Gobi SDK Integration Guide for Windows CE

Gobi 2000, Gobi 3000, AirPrime SL9090, AirPrime MC73xx

SIERRA WIRELESS®

4117228
1.0
April 10, 2015

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 cellular modems are not advised without proper device certifications. These areas include environments where cellular radio can interfere such as explosive atmospheres, medical equipment, or any other equipment which may be susceptible to any form of radio interference. 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.

Limitations of Liability

This manual is provided "as is". Sierra Wireless makes no warranties of any kind, either expressed or implied, including any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. The recipient of the manual shall endorse all risks arising from its use.

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.

Customer understands that Sierra Wireless is not providing cellular or GPS (including A-GPS) services. These services are provided by a third party and should be purchased directly by the Customer.

SPECIFIC DISCLAIMERS OF LIABILITY: CUSTOMER RECOGNIZES AND ACKNOWLEDGES SIERRA WIRELESS IS NOT RESPONSIBLE FOR AND SHALL NOT BE HELD LIABLE FOR ANY DEFECT OR DEFICIENCY OF ANY KIND OF CELLULAR OR GPS (INCLUDING A-GPS) SERVICES.

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®, WISMO®, ALEOS® and the Sierra Wireless and Open AT logos are registered trademarks of Sierra Wireless, Inc. or one of its subsidiaries.

Watcher® is a registered trademark of NETGEAR, Inc., used under license.

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

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

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

Other trademarks are the property of their respective owners.

Contact Information

Sales Desk:	Phone:	1-604-232-1488
	Hours:	8:00 AM to 5:00 PM Pacific Time
	Contact:	http://www.sierrawireless.com/sales
Post:	Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4	
Technical Support:	support@sierrawireless.com	
RMA Support:	repairs@sierrawireless.com	
Fax:	1-604-231-1109	
Web:	http://www.sierrawireless.com/	

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

Document History

Version	Date	Updates
Draft A	November 24, 2010	Creation
Draft B	January 14, 2011	Added firmware image restriction feature
1.0	April 10, 2015	Added information on Gobi 3000, SL9090, SL9090RD and MC73xx

Contents

1. General

1.1. Purpose

This document contains a collection of references and development notes to aid in the software integration of Gobi 2000, Gobi 3000, AirPrime SL9090 and AirPrime MC73xx modules on Microsoft Windows CE, Windows Embedded Compact and Windows Mobile 6.5 platforms. This document is intended for OEMs and third party application developers.

1.2. Scope

This document only pertains to software integration on Microsoft Windows embedded operating systems. Gobi Application Programming Interface (API) and driver design and usage are discussed. Hardware integration, Radio Interface Layer (RIL), and other operation systems are not covered in this document.

1.3. References

Contact your Sierra Wireless representative for a copy of the references listed below as they are not included in the Sierra Wireless Gobi WinCE SDK. Some reference documents may be available for download from www.gobianywhere.com.

[1] Software Glossary for Customers

Reference number: CL93-V3077-1

1.4. Terminology and Acronyms

Term/Acronym	Definition
API	Application Programming Interface
ODM	Original Device Manufacturer
OEM	Original Equipment Manufacturer
QMI	Qualcomm Modem Interface; Qualcomm MSM Interface
SDK	Software Development Kit
SMS	Short Message Service
RIL	Radio Interface Layer

2. Overview

2.1. System Requirements

See release notes for a list of supported development configurations. Microsoft Visual Studio requires a platform SDK to build native code for the target platform. Microsoft provides generic BSP support for various platforms and SDK can be built using the BSP. Third party software developers will need to contact OEMs for specific SDKs unique to their platform. Sierra Wireless uses some reference SDKs for X86 and ARM platforms. Contact your Sierra Wireless representative for a copy of the Sierra Wireless Windows CE Standard SDK if needed.

Sample applications are produced using Visual Studio 2008 along with SP1.

2.2. SDK Installation

The SDK is packaged as a compressed ZIP file and can be uncompressed using the standard extractor included with Windows.

The SDK package does **NOT** include firmware images.

2.3. Folder Structure

2.3.1. Top Level Folder

This folder contains release notes and the integration guide.

The following is the list of key components included in this folder.

Component	Description
Gobi SDK Release Notes for WinCE X.X.pdf	Release notes
Gobi SDK Integration Guide for WinCE.pdf	This document; integration guide
FDT_ReadMe.txt	Documentation for FDT (firmware download tool). Supports MC73xx and SL9090RD.

2.3.2. Binary

This folder contains binaries for drivers and sample applications. Files for different system architectures are available in their own subfolders. Retail and Debug files are also available in separated sub folders. CAB install packages are also available in the CAB sub folder.

The following is the list of key components included in this folder.

Component	Description
SwiGobi.dll	Implements NDIS, serial, and USB driver functionality
SwiQdl.dll	Implements firmware download functionality for Gobi 2000, Gobi 3000, SL9090 and SL9090RD
GobiAPI.dll	SDK library

Component	Description
CEGobiSDK.cab	Driver installation package
FDTCE.exe	The firmware download tool. Supports MC73xx and SL9090RD.

2.3.3. Documentation

This folder contains the Gobi API documentation in doxygen format. Please use index.html as the entry point.

2.3.4. GobiApiHeader

This folder contains header files for Gobi APIs. The APIs are divided into different headers by their function. For new projects, please use GobiAPI.h; it will include all other headers.

The following is the list of key components included in this folder.

Component	Description
GobiAPI.h	Top level header files. It includes all other headers.
QCWWANCMAPI2K.h	Support for legacy Gobi 2000 devices only. No longer maintained.

2.3.5. Integration

This folder contains the sample registry setting file to support the USB driver.

2.3.6. Samples

This folder contains source code for various sample applications. Customers can use them to verify basic SDK functionality, or use them as a starting point to create their own applications.

Component	Description
GobiCarrierFWSelect	Firmware switching for Gobi 2000, Gobi 3000, SL9090 and SL9090RD
GobiTestApp	Directly accesses most of the APIs. Useful for quick testing of a specific API
QMI_NMEA	Access to the NMEA positioning messages with QMI interface instead of the traditional serial interface.
SimpleCM	Starts and stops data connection. Supports CDMA, GSM, UMTS and LTE.
SMSSample	Sends and receives SMS
VoiceDialer	Makes and manages voice call. Controls M2MAV audio interface supported by MC73xx modules.

3. Drivers

3.1. SwiGobi

SwiGobi exposes a USB composite configuration which includes an NDIS adapter and three virtual serial interfaces:

- The NDIS adapter name is SWI Gobi.
- An AT command port is available on port SWI9. RAS/DUN (PPP) connections are supported with the AT command port. This port can be used simultaneously with Gobi API functionality.
- A firmware diagnostic (DM) port is available on port SWI5.
- GPS NMEA port is available on port SWI6.

3.1.1. Selective Suspend

Selective suspend is disabled by default. Use the following registry key to configure selective suspend functionality:

[HKEY_CURRENT_USER\Software\Sierra Wireless\USB]

"IdleTimeout"=dword:0

“IdleTimeout” is the timeout value in milliseconds. Set it to “0” to disable selective suspend. For example, setting “IdleTimeout” to dword:BB8 will enable selective suspend functionality with a timeout of 3 seconds.

If selective suspend is enabled (“IdleTimeout” value is not 0), the minimum value for “IdleTimeout” is 3 seconds. If the “IdleTimeout” value is set to less than 3 seconds, the driver will force it to 3 seconds.

3.1.2. Network Interface Operation Mode

All modules supported by SwiGobi.dll support both serial PPP mode over ATPPP port, and a packet based RMNET mode. Typically, the RMNET port is mapped to an NDIS interface for Windows CE and Windows Embedded Compact, and it is mapped to the RIL interface under Windows Mobile 6.5. It can be controlled by a registry value:

[HKEY_CURRENT_USER\Software\Sierra Wireless\USB]

"UseNDISConfiguration"=dword

Set this value to 0 to select the RIL mode for Windows Mobile 6.5 and set the value to 1 to select NDIS mode.

If this value is missing, the driver defaults to NDIS mode for Windows CE and Windows Embedded Compact, and to RIL mode for Windows Mobile 6.5.

3.1.3. TcpWindowSize

For optimal data throughput performance, TcpWindowSize on the target device must be set to 128k (131072). TcpWindowSize configuration can be found under:

HKEY_LOCAL_MACHINE\Comm\Tcpip\Parms]

3.2. SwiQdl

SwiQdl.dll implements firmware download functionality, supported by Gobi 2000, Gobi 3000 and SL9090.

The following registry values are used to specify the location of available carrier images and the specific carrier image to be downloaded to the Gobi module:

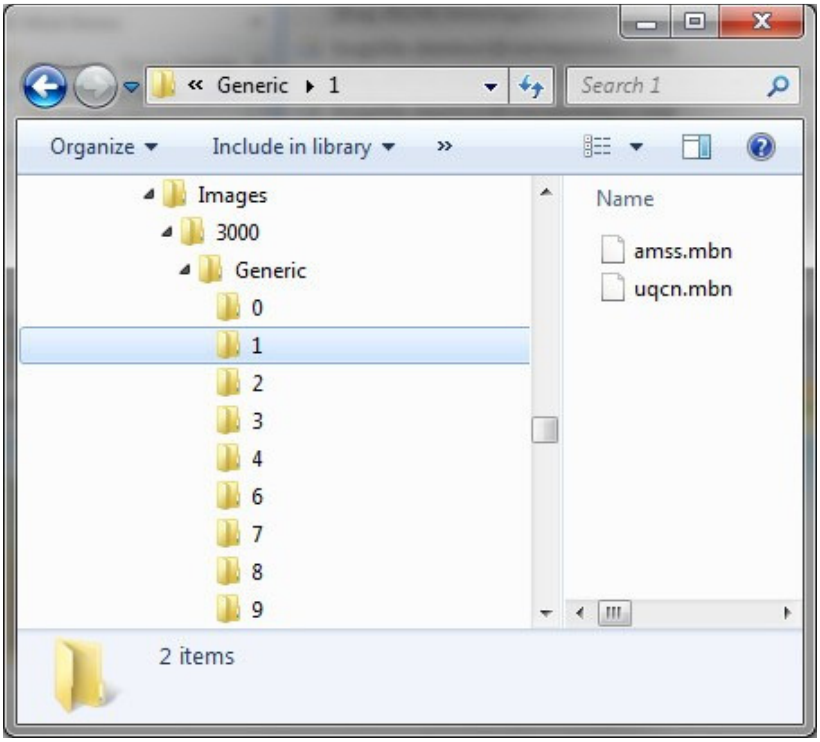
[HKEY_LOCAL_MACHINE\Software\Sierra Wireless\QDL]

"FirmwareLocation"="<path to carrier images>"

"CarrierIndex"=dword:<specific carrier index>

3.2.1. Firmware Image Folder Structure

Gobi firmware images should be placed in a folder structure grouped by different carrier indices. For example:



Where

Folder	Mobile Operator
0	Vodafone
1	Verizon
2	AT&T
3	Sprint
4	T-Mobile
6	Generic EU
7	Telefonica
8	Telecom Italia
9	Orange
12	DoCoMo
14	AMX/Telcel
16	OMH

Folder 6 contains the generic UMTS image. Some folders may not have a carrier specific amss.mbn file. In which case, SwiQdl will automatically download the generic UMTS image. Each folder must contain at least one uqcn.mbn file but no more than one amss.mbn file.

3.2.2. Retry Mechanism

SwiQdl implements a retry mechanism which allows the SwiQdl to re-initiate the firmware downloading process when the previous download is out of sync or failed. The most common reason for download failure is when the modem detects an internal timeout on receiving data packets; the download process has to be terminated at that point.

The maximum number of retries can be configured through a registry key setting as follows:

[HKEY_LOCAL_MACHINE\Software\Sierra Wireless\QDL]

"MaximumRetries"=dword:<maximum number of retries>

The default setting for maximum retries is 2, not including the initial download. If this key is not populated in the registry, SwiQdl will set the maximum retries to 2 by default.

3.2.3. Firmware Image Restriction

OEMs may choose to prevent the user from switching to, or downloading specific firmware images via UpgradeFirmware2K API. By default, there is no restriction and the user is free to select any firmware image present on the host platform. To enable restriction, the following registry keys (one or both) can be specified:

[HKEY_LOCAL_MACHINE\Software\Sierra Wireless\QDL]

"Technology"=string:<technology>

"Carrier"=string:<carrierID>

Where:

technology is

0 – CDMA

1 – UMTS

and CarrierID is identical to Table 2-4 of the Gobi Connection Management API document. Example from table:

1 – Generic

2 – Factory

101 – Verizon

102 – Sprint

201 – AT&T

202 – Vodafone

203 – T-Mobile

204 – Orange

205 – Telefonica

The specified technology and/or carrier type(s) will be restricted. Multiple values can be specified and should be delimited by a space.

UpgradeFirmware2K will reject firmware images that match the restriction criteria with error code 895.

4. GobiApi

4.1. Multiple Client Support

GobiApi supports simultaneous usage by multiple applications (processes). Each application must initialize an instance of GobiApi DLL via procedures outlined in document 80-VF219-N. Multiplexing of application requests are handled by the driver.

4.2. Threading Model

GobiApi can process API requests from multiple threads simultaneously. The order of requests processed is NOT deterministic. That is, if thread ‘A’ issues a request before thread ‘B’, it is not guaranteed that thread ‘A’ will receive a response before thread ‘B’).

4.3. Memory management

GobiApi uses thread local storage to manage memory for each thread. When a worker thread first invokes an API function, memory is allocated—this memory is required to complete all API commands and is unique to each thread. The memory is automatically freed when the worker thread terminates. Figure 4-1 illustrates the recommended application threading model—deviating from this recommendation may result in memory leaks. For example, if GobiApi is unloaded from memory (FreeLibrary()) before the worker thread gracefully terminates, memory is not freed and will result in a memory leak (as shown in Figure 4-2 on page 19).

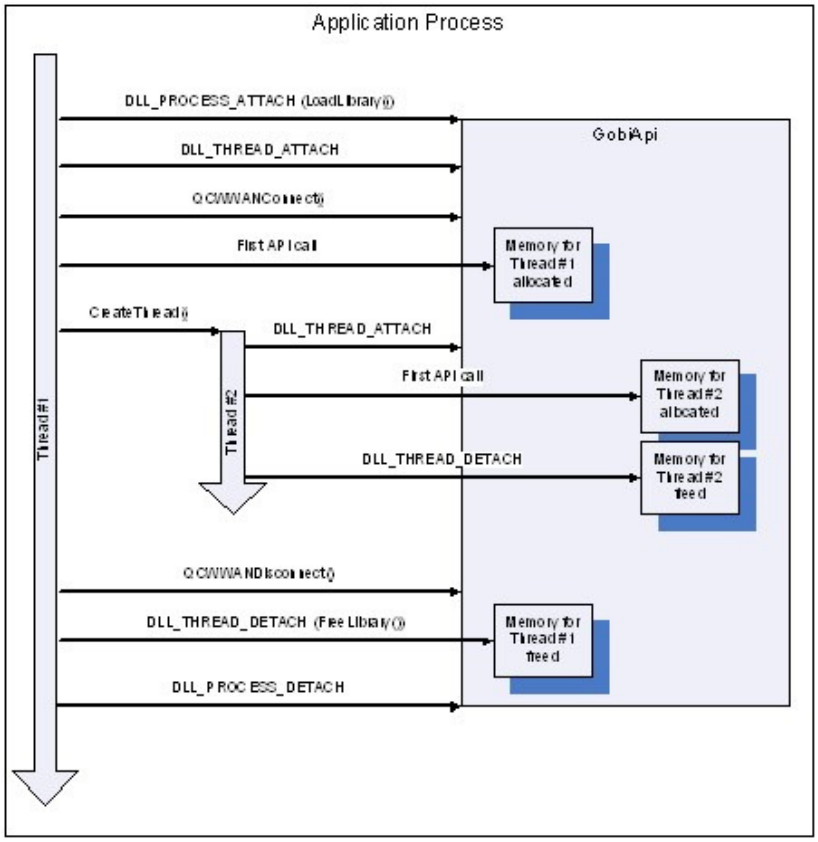


Figure 4-1: Recommended memory management threading model

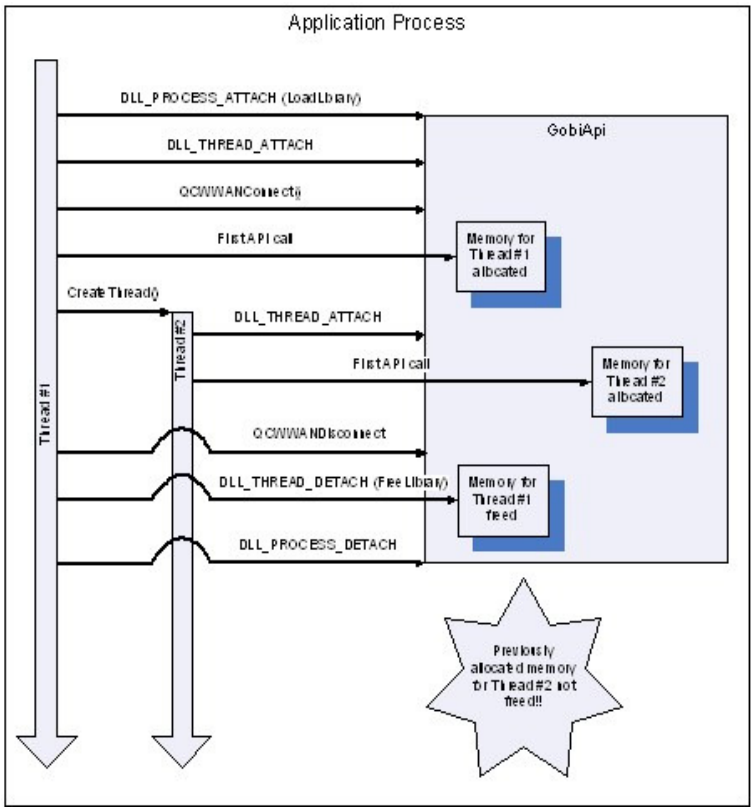


Figure 4-2:Memory leak

4.4. Notification Callback Handling

To receive unsolicited notifications, client applications must register callbacks with GobiApi. Callbacks are then invoked by a thread created and maintained by GobiApi. The same thread used to invoke callbacks is also responsible for processing all modem responses. Therefore, it is imperative that client applications release control of the callback thread immediately to avoid causing timeout errors for other threads invoking API functionality.

The following is the recommended procedure to handle callbacks:

1. Callback handler invoked by GobiApi DLL thread.
2. In the callback handler, cache a copy of relevant information provided by the callback interface.
3. Notify main application thread that notification has been received.
4. Return from callback handler to release control of GobiApi callback thread.

Note: The same thread used to invoke callbacks is also responsible for processing all modem responses. Client applications **must** release control of the callback thread immediately to avoid causing problems for other threads invoking API functionality.

5. Process notification in main application thread.

API calls invoked within the context of an API callback thread will be rejected (see SwiWwanCmApi.h for more information).

See SimpleCM for an example on handling notifications.

4.5. Error handling

GobiApi uses C++ exceptions (try/catch/throw) to handle errors that occur at any level in the processing of QMI packets.

- Each API function is invoked within a try/catch block.
- API functions invoke the throw statement when an error occurs.
- All thrown errors are handled gracefully at the top of the API stack. (Sierra Wireless has verified that the stack unwinds and correctly cleans up allocated stack resources.)

Note: Whenever the throw statement is invoked, a "Raised Exception" message appears in the WinCE debugger. This message is not fatal and can be safely ignored.

4.5.1. Error handling methodology rationale

The error handling method described above was chosen to address issues associated with the QMI packet structure.

Numerous conditional checks are required to validate and process an API transaction—each QMI packet consists of nested TLV structures (with potential for buffer overruns), and multiple packets may be needed to complete any given transaction.

Because the cleanup process is the same wherever an error occurs in a transaction, GobiApi is designed to eliminate duplication of conditional checks and error handlers by relying on the **catch** handler at the top of the stack to process error conditions. This design has the following advantages:

- High quality code—Conditional statements are more likely to contain errors than other statements. By eliminating conditional statements, the resulting code is more robust.
- Decreased time to market—Conditional statements are branch points that typically correlate to test cases that must be executed for white-box testing. By eliminating conditional statements, testing time is decreased.
- Reduced development costs—Bug identification, fixing, and testing are reduced with a simple flow control engine.

5. Logging

5.1. API

To enable GobiApi logging, apply the following registry keys before loading GobiApi:

```
[HKEY_CURRENT_USER\Software\ToolBoxDebug\GobiApi]
    "OutputToFile"=dword:00000001
    "OutputToDebug"=dword:00000001
    "MaxLogSize"=dword:001e8480
    "TraceLevelAll"="TBDOL_INFORMATION"
    "TBDCT_ALL_MEMORY"=dword:00000001
```

Log files are saved in the same directory as the application which loads GobiApi or in the root directory (depending on the platform). Log files use the following naming convention: GobiApi-<app name>.txt.

5.2. Drivers

Drivers use DEBUGMSG or RETAILMSG to output debug information. Use a debug serial port to capture this information. The debug version of each driver will generate extensive debug logging. HKLM\DebugZones can be used to throttle debug messages.

5.3. Firmware

PTClientApp can be used to collect firmware logs. Contact your Sierra Wireless representative for a copy of this tool.