Runtime Boot Loader Update Process for Jetson X1

DA_08065-001 | March 3, 2017

Release 27.1

TABLE OF CONTENTS

DA_08065-001

Version	Date	Authors	Description of Change
1.0	03 Jun 2016	mzenius	Initial release.
1.0	3 Mar 2017	hlang	update release date/moniker for L4T 27.1

INTRODUCTION

This document provides guidelines for enabling runtime boot loader updates with data redundancy features to be used as part of a failure-tolerant system update procedure. The topics discussed are specific to the NVIDIA® Tegra® Linux Driver Package (L4T) R24.2 release and the U-Boot boot loader on the Jetson™ TX1 Developer Kit. Adaptation of these instructions may be required for use on other Tegra X1-based platforms. Implementation of a specific update mechanism or update procedure is outside the scope of this document.

PREREQUISITES

For a description of the Tegra boot flow, see the NVIDIA® Tegra® Public Application Notes. This document assumes an understanding of the Tegra boot process.

For information on the Linux Tegra Driver Package, refer to the Embedded Developer Zone website. This document assumes an understanding of the L4T software and the Jetson TX1 Developer Kit.

BCT, TEGRABOOT, AND BOOTFILESET REDUNDANCY

The Tegra BootROM validates the BCT through an integrated checksum. If the calculated checksum does not match the checksum value within the BCT, the BootROM searches for the next BCT and attempts to validate. Up to 64 copies of the BCT are searched, at mod-16KiB boundaries. The NVIDIA flashing utility, `tegraflash`, writes up to 64 copies of the BCT based on the space allocated for the BCT partition.

A BCT can contain up to four entries for indicating locations (offset and size) and checksums of the tegraboot and bootfileset (BFS). The BootROM computes and validates the checksum for each tegraboot entry. When the first valid checksum is located, the BootROM transfers control (jumps) to the specified tegraboot.

The tegraboot computes and validates the checksum for its companion BFS. When the checksum for the BFS is validated, the tegraboot loads boot files such as `tegraboot-cpu`, `bootloader-DTB`, `kernel-DTB`, `warmboot-vector`, and `TOS`, and takes proper actions on them. When all necessary boot files are loaded, the tegraboot loads a bootloader such as U-Boot and transfers control (jumps) to the bootloader. If the tegraboot fails to validate the bootfileset, then it overwrites itself and resets the board so that the BootROM can validate and load the next tegraboot and bootfileset combination.

OVERVIEW

The procedures for updating the runtime bootloader are as follows:

- Implement Bootloader Redundancy:
  - Modify the Linux kernel to expose the eMMC boot0 and boot1 partitions for runtime access.
  - Modify the partition configuration file to make the tegraboot and bootfileset partitions specified in the `boot0` and `boot1` partitions.
- Deploy devices with bootloader redundancy.
- Prepare new bootloader and BCT images for update:
  - Extract the BCT from the deployed system for offline modification.
  - Use offline, host-based tools to regenerate the BCT and BFS images. (The U-Boot bootloader is part of the new BFS image.)

- Download the new BCT and BFS images into the target device.
- Update the new BCT and BFS images:
    - Copy the BCT and BFS images to eMMC partitions.
    - Reboot to execute the updated bootloader.

FAILURE-TOLERANT BCT AND BOOTLOADER UPDATE

The standard L4T release does not enable bootloader redundancy features. As part of a failure-tolerant bootloader update, you must first implement and deploy bootloader redundancy and then follow the bootloader update procedure.

IMPLEMENTING BOOTLOADER REDUNDANCY

Enabling bootloader redundancy requires modification of the following components:

- Kernel eMMC driver
- Partition configuration file

After these modifications are made, flashed, and deployed, then initial bootloader redundancy is enabled.

Modifying the Linux Kernel

You must modify the Linux kernel to expose the eMMC `boot0` and `boot1` partitions for runtime access. By default, eMMC boot partitions are not exposed during runtime by the Linux kernel.

To expose the eMMC boot partitions

1. Navigate to the kernel driver:

```
<kernel>/drivers/mmc/host/sdhci-tegra.c
```

2. Comment out the following line:

```
host->mmc->caps2 |= MMC_CAP2_BOOTPART_NOACC;
```

When the kernel is booted, the write-protected boot partitions are visible at:

- `/dev/mmcblk0boot0`—BCT, BFS0, and BFS1
- `/dev/mmcblk0boot1`—BFS2 and BFS3

Moving the BFS to the Boot Partitions

You must modify the partition configuration file to move the BFS to the boot partitions.

To modify the partition table

1. Modify the BSP configuration file:

```
bootloader/t210ref/cfg/gnu_linux_tegraboot_emmc_full.xml
```

The BSP configuration file contains the partitioning information for both the Tegra partition table and the GPT.

- Partitions defined after GP1 are visible to Linux.
- Partition NVC contains the tegraboot.
- Partitions NVC-1, NVC-2, and NVC-3 store additional tegraboot instances for redundancy. These partitions must be defined after BCT and before GP1 to be located in boot partitions.
- Partitions PT, PT-1, PT-2, and PT-3 contain layout information for each BFS, and indicate the beginning of each BFS.
- Partitions TBC, TBC-1, TBC-2, and TBC-3 contain the `cpu-portion` of tegraboot.
- Partitions RP1, RP1-1, RP1-2, and RP1-3 contain tegraboot DTBs.
- Partitions DTB, DTB-1, DTB-2, and DTB-3 contain kernel DTBs.
- Partitions EBT, EBT-1, EBT-2, and EBT-3 contain secondary boot loaders such as U-Boot or cboot.
- Partitions WB0, WB0-1, WB0-2, and WB0-3 contain the warmboot vector.
- Partitions BPF, BPF-1, BPF-2, and BPF-3 contain BPMP microcode.
- Partitions TOS, TOS-1, TOS-2, and TOS-3 contain secure monitor code.

2. Verify that each bootfileset is defined immediately after each NVC partition, and that the PT partition is the first within each bootfileset. That is, NVC and bootfileset combinations should be defined in the order: NVC, then PT, then the rest.

3. Verify that sum of partitions sizes of one NVC and matching BFS partitions does not exceed (boot0_size – 1 MiB) / 2. (With standard Jetson TX1, it is 1.5 MiB)

4. Modify the `gnu_linux_tegraboot_emmc_full.xml` file to move the APP entry immediately after the GP1 entry, so that the RootFS is on /dev/mmcblk0p1.

5. Modify all partition entry ID=values to be contiguous.

   The overall layout for the partition configuration file is as follow:

```
…
BCT
NVC
BFS
NVC-1
BFS-1
NVC-2
BFS-2
NVC-3
BFS-3
GP1
APP
…
```

The following is an example of a complete partition configuration file:

```
<?xml version="1.0"?>

<!-- Nvidia Tegra Partition Layout Version 1.0.0 -->

<partition_layout version="01.00.0000">
    <device type="sdmmc" instance="3">
        <partition name="BCT" id="2" type="boot_config_table">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 1048576 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
        </partition>

        <partition name="NVC" id="3" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 188416 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> nvtboot.bin </filename>
        </partition>

        <partition name="PT" id="4" type="partition_table">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 8192 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> flash.bin </filename>
        </partition>

        <partition name="TBC" id="5" type="bootloader">
```

```
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 86016 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> nvtboot_cpu.bin </filename>
    </partition>

    <partition name="RP1" id="6" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 307200 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
    </partition>

    <partition name="DTB" id="7" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 307200 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
    </partition>

    <partition name="EBT" id="8" type="bootloader">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 516096 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> cboot.bin </filename>
```

```
        </partition>

        <partition name="WB0" id="9" type="WB0">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4096 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> warmboot.bin </filename>
        </partition>

        <partition name="BPF" id="10" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 69632 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> bpmp.bin </filename>
        </partition>

        <partition name="TOS" id="11" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 86016 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> tos.img </filename>
        </partition>

        <partition name="NVC-1" id="12" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
```

```
        <size> 188416 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> nvtboot.bin </filename>
    </partition>

    <partition name="PT-1" id="13" type="partition_table">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 8192 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> flash.bin </filename>
    </partition>

    <partition name="TBC-1" id="14" type="bootloader">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 86016 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> nvtboot_cpu.bin </filename>
    </partition>

    <partition name="RP1-1" id="15" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 307200 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
    </partition>
```

```
        <partition name="DTB-1" id="16" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 307200 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
        </partition>

        <partition name="EBT-1" id="17" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 516096 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> cboot.bin </filename>
        </partition>

        <partition name="WB0-1" id="18" type="WB0">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4096 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> warmboot.bin </filename>
        </partition>

        <partition name="BPF-1" id="19" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 69632 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
            <allocation_attribute> 8 </allocation_attribute>
```

```
        <percent_reserved> 0 </percent_reserved>
        <filename> bpmp.bin </filename>
    </partition>

    <partition name="TOS-1" id="20" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 86016 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <partition_attribute> 0 </partition_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> tos.img </filename>
    </partition>

    <partition name="NVC-2" id="21" type="bootloader">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 188416 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> nvtboot.bin </filename>
    </partition>

    <partition name="PT-2" id="22" type="partition_table">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 8192 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> flash.bin </filename>
    </partition>

    <partition name="TBC-2" id="23" type="bootloader">
        <allocation_policy> sequential </allocation_policy>
```

```
            <filesystem_type> basic </filesystem_type>
            <size> 86016 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> nvtboot_cpu.bin </filename>
        </partition>

        <partition name="RP1-2" id="24" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 307200 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
        </partition>

        <partition name="DTB-2" id="25" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 307200 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
        </partition>

        <partition name="EBT-2" id="26" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 516096 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> cboot.bin </filename>
        </partition>
```

```
<partition name="WB0-2" id="27" type="WB0">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 4096 </size>
    <file_system_attribute> 0 </file_system_attribute>
    <allocation_attribute> 8 </allocation_attribute>
    <percent_reserved> 0 </percent_reserved>
    <filename> warmboot.bin </filename>
</partition>

<partition name="BPF-2" id="28" type="data">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 69632 </size>
    <file_system_attribute> 0 </file_system_attribute>
    <partition_attribute> 0 </partition_attribute>
    <allocation_attribute> 8 </allocation_attribute>
    <percent_reserved> 0 </percent_reserved>
    <filename> bpmp.bin </filename>
</partition>

<partition name="TOS-2" id="29" type="data">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 86016 </size>
    <file_system_attribute> 0 </file_system_attribute>
    <partition_attribute> 0 </partition_attribute>
    <allocation_attribute> 8 </allocation_attribute>
    <percent_reserved> 0 </percent_reserved>
    <filename> tos.img </filename>
</partition>

<partition name="NVC-3" id="30" type="bootloader">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 188416 </size>
```

```
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> nvtboot.bin </filename>
        </partition>

        <partition name="PT-3" id="31" type="partition_table">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 8192 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> flash.bin </filename>
        </partition>

        <partition name="TBC-3" id="32" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 86016 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> nvtboot_cpu.bin </filename>
        </partition>

        <partition name="RP1-3" id="33" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 307200 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
        </partition>

        <partition name="DTB-3" id="34" type="data">
```

```
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 307200 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> tegra210-jetson-cv-base-p2597-2180-a00.dtb </filename>
        </partition>

        <partition name="EBT-3" id="35" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 516096 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> cboot.bin </filename>
        </partition>

        <partition name="WB0-3" id="36" type="WB0">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4096 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> warmboot.bin </filename>
        </partition>

        <partition name="BPF-3" id="37" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 69632 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
```

```
        <filename> bpmp.bin </filename>
    </partition>

    <partition name="TOS-3" id="38" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 86016 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <partition_attribute> 0 </partition_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> tos.img </filename>
    </partition>

    <partition name="GP1" id="39" type="GP1">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 2097152 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
    </partition>

    <partition name="APP" id="40" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 15032385536 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> system.img </filename>
    </partition>

    <partition name="LNX" id="41" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 67108864 </size>
```

```
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> boot.img </filename>
    </partition>

    <partition name="SOS" id="42" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 20971520 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
    </partition>

    <partition name="USP" id="43" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 2097152 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
    </partition>

    <partition name="UDA" id="44" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 2097152 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x808 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
    </partition>

    <partition name="GPT" id="45" type="GPT">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 0xFFFFFFFFFFFFFFFF </size>
        <file_system_attribute> 0 </file_system_attribute>
```

```
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
        </partition>
    </device>
</partition_layout>
```

This modification causes the tegraboot and bootfileset partitions to disappear from Linux as `/dev/mmcblk0px`, makes other partitions visible, and causes the root file system partition to stay as `/dev/mmcblk0p1`. No further modifications are required to inform U-Boot and the Linux kernel of the location of the new root file system.

ENABLING BOOTLOADER REDUNDANCY

Flash the Jetson TX1 platform with the above modifications to verify proper loading and functioning of U-Boot, the Linux kernel, and the proper location specified for the root file system. To ensure that all build components are functional, it is recommended that you verify the bootloader and kernel independently prior to enabling bootloader redundancy.

Because enabling redundancy makes eMMC boot partitions very crowded, you must tell `flash.sh` to skip the boot partition check by commenting out the `BOOTPARTSIZE=xxx` line in the `<device name>.conf` file before flashing.

For flashing instructions, see the *Developer Guide* for your device.

PREPARING BOOTLOADER AND BCT IMAGES FOR UPDATE

For convenience, this document refers to BCT image names with four appended numeric values, which denote the bootloader version described by the BCT image. The numeric versions are arbitrary but indicate the bootloader update process.

For example, after flashing, with initial bootloader redundancy enabled, all bootloader versions are identical and referred to as `bct_1111`. When U-Boot is updated and the BCT is modified to contain two new images in slots 0 and 1 (positions 1 and 2), the boot version is referred to as `bct_2211`.

Likewise, for descriptive purposes, a similar numeric value is appended to the name of the BCT binary.

Extracting the BCT from the System

Before updating the bootloader, **read and maintain a copy of the BCT (`bct_1111`) flashed on the production device**. This BCT is used for later modifications to update the bootloader entries (location, size, and hash) within the BCT.

To retrieve the binary BCT

1. Place a production device into forced-recovery mode.

   For instructions on how to place a device in recovery mode, see the *Setting Up Your Platform* topic in the *Developer Guide* for your device.

2. Use the `tegraflash` utility to read `bct_1111` from the device.

   ```
   cd <BSP>/Linux_for_Tegra/bootloader
   ```

```
./tegraflash.py --bl cboot.bin --chip 0x21 --applet nvtboot_recovery.bin --cmd "read
BCT bct_1111;"
```

Note:	The `tegraflash` program reads in only one instance of BCT.

Regenerating the BCT and U-Boot Images

NVIDIA provides a host-based utility, `mkbctpart`, for offline modifications and updates of the BCT. Use `mkbctpart` to specify a new location, size, and hash value for an updated bootloader binary.

The `mkbctpart` syntax is as follows:

```
Usage:  mkbctpart [options] [new BCT file]
where
  <options> are:
    -b|--bctpartition <input BCT file> ------------ default=bct.dump
    -i|--instances <BL update entry CSV> ---------- default=0,1
    -k|--keyfile <RSA Private key file name> ------ default=None
    -l|--listbcts --------------------------------- default=N/A
    -p|--paddedfile <flashable padded BL file> ---- default=<BL>.padded
    -t|--tegratype <Tegra type> ------------------- default=T210
    -B|--Bootloader <new BL file name> ------------ default=u-boot.bin
    -V|--Verbose ---------------------------------- default=0
  <new BCT file> is:
    Output file name for updated BCT partition.

  <BFSBLOB> is:
    The BFSBLONB which includes tegraboot, new BFS image, and U-Boot file is to be downloaded an
```

For example:

```
./mkbctpart -b bct_1111 -i 0,1 -B <…/PATH/>u-boot-dtb.bin.2 -V bct_2211
```

In the above example, `mkbctpart`:

- Takes `bct_1111` as input, describing four version- 1 U-Boot binaries originally flashed in the production device and new U-Boot version-2 (`u-boot.bin.2`) as its input.

- Generates the new BCT file `bct_2211`, which describes U-Boot version-2 in slots 0 and 1, and U-Boot version-1 in slots 2 and 3.

- Generates a new BFSBLOB named as u-boot-dtb.bin.2.padded. Both `bct_2211` and `u-boot-dtb.bin.2.padded` are downloaded to the device prior to the update process.

Note:	**The `mkbctpart` utility expects the ELF image of `u-boot-dtb.bin.2` to exist in the same directory. The name of the ELF image is `u-boot`.**

DOWNLOADING THE UPDATED BCT AND U-BOOT IMAGES

You must copy the updated `bct_2211` and new boot file image `u-boot-dtb.bin.2.padded` to the target device (e.g., with the `scp` command, on a removable storage device, etc.). The mechanism for performing this task is beyond the scope of this document.

COPYING THE BCT AND U-BOOT IMAGES INTO EMMC

The following provides example target commands to overwrite the BCT and `bfsblob.bin` with the updated binaries into BFS instance 0 and 1. (This example is for the standard Jetson TX1 eMMC device with 4 MiB boot0 and 4 MiB boot1)

```
# sudo echo 0 > /sys/block/mmcblk0boot0/force_ro
# sudo dd if=bct_2211 of=/dev/mmcblk0boot0
# sudo dd if=u-boot-dtb.bin.2.padded of=/dev/mmcblk0boot0 seek=2048
# sudo dd if=u-boot-dtb.bin.2.padded of=/dev/mmcblk0boot0 seek=5120
# sudo sync
# sudo sync
# sudo echo 1 >/sys/block/mmcblk0boot0/force_ro
```

Note:	**The BFSBLOB locations for standard Jetson TX1 eMMC (4 MB boot0 and boot1) are:** # **Bfsblob instance 0: boot0, offset=0x100000, maxsize=1572864** # **Bfsblob instance 1: boot0, offset=0x280000, maxsize=1572864** # **Bfsblob instance 2: boot1, offset=0, maxsize=1572864** # **Bfsblob instance 3: boot1. offset=0x180000, maxsize=1572864** **The maxsize of BFSBLOB can be calculated from the formula:** **Maxbfssize = (Boot0_size - 1 MB) / 2**

If and only if you generated `bct_2233`, you can overwrite the BCT and `u-boot-dtb.bin.3.padded` with the updated binaries into BFS instance 2 and 3 as shown in the example below. (This example is for the standard Jetson TX1 eMMC device with 4 MiB boot0 and 4 MiB boot1)

```
# sudo echo 0 > /sys/block/mmcblk0boot0/force_ro
# sudo dd if=bct_2233 of=/dev/mmcblk0boot0
# sudo sync
# sudo sync
# sudo echo 1 > /sys/block/mmcblk0boot0/force_ro
#
# sudo echo 0 > /sys/block/mmcblk0boot1/force_ro
# sudo dd if=u-boot-dtb.bin.3.padded of=/dev/mmcblk0boot1 seek=0
# sudo dd if=u-boot-dtb.bin.3.padded of=/dev/mmcblk0boot1 seek=3072
# sudo sync
# sudo sync
# sudo echo 1 >/sys/block/mmcblk0boot1/force_ro
```

EXECUTING THE UPDATED BOOTLOADER

To execute the updated BCT and bootloader, you must reboot the system.

MKBCTPART UTILITY

The `mkbctpart` utility generates the updated BCT partition file and padded bootloader file from the provided BCT partition file and the new bootloader file, as follows:

1. Reads in the BCT partition file and validates it.
2. Reads in the new bootloader, builds `bfsblob.bin`, and calculates the hash.
3. Updates the bootloader entries in the BCT as specified by the `-i` option.

Note:	**Entries (also known as slots) not specified for modification remain the same, assuming the location, size, and hash of the bootloader being replaced.**

4. Writes out the new updated BCT partition file and `bfsblob.bin` file.

The following shows usage information for `mkbctpart`.

```
Usage:  mkbctpart [options] [new BCT file]
where
  <options> are:
    -b|--bctpartition <input BCT file> ------------ default=bct.dump
    -i|--instances <BL update entry CSV> ---------- default=0,1
    -k|--keyfile <RSA Private key file name> ------ default=None
    -l|--listbcts --------------------------------- default=N/A
    -p|--paddedfile <flashable padded BL file> ---- default=<BL>.padded
    -t|--tegratype <Tegra type> ------------------- default=T210
    -B|--Bootloader <new BL file name> ------------ default=u-boot.bin
    -V|--Verbose ---------------------------------- default=0
  <new BCT file> is:
    Output file name for updated BCT partition.

  <BFSBLOB> is:
    The BFSBLONB which includes tegraboot, new BFS image, and U-Boot file is to be downloaded an
```

Examples

The following are examples of `mkbctpart` usage.

```
./mkbctpart -l -b bct_1111
```

```
./mkbctpart -b bct_1111 -i 0,1 -B <…/PATH/>u-boot-dtb.bin.2 -V bct_2211
```

Notice

Trademarks

Copyright

Legal Information

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OR CONDITION OF TITLE, MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE AND ON-INFRINGEMENT, ARE HEREBY EXCLUDED TO THE MAXIMUM EXTENT PERMITTED BY LAW.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, CUDA, DriveWorks, Jetson, Tegra, TensorRT, Vibrante, and VisionWorks are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

ARM, AMBA, and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc; its operating company ARM Limited; and the regional subsidiaries ARM Inc.; ARM KK; ARM Korea Limited.; ARM Taiwan Limited; ARM France SAS; ARM Consulting (Shanghai) Co. Ltd.; ARM Germany GmbH; ARM Embedded Technologies Pvt. Ltd.; ARM Norway, AS and ARM Sweden AB.

Copyright