the user data used.
- Backup of user data image is encrypted. The encrypted data can only be restored to the original mobile phone.



3 How to build SD upgrade image

3.1 Configure for recovery mode

alps/mediatek/config/<project>/recovery.fstab

| NAND | | | | |
|------|-------------|--------|----------------------|--------------------|
| # | mount point | fstype | device | [device2] |
| | /boot | mtt | boot | |
| | /cache | yaffs2 | cache | |
| | /data | yaffs2 | userdata | |
| | /misc | mtt | misc | |
| | /recovery | mtt | recovery | |
| | /sdcard | vfat | /dev/block/mmcblk0p1 | /dev/block/mmcblk0 |
| | /system | yaffs2 | system | |

| eMMC | | | | |
|------|-------------|--------|----------------------|-----------|
| # | mount point | fstype | device | [device2] |
| | /boot | emmc | boot | |
| | /cache | ext4 | /dev/block/mmcblk0p2 | |
| | /data | ext4 | /dev/block/mmcblk0p3 | |
| | /misc | emmc | misc | |
| | /recovery | emmc | recovery | |
| | /sdcard | vfat | /dev/block/mmcblk0p4 | |
| | /system | ext4 | /dev/block/mmcblk0p6 | |

If use external SD, /sdcard must be set as /dev/block/mmcblk1p1 /dev/block/mmcblk1

3.2 Build full OTA update image step by step

1. Change directory to the root of code base
2. `./makeMtk <project> bm_new`
3. `./makeMtk <project> otapackage`
4. Copy out/target/product/<project>/<project>-ota-<mode>.<user_id>.zip to root directory of SD card and rename it to **update.zip**

3.3 Build differential OTA update image step by step

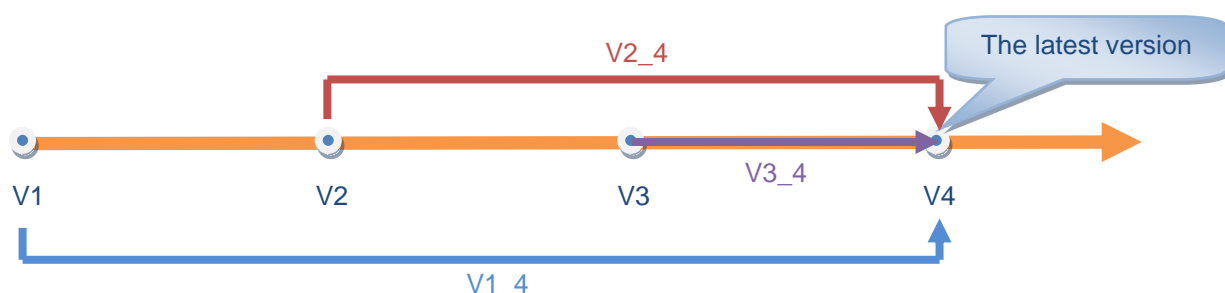
3.3.1 Differential OTA package size

The smaller OTA package size is the only advantage of differential OTA package.

3.3.2 Software maintenance

The software maintenance effort of differential OTA package is huge for customers. They must keep the entire source and OTA package when they release firmware. They also need to build the differential OTA package for every version they had released.

E.g. If the customer's latest version is V4. They want their end user to upgrade to latest version. Totally, they would build 3 versions of differential OTA packages. Which are V1_4, V2_4 and V3_4. The differential OTA package is only suggested for MOTA/FOTA.



3.3.3 Pro and con

| | Full OTA package | Differential OTA package |
|------------------------|------------------|--------------------------|
| Package size | X | O |
| Maintain effort | O | X |
| Changed partition size | O | X |

3.3.4 Step by step to build differential OTA package

Need two codebase to build V2_4 differential OTA package for example:

1. Change directory to root of V2 codebase
2. `./makeMtk <project> bm_new`
3. `./makeMtk <project> otapackage`
4. copy out/target/product/<project>/obj/PACKAGING/target_files_intermediates/<project>-target_files-<user>.zip to the root directory of V4 as **V2_org.zip**
5. Change directory to root of V4 codebase
6. `./makeMtk <project> bm_new`
7. `./makeMtk <project> otapackage`
8. copy out/target/product/<project>/obj/PACKAGING/target_files_intermediates/<project>-target_files-<user>.zip to the root directory of V4 as **V4_new.zip**
9. Find out key path, usually in build/target/product/security/<project>/releasekey
`./build/tools/releasetools/ota_from_target_files -k <key_path> -i V2_org.zip V4_new.zip V2_4.zip`

3.4 Build update image with DSP_BL step by step

Update DSP_BL is not necessary when every time update firmware except DSP_BL needs to be patched.

3.4.1 How to prepare DSP_BL image

Use the new version DSP_BL image, which is located in out/target/product/<project>/DSP_BL

3.4.2 How to prepare update image with DSP_BL

1. Change directory to the root of code base
2. `./makeMtk <project> bm_new`
3. `./makeMtk <project> otapackage`
4. `./build/tools/releasetools/ota_from_target_files -d <dsp_bl.img>
out/target/product/<project>/obj/PACKAGING/target_files_intermediates/<project>-target_files-<mode>.<user_id>.zip update.zip`

3.5 Note and checking

/system/build.prop

ro.product.device

ro.build.product

ro.build.fingerprint (differential OTA package)

Signature key

Differential OTA package

Make sure phone's load is the same as the old version of differential OTA package

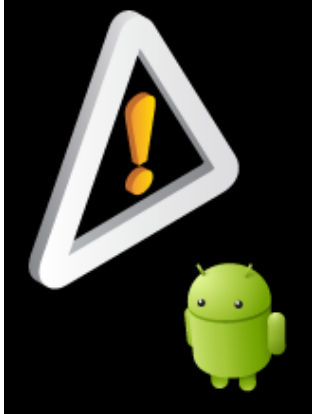
Do not use adb push any files into /system, it will cause upgrade fail

File name (apk, ringtone etc.) CANNOT have a space character or Chinese when build OTA package

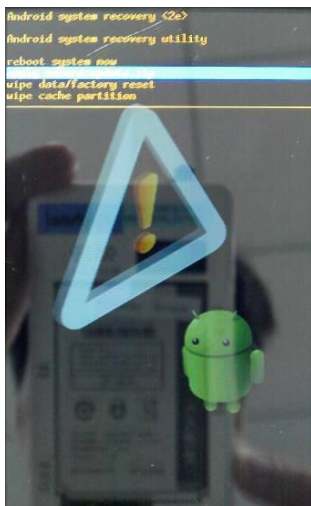
4 How to upgrade firmware by SD card

4.1 Upgrade procedure

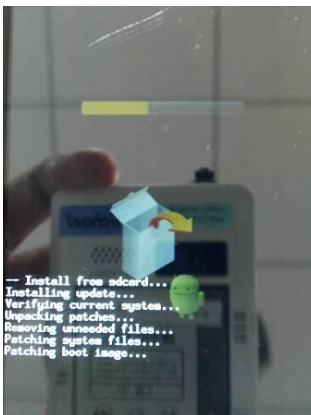
1. Copy **update.zip** to the root directory of SD card and then plug SD card into you handset.
2. Press “Volume up” and “Power key” to enter recovery mode.



3. Press “Home key” to enter main menu



4. Use “Volume down key” to select “**apply update from sdcard**”.
5. Press “Menu key” to enter file browser.
6. Use “Volume up key” or “Volume down key” to select your **update.zip** file.
7. Press “Menu key” to execute upgrade procedure.



8. After upgrade procedure is done, then reboot handset.

NOTE: The key setting is customizable and depends on the phone setting.

4.2 Error handling

If the upgrade process is fail due to power failure. After power on, the system will back step 2 above.

Please repeat through step 3 to step 6 to finish upgrade process.

4.3 Recovery mode debug

- ui_printf
 - Show message in LCD
- printf
 - In recovery mode and eng build
 - ◆ `adb pull /tmp/recovery.log` – submit eService please attach this file
 - Reboot to normal mode
 - ◆ `/cache/recovery/last_log` – submit eService please attach this file
- update.zip
 - `META-INF\com\google\android\updater-script` – submit eService please attach this file

5 Other operation

5.1 Wipe data/factory reset

When execute this item, phone's data and cache partitions will be erased. And phone will be in factory reset state.

5.2 Wipe cache partition

When execute this item, it will erase phone's cache partition.

5.3 Backup user data

When execute this item, SD card must be plugged into phone. It will backup phone's data partition to SD card.

5.4 Restore user data

When execute this item, SD card must be plugged into phone. It can restore the previous backup data from SD card to phone.

6 MTK special factory reset

6.1 Purpose and How-to

In order to preserve pre-installed apks after executing factory reset. Customers need to turn on this special option. Here were also introducing two config files to interact with factory reset process:

- /data/app/.keep_list Use to preserve apks which were not be removed after factory reset
- /data/app/.restore_list Use to restore apks from /system/somewhere to /data/app/

These two files using absolute path to indicate apks and each apk represents in one line.

Customers can set this option as MTK_SPECIAL_FACTORY_RESET=yes in

alps/mediatek/config/[project]/ProjectConfig.mk

6.2 Combinations and Examples

| | | | | |
|--------------------------------|---------------------------------|---|---|---|
| /data/app/.keep_list exists | × | ○ | × | ○ |
| /data/app/.restore_list exists | × | × | ○ | ○ |
| Behavior | Preserve all apks in /data/app/ | ONLY preserve apks which are in keep list | Preserve all apks in /data/app/ then restore apks from /system/ to /data/app/ which are in restore list | ONLY preserve apks which are in keep list then restore apks from /system/ to /data/app/ which are in restore list |

NOTE:

The contains in /data/app/.keep_list must begin with /data/app/ in each line

The contains in /data/app/.restore_list must begin with /system/ in each line

If /data/app/ now contains 10 apks, naming from d1.apk ~ d10.apk; and /system/appbackup/ now contains 5 apks, naming from s1.apk ~ s5.apk.

Example 1:

If you want to /data/app/ to have 8 apks after factory reset, which are d1.apk ~ d3.apk and s1.apk ~ s5.apk.

You can put these two config files when pre-installing the apks.

/data/app/.keep_list:

/data/app/d1.apk

/data/app/d2.apk

/data/app/d3.apk

/data/app/.restore_list:

/system/appbackup/s1.apk

/system/appbackup/s2.apk

/system/appbackup/s3.apk

/system/appbackup/s4.apk

/system/appbackup/s5.apk

Example 2:

If you just want to /data/app/to have 5 apks after factory reset, which are s1.apk ~ s5.apk.

You can put these two config files when pre-installing the apks.

/data/app/.keep_list:

/data/app/.restore_list:

/system/appbackup/s1.apk

/system/appbackup/s2.apk

/system/appbackup/s3.apk

/system/appbackup/s4.apk

/system/appbackup/s5.apk