CDMA Firmware Upgrade Reference

Products supported:

USB modem	USB 598
MiniCard	MC5728V
ExpressCard/PC Card	AirCard 402

SIERRA WIRELESS

2130676
Rev 2

Preface

Limitation of liability

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Patents

This product may contain technology developed by or for Sierra Wireless Inc. This product includes technology licensed from QUALCOMM®. This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group.

Copyright

Trademarks

AirCard® is a registered trademark of Sierra Wireless. Sierra Wireless™, AirPrime™, AirLink™, AirVantage™, Watcher™ and the Sierra Wireless logo are trademarks of Sierra Wireless.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh and Mac OS are registered trademarks of Apple Inc., registered in the U.S. and other countries.

Contact information

Technical support (for **distributors**):	Please contact your Sierra Wireless System Engineer.	
Technical support (for **OEMs**):	Please contact Sierra Wireless Technical Support:	
	Phone:	1-877-231-1144
	E-mail:	support@sierrawireless.com
RMA Support:	repairs@sierrawireless.com	
Post:	Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4	
Fax:	1-604-231-1109	
Web:	www.sierrawireless.com	

For up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, and press releases, consult our website: www.sierrawireless.com.

Revision history

Revision number	Release date	Changes
0.1	Oct 05	Initial document creation
0.2	Oct 05	Added CnS commands for ERI file update
0.3	Feb 06	Cleaned up references to other docs. Added overall flow process
0.4	Feb 06	Added comments from review and notes on work-in-progress
0.5	Mar 06	Added: - API and file format details - Reverse compatibility info - Message flow - New message info
0.6	Apr 06	Message flow clean up. Added references to new versions of firmware.
0.7	Aug 06	Added references for new products, new QC-boot image update step, and HIP Launch Fragment missing information.
0.8	Jun 07	Added SWoC and Minimum Firmware Upgrade revision fields to the Firmware Upgrade Configuration File.
1.1	Dec 07	Added EFS and OMA-DM tree update fields to the Firmware Upgrade Configuration File. Moved SWoC download to end of upgrade in Figure 1 (page 9).
1.2	Feb 10	Edits to Figure 1 “Firmware upgrade process flow” (page 9). Changed the document scope (previously was EM5625 and MC5720). Bugzilla item 11373. Changed the document layout. Miscellaneous wording changes.
2	Aug 12	New template. Changes to the patents section. Removed products that have reached end-of-life (Compass 597, AirCard 595, 595U, 597E, MC 5725, 5725V, 5727, 5727V).

Table of Contents

Introduction

Purpose

This document describes the steps required to perform a field firmware upgrade on the current generation of USB-based Sierra Wireless CDMA WWAN devices (referred to in this document as “modem”).

This document covers the following modems:

- USB 598 modem
- MC5728V MiniCard
- AirCard 402 ExpressCard/PC Card

Scope

This document is intended to help software developers understand the overall process and specific steps that their application must perform, to upgrade the modem’s firmware.

References

1. “*USB Driver Developer's Guide*”, Sierra Wireless, document number 2130634
2. “*CDMA 1xEV-DO CnS Reference Guide*”, Sierra Wireless, document number 2130754

Process overview

Components of a firmware upgrade package

A firmware upgrade package for a CDMA modem is made up of the following components:

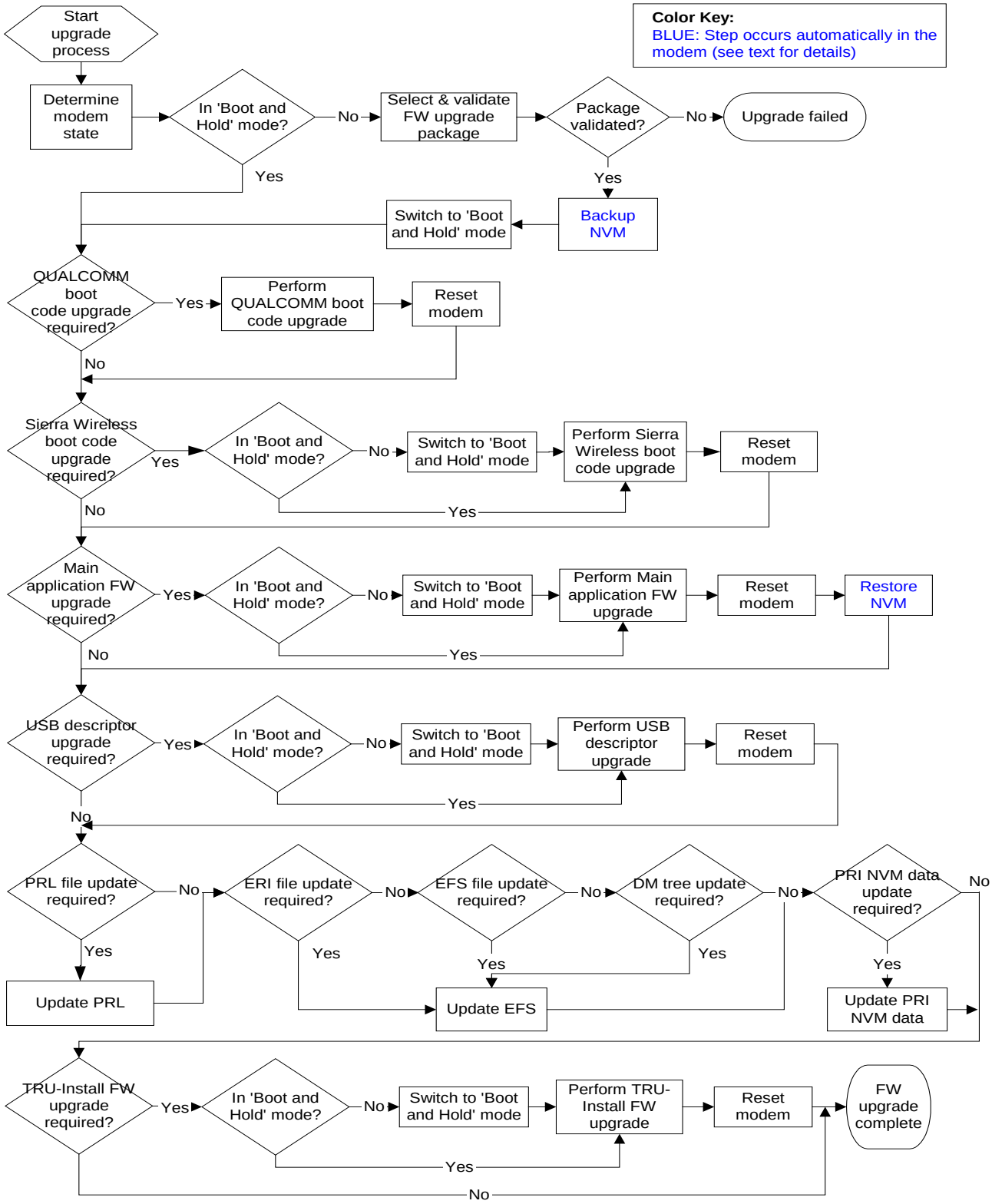
- Firmware upgrade configuration text file (page 26)
- Main application firmware (optional) (page 28)
- QUALCOMM boot code (optional) (page 29)
- Sierra Wireless boot code (optional) (page 29)
- USB descriptor (optional) (page 32)
- TRU-Install™ firmware (optional) (page 32)
- PRL data file (optional) (page 30)
- ERI data file (optional) (page 30)
- EFS data file(s) (optional) (page 31)
- Non-volatile Memory update data file (used for PRI NV update) (page 31)

“Firmware Upgrade Configuration File” (page 26) describes the format of this text file and how to use it to determine which optional components of the firmware upgrade package are included in the package.

Process flow of the firmware upgrade

Figure 1 on page 9 provides an overview of the process for upgrading the firmware of the modem.

Figure 1 Firmware upgrade process flow



Firmware upgrade steps

This section describes the steps that must be completed to properly upgrade the firmware of the modem.

Host preparation

This section describes steps that the host must perform before starting the firmware upgrade. Note that some of these steps are specific to certain software configurations.

Disable NDIS (Windows only)

If the firmware upgrade host application is running on Windows, NDIS must be disabled before starting the firmware upgrade process.

The Windows-based firmware upgrade applications from Sierra Wireless automatically disable NDIS before performing the firmware upgrade.

Disable MUX mode (hosts that use MUX approach only)

The modem's 'boot and hold' mode that is used to perform the firmware image download does not support MUX mode operation (i.e. 3GPP 27.010 multiplexing). If MUX mode is in use on the modem, you must disable MUX mode in the modem before starting the firmware image download. For details on MUX mode, see reference [1].

To disable MUX mode over USB in the modem, the USB driver on the host must:

1. Issue the vendor-specific USB command **Set Mode Non-MUX** (see reference [1]).
2. Tear down the 27.010 protocol stack running on the host.
3. Prepare for non-MUX mode communication with the modem.

The Windows-based firmware upgrade applications from Sierra Wireless automatically disable MUX mode before performing the firmware upgrade.

Disable USB Advanced Power Management (APM)

Sierra Wireless USB-based modems support remote wakeup from a USB suspend state when the modem or host has data to send across the USB bus. This allows the host driver to minimize the modem's power consumption, by putting the USB bus of the modem in a USB suspend state when the host driver determines there is no more data to be sent/received.

The 'boot and hold' mode of the modem does not support USB suspend/remote wakeup operation due to the limited capability of this mode. Given this, during the firmware upgrade process, the host MUST guarantee that it does not allow the USB bus of the modem to enter into a USB suspend state. If this occurs, the modem will not be able to issue a remote wakeup request or recover from any packet loss that could potentially occur during the operation. This is particularly risky, once the modem begins to process image download requests from the host and starts to erase and write memory segments into the memory of the modem.

Determine the modem state

The modem might already be in 'boot and hold' mode (due to a previous firmware image download attempt that failed, or a decision in the modem boot block to not switch to application mode). When the modem is already in 'boot and hold' mode, the firmware upgrade can still be attempted, but the steps to start the firmware upgrade process are different than in the normal case. Given the possibility that the modem may already be in 'boot and hold' mode, the firmware upgrade host application should first determine whether or not the modem is in 'boot and hold' mode before determining how to proceed with the firmware upgrade process.

To determine whether or not the modem is in 'boot and hold' mode, the firmware upgrade host application can issue the HIP **Loop Back Request** message (**0x03**) over the current HIP channel interface to the modem. When in 'boot and hold' mode, the modem is expected to respond with the **Loop Back Response message** (**0x43**) with an indication that the modem is in 'boot and hold' mode. If an indication of 'boot and hold' mode is received, the firmware upgrade host application should proceed to the firmware image upgrade step as described in "Perform the firmware image upgrade" on page 17.

For details of the HIP **Loop Back Request** and **Loop Back Response** messages, see the following section.

Loop Back Request

The Loop Back Request is sent from the host to the modem. The payload of this request is echoed back to the host in the payload of the **Loop Back Response**.

ID	0x03
Message Specific Parameter	None
Payload	Data to be looped back in response

The loopback request is stop-and-wait; after the host has sent a loop back request to the modem, the host should not send another loop back request until it has received a loop back response from the modem.

Loop Back Response

The Loop Back Response is sent from the modem to the host in response to a **Loop Back Request**.

ID	0x043
Message Specific Parameter	Firmware image that is sending the response: • 0x00 – response is from the boot loader image • 0x01 – response is from the application image
Payload	Payload of the **Loop Back Request**

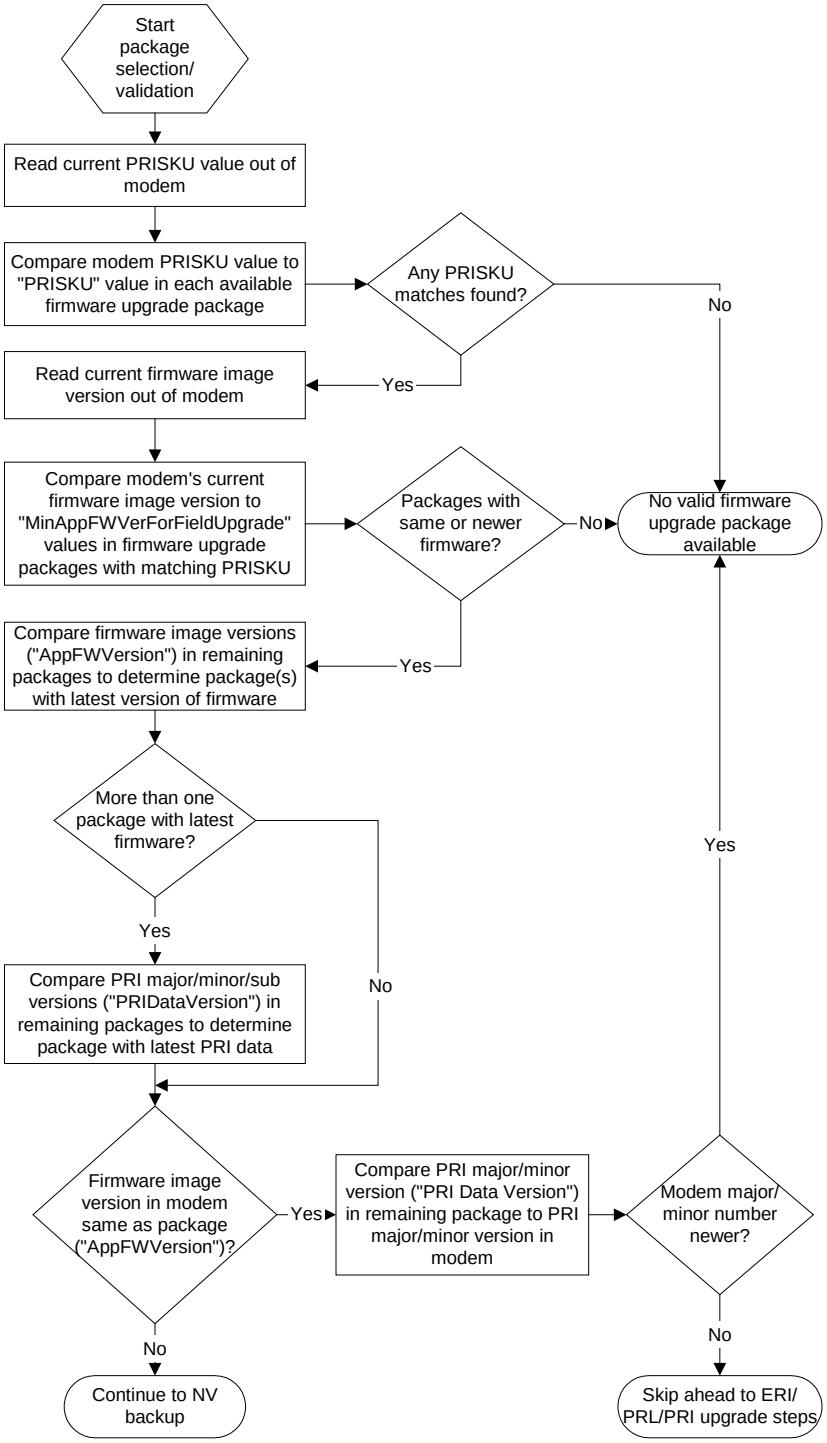
Select/validate the firmware upgrade package

When performing a firmware upgrade, the host may have one or more firmware upgrade packages to choose from. The packages available must first be checked to determine if they have the same PRISKU configuration as the current PRISKU configuration in the modem (e.g. product, carrier, features enabled/disabled, etc.) and whether they contain firmware and PRI data that is newer, the same, or older relative to the firmware and PRI data in the modem. These checks determine whether or not one or more of the packages can be used to perform the firmware upgrade, and which one should be used.

In addition, if a selected package includes the optional "Minimum main application firmware version for field upgrade (MinAppFWVerForFieldUpgrade)" field (page 28) in its Firmware Upgrade Configuration File and if the firmware upgrade process is being performed in a Field Upgrade environment, the main application firmware image currently in the modem is checked to make sure the package can be used for the firmware upgrade process. If the version of the main application firmware image currently in the modem is less than the minimum version required for a field upgrade, the package cannot be used to perform a firmware upgrade in a field environment. This check is a failsafe mechanism to prevent firmware Field Upgrade attempts with packages that are incompatible with the modem.

Figure 2 on page 14 shows the recommended process for determining if there is a valid firmware upgrade package for the modem and, if there is more than one valid package, which one to use.

Figure 2 Select/validate the firmware upgrade package



Some important things to note from this process are as follows:

- The PRISKU is the SKU number of PRI data for a particular PRI configuration (e.g. carrier, features enabled/disabled, etc.). Firmware upgrades should never be performed in the field from one PRISKU to another, because the set of PRI configuration data and carrier approved versions of firmware is different between different PRISKUs. If the firmware of a modem needs to be changed to another PRISKU, a factory-level firmware upgrade must be performed, which is beyond the scope of this document.
- If the firmware upgrade package has the same firmware image version (AppFWVersion) as the modem, the NV backup/restore and firmware image download steps can be skipped, but the PRL file upgrade check, ERI file upgrade check, and PRI update steps must still be performed, because a new firmware upgrade package may have been generated and released to upgrade only the PRL file, ERI file, and/or PRI data values in the modem.
- Only the major/minor portion of the PRI data version number is stored in the modem. The sub portion of the PRI data version number is stored only in the firmware upgrade package to distinguish new versions of the firmware upgrade package that did NOT require changes to the current PRI table that was used to generate the PRI data. This can occur if the package generation process, which uses the PRI table to generate the PRI data, was updated (for example, a new file was added) and a new package was generated without making a change to the PRI table.

To read the PRISKU out of the modem:

- The CnS **Get/Set NV Item** message (**0x1033**, with parameter 0x603E) can be sent on the HIP channel interface, or
- The **AT!SKU** command can be sent on the AT command channel interface of the modem.

To read the PRI major/minor version values out of the modem:

- The CnS **Get/Set NV Item** message (**0x1033**, with parameter 0x6024) can be sent on the HIP channel interface, or
- The **AT!PRIREV** command can be sent on the AT command channel interface of the modem.

To read the firmware image version out of the modem, the CnS **Firmware Version** message (**0x0001**) can be sent on the HIP channel interface of the modem.

For details of the CnS **Firmware Version** and **Get/Set NV Item** messages, see reference [2].

Back up Non-Volatile Memory (NVM)

After a firmware image download to the modem, some or all of the Non-Volatile Memory (NVM) of the modem is rebuilt if the new firmware download image expects different NVM table revision numbers than the ones previously stored in NVM. This typically happens whenever a new firmware revision adds a new item to NVM that did not exist in the previous firmware revision that was running on the modem. To prevent loss of NVM data when a NVM rebuild occurs, the NVM data must be backed up before the firmware download process commences and then restored after it has completed (see “Restore Non-Volatile Memory (NVM)” on page 22).

The NVM backup occurs automatically in the modem; you do not need to send explicit commands to perform the NVM backup step of the firmware upgrade process.

Switch to ‘boot and hold’ mode

To begin the download of a firmware image, the modem must first be placed in ‘boot and hold’ mode; in this mode, the boot loader can process firmware image segment download requests.

There are two mechanisms for requesting the modem to enter ‘boot and hold’ mode:

- Through the AT command interface by issuing the **AT!BOOTHOLD** command.
- Through the HIP channel interface, by sending the CnS **Reset Modem** request message (**0x100F**). There is no corresponding response message to this CnS request.

For details of the CnS **Reset Modem** message, see reference [2].

If the firmware accepts the request to switch to ‘boot and hold’ mode, the modem will first reset and then hold in ‘boot and hold’ mode to prepare for a firmware image download. It is important to note that the modem reset operation causes the device to re-enumerate on the USB bus of the host during the switch to ‘boot and hold’ mode; the firmware upgrade host application must handle this. Once in ‘boot and hold’ mode, the modem communicates only over a single set of endpoints that are configured to provide the HIP service. Firmware download operations are then performed using HIP until the new firmware image is completely downloaded to the modem and executed.

The modem supports the HIP service in ‘boot and hold’ mode on endpoint set 2IN/OUT, endpoint set 4IN/4OUT, or the serial UART interface. After entering ‘boot and hold’ mode, the modem configures the HIP service to the first set of endpoints or serial UART interface that receives a valid HIP request until the modem is reset.

For USB-based hosts that do not use the 27.010 MUX mode capability of the modem, 4IN/4OUT is typically the set of endpoints used by the firmware upgrade host application, since 4IN/OUT is also typically used for HIP service communication during normal modem operation.

For details on the USB endpoint set to service mapping configurations, see reference [1].

To verify the modem is currently in ‘boot and hold’ mode, the firmware upgrade host application can send the HIP **Loop Back Request** message to the modem as is described in “Determine the modem state” on page 11.

Perform the firmware image upgrade

The QUALCOMM boot code, Sierra Wireless boot code, main application firmware, USB descriptor, and TRU-Install firmware images are downloaded to the modem as separate and independent firmware image download steps, as shown in Figure 1 on page 9. The processes for upgrading each of the firmware images are very similar, except for the minor differences described in this document.

The firmware images must be downloaded **in the following order**. Note that not all of the images listed might be downloaded, during a particular firmware upgrade.

1. QUALCOMM boot code
2. Sierra Wireless boot code
3. Main application firmware
4. USB descriptor
5. TRU-Install firmware

After a QUALCOMM boot code or Sierra Wireless boot code image upgrade is performed, if the image is not compatible with the main application firmware image in the modem, then the modem remains in ‘boot and hold’ mode until a compatible main application firmware image is downloaded.

Start the firmware image download

Once the modem is in 'boot and hold' mode, the modem is ready to receive a start firmware download request from the firmware upgrade host application. To start the download of a firmware image, the firmware upgrade host application must send the HIP **Download Start Request** message (**0x00**) over the HIP service along with an initial segment of at least 512 bytes (to include all of the required header information) and up to a maximum of 2000 bytes from the beginning of the firmware download image file.

When this request is received, the modem first validates the header format received in the first 512 bytes of the segment (i.e. correct file format, type, version, etc.). If the modem accepts the header that was received, the modem then proceeds with erasing the current firmware image on the modem. Erasing the firmware takes some time, so the firmware upgrade host application should not expect an immediate response. Once the firmware is erased, the modem returns the HIP **Download Start Response** (**0x40**) to the host with a success or failure indication. At this point, for the modem to be usable again, a successful firmware update process must be completed.

For each segment of the firmware download image file sent to the modem, the firmware upgrade host application should keep a running 32-bit checksum so that, at the end of the process, the checksum can be sent to the modem to be compared and validated with the modem's own internally calculated checksum. For the download operation to be successfully validated, these two running checksums must be equal.

For details of the HIP **Download Start Request** and **Download Start Response** messages, see the following sections.

Download Start Request

ID	0x00
Message Specific Parameter	None
Payload	Image header, followed by data from the firmware download image file. Maximum length of payload field: 2000 bytes

Download Start Response

The Download Start Response is sent in reply to the **Download Start Request**.

ID	0x40
Message Specific Parameter	Return code: • 0x00 – Success; continue the download process • 0x01 – Error; download process aborted
Payload	Optional; NULL-terminated ASCII error string that describes the error reason. For example: “invalid product ID”; “invalid load address”.

Continue the firmware image download

After the modem has sent the HIP **Download Start Response** with a successful indication, the modem is ready to start receiving subsequent segments in sequential order from the firmware download image file, starting with the first sequential byte of the image file that was not sent in the HIP **Download Start Request**. Each subsequent segment should be sent with the HIP **Download Continuation Request** (**0x02**) with a segment of up to 2000 bytes.

After each HIP **Download Continuation Request** is sent, the firmware upgrade host application should wait for the **Download Continuation Response** (**0x42**). As long as each response from the modem includes a successful indication, the firmware upgrade host application should continue to send additional HIP **Download Continuation Request** messages until the complete firmware download image file has been sent to the modem.

For each segment, the firmware upgrade host application should continue to keep a running 32-bit checksum so that, at the end of the process, the checksum can be sent to the modem to be compared against the modem’s internally calculated checksum.

For details of the HIP **Download Continuation Request** and **Download Continuation Response** messages, see the following sections.

Download Continuation Request

ID	0x02
Message Specific Parameter	None
Payload	Portion of the firmware download image file. Maximum length of payload field: 2000 bytes

Download Continuation Response

ID	0x42
Message Specific Parameter	Return code: • 0x00 – Success; the portion of the firmware download image file was downloaded successfully • 0x01 – Failure
Payload	N/a

Finish the firmware image download

After the modem has successfully received all of the segments of firmware download image file, the modem should be notified that the download of the image is complete by sending the HIP **Download End Request** (**0x01**) along with the checksum of the image that was calculated by the firmware upgrade host application. The modem then validates the image that was downloaded, including a comparison of the checksum received against its own internally calculated checksum. The result of the validation is returned in the HIP **Download End Response** (**0x41**) with an indication of success or failure.

For details of the HIP **Download End Request** and **Download End Response** messages, see the following sections.

Download End Request

ID	0x01
Message Specific Parameter	None
Payload	16-bit field containing the checksum for the entire download process.

Download End Response

ID	0x41
Message Specific Parameter	Return code: • 0x00 – Success; the firmware download image file was successfully downloaded • 0x01 – Failure; the download process was aborted.
Payload	Optional; NULL-terminated ASCII error string that describes the error reason.

Reset the modem

After a new QUALCOMM boot code, Sierra Wireless boot code, or main application firmware image has been downloaded and validated by the modem, the firmware upgrade host application should request the modem to try and use the new firmware image to boot up.

If a main application firmware image is downloaded, the firmware upgrade host application should send the HIP **Flash Program Request** (**0x04**). When this request is received, the modem checks that there are no problems with the new main application firmware image and sends the result in the HIP **Flash Program Response** (**0x44**). If a QUALCOMM boot code or Sierra Wireless boot code image is downloaded, the firmware upgrade host application should send the HIP **Launch Fragment Request** (**0x21**).

If the request received by the modem is successful, the modem then resets, triggering a re-enumeration of the USB bus of USB-based hosts, and attempts to boot up using the new firmware image.

The HIP **Launch Fragment Request** message is a single-byte HIP message with a command ID of **0x21**. There is no response to the **Launch Fragment Request**.

For details of the HIP **Flash Program Request**, **Flash Program Response**, and **Launch Fragment Request** messages, see the following sections.

Flash Program Request

ID	0x04
Message Specific Parameter	None
Payload	None

Flash Program Response

ID	0x44
Message Specific Parameter	Return code: • 0x00 – Success; the program was successfully copied to flash memory • 0x01 – Failure; flash programming failed.
Payload	Optional; NULL-terminated ASCII error string that describes the error reason.

Launch Fragment Request

ID	0x21
Message Specific Parameter	Launch code: • 0x00 – Indicates normal launch of the firmware image.
Payload	None

Restore Non-Volatile Memory (NVM)

If some or all of the NVM of the modem was rebuilt as a result of using the new firmware image, the NVM must be restored from the backup repository (see “Back up Non-Volatile Memory (NVM)” on page 16). The modem does this automatically, if necessary, after it boots up the first time with a new firmware image, so the firmware upgrade host application does not need to send a request. However, the restoration of NVM causes a delay in the time required for the modem to boot up, and the firmware upgrade host application must expect and properly handle this.

Update PRL

A Preferred Roaming List (PRL) is stored in a CDMA modem to control the system selection algorithm and behavior of the device. Carriers provide the PRL file to be stored in modems intended to operate on their network. Carriers periodically provide new revisions of the PRL file and sometimes require that the new PRL file be programmed into the modem during a firmware upgrade. This would be required if:

- The carrier does not have the necessary technology to push this new PRL file to the device over-the-air, or
- The PRL change is critical enough to warrant including it as part of a firmware upgrade.

To determine if a PRL file will be written into the modem during the firmware upgrade process, check the PRLUpdate flag (page 30) in the Firmware Upgrade Configuration File of the firmware upgrade package. If this record does not exist or it is not set to TRUE, the PRL file should not be written.

If a PRL file upgrade is required:

- The PRL file to write into the modem is specified in the PRLFile (page 30) of the Firmware Upgrade Configuration File and is included as part of the firmware upgrade package. The exact filename specified in the Firmware Upgrade Configuration File must be used when writing the PRL file into the modem.
- The modem must first be unlocked using the CnS **Interface Unlock** (0x0012) messages on the HIP command interface. After the modem is unlocked, the PRL file can be written into the modem using the CnS **Get/Set PRL Size** (0x1052) and **PRL Write** (0x1054) messages.
- Once the new PRL file is written into the modem, the modem must be reset for the new PRL file to be used. The modem can be reset via the CnS **Reset Device** (0x0010) command over the HIP channel interface.

For details of the CnS **Interface Unlock**, **Get/Set PRL Size**, **PRL Write**, and **Reset Device** messages, see reference [2].

Update ERI

An Enhanced Roaming Indicator (ERI) file is stored in a CDMA modem to extend the roaming indication behavior in the UI beyond what the PRL supports. The usage of an ERI file is optional, and its use is typically determined by the carrier. Carriers that require and provide the ERI file will periodically provide new revisions and require that the new ERI file be programmed into the modem during a firmware upgrade, since currently there is no over-the-air method to push a new ERI file to the modem.

To determine if a ERI file should be written into the modem during the firmware upgrade process, check the ERIUpdate flag (page 30) of the Firmware Upgrade Configuration File. If this record does not exist or is not set to TRUE, the ERI file should not be written.

If a ERI file upgrade is required, the ERI file to write into the modem is specified by ERIFile (page 30) of the Firmware Upgrade Configuration File and is included as part of the firmware upgrade package. The exact filename specified in the Firmware Upgrade Configuration File must be used when writing the ERI file into the modem. Note that the ERI filename *must always* be set to **eri_nam1**. The modem and relevant host applications on the market today use this filename to access the ERI file and thus must be maintained consistent for all CDMA products.

If the ERI file needs to be written into the modem as part of the firmware upgrade process, the modem must first be unlocked using the CnS **Interface Unlock** (0x0012) messages on the HIP command interface. After the modem is unlocked, the ERI file can be written into the modem using the CnS **ERI File Name** (0x104A, “Set” operation) and **ERI Write** (0x104C) messages.

Once the new ERI file is written into the modem, the modem must be reset for the new ERI file to be used. The modem can be reset via the CnS **Reset Device** (0x0010) request over the HIP channel interface.

For details of the CnS **Interface Unlock**, **ERI File Name**, **ERI Write**, and **Reset Device** messages, see reference [2].

Update PRI NV data

The firmware has many configurable options, the values of which are defined by Sierra Wireless, the carrier, and the OEM. The PRI NV data in the modem controls these configurable settings in the modem. Occasionally a value may need to be updated for products in the field; this can be done as part of a firmware upgrade process.

To determine if a PRI NV update is required during the firmware upgrade process, check the NVUpdate flag (page 31) in the Firmware Upgrade Configuration File of the firmware upgrade package. If this record does not exist or is not set to TRUE, the PRI NV update should not be performed.

Before performing a NV update, you must first unlock the modem, using the CnS **Interface Unlock** (0x0012) messages on the HIP command interface.

To perform a NV update, the NV update data file should be written into the modem using the CnS **ERI File Name** and **ERI Write** messages, followed by a CnS **NVRAM Update from File** (0x0011) request.

If a PRI NV update is required, the NV update data file to write into the modem is specified by NVUpdateFile (page 31) of the Firmware Upgrade Configuration File and is included as part of the firmware upgrade package. The exact filename specified in the Firmware Upgrade Configuration File must be used when writing the NV update data file into the modem.

For details of the CnS **Interface Unlock**, **ERI File Name**, **ERI Write**, **NVRAM Update from File**, and **Reset Device** messages, see reference [2].

Firmware Upgrade Configuration File

The configuration file is a text file containing key/value records to describe the contents of the package. The key/value records may appear in any order in the file. Each key/value record ends with a line-feed character.

You can edit this file to, for example, change the download settings.

A sample file is shown on page 33.

Key/value descriptions

Heading

The heading appears before all other records. The heading always remains the same in all packages.

Example

[FWUpgradeConfigFile]

Version

The version record indicates the version of the configuration file syntax. The following table summarizes the changes between versions.

Version	Summary of changes
1.0	Initial release
1.1	Added support for QUALCOMM boot code update flag (QCBootFWUpdate) (page 29).
1.2	Added: • Minimum main application firmware version for field upgrade (MinAppFWVerForFieldUpgrade) (page 28) • USB descriptor update flag (SWIFWUpdate) (page 32) • TRU-Install firmware update flag (SWoCFWUpdate) (page 32)
1.4	Added: • EFS Update flag (page 31) • EFS Files flag (page 31) • OMA-DM Tree Update flag (page 33)

Example

Version = 1.1

PRISKU

The PRISKU value is used to distinguish provisioning packages for different products, carriers, feature sets, etc.

The PRISKU value is an integer value in decimal format.

For more information on PRISKU, see page 15.

Example

PRISKU = 11110

PRI Data Version

This is the version of PRI data associated with this firmware upgrade package.

If this firmware upgrade package includes a NV update data file (see "NV Update File" on page 31), this value is set to the version of PRI data to use to update the modem. If this firmware upgrade package does not include a NV update data file, this value is set to the version of PRI data associated with the other components included with this package.

The PRI data version is expressed numerically as major, minor, and sub version numbers, separated by periods.

Example

PRIDataVersion = 000.001.000

Main application firmware version (AppFWVersion)

This is the version of main application firmware associated with this firmware upgrade package.

If this firmware upgrade package includes a main application firmware download image, this value is set to the version of the main application firmware image to download. If this firmware upgrade package does not include a main application firmware image to download, this value is set to the version of the main application firmware associated with the other components included with this package.

The main application firmware image version is expressed numerically as major, minor, and point-release version numbers, separated by periods

Example

AppFWVersion = 0.32.0

Main application firmware update flag (AppFWUpdate)

This flag indicates whether the package includes main application firmware to be updated.

“TRUE” indicates that the main application firmware is to be updated as part of the firmware upgrade process. “FALSE” indicates that the main application firmware will not be updated.

Example

AppFWUpdate = TRUE

Main application firmware (AppFWFile)

The configuration file may contain this record that indicates the filename of the main application firmware. If the Main application firmware update flag (AppFWUpdate) (above) is FALSE, the main application firmware record is ignored.

Example

AppFWFile = appl.cwe

Minimum main application firmware version for field upgrade (MinAppFWVerForFieldUpgrade)

The configuration file may contain this record that indicates the minimum version of the main application firmware of the modem that is required to allow a firmware upgrade in a field upgrade environment using this firmware upgrade package.

This field provides a failsafe mechanism to prevent field upgrade attempts from older versions of firmware that are not compatible with this package. If this firmware upgrade package does not include this field, there are no upgrade restrictions with this package in relation to the modem’s main application firmware version.

The minimum main application firmware version for field upgrade is expressed numerically as major, minor, and point-release version numbers, separated by periods.

Example

MinAppFWVerForFieldUpgrade = 0.30.0

Note: This field was added in version 1.2 of the configuration file syntax.

QUALCOMM boot code update flag (QCBootFWUpdate)

This flag indicates whether the package includes QUALCOMM boot code to be updated.

“TRUE” indicates that the QUALCOMM boot code is to be updated as part of the firmware upgrade process. “FALSE” indicates that the QUALCOMM boot code will not be updated.

Note: This field was added in version 1.1 of the configuration file syntax.

Example

QCBootFWUpdate = TRUE

QUALCOMM boot code (QCBootFWFile)

The configuration file may contain this record that indicates the filename of the QUALCOMM boot code. If the QUALCOMM boot code update flag (QCBootFWUpdate) (above) is FALSE, the QUALCOMM boot code record is ignored.

Note: This field was added in version 1.1 of the configuration file syntax.

Example

QCBootFWFile = qcom.cwe

Sierra Wireless boot code update flag (BootFWUpdate)

This flag indicates whether the package includes Sierra Wireless boot code to be updated.

“TRUE” indicates that the Sierra Wireless boot code is to be updated as part of the firmware upgrade process. “FALSE” indicates that the Sierra Wireless boot code will not be updated.

Example

BootFWUpdate = TRUE

Sierra Wireless boot code file (BootFWFile)

The configuration file may contain this record that indicates the filename of the Sierra Wireless boot code. If the Sierra Wireless boot code update flag (BootFWUpdate) (above) is FALSE, the Sierra Wireless boot code file record is ignored.

Example

BootFWFile = boot.cwe

PRL Update flag

This flag indicates whether the package includes a PRL file to be written during the firmware upgrade process.

“TRUE” indicates that the PRL is to be written as part of the firmware upgrade process. “FALSE” indicates that the PRL will not be updated.

Example

PRLUpdate = TRUE

PRL File

The configuration file may contain this record that indicates the name of the PRL file. If the PRL Update flag (above) is FALSE, the PRL file record is ignored.

Example

PRLFile = 12345.prl

ERI Update flag

This flag indicates whether the package includes an ERI file to be written during the firmware upgrade process.

“TRUE” indicates that the ERI is to be written as part of the firmware upgrade process. “FALSE” indicates that the ERI will not be updated.

Example

ERIUpdate = TRUE

ERI File

The configuration file may contain this record that indicates the name of the ERI file. If the ERI Update flag (above) is FALSE, the ERI file record is ignored.

Example

ERIFile = eri_nam1

EFS Update flag

The configuration file may contain this flag that indicates whether EFS file(s) will be written to the modem during the firmware upgrade process.

“TRUE” indicates that EFS file(s) will be written. “FALSE” indicates that EFS file(s) will not be written.

Example

EFSUpdate = TRUE

EFS Files

The configuration file may contain this record that indicates the EFS filename(s) to be downloaded to the modem. If the EFS Update flag (above) is FALSE, the EFS file record is ignored.

Example

EFSFiles = file1, file2, …

NV Update flag

This flag indicates whether the package includes NV (non-volatile memory) items to be written during the firmware upgrade process.

“TRUE” indicates that NV items are to be updated as part of the firmware upgrade process. “FALSE” indicates that NV items will not be updated.

Example

NVUpdate = TRUE

NV Update File

The configuration file may contain this record that indicates the filename of the NV (non-volatile memory) update data file. If the NV Update flag (above) is FALSE, the NV update file record is ignored.

Example

NVUpdateFile = nvup

USB descriptor update flag (SWIFWUpdate)

This flag indicates whether the package includes a USB descriptor to be updated.

“TRUE” indicates that the USB descriptor is to be updated as part of the firmware upgrade process. “FALSE” indicates that the USB descriptor will not be updated.

Note: This field was added in version 1.2 of the configuration file syntax.

Example

SWIFWUpdate = TRUE

USB descriptor (SWIFWFile)

The configuration file may contain this record that indicates the filename of the USB descriptor. If the USB descriptor update flag (SWIFWUpdate) (above) is “FALSE”, the USB descriptor record is ignored.

Note: This field was added in version 1.2 of the configuration file syntax.

Example

SWIFWFile = swi_ud_generic_00.cwe

TRU-Install firmware update flag (SWoCFWUpdate)

This flag indicates whether the package includes TRU-Install firmware to be updated.

“TRUE” indicates that the TRU-Install firmware is to be updated as part of the firmware upgrade process. “FALSE” indicates that the TRU-Install firmware will not be updated.

Note: This field was added in version 1.2 of the configuration file syntax.

Example

SWoCFWUpdate = TRUE

TRU-Install firmware file (SWoCFWFile)

The configuration file may contain a TRU-Install firmware record that indicates the filename of the TRU-Install firmware. If the TRU-Install firmware update flag (SWoCFWUpdate) (above) is “FALSE”, the TRU-Install firmware record is ignored.

Note: This field was added in version 1.2 of the configuration file syntax.

Example

SWoCFWFile = swoc.cwe

OMA-DM Tree Update flag

The configuration file may contain an OMA-DM tree update record. OMA-DM tree file name ("dmtree") MUST be specified as part of EFSFiles record.

"TRUE" indicates that the OMA-DM tree is to be updated as part of the firmware upgrade process. "FALSE" indicates that the OMA-DM tree will not be updated and the record will be ignored.

Caution: Unless absolutely needed to update OMA-DM tree in the field, this flag should ALWAYS be set to FALSE or omitted from the record. Otherwise, the current OMA-DM data will be overridden, and modem will become unusable OTA (over the air).

Example

…

EFSFiles = dmtree

…

OMADMTreeUpdate = TRUE

Sample configuration file

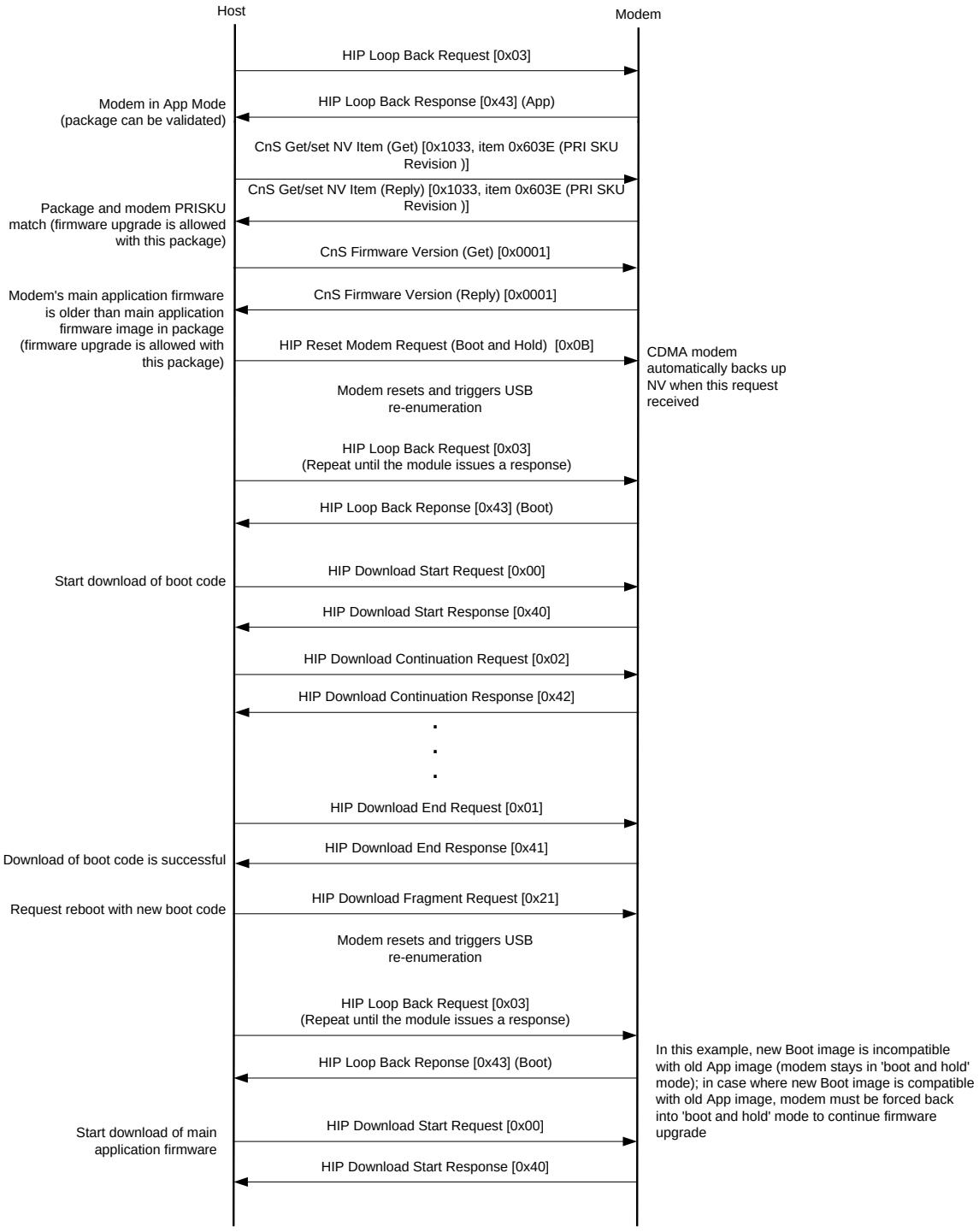
```
[FWUpgradeConfigFile]
Version = 1.2
PRISKU = 11110
PRIDataVersion = 000.001.000
AppFWVersion = 0.32.0
AppFWUpdate = TRUE
AppFWFile = appl.cwe
MinAppFWVerForFieldUpgrade = 0.30.0
QCBootFWUpdate = TRUE
QCBootFWFile = qcom.cwe
BootFWUpdate = TRUE
BootFWFile = boot.cwe
PRLUpdate = TRUE
PRLFile = 12345.prl
ERIUpdate = TRUE
ERIFile = eri_nam1
EFSUpdate = TRUE
EFSFiles = certificate, dmtree
NVUpdate = TRUE
```

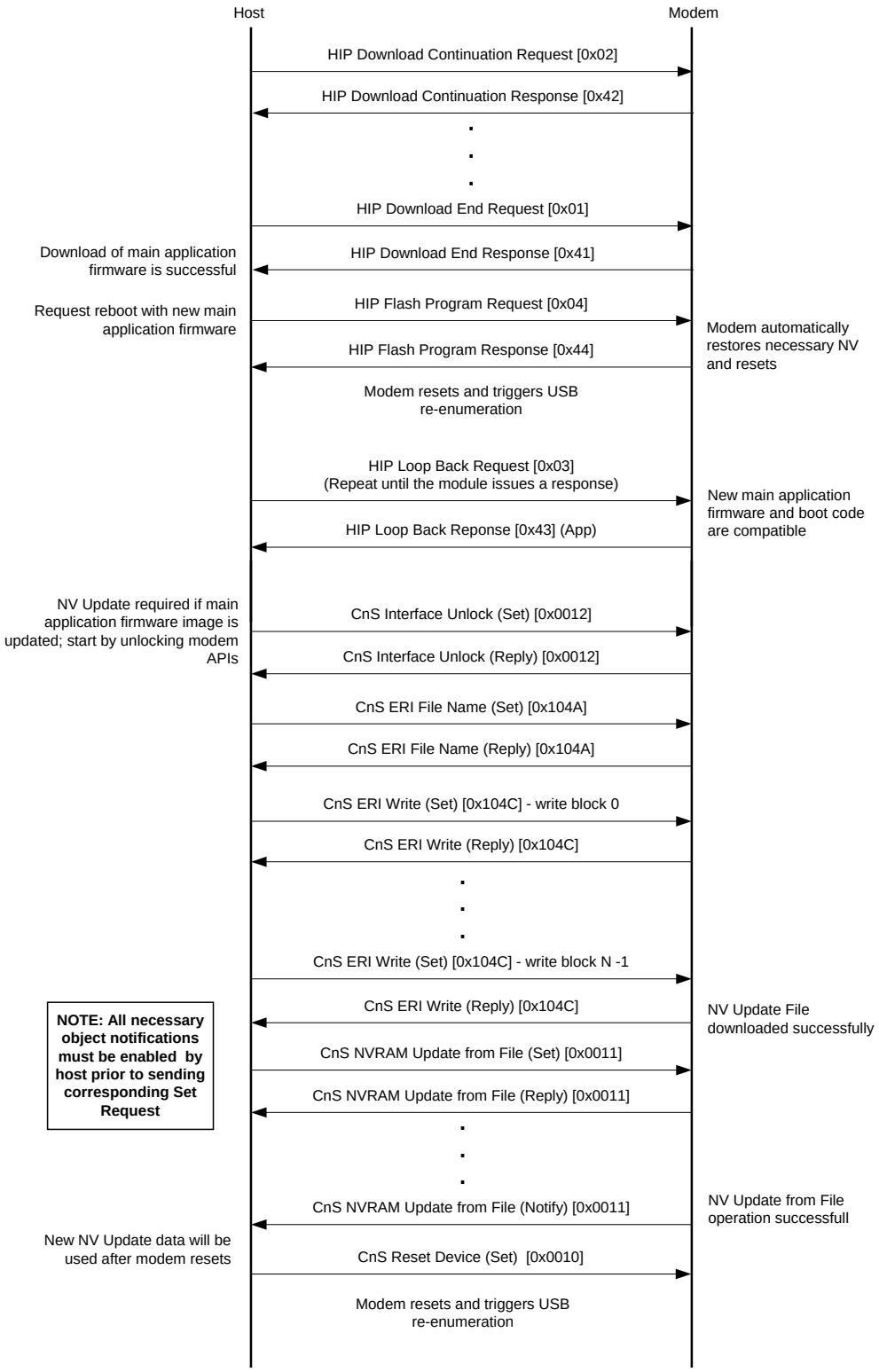
```
NVUpdateFile = nvup
SWIFWUpdate = TRUE
SWIFWFile = swi_ud_generic_00.cwe
SWoCFWUpdate = TRUE
SWoCFWFile = swoc.cwe
OMADMTreeUpdate = TRUE
```

Sample message flow during firmware upgrade

The message flow (shown in the following figures) illustrates a successful firmware upgrade for a CDMA device that requires a main application firmware update, Sierra Wireless boot code update, and an NV update. This is the typical scenario for a CDMA device firmware upgrade. The only major difference is that theSierra Wireless boot code update may not be required or may be replaced with a QUALCOMM boot code update. It is not typical that both a QUALCOMM boot code update and a Sierra Wireless boot code update are required during the same firmware upgrade process.

Figure 3 Message flow during firmware upgrade




Host
Modem
HIP Download Continuation Request [0x02]
HIP Download Continuation Response [0x42]
HIP Download End Request [0x01]
Download of main application firmware is successful
HIP Download End Response [0x41]
Request reboot with new main application firmware
HIP Flash Program Request [0x04]
Modem automatically restores necessary NV and resets
HIP Flash Program Response [0x44]
Modem resets and triggers USB re-enumeration
HIP Loop Back Request [0x03]
(Repeat until the module issues a response)
New main application firmware and boot code are compatible
HIP Loop Back Reponse [0x43] (App)
NV Update required if main application firmware image is updated; start by unlocking modem APIs
CnS Interface Unlock (Set) [0x0012]
CnS Interface Unlock (Reply) [0x0012]
CnS ERI File Name (Set) [0x104A]
CnS ERI File Name (Reply) [0x104A]
CnS ERI Write (Set) [0x104C] - write block 0
CnS ERI Write (Reply) [0x104C]
CnS ERI Write (Set) [0x104C] - write block N -1
CnS ERI Write (Reply) [0x104C]
NV Update File downloaded successfully
NOTE: All necessary object notifications must be enabled by host prior to sending corresponding Set Request
CnS NVRAM Update from File (Set) [0x0011]
CnS NVRAM Update from File (Reply) [0x0011]
CnS NVRAM Update from File (Notify) [0x0011]
NV Update from File operation successfull
New NV Update data will be used after modem resets
CnS Reset Device (Set) [0x0010]
Modem resets and triggers USB re-enumeration

Glossary

Term	Description
APM	Advanced Power Management
CnS	Control and Status (language)—a proprietary protocol for managing the control and status of the modem. CnS is described in detail in the CnS Reference document from Sierra Wireless (document 2130754).
DM	Device Management. See also OMA-DM.
EFS	Encrypted File System
ERI	CDMA/EVDO Enhanced Roaming Indicator file
firmware	Software stored in the modem's ROM or EEPROM; essential programs that remain even when the system is turned off. Firmware is easier to change than hardware but more permanent than software stored on disk.
FW	Firmware. See above.
HIP	Sierra Wireless' Host Interface Protocol—a proprietary protocol for managing the control and status of the modem. Intended to carry a variety of other protocol packets across a single link layer. CnS is one of the types of packets that can be carried inside HIP.
NDIS	Network Driver Interface Specification—a programming interface specification for connecting network interface cards in Windows.
NVM	Non-Volatile Memory—Random Access Memory that retains its contents even if the power is removed.
OMA-DM	Open Mobile Alliance - Device Management. A device management (DM) protocol specified by the Open Mobile Alliance (OMA) Device Management Working Group and the Data Synchronization (DS) Working Group.
PRI	Product Release Instructions—a file that contains the settings used to configure modems for a particular service provider, customer, or purpose.
PRI data	NVM items stored in the modem to hold the values of the PRI

Term	Description
PRISKU	The SKU of PRI data for a particular PRI configuration. Each PRISKU represents a unique combination of product (for example, MC5728V) carrier configuration (for example, Sprint), and feature set options (for example, voice and data support vs. data-only). Each CDMA modem is associated with a specific PRISKU value.
PRL	CDMA/EVDO Preferred Roaming List file. Tells the modem what network carrier(s) to look for; this list determines which CDMA networks a modem can access.
Roaming	A cellular subscriber is in an area where service is obtained from a cellular service provider that is not the subscriber's provider. This may be subject to roaming charges.
SWoC	Software-on-Card. Also known as TRU-Install; a Sierra Wireless feature that installs the necessary software and drivers the first time you insert the modem into a Windows or Mac computer. An installation CD is not required.