Image Switching

M2M Developer's Guide

SIERRA WIRELESS®

4118344
Rev 2

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. The Sierra Wireless modem can transmit signals that could interfere with this equipment.

Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

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 and MMP Portfolio Licensing.

Copyright

Trademarks

Sierra Wireless®, AirPrime®, AirLink®, AirVantage® and the Sierra Wireless logo are registered trademarks of Sierra Wireless, Inc.

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

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.

Contact Information

Sales information and technical support, including warranty and returns	Web: sierrawireless.com/company/contact-us/ Global toll-free number: 1-877-687-7795 6:00 am to 6:00 pm PST
Corporate and product information	Web: sierrawireless.com

Revision History

Revision number	Release date	Changes
1	November 2015	Initial release.
2	October 2016	Added WP75XX modules, removed WP8548-G. Replaced SIM-based Image Selection (AUTO-SIM) on page 15 details with reference to Firmware Image Switching Developer's Guide.

Contents

1: Image Switching

Introduction

The Sierra Wireless AirPrime modules described in Table 1-1 support an 'image switching' feature that allows them to switch between carrier-approved firmware configurations when necessary.

Table 1-1: AirPrime Modules Supporting Image Switching[a]

Module series	Module types
MC73XX	MC7304/MC7305/MC7330/MC7354/MC7354B/MC7355
MC74XX	MC7430/MC7455
EM73XX	EM7330/EM7355
EM74XX	EM7430/EM7455
WP75XX	WP7502/WP7504
WP8458	WP8548

a. Subject to change, list is current as of publication date. Module-specific details may vary for newer firmware versions and module types.

Images consist of pairs of files that are used to configure the module for use with specific carriers. The module is capable of storing a number of these files on the device. The files that make a pair are:

- Firmware—These files are stored in a compressed format on the module.
  File type: .cwe
- Carrier configuration ('PRI'—Product Release Information)—Each of these files contains custom settings for a specific carrier, and is linked internally to a specific firmware file.
  File type: .nvu

When necessary, updated firmware and/or carrier PRI files are released by Sierra Wireless to be loaded on the module.

Note: Occasionally, OEM PRI updates will be provided to update a module's settings. These files also use the ".nvu" extension but are not to be confused with carrier PRIs.

At any time, the module is running an 'active image'—a combination of a single uncompressed copy of a firmware file and a single carrier configuration file.

This document describes how to switch the active image used on the module, as well as how image files are managed.

For additional image switching details, refer to Firmware Image Switching Developer's Guide (Document #4115290).

Customer Use Cases

The methods used to manage the image switching feature depend on customer use cases—a combination of the host device type and module usage requirements. Table 1-2 on page 7 describes typical customer use cases.

Table 1-2: Sample Customer Use Cases

	Host Device Type	
	M2M General	'Headless' M2M
Host device examples	Router, Gateway	Smart meter
Typical Image management methods	Linux SDK AT commands	SIM-based image selection Remote configuration via AVMS
Host storage space for image files	Limited storage space. Image files typically stored only until downloaded to module, then deleted.	None. All image files (firmware and carrier PRI) must be stored on the module.

Image Switch Timing

The time required to switch between an active image and a preferred image depends primarily on whether the active firmware file has to be changed and whether the image file(s) have to be downloaded.

Table 1-3 on page 7 describes the typical time required to switch an MC74XX module from its active image to a different (preferred) image, for several scenarios. At a minimum, the image switch process time includes:

- Host initiating the image switch and rebooting the module (~30 s)
- Module bootup to the point where USB interfaces are communicating (~20 s).

Table 1-3: Typical Image Switch Times

Scenario	Files downloaded from host device	Switch time[a] (MC74XX)
Preferred firmware is active; preferred carrier PRI is on the module but not active	None	55 s
Preferred firmware is active; preferred carrier PRI is not on the module	Carrier PRI (.nvu)	60 s
Preferred firmware is on the module but not active	None	120 s
Preferred image files (firmware and carrier PRI) are not on the module.	Firmware (.cwe) Carrier PRI (.nvu)	130 s

a. Actual time required is affected by factors including whether active firmware is changing, image files are downloading from the host, etc.

Image File Storage

Image Storage—Host-side

The host device can store additional image files (firmware and carrier PRI files) for download to the modem when needed. The number of files that can be stored on the host is limited only by the drive space (hard drive, flash drive) available.

Each carrier's firmware images (including a default OEM image) should be stored in subdirectories of a user-defined storage directory. A typical location for the storage directory is: /home/<username>/SDK

Table 1-4: Carrier image subdirectories[a]

Subdirectory	Description
./0	Vodafone
./1	Verizon
./2	AT&T
./3	Sprint
./4	T-Mobile
./6	Generic
./7	Telefonica
./8	Telecom Italia
./9	Orange
./10	Telstra
./12	Bell
./13	Telus
./14	Rogers
./15	Docomo
./16	Swisscom
./17	Aeris
./OEM	Temporary storage for one-time file used to update OEM settings.

a. Table subject to change as additional carriers are added.

Image Storage—Module-side

Sierra Wireless modules can store multiple image files (firmware and carrier PRI files) to support image switching:

- Firmware files are stored in firmware 'slots' in a compressed format. The currently running firmware is an uncompressed copy of one of the stored files.

- Carrier PRI files are stored as regular files in the module's file system. (Note: Although some AT commands may mention PRI slots in their responses, slots are not actually used.)

The active image running on a module at any time is a combination of:
- a stored carrier PRI file
- an uncompressed copy of a firmware file identified by the carrier PRI.

The number of files of each type that can be stored is module-dependent, as summarized in Table 1-5.

Note: To confirm the number of files that can be stored on your specific device, use the AT!IMAGE command (see !IMAGE on page 18), which lists the "Max FW images" and "Max PRI Images".)

Table 1-5: Module Storage—Firmware and Carrier PRIs[a]

Device Type	Firmware slots	PRI files
EM73XX	1	50
EM74XX	4	50
MC73XX	1	50
MC74XX	4	50
WP75XX	2	50
WP8548	2	50

a. Stated storage capacities are subject to change. Use AT!IMAGE to confirm values if required.

Switching Active Images

Sierra Wireless provides a Linux SDK that includes the API functions required to initiate and manage image switching. For details, see Linux SDK APIs on page 16.)

At any one time, a single active image is running on the module. This is a combination of an uncompressed firmware file and a carrier PRI file.

The general process for switching images is:

1. Host queries the module to determine which images are stored on the module. The host can then determine if a firmware download will be required.
2. Host specifies the preferred image to be used (carrier PRI file and related firmware file).
3. Host reboots the module.
4. If the required firmware is not on the module:
   a. Module asks the host to download the required firmware.
   b. Host downloads the firmware to the module. (Note: The slot chosen for storing the compressed firmware file depends on how slots are assigned—see Firmware Slot Assignment on Module on page 11.)

5. Module uncompresses and loads the requested firmware file.

The general process for switching images using SDK API functions is shown in Figure 1-1 on page 10.

The module can also be configured to automatically switch images to one which is the best match for the active SIM. See SIM-based Image Selection (AUTO-SIM) on page 15.

Warning: *Image switching is intended for occasional use. Because firmware files are stored in compressed format on the module, frequent image switching (involving different firmware files) will stress the device's flash memory.*

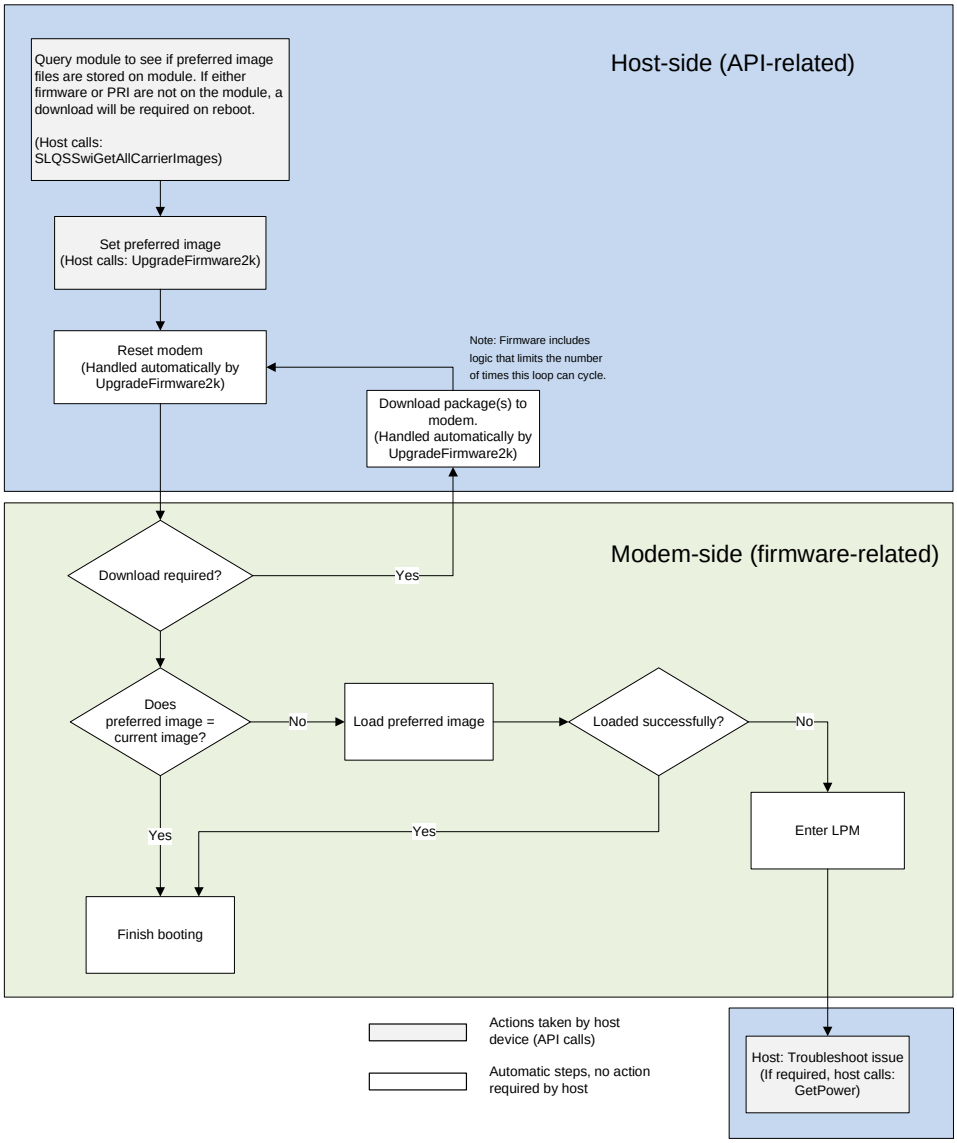

Figure 1-1: Image switching process

Firmware Slot Assignment on Module

As described in Table 1-5 on page 9, each Sierra Wireless module provides multiple storage slots for firmware files. The specific slot used to store a newly-downloaded firmware file depends on whether it is automatically assigned by the module or manually assigned by the host.

Automatic Slot Assignment By Module

During an image switch, if a firmware file is downloaded from the host, the module automatically determines which firmware slot to use as follows:

- Store new firmware file in the first empty slot, if available.
- Otherwise, store it in the first orphaned slot (a slot occupied by a firmware file that is not used by any of the stored carrier PRI files). Figure 1-2 on page 12 illustrates how firmware slots and carrier PRI files are orphaned.
- Otherwise, store in the least recently used slot.
  (Note that the carrier PRI file(s) that are linked to the replaced firmware are now orphaned. If a later switch occurs using an orphaned PRI file, the related firmware will need to be downloaded again to the module.)

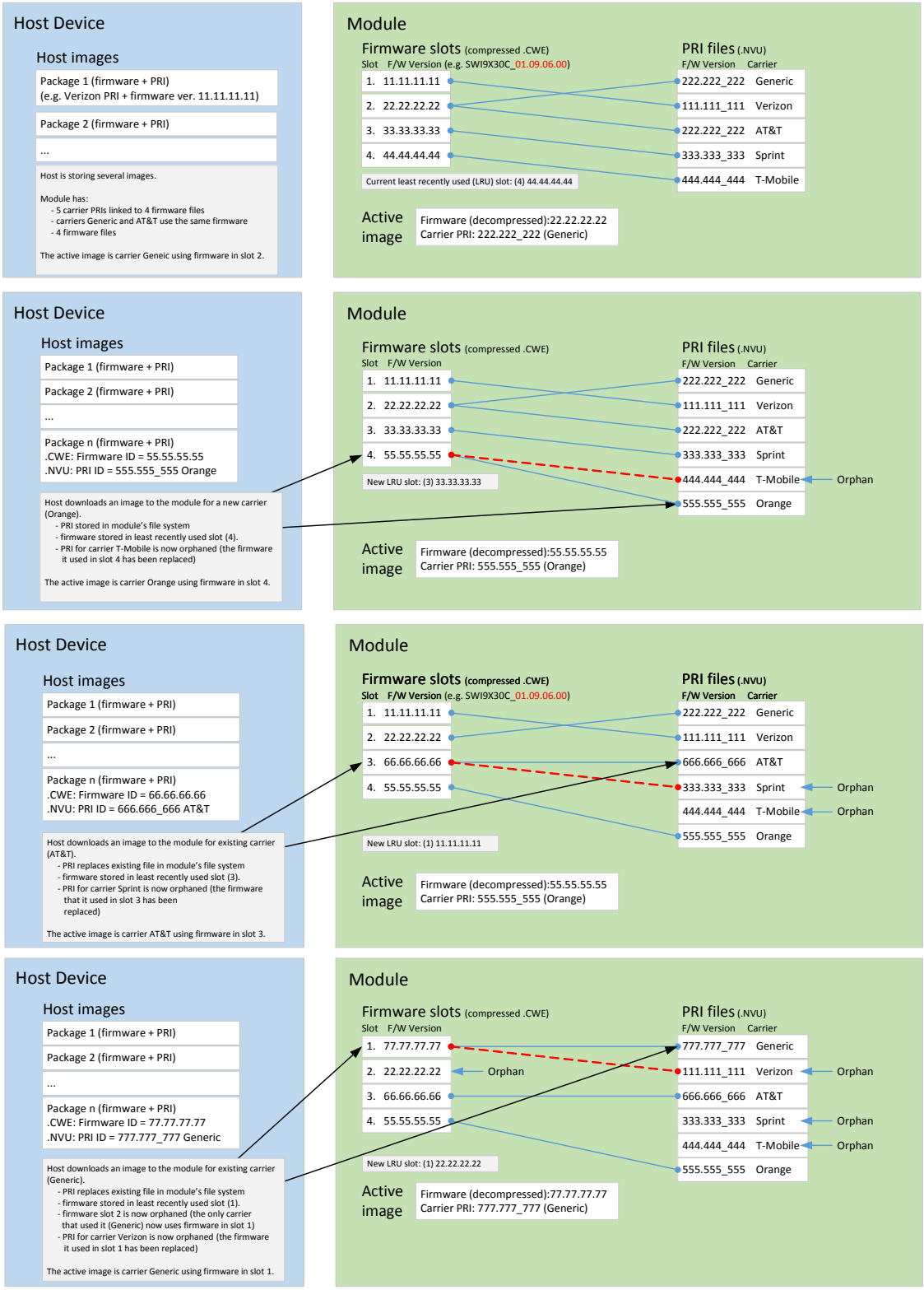

Figure 1-2: Orphaned Firmware and Carrier PRI files

Manual Slot Assignment By Host

If desired, the host can use an API to manually download a firmware file to a specific firmware slot on the module while initiating an image switch. By doing this, the image switch process will not automatically download the firmware file as described in Automatic Slot Assignment By Module on page 11.

To download a firmware file to a specific slot on the module, use SLQSDownloadFirmwareToSlot(). For details, see SLQSDownloadFirmwareToSlot on page 17.

Note: If a firmware file is used by more than one carrier PRI, only one instance of the file needs to be on the module—the PRI files link to the same firmware file. If the host downloads a duplicate firmware file to a specific slot on the device, the module will mark the slot as orphaned and the firmware will be removed during Automatic Slot Assignment By Module on page 11.

Image File Management

The image files (firmware and carrier PRI) stored on the module can be managed in the following ways:

- Host-based Management of Image Files on page 14
- Module-based Management of Image Files on page 14
- SIM-based Image Selection (AUTO-SIM) on page 15
- FOTA-based Management of Image Files on page 15

The image file management method(s) used will typically depend on the type of host device and on how the module is to be used. Table 1-6 on page 13 describes typical image management tasks and indicates the APIs and AT commands available to perform them.

Table 1-6: Image Management Tasks

Task	Linux APIs	AT commands
Get the preferred image details (firmware file, carrier PRI)	SLQSSwiGetFirmwareCurr on page 17	!IMPREF on page 19
Get details about the active image (firmware file, carrier PRI,etc.)	SLQSSwiGetFirmwareCurr on page 17	!IMAGE on page 18 !IMPREF on page 19
Query the module for a list of firmware and PRI images on the module.	SLQSSwiGetAllCarrierImages on page 17 to identify the required PRI.	!IMAGE on page 18
Check carrier PRIs loaded on module	GetStoredImages on page 17	!IMAGE on page 18
Check carrier PRIs stored on the host	GetStoredImages on page 17 SLQSSwiGetAllCarrierImages on page 17	-
Check f/w versions loaded on module	GetStoredImages on page 17 SLQSSwiGetAllCarrierImages on page 17	AT!IMAGE? (See !IMAGE on page 18.)

Table 1-6: Image Management Tasks (Continued)

Task	Linux APIs	AT commands
Check f/w versions stored on the host	GetStoredImages on page 17	-
Download firmware to specific modem slot	SLQSDownloadFirmwareToSlot on page 17	-
Delete image files (firmware and/carrier PRIs) from the module	DeleteStoredImage on page 16	!IMAGE on page 18
Set and switch to a preferred image (firmware file, carrier PRI) that could be stored on the host or the module	UpgradeFirmware2k on page 18	AT!IMPREF="<carrier>" (See !IMPREF on page 19.)
Switch to an image (firmware file, carrier PRI) that is currently stored on the module	UpgradeFirmware2k on page 18 with a pseudo path to select the required PRI (and associated firmware file) and initiate the switch.	!IMPREF on page 19
Enable/disable SIM-based switching	tbd	AT!IMPREF="AUTO-SIM" (enables SIM-based image switching) AT!IMPREF="<carrier>" (disables SIM-based image switching) See !IMPREF on page 19.

Host-based Management of Image Files

Host-based file management is an option for systems that have sufficient storage space on the host for extra firmware files and/or carrier PRI files that won't fit on the module (see Table 1-5 on page 9 for available space by module type).

Firmware files and carrier PRI files are stored on the host device and downloaded to the module:

- Automatically, during an image switch, if they are not already on the module. For details, see Automatic Slot Assignment By Module on page 11.
- Manually, during an image switch. For details, see Manual Slot Assignment By Host on page 13.

Module-based Management of Image Files

Module-based file management is used by systems that have limited host storage space.

All firmware files and carrier PRI files are stored on the module. New files are downloaded from the host when necessary, and the host does not keep copies.

Note: If rarely-used files are removed from the module causing PRIs to be orphaned, and then have to be reloaded, the host has to get them again before they can be downloaded. (See Figure 1-2 on page 12 for an explanation of orphaned files.)

Linux QMI SDK commands and AT commands can be used for on-module image management.

SIM-based Image Selection (AUTO-SIM)

When enabled, the SIM-based image switching feature (also called automatic switching or “AUTO-SIM”) allows the module to automatically switch images when a SIM is inserted that requires a different image than the one currently running.

For a description of the SIM-based image selection process, refer to Firmware Image Switching Developer’s Guide (Document #4115290).

To activate/deactivate SIM-based management, use AT!IMPREF=”AUTO-SIM”. For usage details, see !IMPREF on page 19.

FOTA-based Management of Image Files

If FOTA (which is a SKU and/or carrier customization) is supported by the module, image files may be pushed periodically to the module.

These files are stored by the module as described in Automatic Slot Assignment By Module on page 11.

If any firmware files stored on the module should not be overwritten by a FOTA image, the host can indirectly protect these by making sure an empty slot is available for automatic assignment. The host can use API calls and/or AT commands to delete the firmware from a specific slot, which makes the slot available for automatic assignment when the next FOTA image is pushed to the module.

2: Image Switching Functions

Control of image switching functionality is provided by:

- Linux SDK APIs on page 16
- AT Commands on page 18

These commands can be used to get information about image files on the module and host device, manage storage of image files (download, delete), and initiate image switches.

Note: For current command availability and details, refer to the API Reference Guide included with the SDK and the AT Reference Guide for your module (available at source.sierrawireless.com). Specific commands and/or command details may vary depending on firmware and SDK version.

Linux SDK APIs

Several APIs providing image switching functionality are available in the Linux SDK:

- DeleteStoredImage on page 16
- GetStoredImages on page 17
- SLQSDownloadFirmwareToSlot on page 17
- SLQSSwiGetAllCarrierImages on page 17
- SLQSSwiGetFirmwareCurr on page 17
- UpgradeFirmware2k on page 18

For recommended uses (and comparable AT commands), see Table 1-6 on page 13.

DeleteStoredImage

Purpose: Delete a firmware file from the module by specifying its unique ID and build ID.

Signature: ULONG DeleteStoredImage(
ULONG imageInfoSize,
BYTE *pImageInfo);

GetImagesPreference

Purpose: List the pair of files (firmware and Carrier PRI) that comprise the ‘preferred image’, which will be used on the next module reboot, if available.

Signature: ULONG GetImagesPreference(
ULONG *pImageListSize,
struct PrefImageList *pImageList);

GetStoredImages

Purpose: List the firmware files and carrier PRI files that are stored on the module, showing their file types, unique IDs, and build IDs.

Signature: ULONG GetStoredImages(
ULONG *pImageListSize,
BYTE *pImageList);

SLQSDownloadFirmwareToSlot

Purpose: Download a firmware file from the host device to a specific slot on the module. Note that if any carrier PRI files were linked to the firmware originally in the slot, then the original firmware will have to be downloaded again if any of those PRI files are activated later.

Signature: ULONG SLQSDownloadFirmwareToSlot(
CHAR *pPath,
BYTE slot_index,
BYTE force_download);

SLQSGetFirmwareInfo

Purpose: List the components (firmware file and carrier PRI) of the active image. (Note: This command is used if not in Gobi Image Management Mode; otherwise, use SLQSSwiGetFirmwareCurr on page 17.)

Signature: SLQSGetFirmwareCurr(
CurrentImgList *pCurrentImgList);

SLQSSwiGetAllCarrierImages

Purpose: List all available firmware and carrier PRI files that are stored on the module or host.

Note: On the host device, carrier PRI files should be stored with their matching firmware files in uniquely named sub-folders.(See Image Storage—Host-side on page 8 for details.)

Signature: ULONG SLQSSwiGetAllCarrierImages(
ULONG *pNumOfItems,
struct SWI_STRUCT_CarrierImage *pCarrierImages,
wchar_t *pFolderPath);

SLQSSwiGetFirmwareCurr

Purpose: List the components (firmware file and carrier PRI) of the active image. (Note: This command is used if in Gobi Image Management Mode; otherwise, use SLQSGetFirmwareInfo on page 17.)

Signature: SLQSSwiGetFirmwareCurr(
CurrentImgList *pCurrentImgList);

UpgradeFirmware2k

Purpose: Indicate the image that is to be used (the 'preferred image'), reboot the module, download the preferred image files (firmware and/or carrier PRI) if needed, and complete the image switch by making the preferred image active.

Signature: ULONG UpgradeFirmware2k(
CHAR *path);

SLQSSetSIMBasedImageSwitching

Purpose: Enable SIM-based image switching.

Signature: ULONG SLQSSetSIMBasedImageSwitching(void);

Note: SIM-based image switching can be disabled by downloading a regular image.

AT Commands

Several AT commands providing image switching functionality are supported by the module:

- !IMAGE on page 18
- !IMPREF on page 19

For recommended uses (and comparable Linux APIs), see Table 1-6 on page 13.

!IMAGE

Purpose: List the images that are currently stored on the module, or delete PRI and/or firmware files from the module.

Syntax (parameters listed at end):

Query: AT!IMAGE?[<op>[,<type>]]

Purpose: List all image files, all firmware or PRI files, or the maximum number of firmware slots.

Response (Example shown is for AT!IMAGE? with no parameters):

at!image?

TYPE	SLOT	STATUS	LRU	FAILURES	UNIQUE_ID	BUILD_ID
FW	1	GOOD	25	0 0	?_?	01.08.02.00_?
FW	2	GOOD	22	0 1	?_	01.09.00.00_?
FW	3	GOOD	24	0 0	?_?	00.00.00.00_?
FW	4	GOOD	23	0 0	?_?	01.09.03.00_?

Max FW images: 4
Active FW image is at slot 1

TYPE	SLOT	STATUS	LRU	FAILURES	UNIQUE_ID	BUILD_ID
PRI	FF	GOOD	0	0 0	001.000_000	01.00.04.00_ATT
PRI	FF	GOOD	0	0 0	001.003_000	01.02.00.02_GENEU-4G
PRI	FF	GOOD	0	0 0	001.003_000	01.02.00.02_ORANGE-EU
PRI	FF	GOOD	0	0 0	001.004_000	01.02.00.02_SWISSCOM
PRI	FF	GOOD	0	0 0	001.004_000	01.02.00.02_TELEFONICA

Max PRI images: 50

OK

Assignment: AT!IMAGE?[<op>[,<type>[,<slot>[,<build_id>,<unique_id>]]]

Purpose: Delete PRI and/or firmware files from the module.

Response:

OK

Examples:

AT!IMAGE=0	Deletes all stored image files (firmware and PRIs)
AT!IMAGE=0,0	Delete all stored firmware files
AT!IMAGE=0,0,2	Delete the firmware in slot 2
AT!IMAGE=0,1,, “01.00.01.00_SWISSCOM”,”000.001_000”	Delete a specific PRI using it’s build ID and unique ID.

Parameters:

<op> (Operation):

- 0—Delete (Only for Assignment format)
- 1—List files (Only for Query format)
- 2—Display number of firmware slots supported

<type> (Image type):

- 0—FW (Firmware)
- 1—PRI (Carrier PRI)

<slot> (Firmware image slot ID. Ignore this field for PRI files.)

- 0–FF—Maximum value is SKU-dependent

<build_id> (Build ID)

- ASCII string enclosed by double quotes

<unique_id> (Unique ID)

- ASCII string enclosed by double quotes

!IMPREF

Purpose: Initiate an image switch either by choosing a preferred carrier or by enabling SIM-based image switching, or list the preferred and current firmware versions and carrier configurations.

Note: When setting the preferred carrier using this command, the command will fail if the required firmware for the carrier is not already present on the device. Use the AT!IMAGE? command to see a list of available carrier configurations and firmware.

Syntax (parameters listed at end):

Query: AT!IMPREF?

Purpose: List the preferred image and current image details.

Response:

!IMPREF:
preferred fw version: <firmware_ver>
preferred carrier name: <carrier-name>
preferred config name: <carrier-config>

```
current fw version:      <firmware-ver>
current carrier name:    <carrier-name>
current config name:     <carrier-config>

[<mismatch information>]

OK
```

Assignment: AT!IMPREF=<preference>

Purpose: Choose the preferred carrier, or enable SIM-based image selection. The module will search for a matching carrier PRI and the required firmware for that PRI. If both are found, the new image preference is set.

Response:

```
OK
```

Parameters:

<preference> (Carrier name, or "AUTO-SIM"):

- ASCII string within double quotes
- Use "AUTO-SIM" to enable SIM-based image selection. Disable it by selecting an actual carrier.

<mismatch information> (Type of mismatch between current and preferred image settings):

- ASCII string
- "fw version mismatch"
- "carrier name mismatch"
- <config name mismatch"

3: Frequently Asked Questions

How Do I Check The Active Image?

To check which image is currently running on the module:

- Use SLQSSwiGetFirmwareCurr on page 17.
- Use AT!IMAGE?. This also shows all the firmware and carrier PRI files that are loaded on the module. See !IMAGE on page 18 for details.
- Use AT!IMPREF?. This shows the active (‘current’) image and the preferred image. See !IMPREF on page 19 for details.

How Do I Check Which Image Will Be Switched To After Module Reboots?

To check which image is identified as the ‘preferred’ image (the image that will be used on the next module reboot, if available):

- Use GetImagesPreference on page 16.
- Use AT!IMPREF?. This shows the preferred image and the active (‘current’) image. See !IMPREF on page 19 for details.

How Do I Check Which Image Files Are Available?

To check which image files are stored on the module or host:

- Use SLQSSwiGetAllCarrierImages on page 17.
- Use AT!IMAGE?. This shows all the firmware and carrier PRI files that are loaded on the module or host, and it indicates the active image. See !IMAGE on page 18 for details.

4: Troubleshooting

Module remains in low power (offline) mode

Some common reasons the module enters and remains in low power (offline) mode when image switching is supported include:

- Preferred image firmware settings do not match the current firmware. This can happen if firmware is downloaded without setting the image preference.
- A file download (which occurs when the host changes the image preference) is interrupted.

To check if a problem with image switching is the reason the module is in low power mode, use the API functions described below.

Use API functions to diagnose LPM cause

1. Use **GetPower()** to check if the module is in low power mode.
2. If the module is in low power mode, use **SwiGetCurrentImages()** to get the current image details.
3. Use **GetImagesPreference()** to identify the preferred image.
4. Use **UpgradeFirmware2k()** to download the correct firmware and/or configuration files to match the preferred settings. The module will reboot and attempt the image switch again.

Module starts in boot and hold mode

If the module starts in boot and hold mode after a firmware download is interrupted, the host must re-initiate the image switch (call UpgradeFirmware2k on page 18).

New firmware file not stored after download

If the host downloads new firmware to the modem without setting the image preference, and the module is already running a matching firmware and carrier PRI, the modem will discard the new firmware and restore the existing preferred configuration.

To avoid this problem, make sure when downloading new firmware to set the image preference.

Preferences are not persistent over image switches

When switching between images, factory settings and carrier-specific preferences persist, but certain user-preferences (such as SMS settings) do not.

A: Glossary

A

Term	Definition
active image	The carrier PRI and firmware file that are currently running on the module.
API	Application Programming Interface. A routine included in the SDK.
AT	AT command
automatic switching	A feature that enables the module to automatically switch to an appropriate image when a different carrier's SIM is inserted.
configuration file	See PRI.
.cwe	Filename extension of a firmware file. See firmware.
firmware	Operating software for the module.
host	The device in which a module is installed.
image switching	Process where a module that is running one image (firmware plus PRI) is restarted to run a different image.
module	A Sierra Wireless AirPrime module (MC, EM, etc.)
.nvu	Filename extension of a PRI file. See PRI.
orphaned	A slot containing a file (firmware or PRI) that does not link to a corresponding stored file (PRI or firmware).
preferred image	The carrier PRI and firmware file that should be used when the module reboots.
PRI	Product Release Information—A configuration file containing carrier-specific module settings. One PRI file and one firmware file combine to form an 'image'.
QMI	Qualcomm MSM Interface
SDK	Software Development Kit
SIM	Subscriber Identity Module
SKU	A customer-specific version of a module. (e.g. Carrier#1 may have two variants of an MC74xx module. Each variant is a unique SKU.)
slot	Image file storage location on the module. Device-dependent numbers of slots for firmware files and for PRI files.