Updating an Android Device

Contents

Chapter 1

Creating update.zip

Tip
The following instructions are extrapolated by reading the source code in `/bootable/recovery/` folder of the source tree, especially `recovery.c` and `roots.c`.

1.1 Create directories

First, we want to create a placeholder and structure for our update file.

```
mkdir -p myupdate/META-INF/com/google/android
cd myupdate
```

1.2 Create update-script

Next, create a file with name `update-script` in `META-INF/com/google/android` folder.

update-script

```
show_progress 0.1 0

copy_dir PACKAGE:system SYSTEM:

show_progress 1.0 10
```

The following is the list of commands that is recognized in

1.2.1 copy_dir

Syntax
`copy_dir <src-dir> <dst-dir> [<timestamp>]`

Description
Copy the contents of <src-dir> to <dst-dir>. The original contents of <dst-dir> are preserved unless something in <src-dir> overwrote them.

Example
`copy_dir PACKAGE:system SYSTEM:`

1.2.2 format

Syntax
`format <root>`

Description
Format a partition Example: `format SYSTEM:`, will format entire /system . Note: formatting erases data irreversibly.

1.2.3 delete

Syntax
`delete <file1> [... <fileN>]`

Description
Delete file.

Example
`delete SYSTEM:app/Calculator.apk`, will delete Calculator.apk from system/app directory.

1.2.4 delete_recursive

Syntax
`delete_recursive <file-or-dir1> [... <file-or-dirN>]`

Description
Delete a file or directory with all of it's contents recursively

Example
`delete_recursive DATA:dalvik-cache`, will delete /data/dalvik-cache directory with all of it's contents

1.2.5 run_program

Syntax
`run_program <program-file> [<args> ...]`

Description
Run an external program included in the update package.

Example
`run_program PACKAGE:install_busybox.sh`, will run install_busybox.sh script (shell command) included in the update package.

1.2.6 set_perm

Syntax
`set_perm <uid> <gid> <mode> <path> [... <pathN>]`

Description
Set ownership and permission of single file or entire directory trees, like `chmod', `chown', and `chgrp' all in one

Example
`set_perm 0 2000 0550 SYSTEM:etc/init.goldfish.sh`

1.2.7 set_perm_recursive

Syntax

```
set_perm_recursive <uid> <gid> <dir-mode> <file-moe> <path> [...  <pathN>]
```

Description

Set ownership and permission of a directory with all of it's contents recursively

Example

```
set_perm_recursive 0 0 0755 0644 SYSTEM:app
```

1.2.8 show_progress

Syntax

```
show_progress <fraction> <duration>
```

Description

Use of the on-screen progress meter for the next operation, automatically advancing the meter over <duration> seconds (or more rapidly if the actual rate of progress can be determined).

Example

```
show_progress 0.1 0
```

1.2.9 symlink

Syntax

```
ymlink <link-target> <link-path>
```

Description

Create a symlink (like `ln-s'). The <link-path> is in root:path format, but <link-target> is for the target filesystem (and may be relative)

1.2.10 Partitions

Table 1.1: Disk Partitions

Partition	Linux block device	/mountpoint/	fs	size	Description.
BOOT	(/dev/mtdblock[?])	/	(RAM)	Raw	Kernel, ramdisk and boot config.
DATA	(/dev/mtdblock5)	/data/	yaffs2	91904kb	User, system config, app config, and apps (without a2sd)
CACHE	(/dev/mtdblock4)	/cache/	yaffs2	30720kb	OTA cache, Recovery/update config and temp
MISC	(/dev/mtdblock[?])	N/A	Raw	N/A	TODO Not sure what it does.
PACKAGE	(Relative to package file)	N/A	N/A	N/A	Pseudo-filesystem for update package.

Table 1.1: (continued)

Partition	Linux block device	/mountpoint/	fs	size	Description.
RECOVERY	(/dev/mtdblock[?])	/	Raw	[?]kb	The recovery and update environment's kernel and ramdisk. Similar to BOOT.
SDCARD	(/dev/mmcblk0(p1))	/sdcard/	fat32	32MB-32GB	The microSD card. Update zip is usually here.
SYSTEM	(/dev/mtdblock3)	/system/	yaffs2	92160kb	The OS partition, static and read-only.

1.3 Create signed ZIP file

Generate a new key:

```
openssl genrsa -out key.pem 1024
openssl req -new -key key.pem -out request.pem
openssl x509 -req -days 9999 -in request.pem -signkey key.pem -out certificate.pem
openssl pkcs8 -topk8 -outform DER -in key.pem -inform PEM -out key.pk8 -nocrypt
```

Sign your update.zip

```
zip myupdate.zip myupdate/
java -jar SignApk/signapk.jar certificate.pem key.pk8 myupdate.zip update.zip
```

Note
We assume you have `signapk.jar` downloaded. Instructions on where to get it are here: Generating Keys

Chapter 2

Over The Air updates (OTA)

The recovery tool communicates with the main system through `/cache` files:

- `/cache/recovery/command` - INPUT - command line for tool, one arg per line
- `/cache/recovery/log` - OUTPUT - combined log file from recovery run(s)
- `/cache/recovery/intent` - OUTPUT - intent that was passed in

The arguments which may be supplied in the `recovery.command` file:

- `--send_intent=anystring` - write the text out to recovery.intent
- `--update_package=path` - verify install an OTA package file
- `--wipe_data` - erase user data (and cache), then reboot
- `--wipe_cache` - wipe cache (but not user data), then reboot
- `--set_encrypted_filesystem=on|off` - enables / diasables encrypted fs

After completing, we remove /cache/recovery/command and reboot. Arguments may also be supplied in the bootloader control block (BCB). These important scenarios must be safely restartable at any point:

2.1 Factory Reset

1. user selects `factory reset`
2. main system writes `--wipe_data` to `/cache/recovery/command`
3. main system reboots into recovery
4. `get_args()` writes BCB with `boot-recovery` and `--wipe_data` — after this, rebooting will restart the erase --
5. `erase_volume()` reformats `/data`
6. `erase_volume()` reformats `/cache`
7. `finish_recovery()` erases BCB — after this, rebooting will restart the main system --
8. `main()` calls `reboot()` to boot main system

2.2 OTA Install

1. main system downloads OTA package to `/cache/some-filename.zip`
2. main system writes `--update_package=/cache/some-filename.zip`
3. main system reboots into recovery
4. `get_args()` writes BCB with `boot-recovery` and `--update_package=...` — after this, rebooting will attempt to reinstall the update --
5. `install_package()` attempts to install the update NOTE: the package install must itself be restartable from any point
6. `finish_recovery()` erases BCB — after this, rebooting will (try to) restart the main system --
7. **if install failed**
   a. `prompt_and_wait()` shows an error icon and waits for the user
   b. the user reboots (pulling the battery, etc) into the main system
8. main() calls maybe_install_firmware_update()
9. **if the update contained radio/hboot firmware**
   a. `m_i_f_u()` writes BCB with `boot-recovery` and `--wipe_cache` — after this, rebooting will reformat cache & restart main system --
   b. `m_i_f_u()` writes firmware image into raw cache partition
   c. `m_i_f_u()` writes BCB with `update-radio/hboot` and `--wipe_cache` — after this, rebooting will attempt to reinstall firmware --
   d. `bootloader` tries to flash firmware
   e. `bootloader` writes BCB with `boot-recovery` (keeping `--wipe_cache`) — after this, rebooting will reformat cache & restart main system --
   f. `erase_volume()` reformats `/cache`
   g. `finish_recovery()` erases BCB — after this, rebooting will (try to) restart the main system --
   h. `main()` calls `reboot()` to boot main system

2.3 Secure File System Enabled/Disabled

1. user selects `enable encrypted file systems`
2. main system writes `--set_encrypted_filesystems=on|off` to `/cache/recovery/command`
3. main system reboots into recovery
4. `get_args()` writes BCB with `boot-recovery` and `--set_encrypted_filesystems=on|off` — after this, rebooting will restart the transition --
5. `read_encrypted_fs_info()` retrieves encrypted file systems settings from `/data` Settings include: property to specify the Encrypted FS istatus and FS encryption key if enabled (not yet implemented)
6. `erase_volume()` reformats `/data`
7. `erase_volume()` reformats `/cache`
8. `restore_encrypted_fs_info()` writes required encrypted file systems settings to `/data` Settings include: property to specify the Encrypted FS status and FS encryption key if enabled (not yet implemented)
9. `finish_recovery()` erases BCB — after this, rebooting will restart the main system --
10. `main()` calls `reboot()` to boot main system