



SIERRA
WIRELESS™

Gobi™ 3000 SDK
For Windows CE Platforms

Software Integration Guide

Revision:	Draft B
------------------	----------------

© 2003-2011 Sierra Wireless, Inc.

This document contains information which is proprietary and confidential to Sierra Wireless, Inc. Disclosure to persons other than the officers, employees, agents, or subcontractors of the Company or licensee of this document without the prior written permission of Sierra Wireless, Inc. is strictly prohibited.

QUALCOMM is a registered trademark of QUALCOMM Incorporated. Other product and brand names may be trademarks or registered trademarks of their respective owners. CDMA2000 is a registered certification mark of the Telecommunications Industry Association, used under license.

Document #:		Revision:	Draft B	Page 2 of 13
-------------	--	-----------	---------	--------------

Table of Contents

1	General	3
1.1	Purpose	3
1.2	Scope	3
1.3	Revision History.....	3
1.4	References	3
1.5	Terminology and Acronyms	4
2	Overview	5
2.1	System Requirements	5
2.2	SDK Installation.....	5
2.3	Folder Structure.....	5
2.3.1	Documents	5
2.3.2	Drivers.....	5
2.3.3	GobiApi.....	5
2.3.4	SimpleCM.....	6
2.3.5	SmsSample.....	6
3	Drivers.....	7
3.1	SwiGobi.....	7
3.1.1	Selective Suspend	7
3.1.2	TcpWindowSize.....	7
3.2	SwiQdl.....	7
3.2.1	Firmware Image Folder Structure.....	8
3.2.2	Retry Mechanism	9
3.2.3	Firmware Image Restriction	9
4	GobiApi.....	11
4.1	Multiple Client Support.....	11
4.2	Threading Model.....	11
4.3	Notification Callback Handling.....	11
5	Logging	12
5.1	API	12
5.2	Drivers	12
5.3	Firmware.....	12
6	Known Issues	13

Document #:		Revision:	Draft B	Page 3 of 13
-------------	--	-----------	---------	--------------

1 General

1.1 Purpose

This document contains a collection of references and development notes to aid in the software integration of Sierra Wireless® Gobi™ 3000 modules on Microsoft® Windows operating systems for embedded platforms. OEMs and third party application developers are the intended audience of this document.

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 Revision History

Revision	Date	Summary
Draft A	2010-11-24	Initial draft
Draft B	2011-01-14	Add firmware image restriction feature

1.4 References

The following table lists useful Qualcomm references for SDK integrators. Contact your Sierra Wireless representative for a copy of the references; they are not included in the Sierra Wireless Gobi WinCE SDK. Some are available for download from www.gobianywhere.com.

ID	Title	Comments
CL93-V3077-1	Application Note: Software Glossary for Customers	A list of acronyms and their definitions compiled by Qualcomm

SIERRA WIRELESS, INC.

Document #:		Revision:	Draft B	Page 4 of 13
-------------	--	-----------	---------	--------------

1.5 Terminology and Acronyms

See CL93-V3077-1 for a complete list of acronyms and definitions from Qualcomm.

Term	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

Document #:		Revision:	Draft B	Page 5 of 13
-------------	--	-----------	---------	--------------

2 Overview

2.1 System Requirements

See release notes for a list of supported development configurations. Contact your Sierra Wireless representative for a copy of the Sierra Wireless Windows CE Standard SDK 6.0. Other Platform SDKs can be downloaded from Microsoft. Third party software developers will need to contact OEMs for specific SDKs unique to their platform.

Sample applications are produced using Visual Studio 2005 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 Documents

This folder contains API documentation and integration guides.

The following is a list of key components included in this folder:

Component	Description
GobiConnectionMgmtAPI.pdf	Connection management API
Gobi 3000 SDK for Windows Embedded OEM.pdf	Integration notes

2.3.2 Drivers

This folder contains the driver installer, binaries, and INF files.

The following is a 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
DriverPackage.cab	Driver installation package

2.3.3 GobiApi

This folder contains WWAN connection manager APIs.

The following is a list of key components included in this folder:

Component	Description
------------------	--------------------

SIERRA WIRELESS, INC.

Document #:		Revision:	Draft B	Page 6 of 13
-------------	--	-----------	---------	--------------

QCWWANCMAPI2k.h	Qualcomm wireless WWAN connection manager API
SwiWwanCmApi.h	Sierra Wireless API extensions for Gobi 3000
GobiApi.dll	Implements connection manager APIs

2.3.4 SimpleCM

This folder contains source code for a sample connection manager. Basic data connection functionality, notification callback handling, and device detection are demonstrated.

2.3.5 SmsSample

This folder contains source code for an SMS sample application which utilizes Sierra Wireless SMS helper functions. SMS send, receive and handling of concatenated messages are demonstrated.

Document #:		Revision:	Draft B	Page 7 of 13
-------------	--	-----------	---------	--------------

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 WMP9. 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 WMP5.
- GPS NMEA port is available on port WMP6.

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 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 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

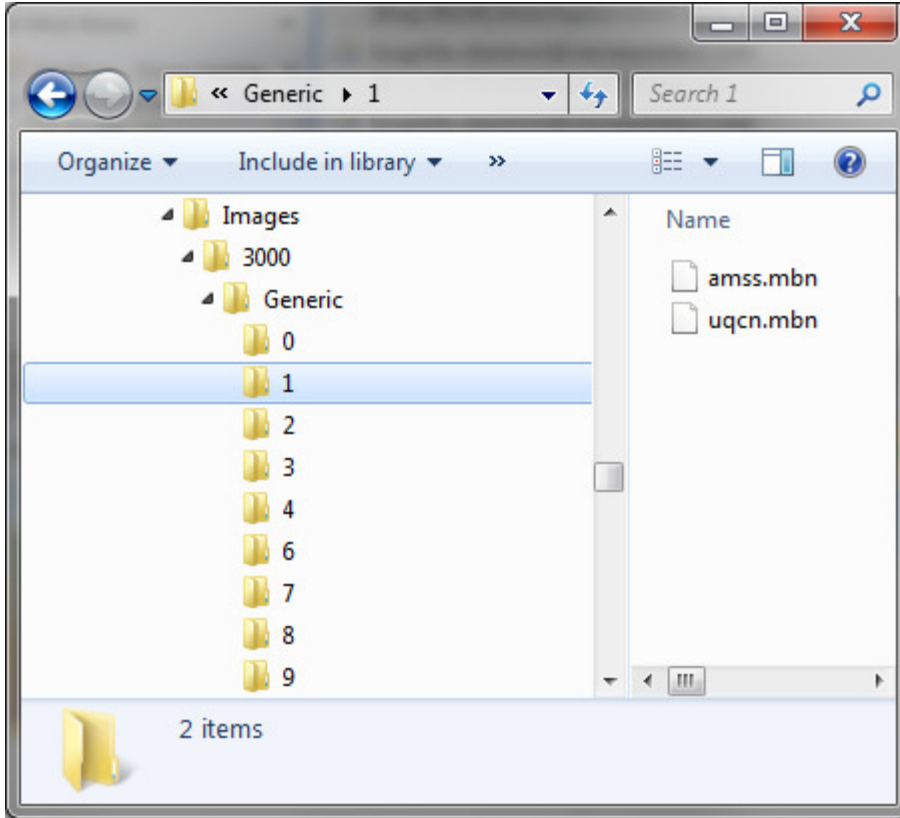
SwiQdl.dll implements firmware download functionality.

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

Document #:		Revision:	Draft B	Page 9 of 13
-------------	--	-----------	---------	--------------

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-initiates the firmware downloading process when the previous download is out of sync or failed. The most common reason for download failure is that 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 the following.

```
[HKEY_LOCAL_MACHINE\Software\Sierra Wireless\QDL]
    "MaximumRetries"=dword:<maximum number of retries>
```

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

3.2.3 Firmware Image Restriction

OEM 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 images present on the host platform. To enable the 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

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

SIERRA WIRELESS, INC.

Document #:		Revision:	Draft B	Page 10 of 13
-------------	--	-----------	---------	---------------

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 which match the restriction criteria with error code 895.

Document #:		Revision:	Draft B	Page 11 of 13
-------------	--	-----------	---------	---------------

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.

4.3 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
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.

Document #:		Revision:	Draft B	Page 12 of 13
-------------	--	-----------	---------	---------------

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.

SIERRA WIRELESS, INC.

Document #:		Revision:	Draft B	Page 13 of 13
-------------	--	-----------	---------	---------------

6 Known Issues
