



**Linux SDK
Version 1.4.0.5
Release Notes**

Document Revision:	1.0
---------------------------	------------

SIERRA WIRELESS INC.

Document #:	N/A	Revision:	1.0	Page 2 of 10
-------------	-----	-----------	-----	--------------

Table of Contents

1	Introduction.....	3
1.1	Trademarks	3
1.2	Change Summary.....	3
	Code Cleanup/Improvements.....	3
1.2.1	New API – SwiTerminateSDK() – Kill the SDK Daemon Process	4
1.2.2	New API – SwiCreateNotifThread() – Create Thread to Handle Notifications 4	
1.2.3	Modifications to GSM FirmwareDownload Utility.....	5
1.2.4	Modifications to GSM Sample Code.....	5
1.2.5	Added Support for New Bands	5
1.2.6	Code Cleanup/Improvements.....	6
1.2.7	Doxygen Documentation Update.....	6
1.3	Compatibility	6
1.3.1	Hardware Compatibility.....	6
1.3.2	Supported Architectures.....	6
1.3.3	Supported Sierra Wireless Embedded Modems	6
1.3.4	Software Compatibility	7
2	SDK Installation.....	9
2.1	Unpacking the Distributed Files.....	9
2.2	Verify the Contents.....	9

Document #:	N/A	Revision:	1.0	Page 3 of 10
-------------	-----	-----------	-----	--------------

1 Introduction

This document contains technical details for the version 1.4.0.5 Linux SDK software release. It identifies the incremental changes to the SDK since its previous release. The previously released version of the SDK was 1.4.0.2.

Release 1.4.0.5 consists of bug fixes and code improvement/cleanup. Please review the Change Summary section in this document to help you become familiar with what's new in this release of SDK.

1.1 Trademarks

Linux is a registered trademark of Linus Torvalds

AirCard® and “Heart of the Wireless Machine®” are registered trademarks of Sierra Wireless.

Other trademarks are the property of the respective owners.

1.2 Change Summary

The table below contains hyperlinks to subsections of this document describing the changes.

Change Summary	
Item	Title
1	New API – SwiTerminateSDK() – Kill the SDK Daemon Process
2	New API – SwiCreateNotifThread() – Create Thread to Handle Notifications
3	Modifications to GSM FirmwareDownload Utility
4	Modifications to GSM Sample Code
5	Added Support for New Bands
6	Code Cleanup/Improvements
7	Doxygen Documentation Update

Document #:	N/A	Revision:	1.0	Page 4 of 10
-------------	-----	-----------	-----	--------------

Additional details about these changes follow in the next subsections.

Throughout this document, if necessary, we refer to the root directory in which you have installed the SDK as

<SDKROOT>

1.2.1 New API – SwiTerminateSDK() – Kill the SDK Daemon Process

This new API is provided for occasions when the SDK daemon is required to accomplish a task but is not required for ongoing operation. An example of this situation would be to support downloading firmware to a Sierra Wireless device installed in a Linux-based platform but which is controlled by means other than the Linux SDK.

Internal changes to the SDK in support of this API are to record the Process ID (PID) of the SDK daemon at the time it is created. The PID is stored in memory in the API-side. Calling this new API causes the SDK to be killed using the “kill()” Linux System call. Therefore, if the PID of the running SDK is different from the one stored the API side memory, the API call will fail.

As part of this change, the SDK daemon now logs a shutdown message in response to a Linux SIGTERM signal:

SDK daemon received SIGTERM - shutting down

Prior to this change the SDK generated a stack dump in response to the signal.

For an example of where this new API is used, the reader is referred to the handling of the new “-x” switch in:

Utilities/gsm/FirmwareDownload/FirmwareDownload.c

1.2.2 New API – SwiCreateNotifThread() – Create Thread to Handle Notifications

A new API has been added to create the Notification thread for API-side applications. Prior to this API applications were left to create the notification thread using their own methods – e.g. via direct calls to pthread_create(). This method has been observed to cause applications to crash when used on certain platforms – e.g. Android.

The root cause of the crash is an assumption about how the scheduler works when creating new threads. The assumption is that the new thread will always run as soon as it is created and the calling thread will be blocked until the new thread reaches a point in its operation where it will become blocked. This assumption has been successfully tested on many Linux operating systems. However, on some Android platforms, the opposite situation is the case, the newly-spawned thread is blocked until the caller blocks. In the event the caller proceeds to make additional API calls to the notification facility, the crash results. The crash happens because the notification API calls are made before the notification facility has had a chance to initialize itself.

Document #:	N/A	Revision:	1.0	Page 5 of 10
-------------	-----	-----------	-----	--------------

The new API prevents the crash by blocking the calling thread until the notification thread has completed its essential initialization steps.

For an example of how this new API is used, the reader is referred to:

Utilities/gsm/FirmwareDownload/FirmwareDownload.c
in function initializeAPI().

1.2.3 Modifications to GSM FirmwareDownload Utility

The FirmwareDownload.c utility, located in:

Utilities/gsm/FirmwareDownload/FirmwareDownload.c

has been modified to use the new APIs noted in 1.2.1 and 1.2.2. A new command line switch, -x and -exit, have been added to instruct the firmware download application to kill the SDK daemon when the firmware download is complete. Without this switch added to the command line the firmware download operation is the same as it always has been. With the addition of the switch, the firmware download will kill the SDK daemon using the new API noted in 1.2.1.

There are some cases where killing the SDK daemon will fail and these are noted:

1. The SDK daemon was already running or created by another application. In this case, when the firmware download application tries to kill the daemon it will fail
2. The firmware download application was aborted before completion using the ^C operation from the command line. Subsequent restarts of the firmware download application would run afoul of condition 1, above.

1.2.4 Modifications to GSM Sample Code

The GSM Sample Code Main Routine, located in:

Sample_Code/gsm/Common/Main/CommonMain.c

has been modified to use the new APIs noted in 1.2.1 and 1.2.2.

The new notification creation mechanism (modeled after FirmwareDownload.c) was applied. This affects all GSM Sample Code Applications.

1.2.5 Added Support for New Bands

Added entries to the Linux V2 SDK for the following band masks:

WcdmaEu:

WCDMA 900/2100 (0x00000008100000000)

WcdmaAll:

WCDMA 2100/1900/850/800/900 (0x0000000b300000000)

SIERRA WIRELESS INC.

Document #:	N/A	Revision:	1.0	Page 6 of 10
-------------	-----	-----------	-----	--------------

Note that there are now multiple references to CDMA ALL (due to different CDMA band support on different devices).

1.2.6 Code Cleanup/Improvements

Removed warnings related to potential use on non-initialized variables throughout the code.

1.2.7 Doxygen Documentation Update

Updated Doxygen documentation for version 1.4.0.5.

1.3 Compatibility

1.3.1 Hardware Compatibility

There are two Sierra Wireless software components which have hardware dependencies built into them, the Sierra Wireless USB Driver and the SDK itself. This dependency is predominantly restricted to the USB Vendor ID and Product ID that the driver and SDK will detect, although there are additional dependencies in the driver as well, not discussed in this document.

1.3.2 Supported Architectures

This build of the Linux SDK contains object files for 32-bit Intel 80x86 architectures and for the recommended ARM 9 development platform. Sierra Wireless recommends the use of the [Technologies TS-7800](#) single board computer for your early-stage development of ARM 9-based applications.

1.3.3 Supported Sierra Wireless Embedded Modems

The following Sierra Wireless modems are compatible with this version of the SDK. Note, not all the listed modems have been tested against this version of firmware:

Product Name
GSM
MC8775
MC8777
MC8704
MC8705
MC8780 / MC8781

SIERRA WIRELESS INC.

Document #:	N/A	Revision:	1.0	Page 7 of 10
-------------	-----	-----------	-----	--------------

MC8785
MC8790
MC8791/MC8792
MC8700
MC8801
SL8080
SL8081
SL8082
SL8083
SL8084
SL8085
SL8090
SL8091
SL8092
SL8093
CDMA
MC5725
MC5727
MC5727V
MC5728V
SL5010
SL5011
AC802

1.3.4 Software Compatibility

This version of the Linux SDK has been developed and tested using Ubuntu 8.04 LTS as installed from the Live CD *without* any subsequent upgrades applied. The as-delivered image contains source code, objects and executables for the supported i386/32 bit architecture.

1.3.4.1 C – Language

The Linux SDK is coded in the C-language and all external entry points (APIs) are callable by any language that can call C-Language functions. Some small changes to header files have been made to make calling C functions from the C++ language easier. The top and bottom of these header files have had the following code snippets added:

Top:

```
#ifndef __cplusplus
extern "C" {
#endif
```

Bottom:

```
#ifdef __cplusplus
```

SIERRA WIRELESS INC.

Document #:	N/A	Revision:	1.0	Page 8 of 10
-------------	-----	-----------	-----	--------------

```
}  
#endif
```

1.3.4.2 Libraries

Customer applications can be built entirely from the library functions located in the *build/lib/<architecture>* directory.

Libraries are created using the Linux archive tool, “ar”, version 2.18.0.20080103

Document #:	N/A	Revision:	1.0	Page 9 of 10
-------------	-----	-----------	-----	--------------

2 SDK Installation

This release of the SDK has been distributed as a tar'ed, zipped image. The steps for unpacking this software are described in detail in the following subsection.

2.1 Unpacking the Distributed Files

Use the following steps to unpack your Linux SDK release.

- Place the distribution file into a directory which will become the root directory for your Linux SDK directory tree.
- If you are using the command line, unzip the contents of the distributed file using the command:

```
tar -xvvzf LinuxSDK_V1_4_0_5.tar.gz
```

- When this operation is complete, you will have a new directory with the same name as this Linux SDK distribution.

2.2 Verify the Contents

Executables for the Ubuntu 8.04 desktop environment are included with this distribution and should work out-of-the-box. A quick check to ensure the executables are able to run on your machine can be accomplished as follows:

- Open two shell windows and position them so they can be seen simultaneously. In one of the windows enter the following Linux command:

```
tailf /var/log/user.log
```

this will cause the contents of the user.log file to be printed to the console whenever any changes are made to that file.

- In the other shell window, ensure there is not already a previous version of the Linux SDK process running with the command:

```
killall swisdk1
```

This will cause an sdk process to be terminated immediately if there is one running.

- Change directory to

¹ Note, if you have been using older versions of the SDK you might have started the SDK daemon using its as-linked name, e.g. swisdk386. If there is a chance you have been running this executable then you must also make sure it is killed before attempting to use this new release.

SIERRA WIRELESS INC.

Document #:	N/A	Revision:	1.0	Page 10 of 10
-------------	-----	-----------	-----	---------------

LinuxSDK_V1_4_0_5/pkg/ap

- Start this new version of the SDK by entering the following command into the same shell window you used to kill the currently-executing SDK (if any) in the previous step:

```
./aptesti386 -n t1 -p <SDKROOT>/build/bin/i386/swisdk
```

- If there is no Sierra Wireless modem installed on your computer, the test application will start and continuously display the following message until the SDK detects the presence of a modem:

Air Server not available – SLEEPING

- You should be able to observe the log file (displayed in the other shell window) come alive with the display of several messages indicating the SDK process has started

If you have a compatible Sierra Wireless modem installed, after several seconds the aptesti386 command you entered above will terminate and the firmware version string from the attached modem will be displayed. The output will look similar to:

API Opened Successfully

Modem's boot and hold state is 0

Invoking test: t1, SwiGetFirmwareVersion

Test A: successful fetching of firmware version

Modem Firmware Version:

J0_0_3_5AP C:/WS/FW/J0_0_3_5AP/MSM7200A/SRC/AMSS 2008/02/26 16:59:19

Test B: Empty firmware version buffer given

Buffer too small!

Test C: Small firmware version buffer given

Modem Firmware Version:

J0_0_3_5AP

Test Completed -----