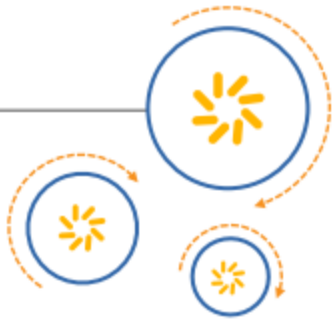
Qualcomm Technologies, Inc.



QUALCOMM®
2018-01-08 17:47:19 PST
kh.lee@lge.com

00005.1

Release Notes

MSM8996AU.LA.1.2

RNO-171218042050-2883 Rev. 1 (LG Electronics a.k.a. LGE)

December 20, 2017

Confidential and Proprietary - Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

© Qualcomm Technologies, Inc. All rights reserved.

1. Download

A. ChipCode Location:

https://chipcode.qti.qualcomm.com/software_locator?ref=r00005.1&project=MSM8996AU-LA-1-2_HLOS_DEV_BOOT
https://chipcode.qti.qualcomm.com/software_locator?ref=r00005.1&project=MSM8996AU-LA-1-2_TEST_DEVICE
https://chipcode.qti.qualcomm.com/software_locator?ref=r00005.1&project=MSM8996AU-LA-1-2_AMSS_Standard_OEM
https://chipcode.qti.qualcomm.com/software_locator?ref=r00005.1&project=MSM8996AU-LA-1-2_HLOS_DEV

B. ChipCode Commands:

git clone -depth 1 <repo location>

QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com

2. Release History

ChipCode Release History:
https://createpoint.qti.qualcomm.com/planner/link/search/releaseHistory/preset/MSM8996AU-LA-1-2_HLOS_DEV_BOOT/master/r00005.1
https://createpoint.qti.qualcomm.com/planner/link/search/releaseHistory/preset/MSM8996AU-LA-1-2_TEST_DEVICE/master/r00005.1
https://createpoint.qti.qualcomm.com/planner/link/search/releaseHistory/preset/MSM8996AU-LA-1-2_AMSS_Standard_OEM/master/r00005.1
https://createpoint.qti.qualcomm.com/planner/link/search/releaseHistory/preset/MSM8996AU-LA-1-2_HLOS_DEV/master/r00005.1

QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com

3. Getting Started

This document provides information for Release Notes: MSM8996AU.LA.1.2 Linux Android Release 00005 for MSM8996AU Devices (MSM8996AU.LA.1.2-00005-STD.PROD), a source code release of software.

This release is an Engineering Sample (ES5) release.

3.1 Download QTI proprietary software from Qualcomm ChipCode™ portal

Refer to MSM8996 LINUX ANDROID SOFTWARE USER MANUAL (SP80-NV396-4)

3.2 Download HLOS software

Refer to MSM8996 LINUX ANDROID SOFTWARE USER MANUAL (SP80-NV396-4)

Note: The CAF link and the manifest are as follows:

CAF manifest @ https://source.codeaurora.org/quic/la/platform/manifest/tag/?id=LA.HB.1.1.5-03710-8x96.0

To sync OSS from CAF, run the following command:

```
repo init -u git://codeaurora.org/platform/manifest.git -b release -m LA.HB
.1.1.5-03710-8x96.0.xml --repo-url=git://codeaurora.org/tools/repo.git --
repo-branch=caf-stable
```

QUALCOMM®
2018-11-08 17:17:19 PST
km.lee@lge.com

4. Build Instructions

4.1 Metabuild components

Name	Build ID
META build	MSM8996AU.LA.1.2-00005-STD.PROD
ADSP	ADSP.8996.2.7.2-00224-00440
BOOT	BOOT.XF.1.1-00145-M8996LZB
BTFM	BTFM.RM.2.4.1-00010-QCABTFMSWPZ
WLAN	WLAN.RM.4.5.1-00006-QCARMSWCZ-1
CPE	CPE.TSF.1.0-00035-W9335AAAAAAAZQ
Android	LA.HB.1.1.5-03710-8x96.0
MPSS	MPSS.TH.2.0.2.c4-00020-M8996FAAAANAZM
RPM	RPM.BF.1.6.c4-00013-M8996AAAAANAZR
TZ	TZ.BF.4.0.1-00345-M8996AAAAANAZT
Video	VIDEO.VE.4.4-00031-PROD
SLPI	SLPI.HB.1.0-00275-M8996AZL

For the build components and IDs, refer to the about.html file in the root of the repository on Qualcomm ChipCode™ portal.

4.2 Generate HLOS sparse images (Windows build machine)

1. Open the Windows command prompt and navigate to the following directory:

   ```
   cd \common\build
   ```

1. Run the following script:

   ```
   build.py
   ```

The command prompt window displays a large amount of output text. After approximately 5 minutes, a message similar to the following prints and control is handed back to the command prompt.

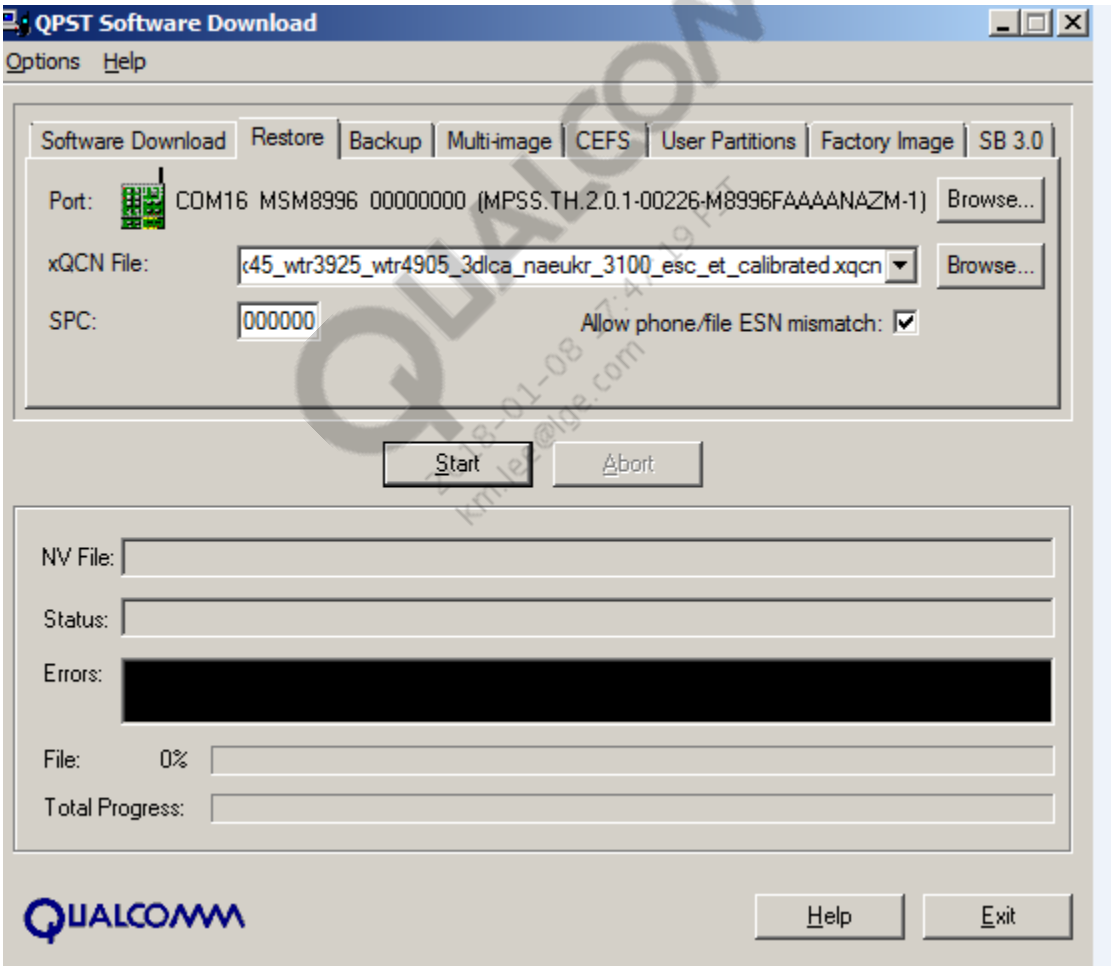
```
SUCCESS: Created "rawprogram_unsparse.xml" update_common_info.py:============ UPDATE
COMMON INFO COMPLETE========== ========== If the script fails and a build issue
exists, contact QTI support
```

4.3 Load QCN files for modem

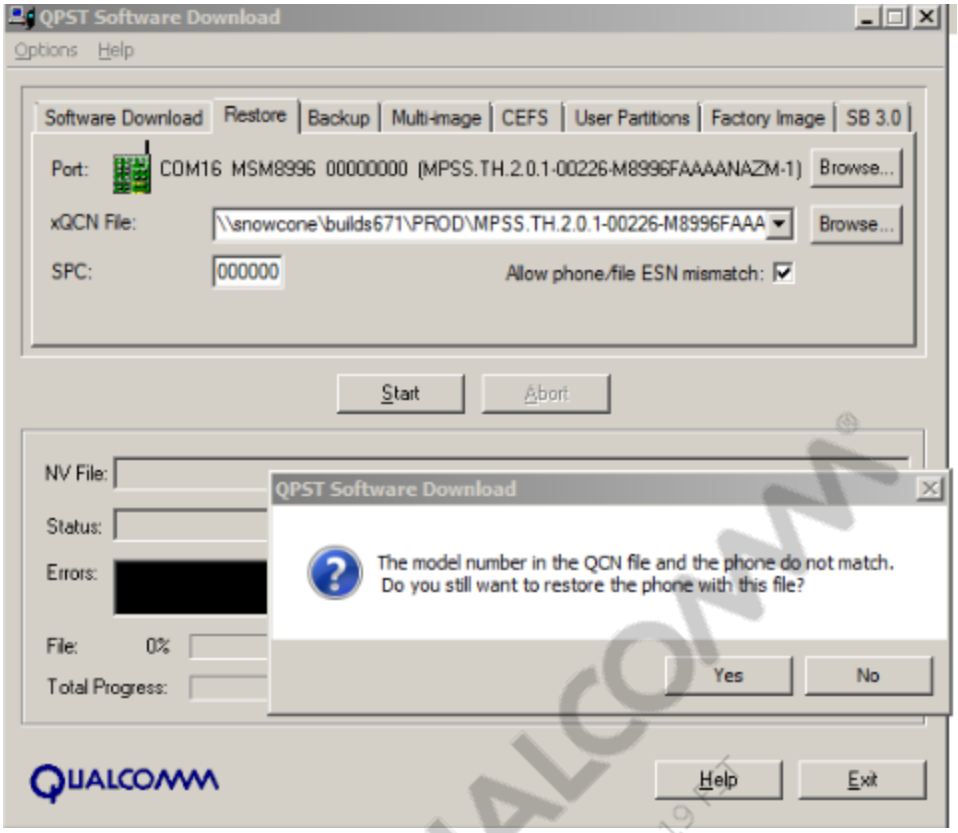
1. Download latest QPST and QXDM tools from CreatePoint.

RF Card MCN	QCN Files
MP25-NM506-4	\modem_proc\rftarget_thor\mdm9x45\qcn\wtr3925_3dlca_naeukr_3100v2\mdm9x45_wtr3925_wtr4905_3dlca_naeukr_3100_esc_et_calibrated.xqcn \modem_proc\rftarget_thor\mdm9x45\qcn\wtr3925_3dlca_naeukr_3100v2\NAEUKR_3100V2_Supplemental_CA_NVs.QCN
MP25-W4242-4	modem_proc\rftarget_thor\mdm9x45\qcn\wtr3925_3dlca_na_auto\mdm9x45_wtr3925_wtr4905_3dlca_na_auto_esc_apt_calibrated.xqcn
20-W4243-H1/ W4557-H1/ W4559 -H1	modem_proc\rftarget_thor\mdm9x45\qcn\wtr3925_3dlca_na_auto_v2\mdm9x45_wtr3925_wtr4905_3dlca_na_auto_v2_esc_apt_calibrated.xqcn

1. Open QPST Software Download.
2. Browse and select the COMPORT of the device.
3. Browse and select the relevant xQCN file.

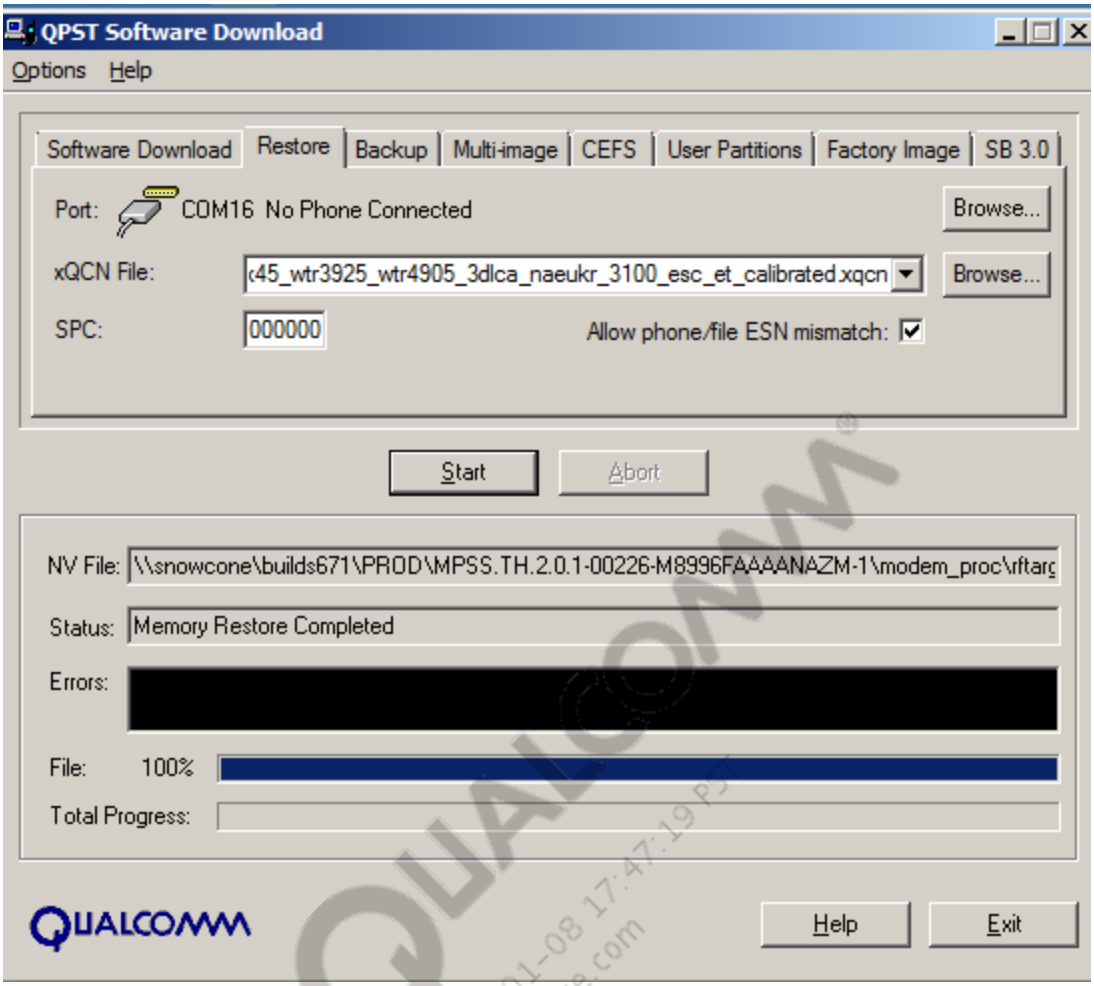


1. Press Start and select Yes, if you see the following message:



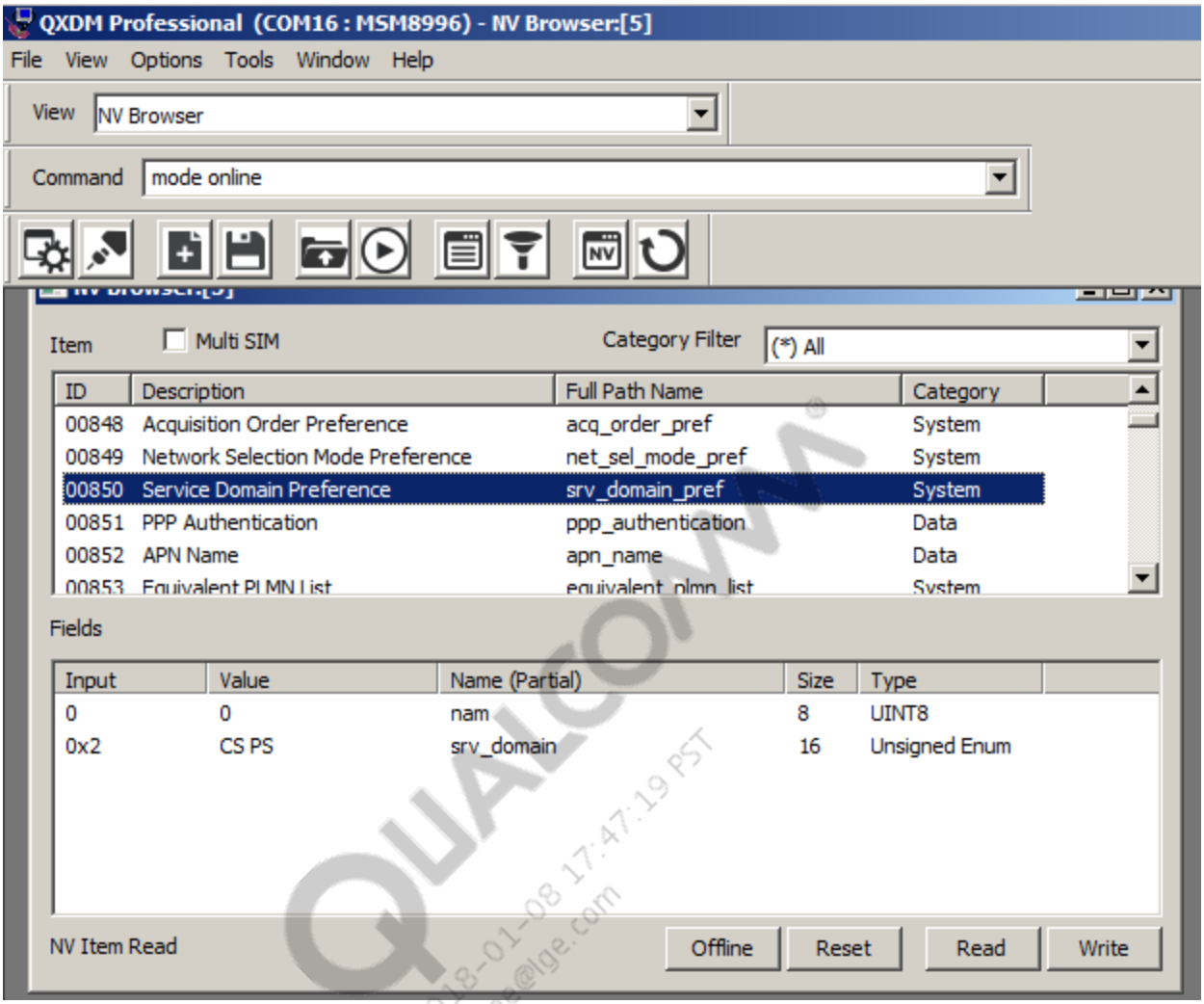
You can see the Memory Restore Completed message on the Status bar (as shown), which means the QCN file is loaded successfully.

QUALCOMM®
2018-01-08 17:47:19 KST
km.lee@lge.com

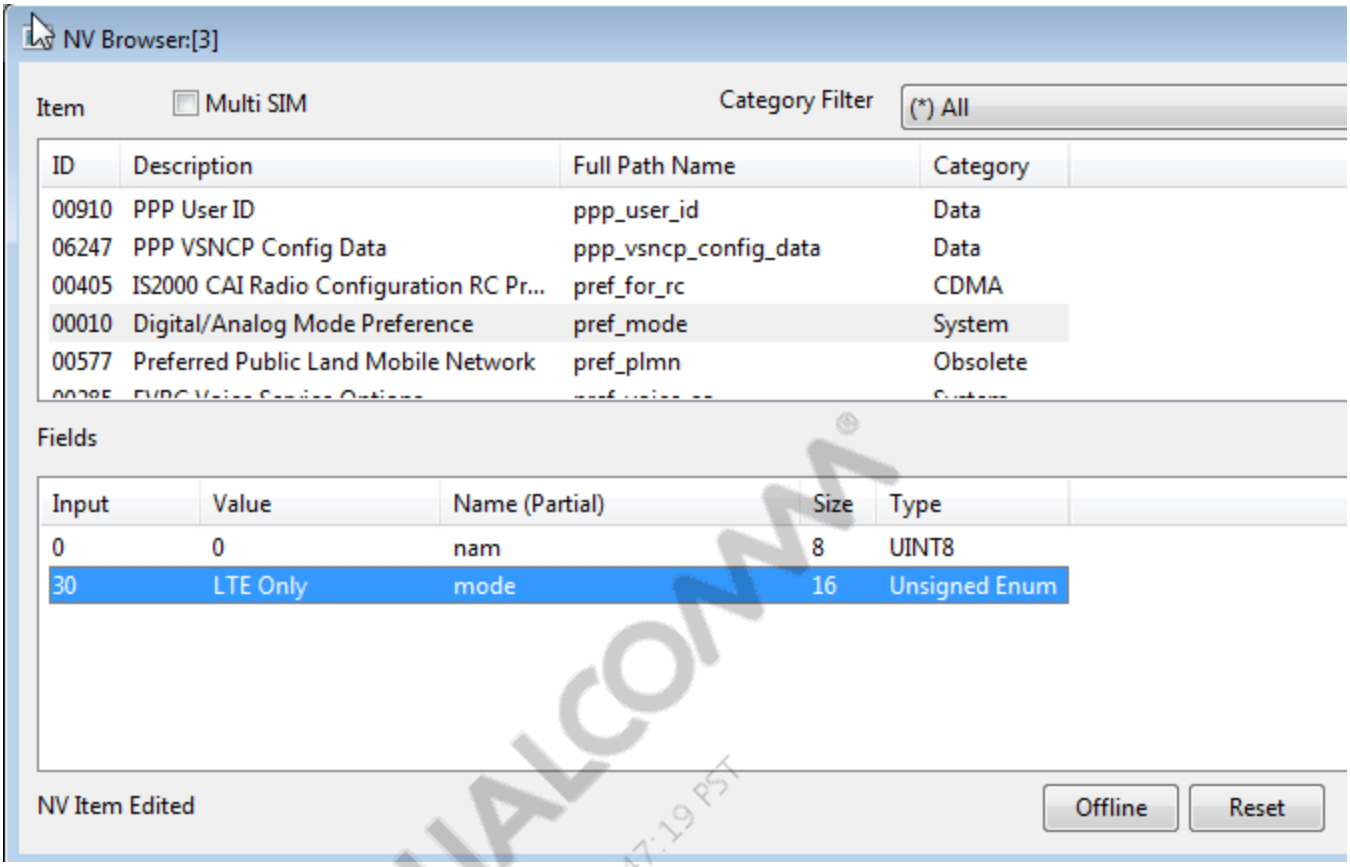


1. Repeat the steps 1, 2, 3, 4 and 5, if the RF card requires more than one QCN.
2. Reset the device.
3. Set the NVItem 00453 - QXDM > view > NV Browser > set 00453 NV item to 0.
4. For LTE, set the below NV items from QXDM:

- QXDM > view > NV Browser > set 00850 NV item value as follows:



1. Press Write after changing the value.
2. Set 00010 to LTE Only as follows:

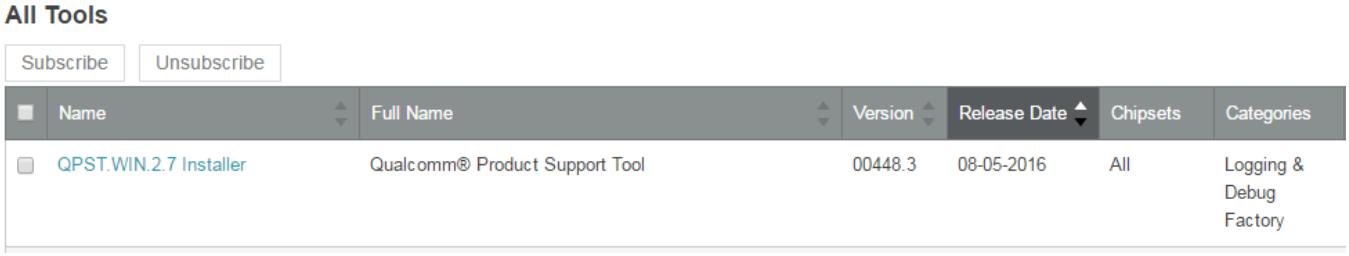


1. Press Write after changing the value.
2. Reset the device after all NV Item settings.

4.4 Flash images using QIPL loading on S820 ADP ver 3

4.4.1 QPST

QFIL is part of Qualcomm Product Support Tool (QPST). Download the QPST tool from https://createpoint.qti.qualcomm.com



Install a Qualcomm USB WWAN Driver. Download the driver from https://createpoint.qti.qualcomm.com

QUD.WIN.1.1 INSTALLER

System OS: **Windows** Last Updated: **11-20-2015** Version: **10039.2**

Chipset: **All**

Category: **Bring-up**

Reference Number: **QUD.WIN.1.1-10039-GENERAL-2**

Full Name: **Qualcomm® USB Driver**

Qualcomm USB WWAN Driver Installer for Microsoft Windows - External

4.4.2 S820 ADP Ver 3 board

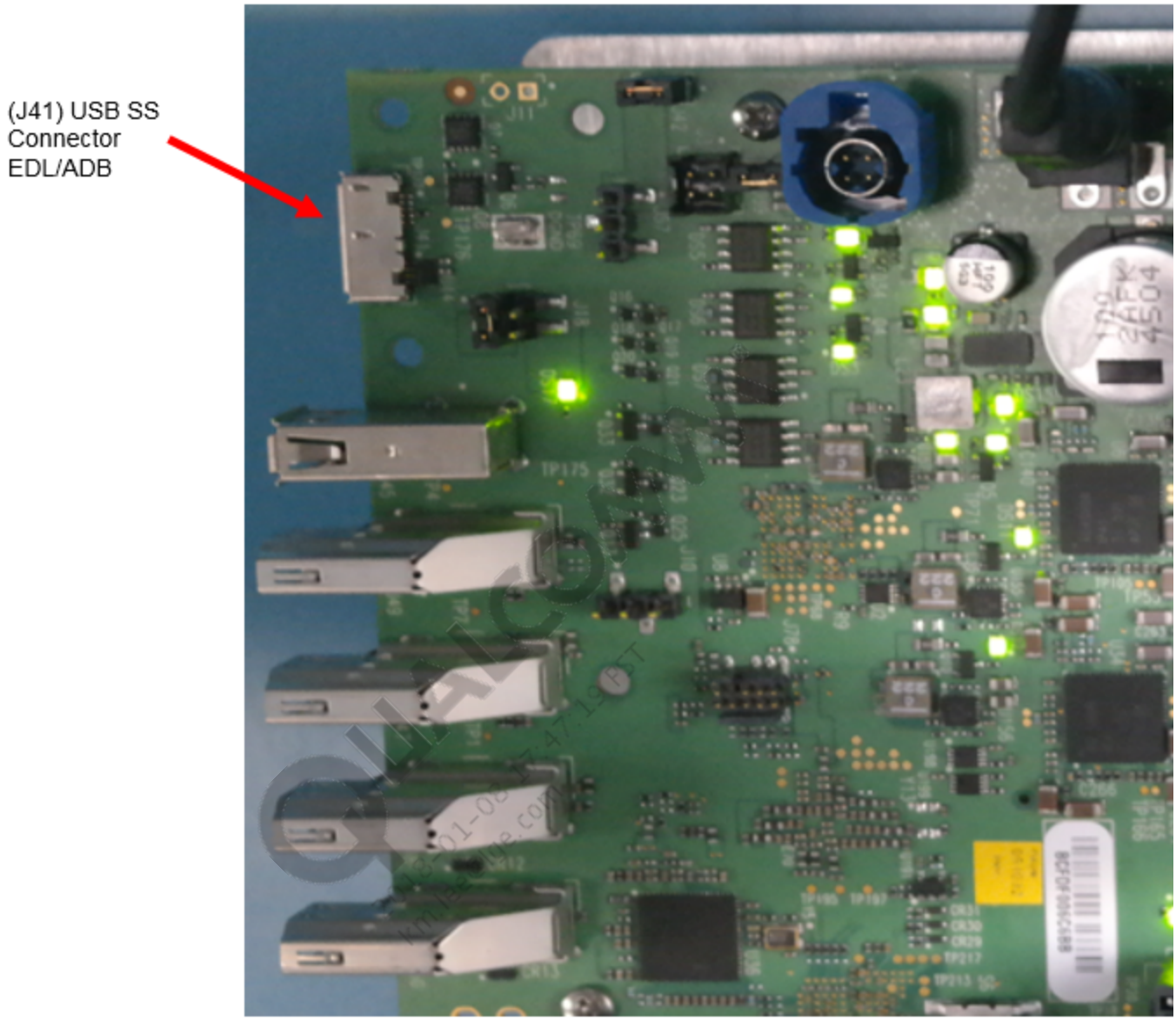
4.4.2.1 Switch for an EDL mode

The fourth switch of S1 is used to put a S820 ADP Ver 3 into EDL (emergency download) mode, as shown.

S1-4: forced USB boot

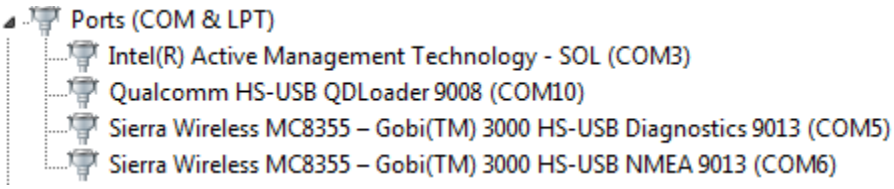
ON	ENABLED
OFF	DISABLED

4.4.2.2 USB connection

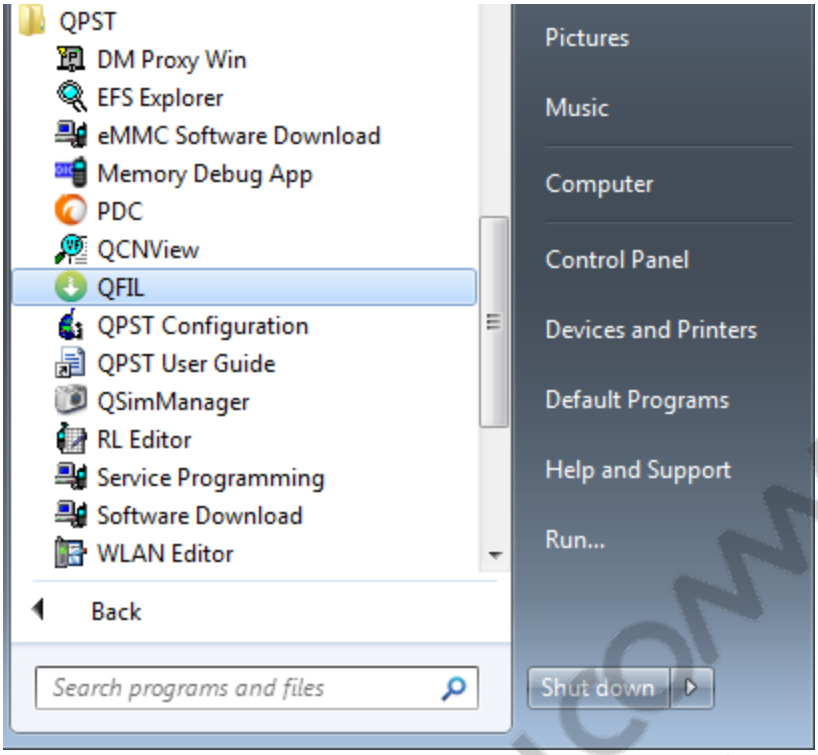


4.4.3 Flashing Android

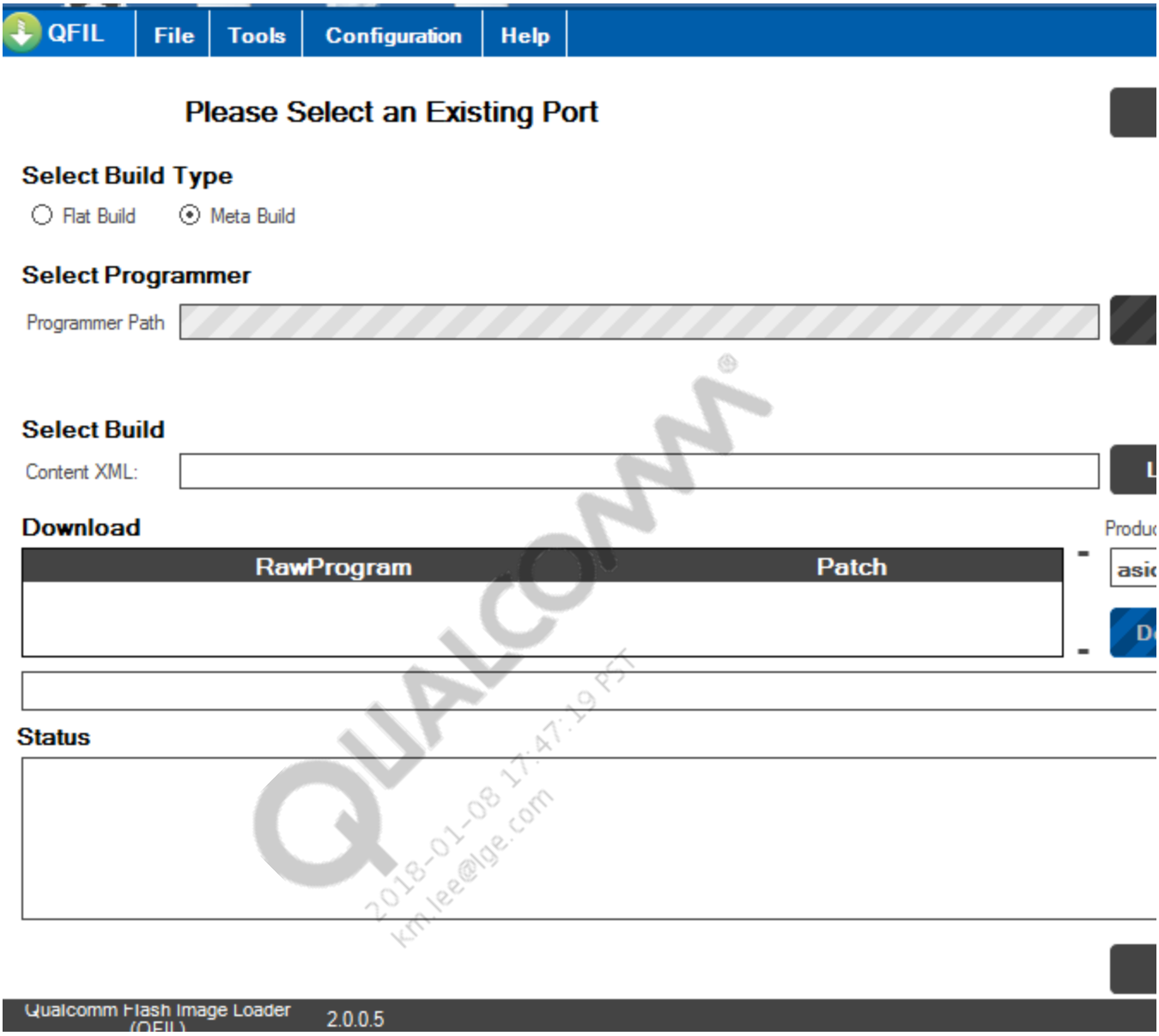
1. Ensure that you are using QPST build 445 or later, on your Windows 7 PC.
2. Check for Emergency Download mode enumeration on the Device Manager.



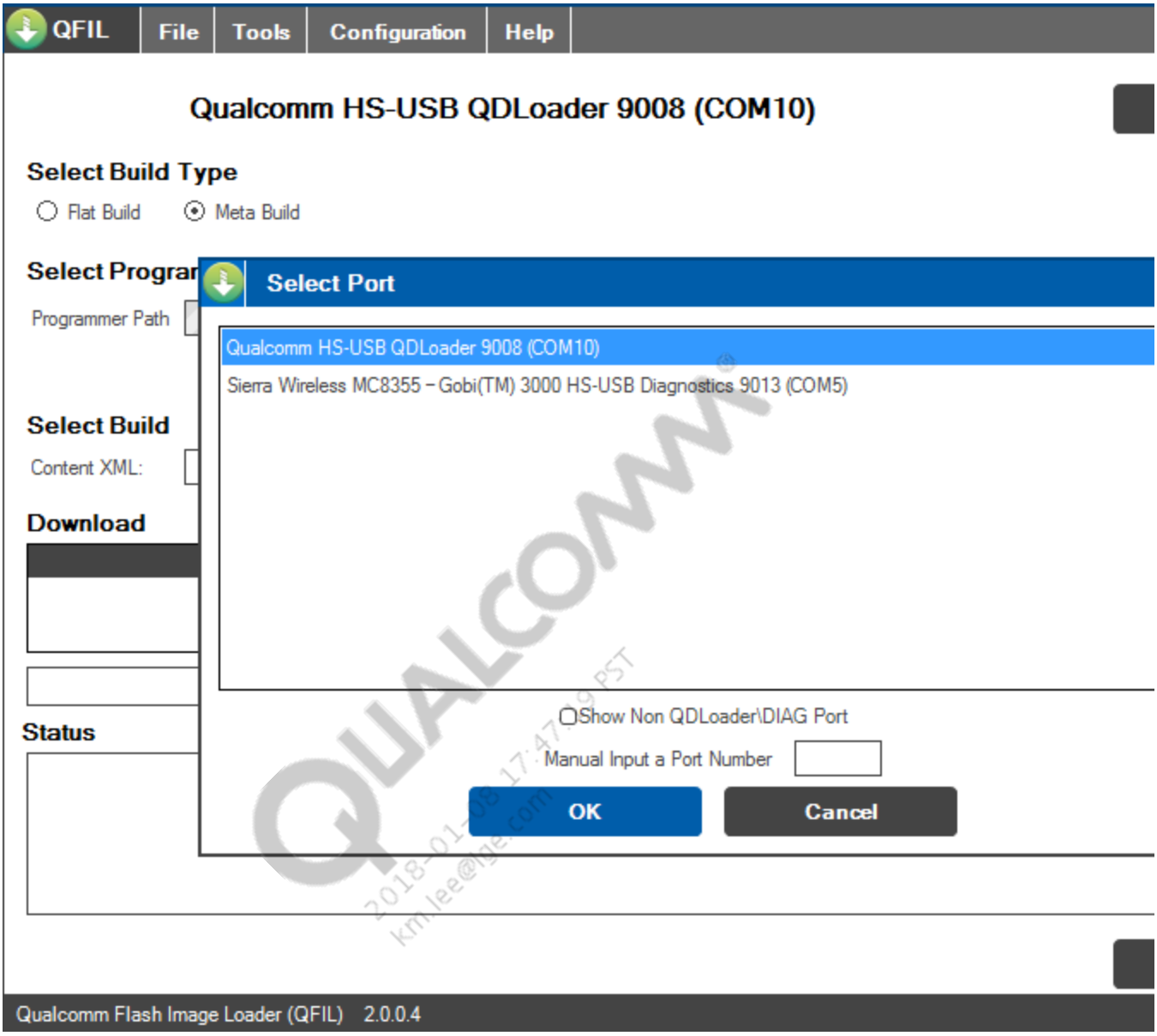
1. Launch the QFIL/firehose app from C:\Program Files (x86)\Qualcomm\QPST\bin



QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com



1. Click **SelectPort** and select the **COM** port. Ensure that the correct COM port is selected, then click **OK**.



1. A new device automatically boots in Emergency Download mode. To force a device with software already loaded to boot in Emergency Download mode, use the following switch settings:

- Turn ON the fourth switch on S1 for S820 ADP Ver 3

1. Select a build type. It is recommended to use the Meta Build type.
2. Click **Load Content** and locate contents.xml at a top-level directory for your project tree. For example , a directory, Project_8996, as shown, is the top-level directory. Place contents.xml and other directories that were downloaded and extracted from QTI's ChipCode portal, under this top-level directory.

   A path to a flash programmer is populated automatically. The flash programmer, prog_emmc_firehose_8996_ddr.elf, is located in <project dir> \boot_images\QcomPkg\Msm8996Pkg\ Bin64\.
   Additional .xml files are populated automatically in a Download box.

1. Choose **asic** for Product Flavors.

Note: Select **modemlite** for APQ8096AU boards.

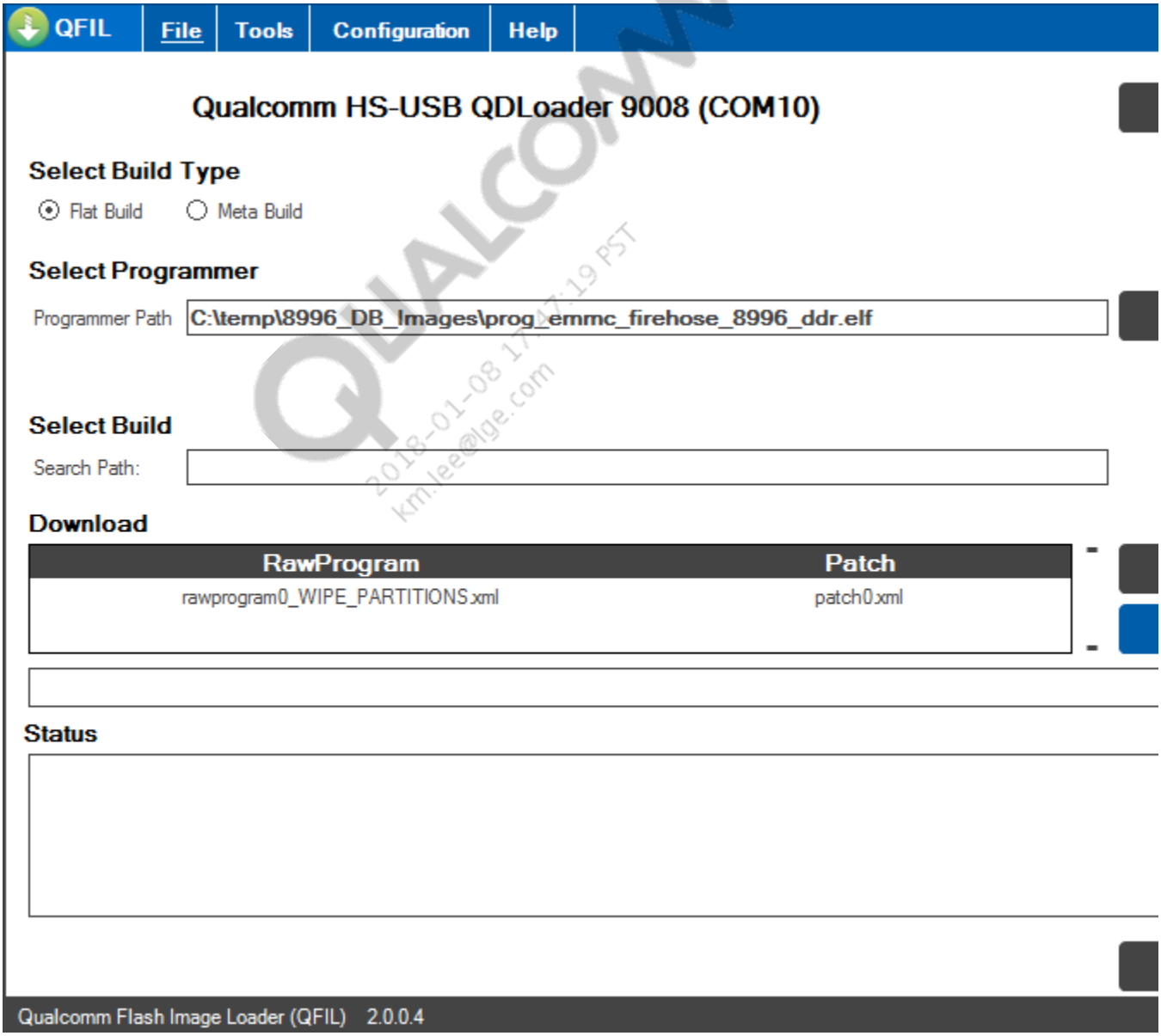
1. Click **Download Content**. The software starts downloading automatically after it goes through all the search paths.
2. When downloading is complete, change the switch as follows and then reboot:

- Turn OFF the fourth Switch on S1 for S820 ADP Ver 3

Android comes up.

4.4.4 Erase Android

1. Follow steps 1 and 6 in Section 4.4.3.
2. Switch the Build Type from Meta Build to Flat Build.
3. Click **LoadXML...** and select `common\build\emmc\rawprogram0_WIPE_PARTITIONS.xml` and `patch0.xml`.



1. Click Download to erase the eMMC.
2. Change the switches to:
    - Turn OFF the first switch on S2 for S820 ADP V1
    - Turn OFF the fourth switch on S1 for S820 ADP V3
1. Reboot. Android should not boot.

4.5 Flash images using fastboot on S820 ADP Ver 3

Before the system images are programmed using Fastboot, the Android boot loaders must already be flashed on the target and must be functional.

1. Plug the USB cable into the target.
2. Put the device into Fastboot mode.
3. On the S820 ADP Ver 3, hold the volume-down key and power-cycle the device.
4. Confirm that Fastboot is active from the Windows command shell:

   fastboot devices

The list of registered devices is shown.

1. After the device is detected, flash the binaries to the target. The following commands run all the fastboot steps at once.

   ```
   cd <target_root>/common/build
   ```

   ```
   cd <target_root>/common/build
   ```

   ```
   fastboot_complete.py -pf=asic -st=<storage_type> (will load only _A partitions)
   ```

   ```
   <storage_type> can be "emmc"
   ```

Note: Select product flavor as **modemlite** for APQ8096AU as follows:

```
fastboot_complete.py --pf=modemlite --st=<storage_type>
```

Use -bp option to load both A & B partitions (fastboot_cpmplete.py -st=asic -bp)

Each binary can be flashed selectively through the following Fastboot commands:

```
fastboot flash partition common/build/emmc/gpt_both0.bin
```

```
fastboot flash aboot_a LINUX/android/out/target/product/msm8996/
emmc_appsboot.mbn
```

```
fastboot flash bluetooth_a common/build/emmc/bin/BTFM.bin
```

```
fastboot flash dsp_a adsp_proc/build/dynamic_signed/adspso.bin
```

```
fastboot flash modem_a common/build/emmc/bin/asic/NON-HLOS.bin
```

Note: for APQ8096AU boards, the above command will be:

```
fastboot flash modem_a common/build/emmc/bin/modemlite/NON-HLOS.bin
```

```
fastboot flash pmic_a boot_images/QcomPkg/Msm8996Pkg/Bin64/pmic.elf
```

```
fastboot flash xbl_a boot_images/QcomPkg/Msm8996Pkg/Bin64/xbl.elf
```

```
fastboot flash tz_a trustzone_images/build/ms/bin/IADAANAA/tz.mbn
```

```
fastboot flash hyp_a trustzone_images/build/ms/bin/IADAANAA/hyp.mbn

fastboot flash devcfg_a trustzone_images/build/ms/bin/IADAANAA/devcfg.mbn

fastboot flash keymaster_a trustzone_images/build/ms/bin/IADAANAA/keymaster.mbn

fastboot flash cmnlib64_a trustzone_images/build/ms/bin/IADAANAA/cmnlib64.mbn

fastboot flash cmnlib_a trustzone_images/build/ms/bin/IADAANAA/cmnlib.mbn

fastboot flash rpm_a rpm_proc/build/ms/bin/AAAAANAAR/rpm.mbn

fastboot flash boot_a LINUX/android/out/target/product/msm8996/boot.img

fastboot flash system_a LINUX/android/out/target/product/msm8996/system.img

fastboot flash vendor_a

LINUX/android/out/target/product/msm8996/ vendor.img

fastboot flash persist LINUX/android/out/target/product/msm8996/persist.img

fastboot flash userdata LINUX/android/out/target/product/msm8996/userdata.img

fastboot flash lksecapp_a trustzone_images/build/ms/bin/IADAANAA/lksecapp.mbn
```

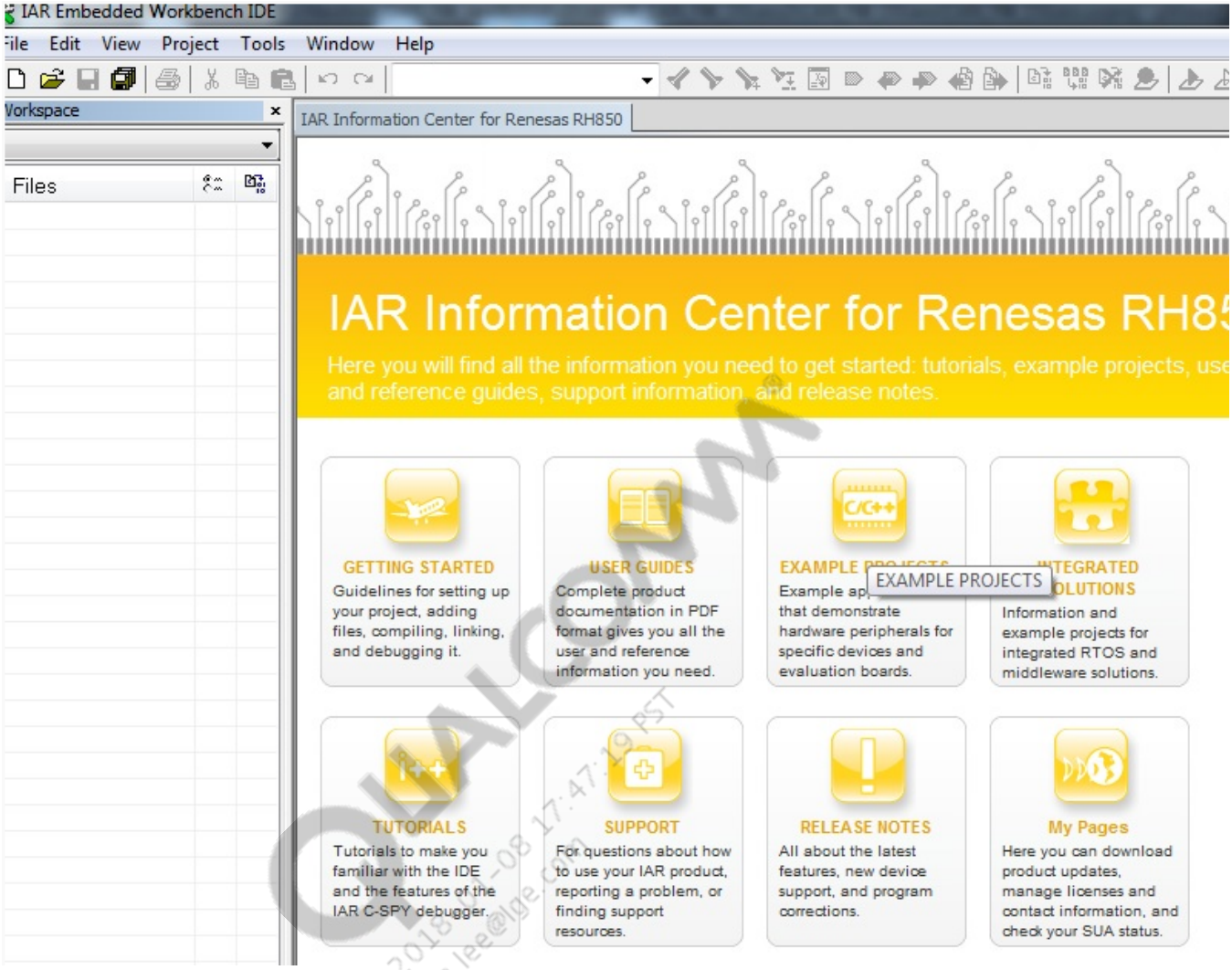
1. After the intended binaries are loaded, power cycle the device.
2. The device boots up into Android.

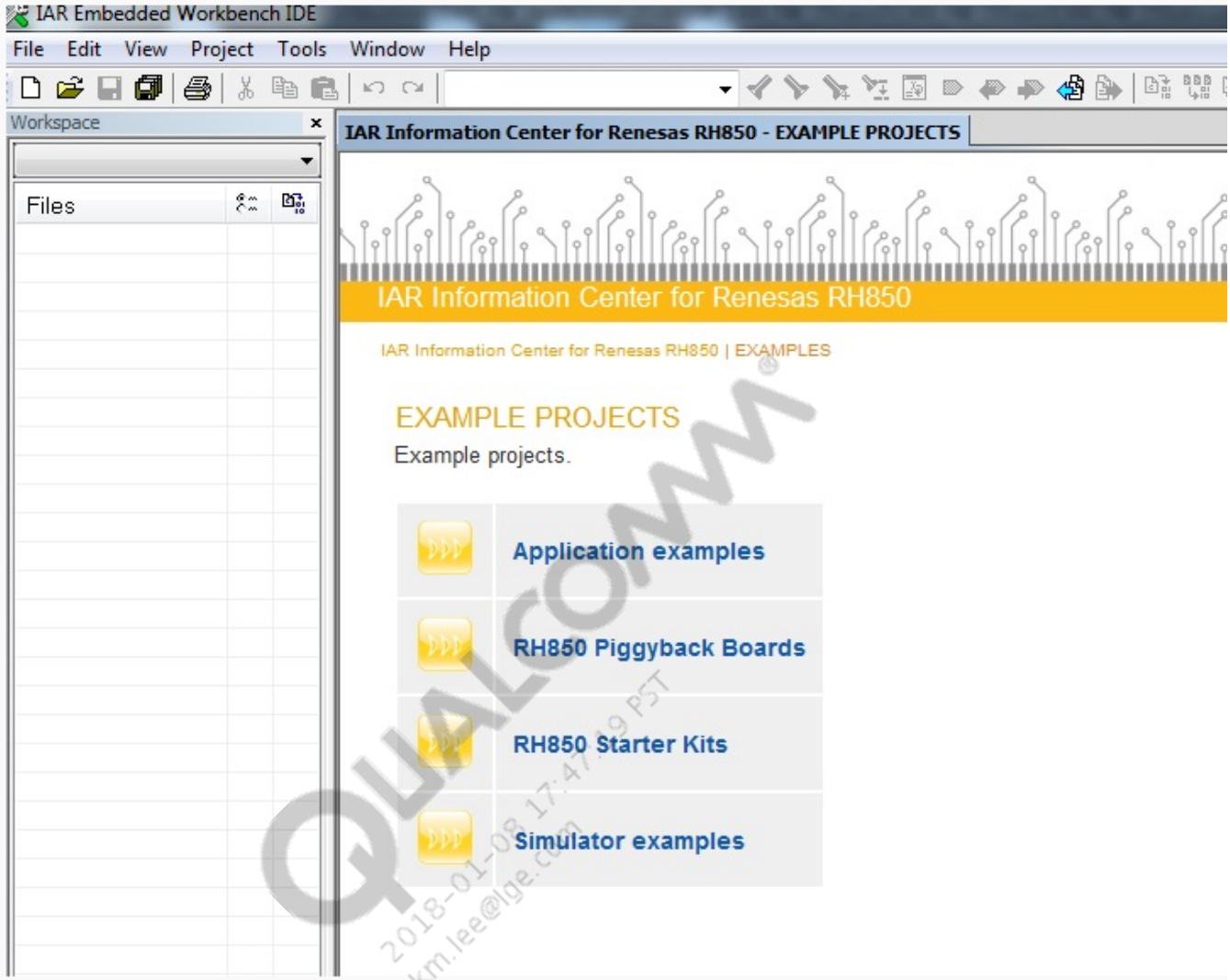
4.6 RH850 firmware

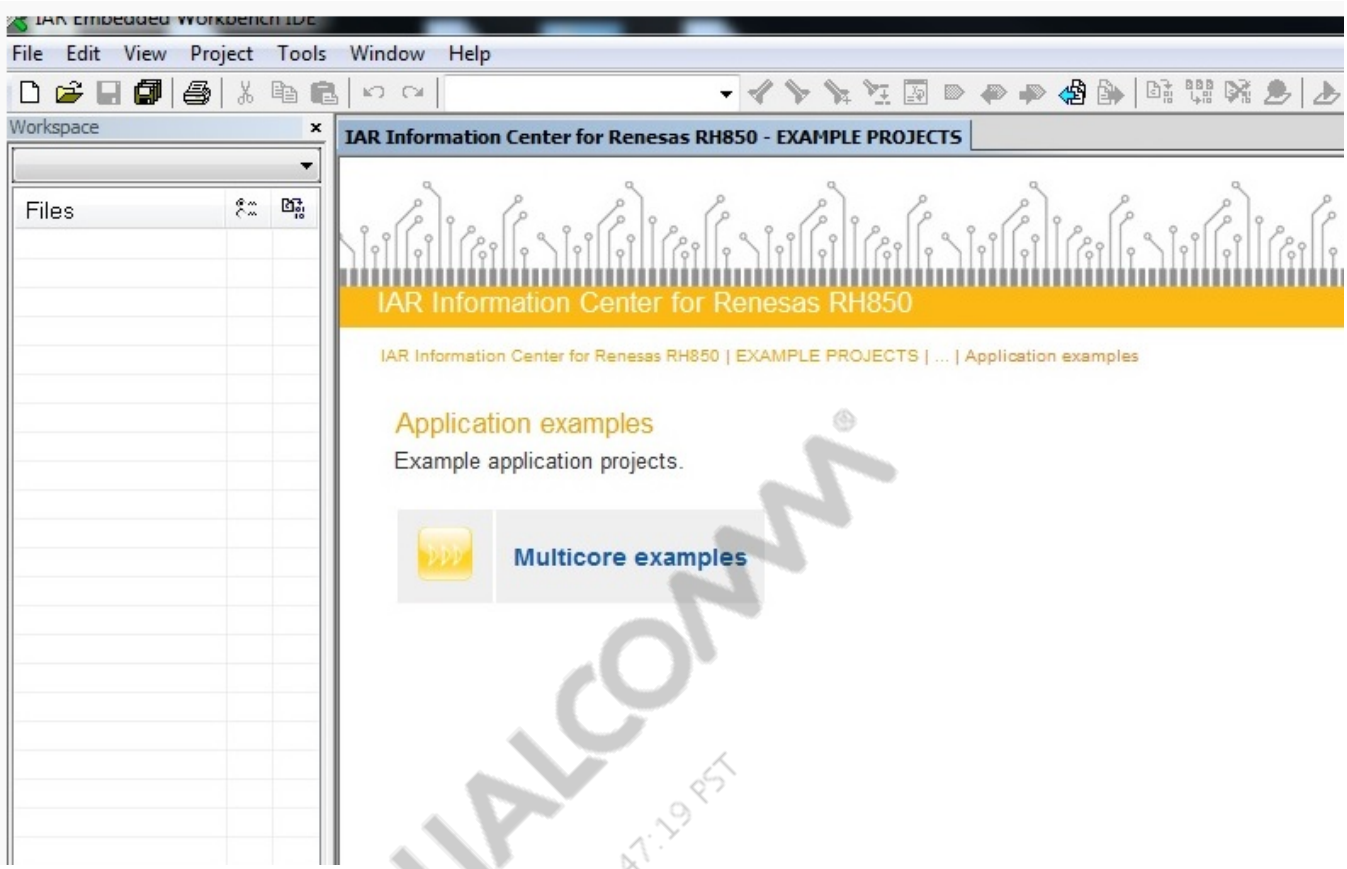
Install IAR embedded workbench tool from: https://www.iar.com/iar-embedded-workbench/renesas/rh850/

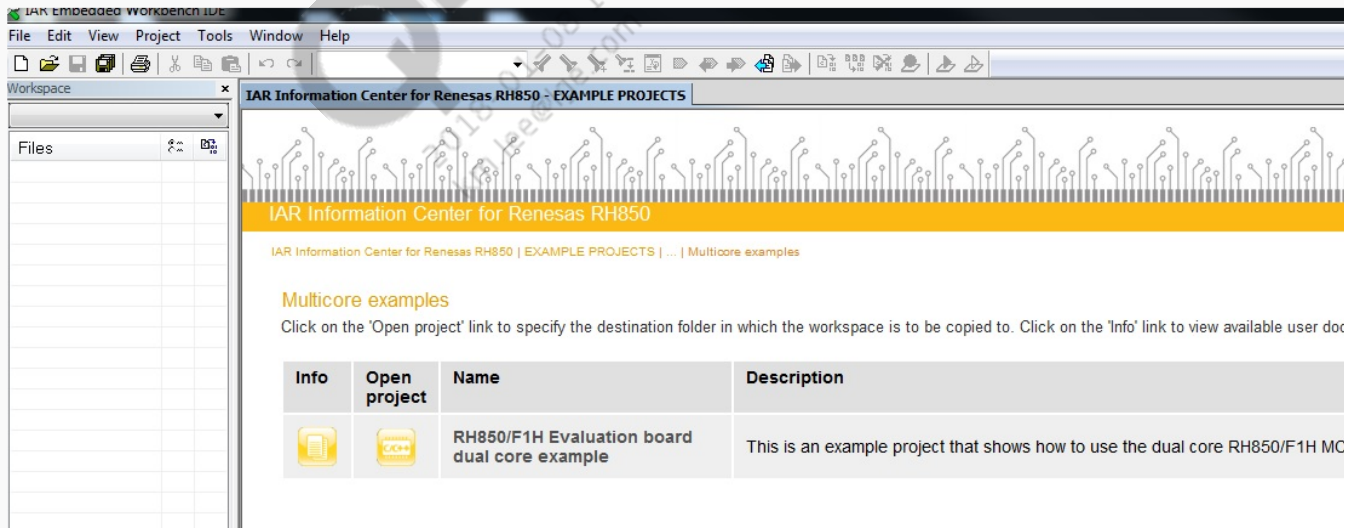
If you need full CAN functionality with firmware upgrade feature from APQ, select the Time limited (30 days) license or buy the full license from IAR. If you only need basic CAN and audio functionality, select the 32K Kickstart, size limited evaluation version.

1. Open IAR embedded Workbench IDE
2. Select Example Projects>RH850 Starter Kits>Application examples>Multicore examples>Open Project

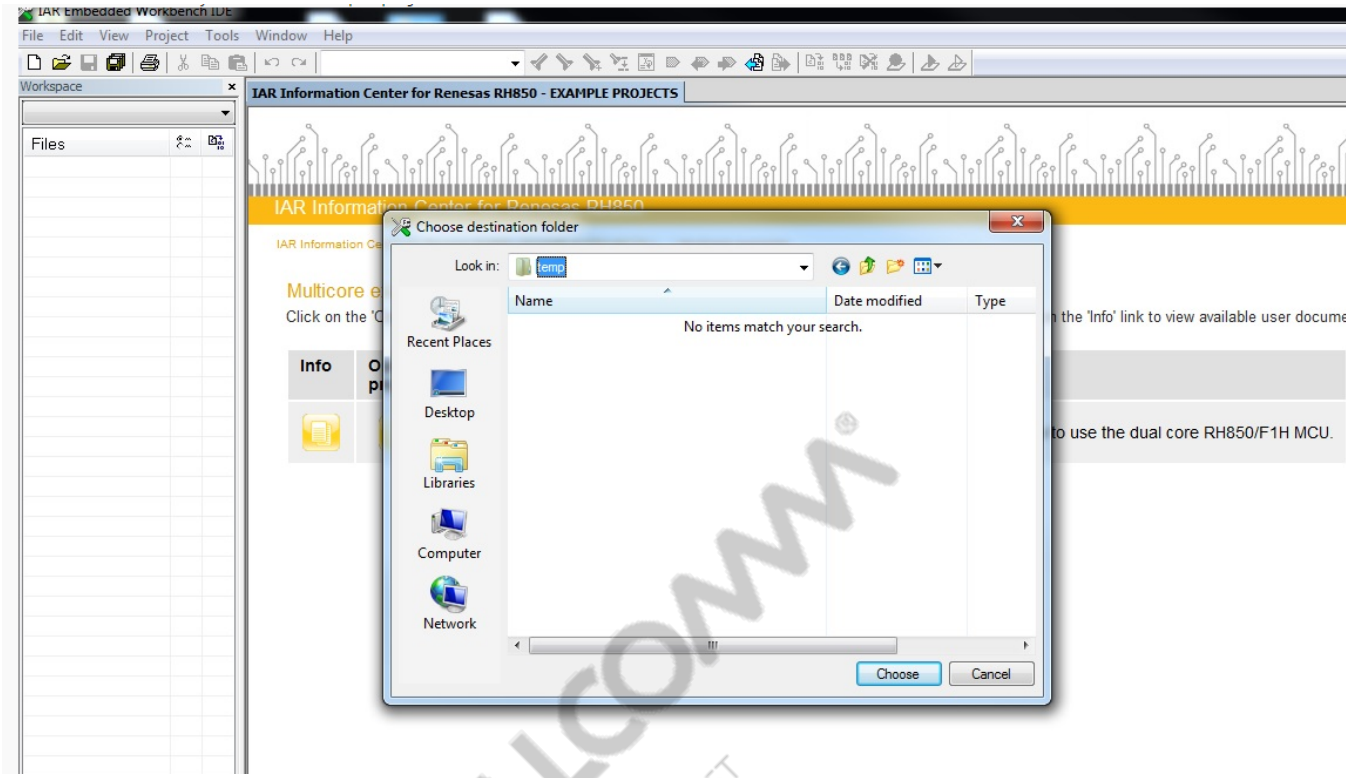

IAR Embedded Workbench IDE
File Edit View Project Tools Window Help
Workspace
Files
IAR Information Center for Renesas RH850
IAR Information Center for Renesas RH85
Here you will find all the information you need to get started: tutorials, example projects, use
and reference guides, support information, and release notes.
GETTING STARTED
Guidelines for setting up your project, adding files, compiling, linking, and debugging it.
USER GUIDES
Complete product documentation in PDF format gives you all the user and reference information you need.
EXAMPLE PROJECTS
Example ap
that demonstrate hardware peripherals for specific devices and evaluation boards.
EXAMPLE PROJECTS
INTEGRATED
OLUTIONS
Information and example projects for integrated RTOS and middleware solutions.
TUTORIALS
Tutorials to make you familiar with the IDE and the features of the IAR C-SPY debugger.
SUPPORT
For questions about how to use your IAR product, reporting a problem, or finding support resources.
RELEASE NOTES
All about the latest features, new device support, and program corrections.
My Pages
Here you can download product updates, manage licenses and contact information, and check your SUA status.


IAR Embedded Workbench IDE
File Edit View Project Tools Window Help
Workspace
Files
IAR Information Center for Renesas RH850 - EXAMPLE PROJECTS
IAR Information Center for Renesas RH850
IAR Information Center for Renesas RH850 | EXAMPLES
EXAMPLE PROJECTS
Example projects.
Application examples
RH850 Piggyback Boards
RH850 Starter Kits
Simulator examples

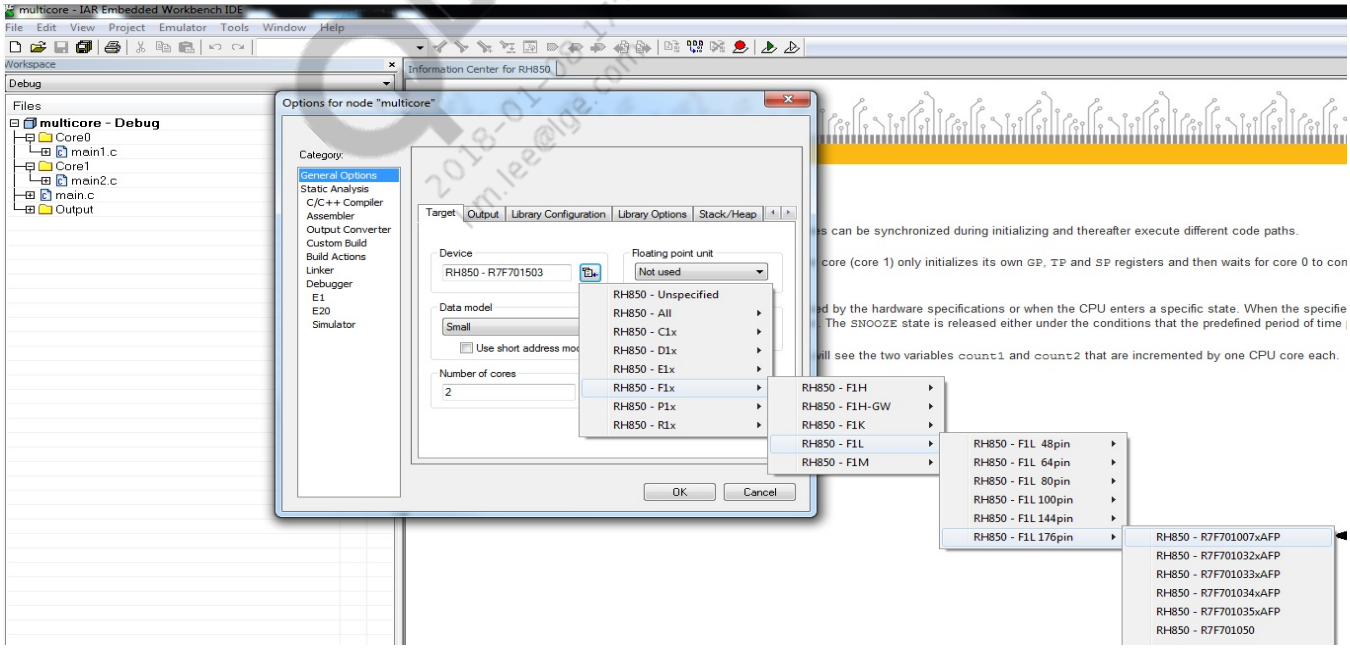




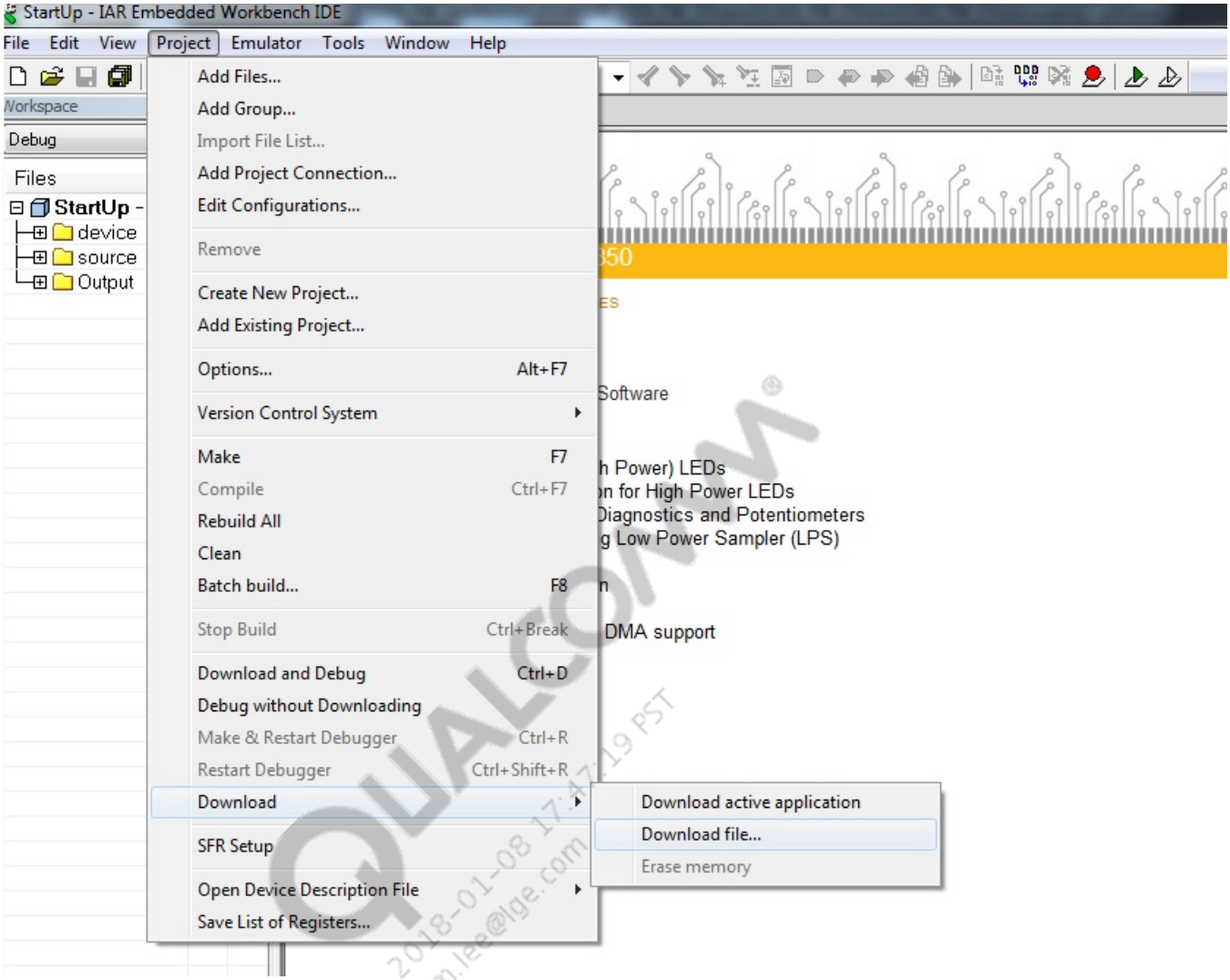
1. Choose a folder locally to save the example project



1. Once the project is saved, select Project->options and change the "Device" to ""



1. Select Project>Download>Download file.



For basic CAN and audio functionality point to the firmware binary at:
`./vendor/qcom/proprietary/vnw/firmware/RH850_Firmware.out`

For full CAN features including firmware upgrade feature over adb, select and flash the following binaries located at:

`./vendor/qcom/proprietary/vnw/firmware/rh850_fw.out`

`./vendor/qcom/proprietary/vnw/firmware/rh850_boot_fw.out`

This is a one-time only requirement. Once you have these two binaries flashed on the target, you can update successive releases of firmware using canflasher tool over adb using the following command.

```
canflasher can0
```

For S820 ADP Ver 3, TURN ON switches S8, S9, and S11 before running canflasher tool. Turn OFF S8, S9, and S11 after flashing.

5. Memory configuration

This software can be used with the MSM™ chipsets ASICs and revisions, with the indicated release quality. ASIC revisions available at the time of this release are assumed to be supported, unless otherwise indicated .

5.1 Supported ASICs

5.1.1 Platform information

The MSM8996 Automotive platform is supported in this release.

5.1.2 Memory configuration and usage

Figure 2-1 shows the memory map configurations for the MSM8996 ASIC.

NOTE: This is a preliminary memory map and may change in the future releases. It is for reference-only in this release and should not be used for optimization.

QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com

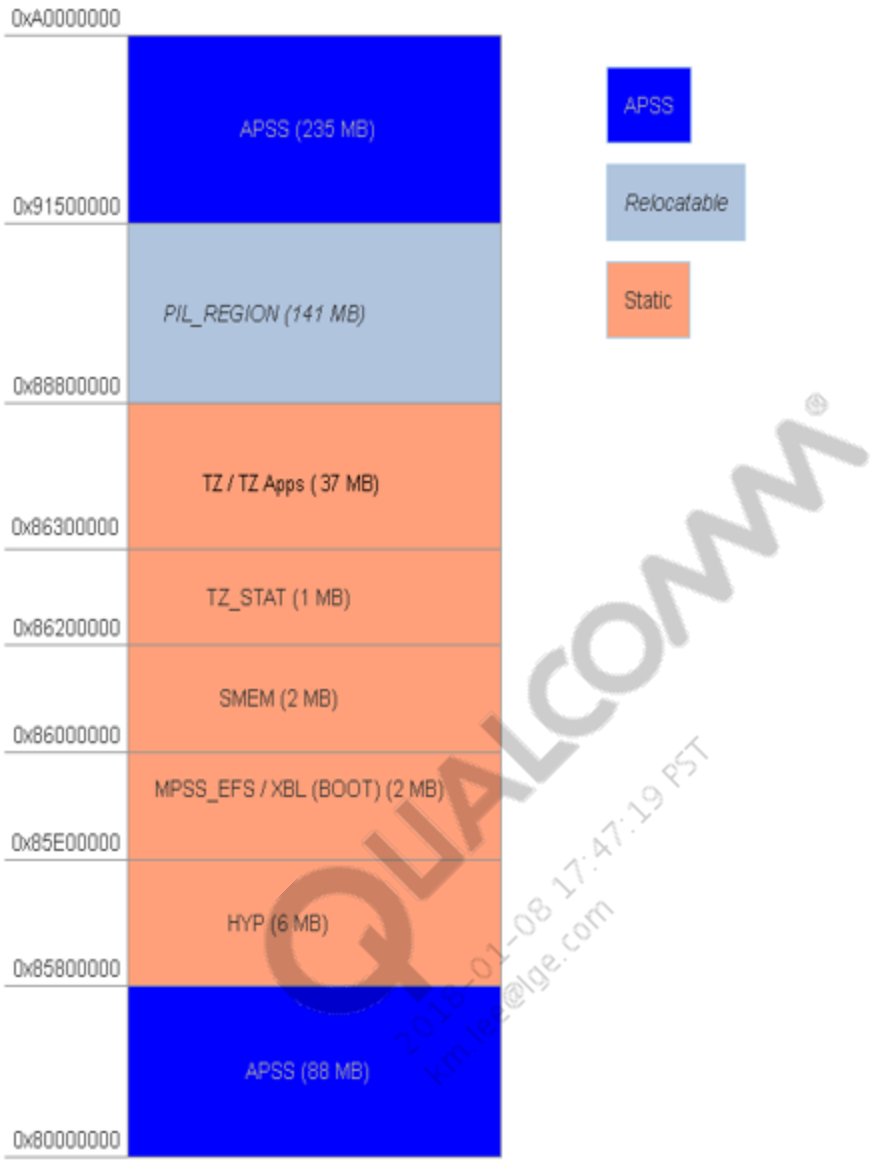


Figure 2-1 MSM8996 memory map diagram

6. New Features

This chapter describes the new software features introduced in this release and a description of those features.

6.1 New features

- Treble enablement
- dm-verity enablement in Linux
- 6/8GB DDR support
- SPI slave support
- User build

6.2 Features not part of this release

This section is not applicable to this release.

QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com

7. Limitations

This chapter lists the bugs and limitations reported for this product line:

- New - Newly reported bugs and limitations
- Ongoing - Previously reported bugs and limitations
- Resolved - Previously reported bugs and limitations that have been resolved and are no longer relevant

Note: For a list of all completed Change Requests (CRs), see the file named fixed_crs.xls, which is located in the root directory for each software image. In addition, a file named Known_Issues.xls (in the root directory for each software image) contains the known CRs in this release.

7.1 New

7.1.1 Bugs Mojave V3

- CR#2159174: Automotive [Android "O" 1.1.5] : Native HDMI display is not up after flashing the build, need to do hard reboot to make the display UP
- CR2160333: Automotive [Andorid "O" 1.1.5] : Bad page state crash causing device to reboot
- CR#2158079: [WLAN][Performance] SAP 5G HT40 mode was actually brought up into HT20

7.1.2 Bugs Mojave-Lite

- CR#2159160: Remove and insert USB cable, device going to unknown state

7.2 Ongoing

7.2.1 Bugs Mojave V3

- CR#2084940 : Device not going to TCXO mode, XO functionality not working
- CR#2084948 : When we change the volume on device, the changes in volume reflects over the speaker but not on volume bar on the target display and vice versa.
- CR#2104351 : Automotive [Android "O" 1.1.5] : Surround ECNS (SEC) failed
- CR#2104311: CS MO/MT Call**:** Call duration is not showing on target display during CS MO/MT calls : Tested when we were able to make a MO call.
- CR#2131545: Automotive [Android O 1.1.5] : Garbled image observed during flashing the system image
- CR#2147812 : Automotive [Android O 1.1.5] :Native HDMI touch is not working once video starts playing on DSI0 & DSI1
- CR#2140897**:** Automotive [Android O 1.1.5] : FM Audio is not heard over the speakers when user inputs a number from the dial pad

7.3 Resolved

- Need to connect primary display prior to boot

7.3.1 Bugs

For a list of all completed Change Requests (CRs), see the file named fixed_crs.xls, which is in the root directory for each software image.

7.3.2 Limitations

- ES5 build will work with / without the external Rome card MCNs: 20-YA734-H1 + YA788-010 or 20-YA734-H1 + 30-YA451-010.
- To enable new vehicle HAL feature, upgrade RH850 firmware to version 3.0. Refer to Section 4.6 for RH850 flash instruction.
- Bluetooth phone/PBAP may require a restart of the auto board to sync call logs/PBAP contacts. First connection of new phone may not display proper sync, but subsequent reconnection has all information displayed.
- WLAN does not support roaming/handoff scenarios
- This is the first HY11 based user build, only boot has been validated. Full functionality will be verified in later release

QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com

8. Additional Information

8.1 Change request

This section is not applicable for this release.

8.2 Display Information

From ES3 release HDMI is the primary display. Use

fastboot oem select-display-panel hdmi

8.3 Application Notes

8.3.1 A/B slot switching options

Get the number of slots

```
bootctl get-number-slots
```

Get the current slot

```
bootctl get-current-slot
```

Set the active slot (0/1)

```
bootctl set-active-boot-slot 1
```

Reboot device

Fastboot option to set active slot can be used as follows:

```
fastboot set_active <slot>
```

Reboot device and use bootctl to validate current boot slot. Fastboot option can be used if bootctl fails to set slot correctly.

8.3.2 A/B update flashing options

Open a command window on the PC. Run the following command options from the Meta build directory:

Option 1: Flash _a partition only:

```
fastboot_complete.py
```

Option 2: both _a and _b partitions:

```
fastboot_complete.py -bp
```

Option 3: Flash with built APSS images:

QUALCOMM®
2018-01-08 17:47:19 PST
km.lee@lge.com

```
fastboot_complete.py -ap=\\path-to-source-code-root -bp
```

Option 4: Replace a self-built APSS images:

Partitions	Flashing commands
System_a, system_b	fastboot.exe **erase** system_a (or system_b) fastboot.exe **flash** system_a (or system_b) system.img
Vendor_a, vendor_b	fastboot.exe **erase** vendor_a (or vendor_b) fastboot.exe **flash** vendor_a (or vendor_b) vendor.img
boot_a, boot_b	fastboot.exe **erase** boot_a (or boot_b) fastboot.exe **flash** boot_a (or boot_b) boot.img
aboot_a, aboot_b	fastboot.exe **erase** aboot_a (or aboot_b) fastboot.exe **flash** aboot_a (or aboot_b) emmc_appsboot.mbn
Persistent	fastboot.exe **erase** persistent fastboot.exe **flash** persistent persistent.img
Userdata	fastboot.exe **erase** userdata fastboot.exe **flash** userdata userdata.img
Recovery	Partition does not exist any more
Cache	Partition does not exist any more

For partition info, see Google documentation on A/B update: https://source.android.com/devices/tech/ota/ab_updates

To validate the flashing success on the target after reboot:

```
$ bootctl get-number-slots
"2"
$ bootctl get-current-slot
"0"
$ bootctl get-suffix SLOT
"_a"
```

8.3.3 A/B OTA end to end update

1. Sync and build an A/B enabled source code tree (eg: MSM8996.LA.1.2)
2. Make an OTA zip package (msm8996-ota-eng.zip as an example):
3. Copy following Meta images to "<source-code-tree>/device/qcom/msm8996/radio/"

Meta image name	Rename to .img and put into /radio directory
xbl.elf	xbl.img
hyp.mbn	hyp.img
NON-HLOS.bin	modem.img
BTMF.mbn	bluetooth.img
emmc_appsboot.mbn	aboot.img
cmnlib64.mbn	cmnlib64.img

Meta image name	Rename to .img and put into /radio directory
cmnlib.mbn	cmnlib.img
devcfg_auto.mbn	devcfg.img
adspso.bin	dsp.img
keymaster.mbn	keymaster.img
pmic.elf	pmic.img
rpm.mbn	rpm.img
tz.mbn	tz.img

Note: system.img, vendor.img, and boot.img need NOT to be copied to /radio directory.

1. Make OTA package as follows (msm8996-ota-eng.zip is generated in the */out folder):

   ```
   <source-code-tree>$ make otapackage AB_OTA_PARTITIONS="xbl rpm tz hyp pmic modem
   aboot keymaster cmnlib cmnlib64 devcfg bluetooth dsp system boot vendor"
   ```

1. Copy the msm8996-ota-eng.zip from */out folder to a PC folder PC_*path_to*. On that PC:

   ```
   adb push /PC_path_to/msm8996-ota-eng.zip /data/ota_package
   ```

Note: To resolve pushing permission issues, run:

```
adb shell setenforce 0
```

1. Using the python script listed on Reference, generate the command string:

   ```
   PC_path_to_ota_zip_file> python ota_update.py msm8996-ota-eng.zip
   ```

This will generate a command string to be run on target. For example:

```
update_engine_client --update --follow --payload=file:///data/ota_package/
msm8996-ota-eng.zip --offset=5596 --size=936076432

--headers="FILE_HASH=r9wL5Q+Jt1kbgB/ETpQkpRF8KKEK24TKgG4eoswFi6o=

FILE_SIZE=936076432

METADATA_HASH=LE0lsch9HCWegAViz/bKHAJEWQDFmfDmCgkbCHFXIIk=

METADATA_SIZE=119827

"
```

1. Copy and paste the command sting generated from step 6 to target shell.
2. Once the images are downloaded and partition update is completed, reboot the target. It should boot from the newly updated _b partition.
3. To switch between active boot SLOT, execute the following command.

   ```
   $ bootctl set-active-boot-slot x (0 or 1).
   ```

1. Reboot the target.
2. Check the current slot:

```
$ bootctl get-current-slot.
```

The boot slot number should be slot x.

8.3.4 Reference: ota_update.py

```
#!/usr/bin/env python

import sys

import zipfile

def main():

if len(sys.argv) != 2:

sys.stderr.write('Use: %s <ota_file.zip>\n' % sys.arv[0])

return 1

otazip = zipfile.ZipFile(sys.argv[1], 'r')

payload_info = otazip.getinfo('payload.bin')

payload_offset = payload_info.header_offset + len(payload_info.FileHeader())

payload_size = payload_info.file_size

payload_location = '/data/ota_package/msm8996-ota-eng.c_chenq.zip'

headers = otazip.read('payload_properties.txt')

print (

'update_engine_client --update --follow --payload=[[[1]]]'

' --offset={payload_offset} --size={payload_size}'

' --headers="{headers}"').format(**locals())

return 0

if __name__ == '__main__':

sys.exit(main())
```

9. Test Reports

9.1 MPSS test report

This section is not applicable for this release.

9.2 CNSS test report

This section is not applicable for this release.

9.3 APSS test report

9.3.1 Multimedia test report

This section is not applicable for this release.

9.3.2 Platform test report

Domain	Test case	Test status	Comments
Common	ADB Shell	Pass	-
	Boot	Pass	-
	Touch	Pass	-
	QPST/QXDM Professional™ tool	Pass	-
MM/GFX	Audio Playback	Pass	-
	Video Playback	Pass	-
	Early Camera	Pass	
	Late Camera	Pass	
	SDCard_AudioPlayback	Pass	-
	AV Playback	Pass	-
	Playback - PCM, MP3, FLAC, AAC	Pass	
	Recording	Pass	-
	USB_AudioPlayback	Pass	-
Modem	Voice call	Pass	
	Data call -web browser	Pass	-
Connectivity	Bluetooth A2DP	Pass	
	Bluetooth AVRCP	Pass	
	Bluetooth PBAP	Pass	-
	Bluetooth HFP	Pass	
	Wi-Fi Station Mode	Pass	

Domain	Test case	Test status	Comments
	Wi-Fi AP Mode	Pass	
CoreBSP	USB	Pass	-
	SD Card	Pass	-
Location Service	GPS	Pass	-
Vehicle Network	HVAC	Pass	-
	Driver side playback	Pass	
	FM Radio	Fail	FM Audio is not heard over the speakers
Power	TCXO	Fail	XO not functional.
Display	Single DSI	Pass	
	Two Display	Pass	
	Three Display	Pass	
UFS	UFS boot test	Pass	
A/B Slot switching	A/B Slot switching	Pass	

9.3.3 CTS/CTSV/GTS report

This section is not applicable for this release.

9.3.4 CTS documented failures

This section is not applicable for this release.

9.4 Product stability report

This section is not applicable for this release.

9.5 Cumulative features

This section describes the software features introduced in this release and previous releases.

The following table provides a list of supported software features from previous releases.

Technology	Feature	Comments
Audio	Music attribute	-
	GFX	-
Display	Single Display	-
	Basic Car HMI	-
	Boot	-
	Initial O Support (Early Access)	-

Technology	Feature	Comments
	Wi-Fi (STA mode)	-
	GPS/GNSS support	-
	HVAC fix	-
Modem	Modem datacall	-
Video	Video encode	-
	Video decode	-
Bluetooth	A2DP	-
	PBAP	-
	HFP	-
Wi-Fi	AP mode	-
Camera	qcarcam_test	-
Early services	Camera, Splash	-
	Suspend-to-RAM	-
	eAVB support	-
	System/Vendor image split	-
	New Camera HAL implementation (AIS)	-
Vehicle Network	Vehicle HAL Service (QC Implementation)	-

9.5.1 Testing AIS Camera

1. Connect target using adb shell with root access.

```
$adb root
$adb remount
$adb shell
```

1. Launch AIS server

```
$cd /system/bin
$ais_server &
```

1. Launch AIS camera test application

```
$cd /system/bin
$qcarcam_test -config=qcarcam_config_single.xml
```

Note: qcarcam_config_single.xml runs camera input stream 0, to change this edit "input_id" field in qcarcam_config_single.xml

10. References

Title	Number
Qualcomm Technologies, Inc.	
GUID Partition Tables and Programming	80-N7350-1
Introduction to Qualcomm Createpoint	80-NC193-1
Qualcomm CreatePoint User Guide	80-NC193-2
MSM8996 Linux Android Software User Manual	SP80-NV396-4
Using Qualcomm Flash Image Loader (QFIL) on APQ8096 DragonBoard	80-NL475-9
Software Configuration Data Table (CDT)	80-N3411-1
Qualcomm Snapdragon 820 Automotive Development Platform Quick Start User Guide	80-NU574-123

QUALCOMM
2018-01-08 17:47:19 PST
km.lee@lge.com