Qualcomm Technologies, Inc.

Qualcomm Firehose Validated Image

Programming Guide

80-P9116-1 C

February 7, 2019

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	October 2016	Initial release
B	January 2018	Updated VIP download procedure in Chapter 3 and 4.
C	February 2019	Updated the title and Section 3.1.5, *Load signed digests and signed images to device.*

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document explains validated image programming (VIP) and introduces the process of VIP.

1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

Code variables appear in angle brackets, for example, `<number>`.

Commands to be entered appear in a different font, for example, **`copy a:*.* b:`**.

Button and key names appear in bold font, for example, click **Save** or press **Enter**.

Shading indicates content that has been added or changed in this revision of the document.

1.3 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

2 Background

When secure boot is enabled, the primary boot loader (PBL) loads the secondary boot loader (SBL) when the SBL is digitally signed and the signature passes. Likewise, the SBL executes the next stage of boot when that code is digitally signed and its signature passes. This process continues with each stage checking the signature of the next and if any of those signatures along the boot chain do not pass, the phone abruptly halts. As such, the phone is secure, since it only executes the signed codes.

VIP controls which packets are allowed to be issued to the target. This funcionality, not only allows you to control which files are programmed, but also allows the customers to designate which commands are allowed or not to be executed on the target. For more details, refer to *Firehose Protocol v2.0 Definition Document (*80-NG319-1).

When the fuse of chip is blown, security boot automatically enables after the device boots up. The image must be signed before the download. Otherwise, the device is bricked. The following documents introduce how to sign image files. Contact the security team at CDMATech Support if necessary.

- *Sectools: SecImage Tool User Guide* (80-NM248-1)
- *Sectools: FuseBlower User Guide* (80-NM248-3)

3 Download VIP using commandline tool

This section describes the process to download VIP with commandline tool.

3.1 Prerequisites

1. Fh_loader.exe and QSaharaServer.exe from the Qualcomm product suppot tool (QPST) package (available in C:\Program Files (x86)\Qualcomm\QPST\bin). Use QPST version 2.7.471.0 or later. QPST installation package is available at CreatePoint.
2. Images signed using the Sectools.
3. Device in 9008 (in EDL mode).

3.1.1 Create a folder

Create a folder with all the signed images for secure boot. Regenerate non-HLOS.bin with the signed pil split binaries and copy to this folder.

For example: C:\Projects\8909\images

Mandatory images: sbl1, TZ, rpm, appsbl, NON-HLOS.bin

NOTE: All the split binaries of mba, modem, pil images, TZ apps and so on are merged into a non-HLOS.bin. Once all these images are signed, replace the signed split binaries in the respective Meta build path and regenerate the non-HLOS.bin. This should be used for generating the digests. For more details on how to generate non-HLOS.bin, please refer the release notes of corresponding chipset

NOTE: If FLAT build is generated, copy the signed images and regenerated non-HLOS.bin into the FLAT build folder

3.1.2 Create a table of digests

1. Perform the dry run to create the table of digests using the following command.

```
fh_loader.exe --erase=0 --port=\\.\COM12 --contentsxml=\\diwali\NSID-
HYD-01\MSM8909.LA.3.0.1-00019-STD.PROD-1\contents.xml --noreset --
showpercentagecomplete --createdigests --
search_path=C:\Projects\8909\images\
```

   - --port: The port on which the device is connected

- --contents.xml: Path to the contents.xml
- --search_path: The path in which signed images are copied. Signed images are taken from this path for programming. For other images, (such as boot.img, system.img) path from contents.xml is used.

2. For FLAT build, use the following command.

```
fh_loader.exe --erase=0 --port=\\.\COM12 --
sendxml=rawprogram_unsparse.xml,patch0.xml --noreset --
showpercentagecomplete --createdigests --
search_path=C:\Users\VIP_testing\Flat_build\eMMC\
--search_path: Path of the FLAT builds
```

```
C:\Program Files (x86)\Qualcomm\QPST\bin>fh_loader.exe --erase=0 --port=\\.\COM12 --contentsxml=\\
diwali\NSID-HYD-01\MSM8909.LA.3.0.1-00019-STD.PROD-1\contents.xml --noreset --showpercentagecomple
te --createdigests --search path=C:\Projects\8909\images\
```

3. The output is shown in the following figure, which creates three files at the same path with fh_loader.
   - DigestsToSign.bin.mbn, this file must be digitally signed.
   - ChainedTableOfDigests.bin
   - DIGEST_TABLE.bin

The digests table is created successfully and is shown in Figure 3-1.

Figure 3-1 Create digest table

3.1.3 Digitally sign the digests and the VIP device programmer

Sign the below two files using same certificates, keys and Secure boot method used for signing the other component images and put them to the images folder. Also copy the ChainedTableOfDigests.bin to the same folder

1. VALIDATED_emmc_firehose_8xxx.mbn or prog_emmc_ufs_firehose_SdmXXX_ddr.elf
2. DigestsToSign.bin.mbn (use -g vip in signing command)

3.1.4 Load device programmer into the device

Send over the signed validated Firehose programmer via Sahara protocol to the device, which is in the EDL mode as follows, and is shown in Figure 3-2.

```
QSaharaServer.exe -p \\.\COM12 -s 13:validated_emmc_firehose_8909_ddr.mbn -
b C:\Projects\8909\images\
```

```
C:\Program Files (x86)\Qualcomm\QPST\bin>QSaharaServer.exe -p \\.\COM12 -s 13:validated_emmc_fireh
ose_8909_ddr.mbn -b C:\Projects\8909\images\
```

Figure 3-2 Load device programmer

The process to transfer the device programmer takes less than one second, and a successfully transferred device programmer is shown in Figure 3-3.

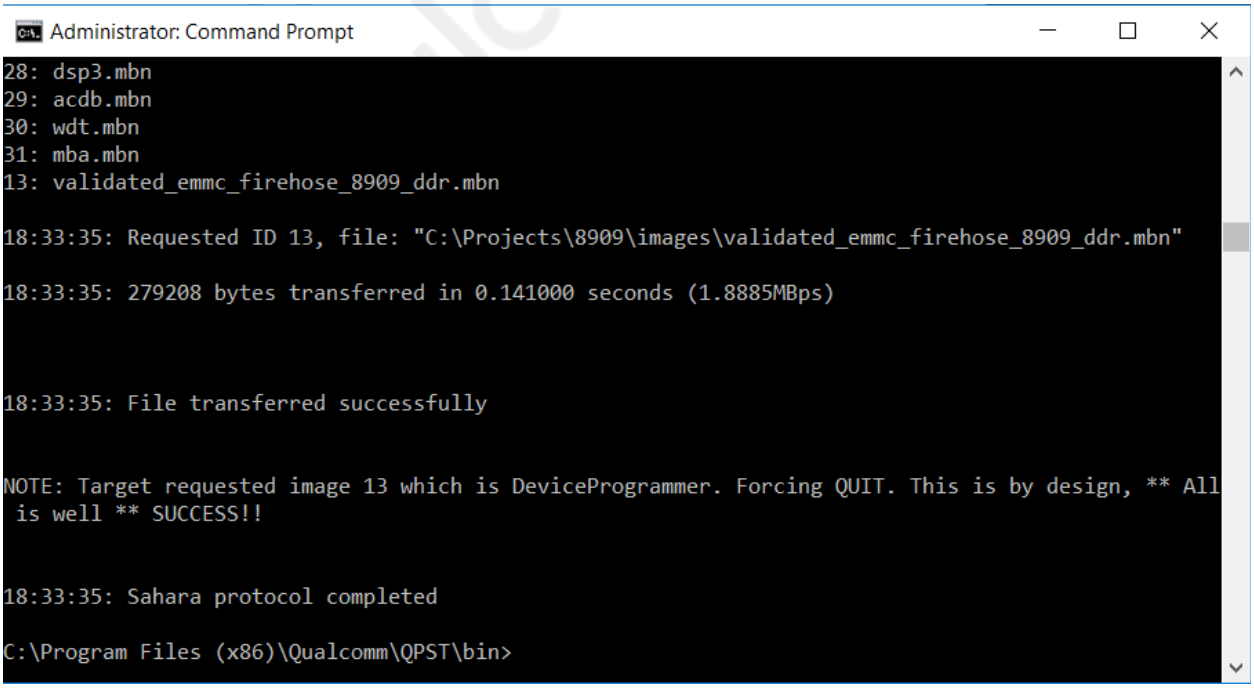


Figure 3-3 Transfer device program

3.1.5 Load signed digests and signed images to the device

Send over the DigestsToSign.bin.mbn signed image files and ChainedTableOfDigests.bin files via Firehose protocol, using the same command as dry run, and replace `--createdigest` with the signed files and chained hash as follows (Figure 3-4):

```
fh_loader.exe --erase=0 --port=\\.\COM12 --contentsxml=\\diwali\NSID-HYD-
01\MSM8909.LA.3.0.1-00019-STD.PROD-1\contents.xml --noreset --
showpercentagecomplete --
```

```
signeddigests=C:\Projects\8909\images\DigestsToSign.bin.mbn --
chaineddigests=C:\Projects\8909\images\ChainedTableOfDigests.bin --
search_path=C:\Projects\8909\images\
```

```
C:\Program Files (x86)\Qualcomm\QPST\bin>fh_loader.exe --erase=0 --port=\\.\COM12 --contentsxml=\\
diwali\NSID-HYD-01\MSM8909.LA.3.0.1-00019-STD.PROD-1\contents.xml --noreset --showpercentagecomple
te --signeddigests=C:\Projects\8909\images\DigestsToSign.bin.mbn --chaineddigests=C:\Projects\8909
\images\ChainedTableOfDigests.bin --search_path=C:\Projects\8909\images\
```

Figure 3-4 Send signed files with digest table

For FLAT build, use the following command:

fh_loader.exe --erase=0 --port=\\.\COM12 --
sendxml=rawprogram_unsparse.xml,patch0.xml --noreset --showpercentagecomplete --
signeddigests=C:\Users\Flat_build\eMMC\DigestsToSign.bin.mbn --
chaineddigests=C:\Users\Flat_build\eMMC\ChainedTableOfDigests.bin --
search_path=C:\Users \Flat_build\eMMC\

The signature is passed for digest binary as shown in Figure 3-5.

```
18:51:57: INFO: Looking for file 'C:\Projects\8909\images\DigestsToSign.bin.mbn'
18:51:57: INFO: Signed Digest Table (8168 bytes) 'C:\Projects\8909\images\DigestsToSign.bin.mbn'
18:51:57: INFO: SIGNED Digest table had 54 digests
18:51:57: INFO: MaxNumDigestsPerTable is 256
18:51:57: INFO: Setting CurrentDigestLocation to number 202, meaning 54 hashes to go before a new
table must be sent
18:51:57: INFO: Calling sendTransmitBufferBytes(8168 bytes)
18:51:57: INFO:          _____ _                    _
18:51:57: INFO:         /  ___(_)                  | |
18:51:57: INFO:         \ `--. _  __ _ _ __   __ _| |_ _   _ _ __ ___
18:51:57: INFO:          `--. \ |/ _` | '_ \ / _` | __| | | | '__/ _ \
18:51:57: INFO:         /\__/ / | (_| | | | | (_| | |_| |_| | | |  __/
18:51:57: INFO:         \____/|_|\__, |_| |_|\__,_|\__|\__,_|_|  \___|
18:51:57: INFO:                   __/ |
18:51:57: INFO:                  |___/

18:51:57: INFO:          ______                        _
18:51:57: INFO:         | ___ \                      | |
18:51:57: INFO:         | |_/ /_ _ ___ ___  ___  __| |
18:51:57: INFO:         |  __/ _` / __/ __|/ _ \/ _` |
18:51:57: INFO:         | | | (_| \__ \__ \  __/ (_| |
18:51:57: INFO:         \_|  \__,_|___/___/\___|\__,_|

18:51:57: INFO: Looking for file 'NON-HLOS.bin'
```

Figure 3-5 Signature passed for digest binary

The VIP programming is performed successfully as shown in Figure 3-6.

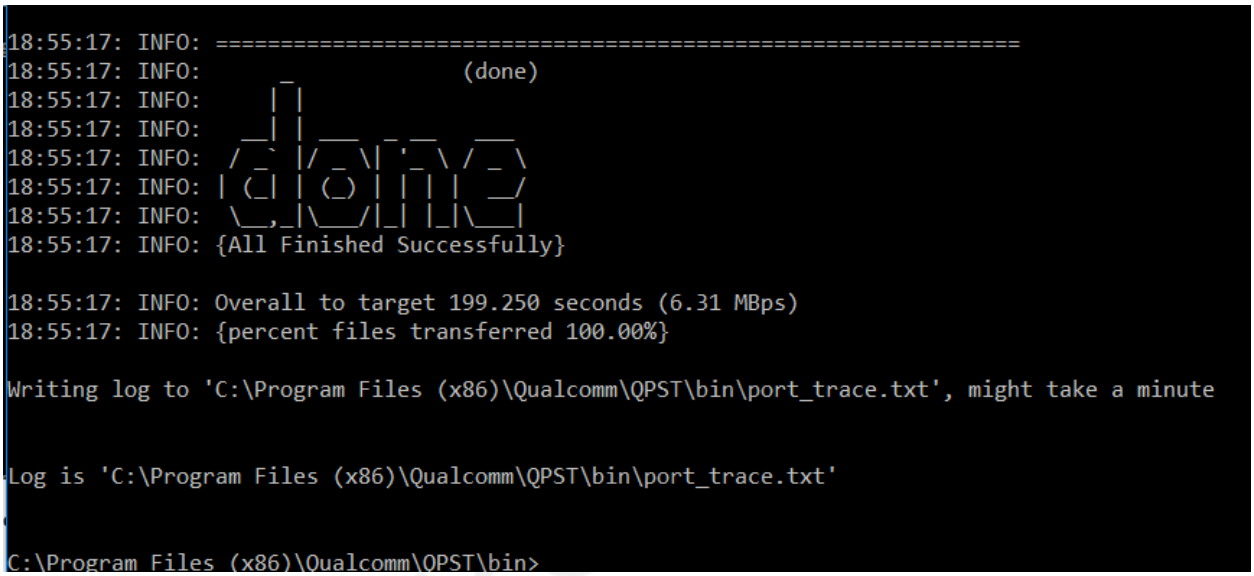


Figure 3-6 VIP programming successful

The VIP controls which packets are allowed to be issued to the target. So it needs to create digests before VIP download. The digest value generated by the dry run must match with the one calculated during VIP download. The VIP needs to:

- Recreate the digest if the configuration is changed, or if the image file is modified.
- Use the singed validated (VIP enabled) Firehose programmer to download VIP.

4 Download VIP using factory tool

There are two xtt files used for VIP download. The DryRun xtt file is used to create the digest table. The Download xtt file is used to download signed images to the devices as shown in Figure 4-1.

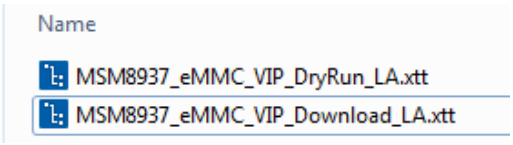
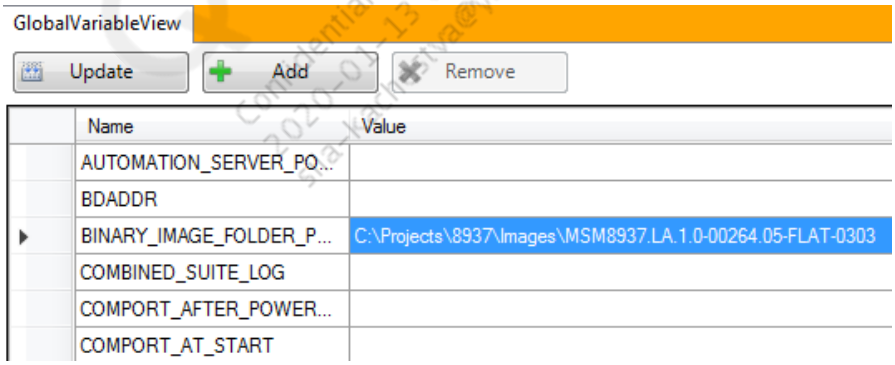


Figure 4-1 VIP download xtt files

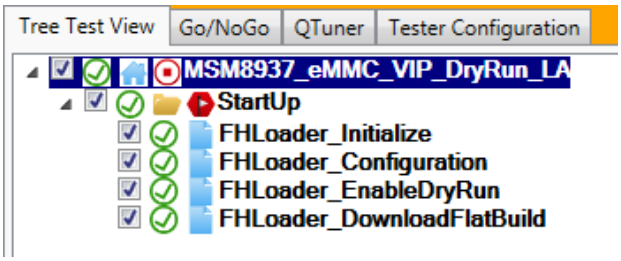
4.1.1 Perform dry run

Perform the dry run using the DryRun xtt file to create the table of digests. The image files must be signed before the dry run.

1. Set up image path in the **GlobalVariableView**.



2. Run the xtt file to create the following files:
   - DigestsToSign.bin.mbn (needs to be signed digitally)
   - ChainedTableOfDigests.bin



The callback log for dry run is shown in Figure 4-2.

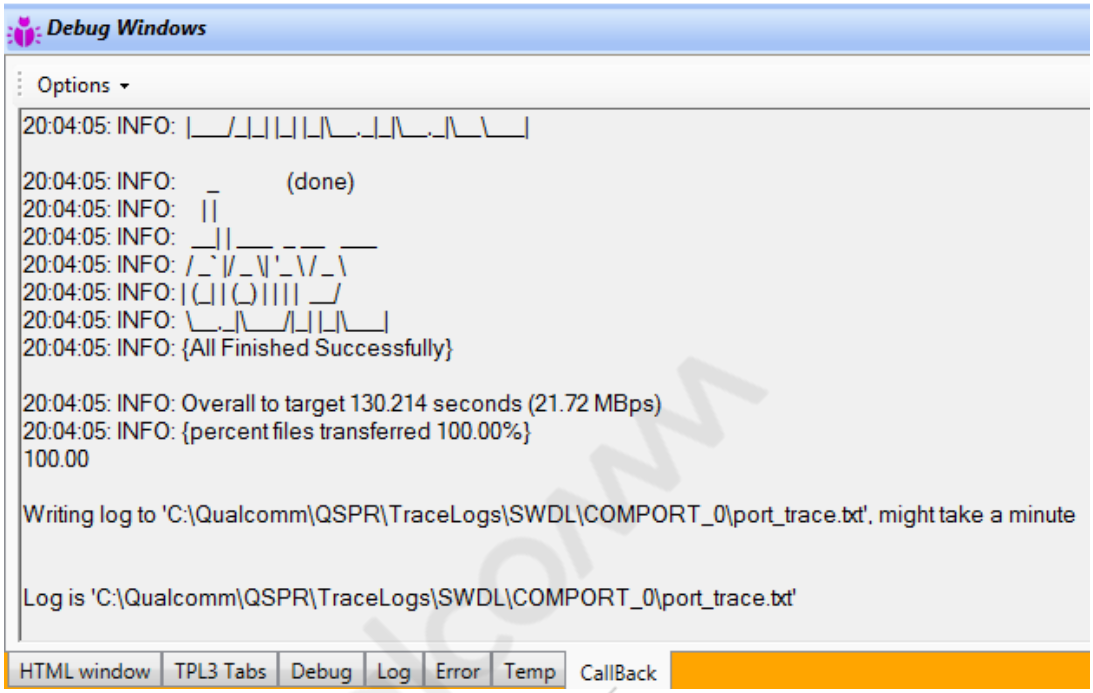


Figure 4-2 Callback log for dry run

The digests files are created at `C:\Qualcomm\QSPR\TraceLogs\SWDL\COMPORT_0`, as shown in Figure 4-3.

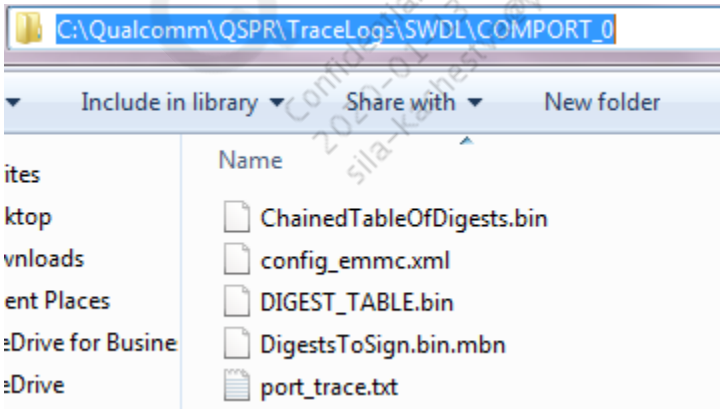


Figure 4-3 Created digests table

4.1.2 Digitally sign two files

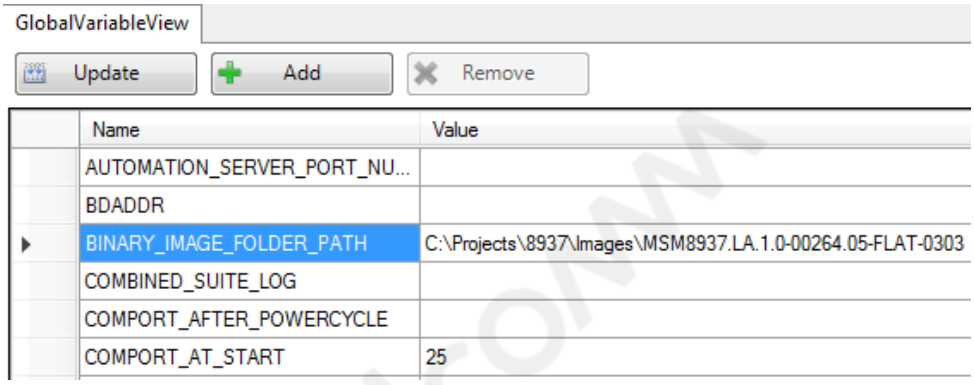
Digitally sign the two files using the same method as SecureBoot, and add them to the image folder:

- VALIDATED_emmc_firehose_8xxx.mbn (device programmer with VIP)
- DigestsToSign.bin.mbn

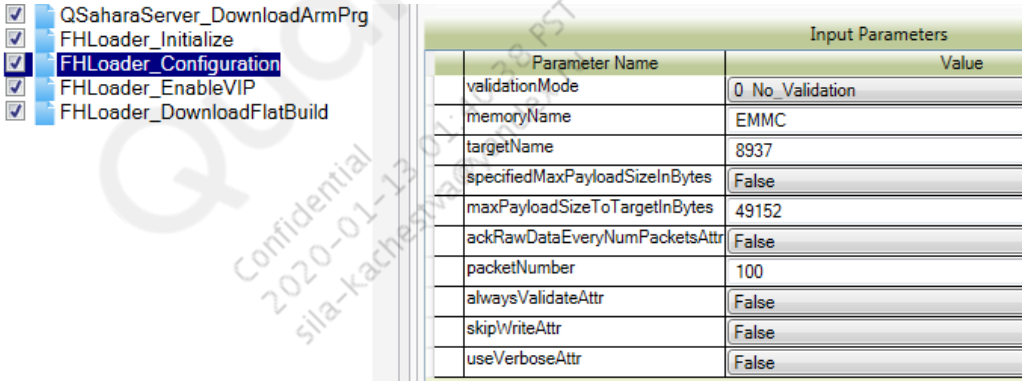
4.1.3 Send over signed image and digest table

Send over the DigestsToSign.bin.mbn signed image files, and the ChainedTableOfDigests.bin via Download xtt files.

1. Force the device to emergency download (EDL) mode
2. Set up the image path and com port number in **GlobalVariableView.**



3. Ensure that the configuration setup is the same as the configuration in dry run.



4. Run Download xtt.

If the hash value of any packet is different from the value in the digest table, VIP fails as shown in Figure 4-4.

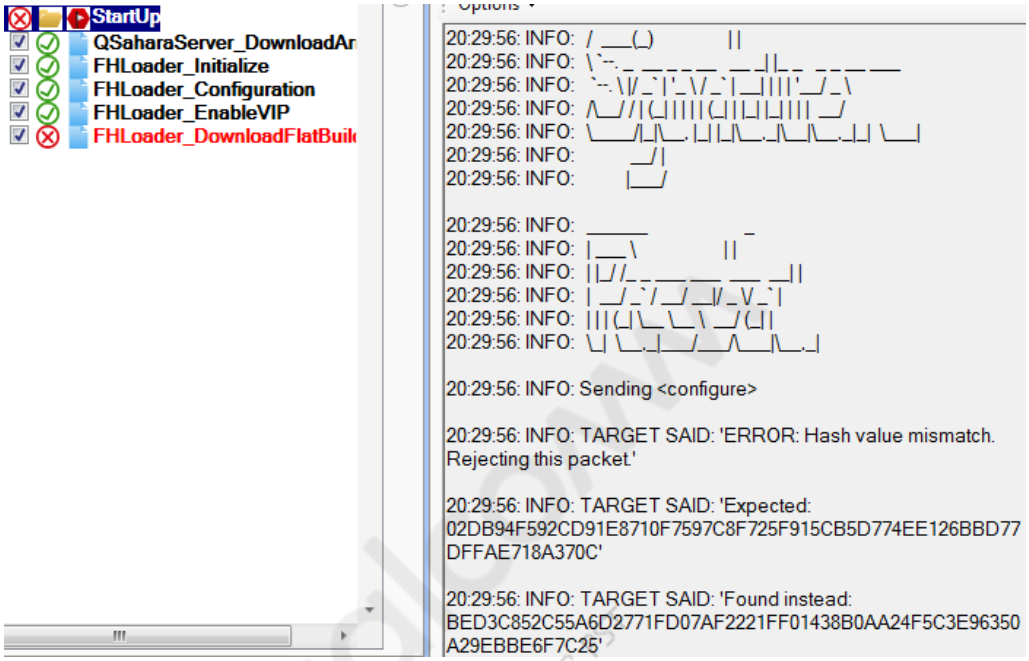


Figure 4-4 Failed VIP

If the hash value of any packet is same as the value in the digest table, then the VIP download is successful as shown in Figure 4-5 and Figure 4-6.

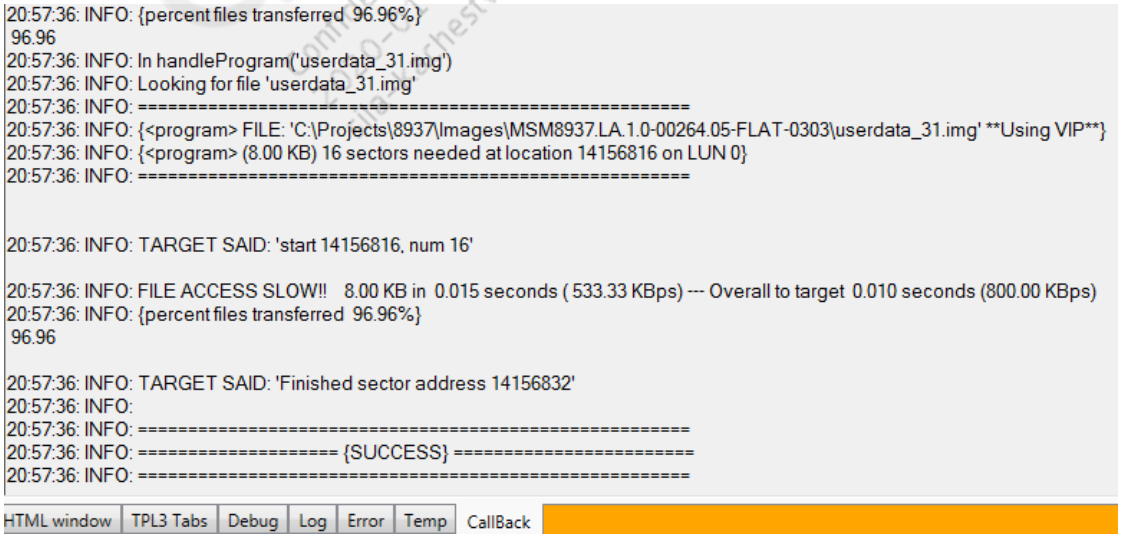


Figure 4-5 Downloading VIP

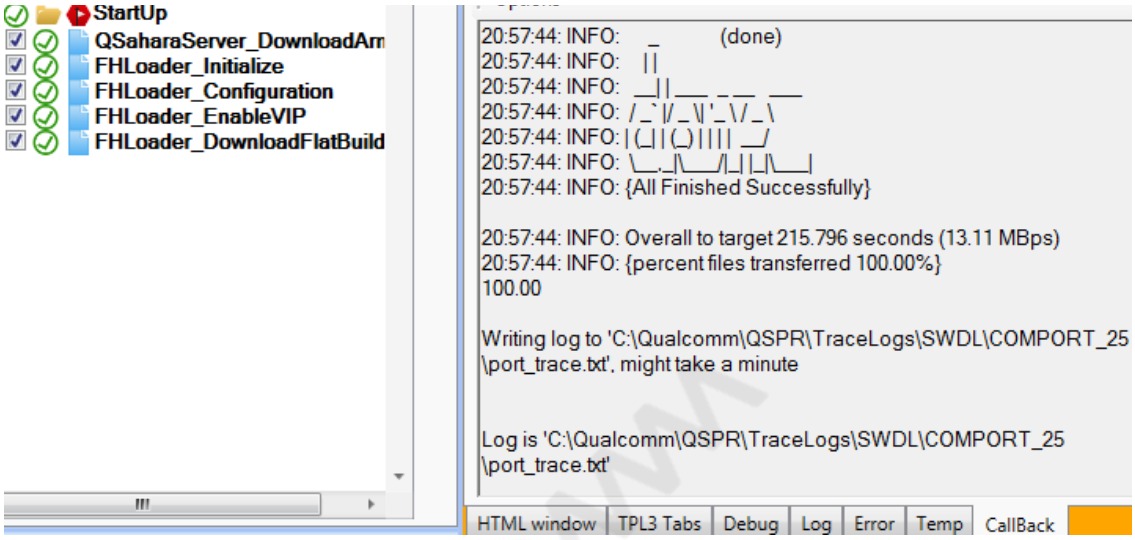


Figure 4-6 Successful VIP download

5 Throughput

Use the following table as reference for the VIP download.

Table 5-1 VIP download throughput

CPU	Intel(R) Core(TM) i5-2540M CPU @ 2.60 GHz
RAM	DDR2 4.0 GB
OS	Windows 7 Enterprise SP1 64 bit
Image size	2.76 GB
Download Time (VIP)	318 seconds
Download Time (No-VIP)	155 seconds
Throughput (VIP)	8.89 Mbps
Throughput (No-VIP)	18.20 Mbps

A References

A.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Firehose Protocol V2.0 Definition Document	80-NG319-1
Qualcomm Firehose Loader User Guide	80-P6549-1
Sectools: SecImage Tool User Guid	80-NM248-1
Sectools: FuseBlower User Guide	80-NM248-3

A.2 Acronyms and terms

Acronym or term	Definition
EDL	Emergency download
PBL	Primary boot loader
QPST	Qualcomm product suppot tool
SBL	Secondary boot loader
VIP	Validated image programming