



Sierra Wireless

Location Based Services (CDMA)



SIERRA
WIRELESS™

2130907
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.

Copyright

© 2012 Sierra Wireless. All rights reserved.

Trademarks

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

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 |
| | E-mail: | sales@sierrawireless.com |
| RMA: | repairs@sierrawireless.com | |
| Post: | Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4 | |
| Fax: | 1-604-231-1109 | |
| Web: | 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

Revision History

| Revision number | Release date | Changes |
|-----------------|--------------|------------------------|
| 1.0 | August 2007 | Initial release |
| 1.1 | April 2008 | Added MC5727 Mini Card |

| Revision number | Release date | Changes |
|-----------------|--------------|--|
| 1.2 | June 2009 | <p>Added:</p> <ul style="list-style-type: none"> • MC5727V and MC5728V Mini Cards. • “Smart Mode” (page 15). • Query satellite information: page 33 (API), page 52 (CnS), page 63 (AT). • Query base station information: page 33 (API), page 52 (CnS), page 64 (AT). • Enable the NMEA stream: page 58 (AT). • Clear assistance data: page 30 (API), page 51 (CnS), page 61 (AT). • “gpsOne, 7-th generation engine” (page 87) • Index (page 95). • List of Tables. • Modem must be unlocked before a successful query with AT commands !GPSIPADDR and !GPSPORTID. <p>Removed references to adapter products (“AirCard” PC Cards or ExpressCards, USB modems).</p> <p>Adjusted “Setting up Watcher” (page 77) to reflect the current GUI.</p> <p>Renamed:</p> <ul style="list-style-type: none"> • SwiGetPdsm* functions to SwiGetLbs* • SwiSetPdsm* functions to SwiSetLbs* • SWI_NOTIFY_PDSM* notifications to SWI_NOTIFY_Lbs* <p>“GPS fix modes” (page 13) now lists six modes.</p> <p>Miscellaneous formatting changes.</p> <ul style="list-style-type: none"> • Added hyperlinks to glossary items and sections that describe the fix modes. |
| 2 | August 2012 | <ul style="list-style-type: none"> • Removed products that have reached end-of-life (MC 5725, 5725V, 5727, 5727V). Removed related sections “Getting or setting the transportation mechanism for location processing: AT!GPSPTLM command”, “Getting or setting the privacy level for location processing: AT!GPSPRIV command”, and “Getting or setting the network access value for location processing: AT!GPSNETACC command”. • Updated to new template. • Removed Patent information. |



Contents

| | |
|---|-----------|
| Location Based Services Primer | 13 |
| GPS fix modes | 13 |
| Standalone | 13 |
| Conditions for use | 13 |
| MS-based | 14 |
| Conditions for use | 14 |
| Enhanced observed time differential | 14 |
| Assisted GPS | 14 |
| Conditions for use | 15 |
| Advanced Forward Link Trilateration | 15 |
| Speed Optimal | 15 |
| Accuracy Optimal | 15 |
| Data Optimal | 15 |
| Smart Mode | 15 |
| MS-based Smart Mode | 15 |
| Standalone Smart Mode | 16 |
| gpsOne | 16 |
| Position determination entity | 17 |
| Wireless enhanced 911 services | 17 |
| Fix initiation | 17 |
| Control plane | 17 |
| User plane | 18 |
| Secure User Plane Location | 18 |
| Continuous tracking | 18 |
| | |
| Supported Applications | 19 |
| Device-initiated applications | 19 |
| Network-initiated applications | 19 |
| E911 | 20 |

| | |
|--|-----------|
| LBS uses | 20 |
| Laptop local | 20 |
| Laptop served from network | 20 |
| Enterprise served from network | 21 |
| Common software applications | 21 |
| Watcher | 21 |
| Microsoft Streets and Trips | 21 |
| Software Interface Options | 23 |
| Software interface comparison | 23 |
| API use cases | 25 |
| Opening and unlocking the modem: SwiSetLbsPaGpsLock API | 25 |
| Using APIs to request the location of the modem | 26 |
| Requesting the last known location: SwiGetLbsPdData API | 26 |
| Requesting a single location fix: SwiSetLbsPdGetPos API | 26 |
| Requesting frequent tracking: SwiSetLbsPdTrack API. | 28 |
| Clear assistance data: SwiSetLbsClearAssistance API | 30 |
| Downloading ephemeris and almanac data: SwiSetLbsPdDownload API | 31 |
| Query satellite information: SwiGetLbsSatInfo API | 33 |
| Getting base station information for location processing: SwiGetLbsPaBsInfo API | 33 |
| Terminating a position determination session: SwiSetLbsPdEndSession API | 34 |
| Terminating a location fix session: SwiSetLbsPdEndSession API | 34 |
| Terminating a download session: SwiSetLbsPdEndSession API | 35 |
| Requesting default position determination parameters: SwiGetLbsPaParam API | 35 |
| Using APIs to modify default position determination parameters | 37 |
| Modifying the transport mechanism parameter: SwiSetLbsPaPtlmMode API | 37 |
| Modifying the privacy parameter: SwiSetLbsPaPrivacy API. | 38 |
| Modifying the network access parameter: SwiSetLbsPaNetAccess API | |

| | |
|---|----|
| 38 | |
| Error conditions for location fixes | 38 |
| API FAQ | 40 |
| Configuring the modem and developer's kit for LBS | 42 |
| Software components | 42 |
| Modem setup | 42 |
| Using CnS commands | 43 |
| CnS message interface | 43 |
| Query, response, and notification | 43 |
| Limitations | 44 |
| Types of LBS clients | 45 |
| Enabling notifications: Location notification status and Download notification status objects | 46 |
| Opening and unlocking the modem: GPS Lock object | 46 |
| Getting the current status of location based services: PDSM Status object | 46 |
| Requesting the location of the modem | 49 |
| Requesting the last known location: Position Fix Data object | 49 |
| Requesting a single location fix: PD Get Position object | 49 |
| Requesting frequent tracking: PD Track object | 50 |
| Clear assistance data: Clear GPS Assistance Data object | 51 |
| Downloading ephemeris and almanac data: PD Download object | 51 |
| Query satellite information: Satellite information object | 52 |
| Getting base station information for location processing: Base Station information object | 52 |
| Terminating a position determination session | 52 |
| Terminating a location fix session: End Session object | 52 |
| Terminating a download session: End Session object | 52 |
| Accessing default position determination parameters: PA Parameter Settings object | 53 |
| Accessing individual default position determination parameters | 53 |
| Modifying default position determination parameters | 54 |

| | |
|---|-----------|
| Using AT commands | 54 |
| Query and response | 55 |
| Enabling notifications | 55 |
| Opening and unlocking the modem: AT!GPSLOCK command | 55 |
| Getting the current status of location based services: AT!GPSSTATUS command | 56 |
| Enabling the NMEA stream: AT!GPSNMEAENABLE command | 58 |
| Requesting the location of the modem | 58 |
| Requesting the last known location: AT!GPSLOC command | 58 |
| Requesting a single location fix: AT!GPSFIX command | 59 |
| Requesting frequent tracking: AT!GPSTRACK command | 60 |
| Clear assistance data: AT!GPSCLRASSIST command | 61 |
| Downloading ephemeris and almanac data: AT!GPSDLOAD command | 62 |
| Query satellite information: AT!GPSSATINFO command | 63 |
| Getting base station information for location processing: AT!GPSBSINFO command | 64 |
| Terminating a location fix or download session: AT!GPSEND command | 64 |
| Accessing default position determination parameters | 65 |
| Getting or setting the IP address: AT!GPSIPADDR command | 65 |
| Getting or setting the port ID for location processing: AT!GPS PORTID command | 66 |
| Placing the modem in diagnostic mode: AT!DIAG command | 67 |
| NMEA interface | 67 |
| NMEA port configuration | 67 |
| Controlling data transmission | 68 |
| NMEA sentences | 68 |
| Sentence one: \$GPGGA Global Positioning System Fix Data | 68 |
| Sentence two: \$GPGSA, GPS dilution of precision and active satellites 69 | |
| Sentence three: \$GPGSV, satellites in view | 70 |
| Sentence four: \$GPRMC, recommended minimum specific GPS/transit data | 71 |
| Sentence five: \$GPVTG, track made good and ground speed | 71 |
| Testing | 73 |
| Testing for accuracy | 74 |

| | |
|--|-----------|
| Testing for yield | 74 |
| Testing for latency | 74 |
| Standard test equipment | 75 |
| Modem Setup | 77 |
| Setting up Watcher. | 77 |
| Setting up HyperTerminal for NMEA interface operation | 78 |
| Setting up HyperTerminal to run AT commands | 81 |
| Modem Integration Notes | 85 |
| Privacy control | 85 |
| LBS usage interruptions | 85 |
| Incoming call interruptions | 86 |
| Outgoing call interruptions | 86 |
| Automatic and original equipment manufacturer-required actions | 86 |
| Configuration | 86 |
| Behavior during operation | 87 |
| Modem call flows and messages | 87 |
| Data returned | 87 |
| gpsOne. | 87 |
| gpsOne, 7-th generation engine. | 87 |
| Resources | 89 |
| Glossary and Acronyms | 91 |
| Index | 95 |

List of Tables

| | |
|---|----|
| Software interface comparison..... | 23 |
| Modem lock parameters | 25 |
| Location parameters for single location fixes..... | 27 |
| Location parameters for frequent tracking | 29 |
| Parameters for downloading ephemeris and almanac data..... | 32 |
| Location fix termination parameters | 34 |
| Location parameters..... | 36 |
| Error conditions for location fixes..... | 38 |
| CnS commands used in PDSM transactions | 44 |
| GPS lock status settings | 46 |
| Current status information of Location Based Services | 47 |
| GPS lock settings | 55 |
| Location based services status values..... | 56 |
| Location based services failure codes..... | 57 |
| NMEA settings..... | 58 |
| Parameters for single location fixes..... | 60 |
| Fix parameters for frequent tracking | 61 |
| Fix parameters for clearing assistance data..... | 62 |
| Download parameters | 63 |
| Session parameters | 64 |
| Sentence elements for sentence one..... | 68 |
| Sentence elements for sentence two..... | 69 |
| Sentence elements for sentence three | 70 |
| Sentence elements for sentence four | 71 |
| Sentence elements for sentence five..... | 71 |

1: Location Based Services Primer

1

- GPS fix modes
- Smart Mode
- gpsOne
- Wireless enhanced 911 services
- Fix initiation

Location based services (LBS) is a global positioning system (GPS) location feature supported by some carriers and networks.

This chapter provides an overview of LBS, and details on Sierra Wireless' implementation of it.

GPS fix modes

There are six modes for GPS fixes:

- [Standalone](#) (described below)
- [MS-based](#) (page 14)
- [Assisted GPS](#) (page 14)
- [Speed Optimal](#) (page 15)
- [Accuracy Optimal](#) (page 15)
- [Data Optimal](#) (page 15)

Typically, upon invocation of LBS in the Sierra Wireless modem, and based on specified parameters, the LBS engine chooses the appropriate fix method. LBS software interfaces also allow for invocation of specific modes.

Note: The modem supports all six modes; however, due to carrier restrictions, certain modes may be unavailable. For details, contact Sierra Wireless.

Standalone

A standalone (or mobile-autonomous positioning) fix is based only on information from GPS satellites and does not incorporate a network or a position determination entity (PDE) for location determination.

Conditions for use

Standalone fixes do not require code division multiple access (CDMA) coverage. However, they do require a clear line of sight to the satellite and are therefore ineffective for use indoors or in any environment where the line of sight to a satellite is obstructed. This mode is used for continuous tracking.

MS-based

Mobile station-based location **fixes** make use of the CDMA network to download assistance data when needed (for example, cold start, or stale/incomplete). If the assistance data is current, the device operates autonomously—without communicating with the **PDE**, and the device itself computes the GPS position.

Enhanced observed time differential (**EOTD**) is an example of an **MS**-based mode.

Conditions for use

MS-based tracking requires **CDMA** coverage. This mode is typically used for continuous tracking. Coverage is generally poor indoors and in rural areas where base stations do not overlap. Coverage also may be poor where there are terrestrial obstructions likely to result in multi-path interference. MS-based tracking requires simultaneous visibility to three or more satellite vehicles and higher transmit power to overcome interference.

Enhanced observed time differential

EOTD is one method for calculating position location in a terrestrial-based LBS system. It uses a triangulation method with multiple base stations to determine how far away a handset is from a tower by timing the signal. Initially rolled out to satisfy **E911** requirements, EOTD is a popular solution in GSM applications, where QUALCOMM's **gpsOne** solution is not widespread. In an EOTD system, devices support positioning in a network where base stations are asynchronous. An example of this implementation is a device reporting measured times from three base stations. The times are then combined with timing data from the network to determine the location of the device.

Assisted GPS

An assisted GPS (**AGPS**) **fix** is based on information from satellites and a mobile network. This method uses an assistance server, or a **PDE**. The PDE provides GPS data (referred to as assistance data) to the device, and the device can either derive a location solution based on that data, or the PDE can derive the location solution based on data supplied by the device. The network supplies a location signal which is combined with satellite data. The satellite signal is sent to the PDE and is combined with signal strength measurements relative to a tower to position. Advanced Forward Link Trilateration (**AFLT**) is an example of an **MS**-assisted mode.

AGPS supports static navigation and network-initiated applications. Because this method relies on GPS satellites and a mobile network, it takes less time to determine a position than required for a standalone fix.

Conditions for use

AGPS requires [CDMA](#) coverage. This method provides the most accurate tracking in a variety of environments because it makes use of weaker signals. Triangulation with cell-site locations also makes it possible to provide a location indoors and in other locations without a direct line of sight to a satellite.

Advanced Forward Link Trilateration

AFLT is a location determination solution that uses [CDMA](#) pilot phase measurements in its final calculation. AFLT measures the phase delay between signals sent to two base stations and then compares this information to information taken from two other base stations.

Speed Optimal

This mode uses either an [MS-based](#) or [Assisted GPS](#) fix based on the fastest possible time to fix. The device determines which fix type to attempt based on fix rate and initial uncertainties.

Accuracy Optimal

This mode uses an [Assisted GPS](#) fix first, and then an [MS-based](#) fix if the specified level of accuracy cannot be achieved by the AGPS fix.

Data Optimal

This mode uses an [MS-based](#) fix first, and then an [Assisted GPS](#) fix. The device attempts to minimize the amount of data exchanged over the network.

Smart Mode

Also known as “GPS dynamic mode”, this feature increases the likelihood of successfully obtaining the location position by using a fall back method. “Fall back” means that, if the original fix method fails, the modem retries obtaining the fix by using the alternative method.

When enabled, this feature can be configured so that:

- The modem falls back to [Standalone](#) mode if an [MS-based fix](#) attempt fails. (“[MS-based Smart Mode](#)”; page 15)

The Smart Mode settings are enabled via an [NV](#) item in [PRI](#).

MS-based Smart Mode

If an [MS-based fix](#) is requested and assistance data is needed but the [PDE](#) is not available, then the MS-based Smart Mode setting causes the modem to fall back to a [Standalone](#) fix to demodulate the assistance data from the satellites directly.

Had the modem failed to contact the PDE, the fix attempt would have failed. Instead, the fall back allows the modem to continue processing the fix request. Hence, a "smart" MS-based solution.

The PDE may be unavailable for any of the following reasons:

- The modem may be out of CDMA network coverage.
- The PDE IP Address/Port ID used for authentication is incorrect.
- The network or the PDE is temporarily down.
- The network does not have a PDE (more likely with smaller carriers).

For more information on the PDE, see [Position determination entity](#) on page 17.

One of the primary advantages of enabling MS-based Smart Mode is it provides a common configuration to allow both activated and unactivated modems to initiate a fix with a reasonable chance of returning a successful position. If the modem is **unactivated**, then the PDE contact is impossible, but the fall back allows the modem to do a [Standalone](#) search in hopes of obtaining a successful position. If the modem is **activated**, then the modem can use the PDE to download the assistance data to improve the TTFF (Time To First Fix). If however, the activated modem is out of CDMA network coverage, then the fall back allows the modem to have a chance of returning a standalone fix if conditions allow (clear sky view, sufficient number of satellites, and sufficient signal strength).

Standalone Smart Mode

If a [Standalone fix](#) is requested and the modem needs assistance data (due to a cold start fix or stale/incomplete data), then the Standalone Smart Mode setting causes the modem to fall back to an [MS-based](#) call flow to download the missing assistance data from the [PDE](#). This would improve the [TTFF](#) compared to demodulating assistance data from the satellites directly. Hence, a "smart" standalone solution.

gpsOne

Sierra Wireless' implementation of LBS features is built upon the capabilities of [gpsOne™](#).

gpsOne is a QUALCOMM wireless-assisted technology that uses data from both GPS satellites and [CDMA](#) networks to provide high availability and high accuracy across a wide variety of environments. The gpsOne solution incorporates both [MS-based](#) and [Assisted GPS](#) approaches.

Originally deployed to meet requirements for [E911](#) location, gpsOne technology allows emergency dispatchers to identify the location of emergency calls to within 5 to 50 meters in most environments, and with industry precision indoors.

The ability to make position-location measurements is embedded directly into the module wireless chipset, thus eliminating the need for a separate, more expensive GPS-only receiver. Because gpsOne technology is an integrated solution, it provides the lowest-cost GPS-based technology available for wireless devices.

The gpsOne architecture includes both device and network components. GPS receivers detect signals from a constellation of more than twenty-four GPS satellites operated by the United States Air Force. These signals are decoded in the mobile device, and the network (**PDE**) or the modem calculates the latitude and longitude of the device. For more information about PDEs, see [Position determination entity](#) below.

Position determination entity

A position determining entity is deployed in the carrier network in a mated pair configuration. The **PDE** assists the GPS receiver inside the mobile device in two ways:

- On request, the PDE sends assistance data ([almanac data](#) and [ephemeris data](#)) to the device. This allows the device to search only for GPS satellites that should be visible to it, thus reducing the time required for the GPS receiver to perform the first location **fix**. Subsequent location fixes occur much faster, because the receiver is then “tuned” to visible satellites.
- The PDE may also assist in the calculation of latitude and longitude. The PDE has access to the Carrier Base Station Almanac (BSA), allowing the PDE to calculate the location of a device by triangulating its location among the base stations that are visible to the device. This technique is known as **AFLT**. The PDE may calculate the latitude and longitude of a device exclusively through AFLT, or it may perform a hybrid calculation based on AFLT in conjunction with processing raw GPS data sent from the device.

Wireless enhanced 911 services

The Federal Communication Commission's wireless Enhanced 911 (**E911**) requirements were instituted to make it easier to determine the location of wireless callers making of emergency calls by providing emergency dispatchers with location information. Phase I of the requirements mandated that, at the request of a local Public Safety Answering Point (PSAP), carriers be able to provide the wireless telephone number originating an emergency call and the location of the mobile antenna that received the call. Phase II added the requirement that carriers be able to provide the origin of emergency calls with a range of accuracy of 50 to 300 meters.

Fix initiation

GPS **fixes** can be initiated by a network or a device user. Control plane fixes are initiated by the network, whereas user plane fixes are initiated by the device user with the help of a mapping application.

Control plane

Designed to provide highly accurate, high availability emergency location, control plane initiation enables network-initiated queries for device location coordinates. Control plane data is sent on the control channels of the wireless network.

User plane

User plane initiation enables subscribers to initiate location based services using various GPS applications. A location server communicates directly with the mobile device via an IP (Internet Protocol) link integrated into the mobile communication network.

Secure User Plane Location

Secure User Plane Location (SUPL) is a standard developed by the Open Mobile Alliance (OMA) to enable LBS on any [AGPS](#)-capable phone on any mobile network.

Continuous tracking

In continuous tracking mode, an application requires multiple location [fixes](#) over a small period of time (a fix between from every 1 to 240 seconds.) Assuming that the [ephemeris data](#) and [almanac data](#) are valid (based on a location fix within the previous 4 minutes), the modem will first attempt to obtain [MS-based](#) fixes using GPS data only. If the uncertainty of the GPS location fix is greater than the specified value, or the coarse location data is invalid, then an [Assisted GPS](#) fix may be necessary. The application can specify how many fixes should be done and at what rate they should occur.

Because of the potential need for updating the [ephemeris data](#), [almanac data](#) or location data, it may require more time to achieve a first fix than to achieve subsequent fixes. Tracking applications must consider this possible initial delay, and may want to behave proactively by initiating a data download session or a single location fix as part of their startup sequence.

Note: The [almanac data](#) is valid for up to 6 months, [ephemeris data](#) is valid for 30 to 120 minutes, and coarse location data is valid for 4 minutes.

2: Supported Applications

- Device-initiated applications
- Network-initiated applications
- LBS uses
- Common software applications

There are two types of LBS applications: device-initiated and network-initiated.

Device-initiated applications

Device-initiated applications are launched by the subscriber from the mobile device. Device-initiated applications fall into the following categories:

- **Mapping applications** provide graphical map views and general map manipulation
- **POI (point of interest) applications** provide search or browse capability to locate nearby points of interest such as restaurants, teller machines, movie theatres, parks, and so on.
- **Navigation or turn-by-turn (TBT) applications** provide static directions or real-time TBT driving directions
- **Finder applications** locate people from a mobile device
- **Other** applications are emerging rapidly. Examples are location-aware blogging and tracking services for runners and cyclists

Network-initiated applications

Network-initiated applications reside outside the carrier network and seek to obtain location updates for devices operating on the carrier network. Most applications offered by wholesale carrier partners fall into this category. These applications are enabled by a location proxy server (LPS). In a typical carrier architecture, the LPS contains a user privacy center to authenticate location requests.

Typical network-initiated applications include:

- Vehicle tracking
- Field force automation (FFA)
- Sales force automation (SFA)

The LPS serves two purposes:

- **Privacy management and access control:** The LPS allows subscribers to configure individual privacy profiles that restrict which applications and services may access their location. The LPS controls access to location data by both device-initiated and network-initiated applications. Device-initiated applications must

authenticate themselves to the LPS prior to performing a location [fix](#). Network-initiated applications, such as a tracking application, must login to the LPS and authenticate themselves, and then may proceed with a location look-up only if the subscriber has explicitly given their permission to be located by that application.

- **Geo-services and geo-data:** The LPS provides common geographic information services (GIS) and data to LBS applications. Geo-data includes maps, point-of-interest information, and driving routes. Geo-services include geocoding (translating a street address into latitude/longitude), reverse geocoding, route planning, and other geo-spatial calculations. Carriers require that all device-initiated applications use this data to ensure that consistent, high-quality maps and mapping services are provided to subscribers. (Use by network-initiated applications is optional.)

A network-initiated application submits a location request to the LPS. After the application is authenticated and the end user's permission is verified, the LPS sends a message to the end user's device that invokes a host application called the Location Manager. The Location Manager, which trusts all messages from the LPS, immediately performs a location fix and returns the result to the LPS as soon as it is available.

E911

Requirements are based on FCC [E911](#) mandate requirements, which were designed to improve the reliability of wireless 911 services and to provide emergency services personnel with location information more quickly.

LBS uses

LBS can be implemented in several ways:

- Laptop local
- Laptop served from network
- Enterprise served from network

Laptop local

One laptop local application of LBS is identifying your current location on a map. (Microsoft Streets & Trips is one example.) This makes it possible to find nearby points of interest.

Geocaching is another popular LBS application. Geocaching is similar to treasure hunting based on GPS coordinates.

Laptop served from network

One example of a laptop served from a network application is locating children or friends to ensure safety or arrange a meeting.

Enterprise served from network

An example of an enterprise served from network application is locating employees during business hours and locating equipment.

Common software applications

Commonly integrated applications include Watcher™ from Sierra Wireless, and Microsoft Streets and Trips.

Watcher

Watcher allows you to manage and monitor the connection between the modem and the **CDMA** network. Watcher is used to:

- Determine your signal strength, roaming status, 1X high-speed data availability, and other network connection parameters
- Initiate voice and data calls
- View call statistics
- Receive and send SMS messages
- Customize features and options

Without running Watcher, you can make a high-speed data connection simply by launching a web browser or e-mail application. However, to make use of other features of the Watcher software, you must run Watcher.

For information on setting up Watcher to run in standalone mode, see [Setting up Watcher](#) on page 77.

Microsoft Streets and Trips

Compatible with Sierra Wireless modems, Microsoft Streets and Trips is comprehensive trip planning software. Features include points of interest location, and turn by turn driving directions. Refer to Microsoft's documentation for information on integrating Microsoft Streets and Trips.

3: Software Interface Options

- Software interface comparison
- API use cases
- Using CnS commands
- Using AT commands
- NMEA interface

There are four options for interfacing with the LBS capability provided by Sierra Wireless modules:

- Application programming interfaces (part of the Sierra Wireless host SDK) ([page 25](#))
- CnS commands ([page 43](#))
- AT commands ([page 54](#))
- NMEA interface ([page 67](#))

Either CnS or AT commands can be used to configure the modem for LBS services, while NMEA is intended as a read-only output for applications that require NMEA format. Location information will also flow back by CnS or AT commands, which are command and response protocols.

Software interface comparison

[Table 3-1](#) provides a comparison of software interface options for LBS functionality.

Table 3-1: Software interface comparison

| Function | API | CnS | AT |
|---|--|--|--|
| Open and unlock the modem | SwiSetLbsPaGpsLock (page 25) | 0x0F14, GPS Lock (page 46) | AT!GPSLOCK (page 55) |
| Enable notifications | N/A | <ul style="list-style-type: none"> • 0x0F707, Location notification status • 0x0F08, Download notification status • 0x0F09, Parameter notification status (page 46) | N/A |
| Get current status of location based services | N/A | 0x0F03, PDSM Status (page 46) | AT!GPSSTATUS (page 56) |
| Enable NMEA stream | N/A | N/A | AT!GPSNMEAENABLE (page 58) |

Table 3-1: Software interface comparison (Continued)

| Function | API | CnS | AT |
|--|--|--|---|
| Request the last known location of modem | SwiGetLbsPdData (page 26) | 0x0F0A, Position Fix Data (page 49) | AT!GPSLOC (page 58) |
| Request a single location fix | SwiSetLbsPdGetPos (page 26) | 0x0F02, PD Get Position (page 49) | AT!GPSFIX (page 59) |
| Frequent tracking of the modem host | SwiSetLbsPdTrack (page 28) | 0x0F04, PD Track (page 50) | AT!GPSTRACK (page 60) |
| Clear assistance data | SwiSetLbsClearAssistance (page 30) | 0x0F20, Clear GPS Assistance Data (page 51) | AT!GPSCLRASSIST (page 61) |
| Download ephemeris data and almanac data | SwiSetLbsPdDownload (page 31) | 0x0F01, PD Download (page 51) | AT!GPSDLOAD (page 62) |
| Query satellite information | SwiGetLbsPaBsInfo (page 33) | 0x0F1F, Satellite Information (page 52) | AT!GPSSATINFO (page 63) |
| Query base station information | SwiGetLbsSatInfo (page 33) | 0x0F19, Base Station information (page 52) | AT!GPSBSINFO (page 64) |
| Terminate a location fix session | SwiSetLbsPdEndSession (page 34) | 0x0F05, PD End Session (page 52) | AT!GPSEND (page 64) |
| Terminate a download session | SwiSetLbsPdEndSession (page 35) | 0x0F05, PD End Session (page 52) | AT!GPSEND (page 64) |
| Access the default position determination parameters | <ul style="list-style-type: none"> Transport mechanism: SwiSetLbsPaPtlmMode Privacy: SwiSetLbsPaPrivacy Network access permissions: SwiSetLbsPaNetAccess IP Address: SwiGetLbsPaIpAddr Port ID: SwiGetLbsPaPortId Lock: SwiGetLbsPaGpsLock (page 35) | <ul style="list-style-type: none"> Retrieve all: 0x0F06, PA Parameter Settings. Returns IP Address, Port ID, GPS Lock, Transport Mechanism, Privacy, and Network Access Access parameters individually using 0x0F13, TCP/IP Address; 0x0F14, GPS Lock; 0x0F15, Transport Mechanism; 0x0F16, Port ID; 0x0F17, Privacy Level; and 0x0F18, Network Access Permissions (page 53) | <ul style="list-style-type: none"> TCP/IP address: AT!GPSIPADDR (page 65) Port ID: AT!GPSPORTID (page 66) GPS lock: AT!GPSLOCK (page 55) |

API use cases

This section demonstrates the application programming interface (API) with use cases for using the sample application (SWI LBS Sample) to open and unlock the modem, request the location of the modem, download [ephemeris data](#) and [almanac data](#), terminate a position location session, and access and modify default position determination parameters.

Opening and unlocking the modem: SwiSetLbsPaGpsLock API

The following code sample demonstrates using the SwiSetLbsPaGpsLock API to unlock the modem before running GPS fixes.

Warning: Before opening and unlocking the modem, refer to carrier LBS requirements for fix frequency restrictions.

```
SWI_STRUCT_PDSM_PA_Gps_Lock    SetLock;

    SetLock.sizeStruct= sizeof
    ( SWI_STRUCT_PDSM_PA_Gps_Lock );

    SetLock.iGpsLock =
    SWI_PDSM_PA_No_GPS_Lock_Active;

    rCode = SwiSetLbsPaGpsLock(&SetLock, TIMEOUT);

    if (SWI_RCODE_OK != rCode)
    {
    ProcessError(rCode);
    }
```

Note: For a more detailed discussion of the CDMA 1xEV-DO API, refer to the CDMA API Reference (document 2131051).

[Table 3-2](#) lists and describes modem lock parameters.

Table 3-2: Modem lock parameters

| Parameter | Value | Description |
|-----------|-------|--|
| GPS Lock | 0 | Modem is unlocked |
| | 1 | Mobile-initiated sessions are not allowed |
| | 2 | Mobile-terminated sessions are not allowed |
| | 3 | E911 only allowed |

See [Table 3-8](#) on page 38 for a list of error conditions.

Using APIs to request the location of the modem

You can request the last known location or a single location [fix](#).

Requesting the last known location: SwiGetLbsPdData API

The following code sample demonstrates using the SwiGetLbsPdData API to request the last known location. Note that no [notifications](#) are generated.

```
SWI_STRUCT_PDSM_PD_Data  LastKnownLoc;
LastKnownLoc.sizeStruct= sizeof(SWI_STRUCT_PDSM_PD_Data);
m_rCode = SwiGetLbsPdData(&LastKnownLoc, TIMEOUT);
if (SWI_RCODE_OK != m_rCode)
{
    ProcessError(m_rCode);
}
```

See [Table 3-8](#) on page 38 for a list of error conditions.

Requesting a single location fix: SwiSetLbsPdGetPos API

The following code sample demonstrates using the SwiSetLbsPdGetPos API to request a single location [fix](#).

```
SWI_STRUCT_PDSM_PD_GetPos  GetLocFix;
GetLocFix.sizeStruct= sizeof(SWI_STRUCT_PDSM_PD_GetPos);

GetLocFix.iFixType= m_sFixType;
GetLocFix.iPerformance = m_editPerformance;
GetLocFix.iAccuracy = m_editAccuracy;

m_rCode = SwiSetLbsPdGetPos(&GetLocFix,
TIMEOUT);

if (SWI_RCODE_OK != m_rCode)
{
    ProcessError(m_rCode);
}
```

Table 3-3 lists and describes parameters for single location fixes and their acceptable values.

Table 3-3: Location parameters for single location fixes

| Parameter | Value | Description |
|----------------|-------------------|---|
| FixType | 1 | Standalone |
| | 2 | MS-based |
| | 3 | Assisted GPS (MS-assisted) |
| | 4 | Optimized for speed. This option specifies either an MS-based or Assisted GPS fix based on the fastest possible time to fix. The device determines which fix type to attempt based on fix rate and initial uncertainties. |
| | 5 | Optimized for accuracy. This option specifies an Assisted GPS fix first, and then an MS-based fix if the specified level of accuracy cannot be achieved by the AGPS fix. |
| | 6 | Optimized for data. This option specifies an MS-based fix first, and then an Assisted GPS fix. The device attempts to minimize the amount of data exchanged over the network. |
| Performance | 0 to 255 seconds | Amount of time (in seconds) the modem will wait for a satellite to be discovered |
| Accuracy | 1 to 10000 meters | Distance (in meters) to which data must be accurate |

See Table 3-8 on page 38 for a list of error conditions.

Note: The modem makes extensive use of notifications. To properly manage the modem, notifications must be implemented.

When using the SwiSetLbsPdGetPos function call, your application can be notified of the results of the position fix using the SWI_NOTIFY_Lbs_PD_Data notification. This notification reports to registered host clients the results of the position fix.

Figure 3-1 shows a location fix in the sample application.

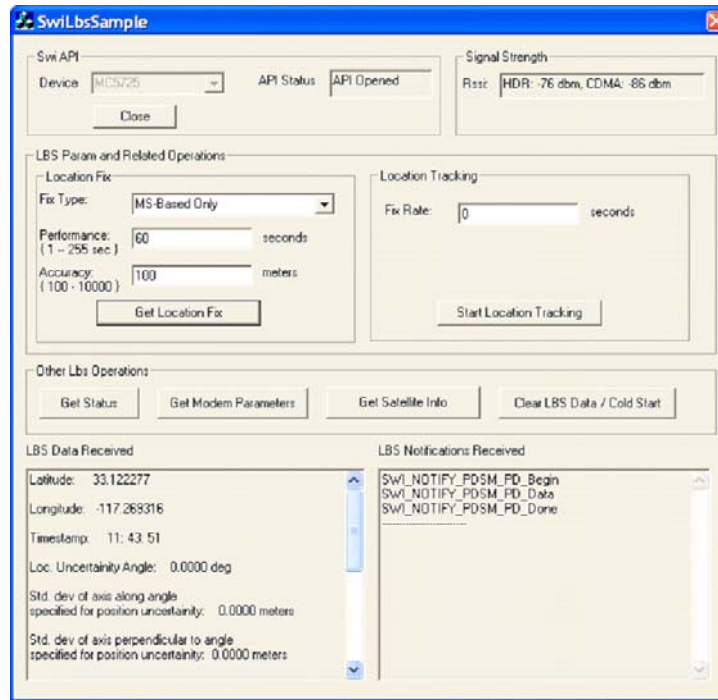


Figure 3-1: Location fix in the sample application

Requesting frequent tracking: SwiSetLbsPdTrack API

The following code sample demonstrates using the SwiSetLbsPdTrack API to request frequent tracking.

```
// TODO: Add your control notification handler code here
if( !m_bTracking )
{
    SWI_STRUCT_PDSM_PD_Track GetTrack;
    GetTrack.sizeStruct= sizeof(SWI_STRUCT_PDSM_PD_Track);

    GetTrack.nFixType= m_sFixType;
    GetTrack.nPerformance = m_editPerformance;
    GetTrack.nAccuracy = m_editAccuracy;
    GetTrack.nFixRate = m_editFixRate;
    GetTrack.nFixCount = 1000; ///Range is 0-1000. 1000 =
    continuous

    m_rCode = SwiSetLbsPdTrack(&GetTrack, TIMEOUT);

    if (SWI_RCODE_OK != m_rCode)
    {
```

```

ProcessError(m_rCode);
    }
    else
    {
m_bTracking = TRUE;
    }
}

```

Note: Continuous tracking does not terminate automatically. Using the sample application, a continuous tracking session is initiated by clicking the Start tracking button, which has a toggle functionality. Once a session has started, the button transforms to the End tracking button. To terminate the session, click the End tracking button.

Table 3-4 lists and describes parameters for frequent tracking and their acceptable values.

Table 3-4: Location parameters for frequent tracking

| Parameter | Value | Description |
|-------------|-------------------|---|
| FixType | 1 | Standalone |
| | 2 | MS-based |
| | 3 | Assisted GPS (MS-assisted) |
| | 4 | Optimized for speed. This option specifies either an MS-based or Assisted GPS fix based on the fastest possible time to fix. The device determines which fix type to attempt based on fix rate and initial uncertainties. |
| | 5 | Optimized for accuracy. This option specifies an Assisted GPS fix first, and then an MS-based fix if the specified level of accuracy cannot be achieved by the AGPS fix. |
| | 6 | Optimized for data. This option specifies an MS-based fix first, and then an Assisted GPS fix. The device attempts to minimize the amount of data exchanged over the network. |
| Performance | 0 to 255 seconds | Amount of time (in seconds) the modem will wait for a satellite to be discovered |
| Accuracy | 1 to 10000 meters | Distance (in meters) to which data must be accurate |
| FixRate | 1 to 1800 | Interval (in seconds) determining how often fixes are requested |
| FixCount | 1 to 1000 | Number of fixes. 1000 = continuous. Continuous fixes do not terminate automatically. |

See Table 3-8 on page 38 for a list of error conditions.

Note: The modem makes extensive use of notifications. To properly manage the modem, notifications must be implemented.

Figure 3-2 shows frequent tracking in the sample application.

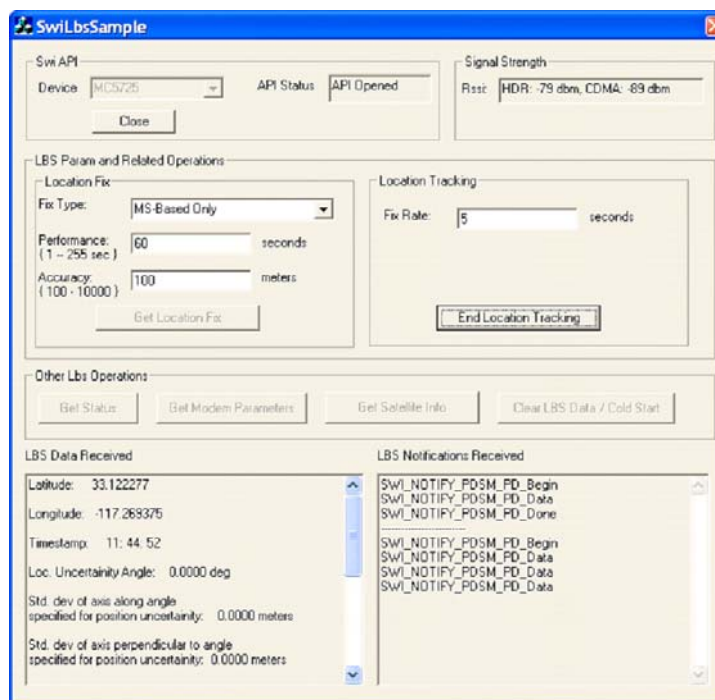


Figure 3-2: Frequent tracking in the sample application

Clear assistance data: SwiSetLbsClearAssistance API

The following code sample demonstrates using the SwiSetLbsClearAssistance API to clear various assistance data in the modem, which forces a cold start. Individual items of the assistance data can be specified.

```
void CSwiLbsSampleDlg::OnColdStart()
{
// TODO: Add your control notification handler code here
if (m_bConnected)
{
SWI_STRUCT_LbsClearAssistance ClearData;
ClearData.sizeStruct =
sizeof(SWI_STRUCT_LbsClearAssistance);
//Clear everything

ClearData.bClrEphemerisData= 1;
// 0: Do nothing 1: Clear Satellite Ephemeris Data

ClearData.bClrAlmanacData= 1;
// 0: Do nothing 1: Clear Satellite Almanac Data
```

```

ClearData.bClrPosData= 1;
// 0: Do nothing 1: Clear Position and Time information

ClearData.bClrSelfLearnDb= 1;
// 0: Do nothing 1: Clear Standalone Self Learning Data base

ClearData.bClrServSysDb= 1;
// 0: Do nothing 1: Clear Standalone Serving System Data Base

ClearData.bClrTimeRef= 1;
// 0: Do nothing 1: Clear Standalone Time Reference

ClearData.bResetThrottleParams= 1;
// 0: Do nothing 1: Reset MS Based Throttling Parameters

m_rCode = SwiSetLbsClearAssistance(&ClearData, TIMEOUT);
if (SWI_RCODE_OK != m_rCode)
{
ProcessError(m_rCode);
}
}
}
}

```

Downloading ephemeris and almanac data: SwiSetLbsPdDownload API

Warning: Before downloading *almanac data* and *ephemeris data* from a carrier's PDE, refer to carrier documentation for restrictions on PDE access frequency.

If the modem is in [Assisted GPS](#) or [MS-based](#) mode, it will automatically request [ephemeris data](#) and [almanac data](#) when it determines that the existing data is outdated. Your application can also request that ephemeris and almanac data be downloaded using the SwiSetLbsPdDownload function call.

The following code sample demonstrates using the SwiSetLbsPdDownload API to download ephemeris and almanac data.

```

SWI_STRUCT_PDSM_PD_Download GetData;
GetData.sizeStruct= sizeof(SWI_STRUCT_PDSM_PD_Download);
GetData.nOptions = 2 //1 - for periodic download and 2 for single
//download.
m_rCode = SwiSetLbsPdDownload(&GetData, TIMEOUT );

```

```

if (SWI_RCODE_OK != m_rCode)
{
    ProcessError(m_rCode);
}

```

Table 3-5 lists and describes parameters for downloading [ephemeris data](#) and [almanac data](#) and their acceptable values.

Table 3-5: Parameters for downloading ephemeris and almanac data

| Parameter | Value | Description |
|-----------|--------------------|---|
| Option | 1 | Option 1 specifies periodic data download. |
| | 2 | Option 2 specifies a single data download. |
| Duration | No range specified | If option 1 is specified, duration defines the interval between two downloads in minutes. |

Note: Almanac data is valid for up to 6 months. The ephemeris data refresh time frame is determined by how long satellites with ephemeris data are in view. Validity typically varies from 30 to 120 minutes, after which accuracy will slowly degrade unless new ephemeris data is obtained.

See [Table 3-8](#) on page 38 for a list of error conditions.

Note: The modem makes extensive use of [notifications](#). To properly manage the modem, notifications must be implemented.

Your application can be notified that a download session of [ephemeris data](#) and [almanac data](#) has begun or completed.

When using the `SwiSetLbsPdDownload` function call, your application can receive a `SWI_NOTIFY_Lbs_PD_Dload_Begin` [notification](#). This notification reports to [registered](#) host clients that a data download session has begun.

When using the `SwiSetLbsPdDownload` function call, your application can receive notification that a position determination download session has completed using `SWI_NOTIFY_Lbs_PD_Dload_End` and `SWI_NOTIFY_Lbs_PD_Dload_Done`.

When using the `SwiSetLbsPdDownload` function call, your application can receive a `SWI_NOTIFY_Lbs_PD_Dload` notification. This notification reports to [registered](#) host clients that ephemeris and almanac data have been downloaded.

Query satellite information: SwiGetLbsSatInfo API

The following code sample demonstrates using the SwiGetLbsSatInfo API to retrieve information about satellites visible during the latest position fix session and satellites used in the latest position fix.

```
SWI_STRUCT_PDSM_SAT_Info SatInfo;

SatInfo.sizeStruct =
sizeof(SWI_STRUCT_PDSM_SAT_Info);

if (m_bConnected)
{
m_rCode = SwiGetLbsSatInfo(&SatInfo,
TIMEOUT);
if (SWI_RCODE_OK != m_rCode)
{
ProcessError(m_rCode);
}
}
```

Getting base station information for location processing: SwiGetLbsPaBsInfo API

The following code sample demonstrates using the SwiGetLbsPaBsInfo API to retrieve base station information for location processing.

```
SWI_STRUCT_LbsPaBsInfo BsInfo;

BsInfo.sizeStruct =
sizeof(SWI_STRUCT_LbsPaBsInfo);

if (m_bConnected)
{
m_rCode = SwiGetLbsPaBsInfo(&BsInfo,
TIMEOUT);
if (SWI_RCODE_OK != m_rCode)
{
ProcessError(m_rCode);
}
}
```

Terminating a position determination session: SwiSetLbsPdEndSession API

Your application can terminate a location **fix** session or a download session.

Terminating a location fix session: SwiSetLbsPdEndSession API

If a location fix session is in progress, the following code sample demonstrates using the SwiSetLbsPdEndSession API to terminate a location fix session when the End tracking button is clicked in the sample application.

```
{
SWI_STRUCT_PDSM_PD_EndSession Ending;
Ending.sizeStruct = sizeof( SWI_STRUCT_PDSM_PD_EndSession);
Ending.iSessionType= 0; //Location Service

    m_rCode = m_pSwiSetLbsPdEndSession(&Ending,
    TIMEOUT);
    if (SWI_RCODE_OK != m_rCode)
    {
        ProcessError(m_rCode);
    }
}
```

[Table 3-6](#) lists and describes parameters for terminating location fixes.

Table 3-6: Location fix termination parameters

| Parameter | Value | Description |
|-------------|-------|--------------------------|
| SessionType | 0 | End location fix session |
| | 1 | End download session |

See [Table 3-8](#) on page 38 for a list of error conditions.

*Note: The modem makes extensive use of **notifications**. To properly manage the modem, notifications must be implemented.*

When using the SwiSetLbsPdEndSession function call, your application can receive **notification** of the termination of a location **fix** session using SWI_NOTIFY_Lbs_PD_End. The SWI_NOTIFY_Lbs_PD_End notification reports to the **requesting** host client when a position determination fix session has completed.

Figure 3-3 shows terminating a location fix session in the sample application.

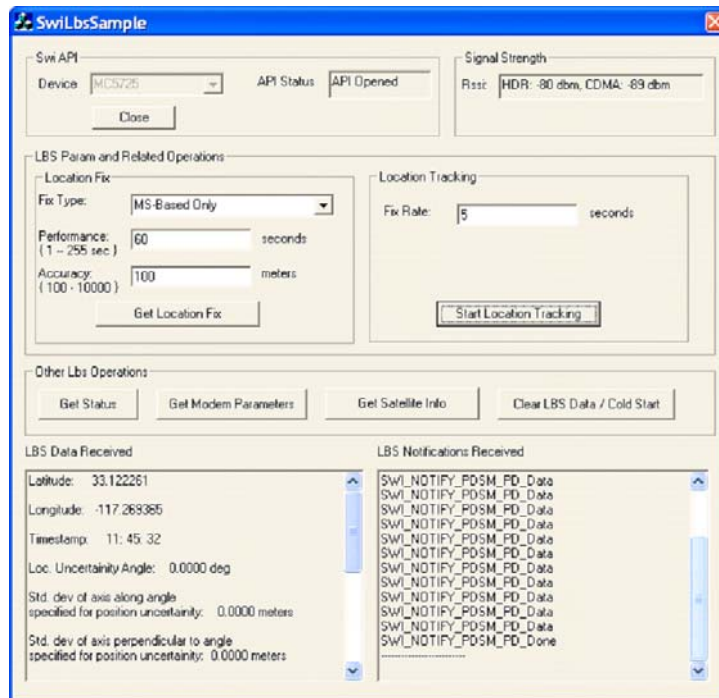


Figure 3-3: Terminating a location fix session in the sample application

Terminating a download session: SwiSetLbsPdEndSession API

Terminating a download session uses the same function call used to terminate a location fix session. See [Terminating a location fix session: SwiSetLbsPdEndSession API](#) on page 34 for details.

Requesting default position determination parameters: SwiGetLbsPaParam API

The following code sample demonstrates using the SwiGetLbsPaParam API to request default position determination parameters.

```
void CSwiLbsSampleDlg::OnGetParam()
{
    // TODO: Add your control notification handler code here
    SWI_STRUCT_PDSM_PA_GetParam GetParam;
    GetParam.sizeStruct =
        sizeof(SWI_STRUCT_PDSM_PA_GetParam);

    if (m_bConnected)
    {
```

```

m_rCode = SwiGetLbsPaParam(&GetParam,
TIMEOUT);
if (SWI_RCODE_OK != m_rCode)
{
    ProcessError(m_rCode);
}

```

Note: Values other than those shown are carrier specific. Contact a Sierra Wireless field application engineer for additional information.

Table 3-7 lists and describes location parameters and their acceptable values.

Table 3-7: Location parameters

| Parameter | Value | Description |
|---------------------|-------|------------------------------|
| Network Access | 0 | Network only |
| | 1 | No network access allowed |
| | 2 | Network with demodulation |
| | 3 | Network without demodulation |
| Privacy values | 0 | Low |
| | 1 | Medium |
| | 2 | High |
| Transport mechanism | 0 | Default |
| | 1 | TCP/IP |
| | 2 | DBM (databurst mechanism) |

See Table 3-8 on page 38 for a list of error conditions.

Alternatively, you can retrieve each parameter individually. To retrieve GPS IP address, call SwiGetLbsPaIpAddr. To retrieve GPS Lock, call SwiGetLbsPaGpsLock. To retrieve GPS PTLM mode, call SwiGetLbsPaPtlmMode. To retrieve GPS PortID, call SwiGetLbsPaPortId. To retrieve GPS Privacy, call SwiGetLbsPaPrivacy. To retrieve GPS NetAccess, call SwiGetLbsPaNetAccess.

Figure 3-4 shows position determination parameters in the sample application.

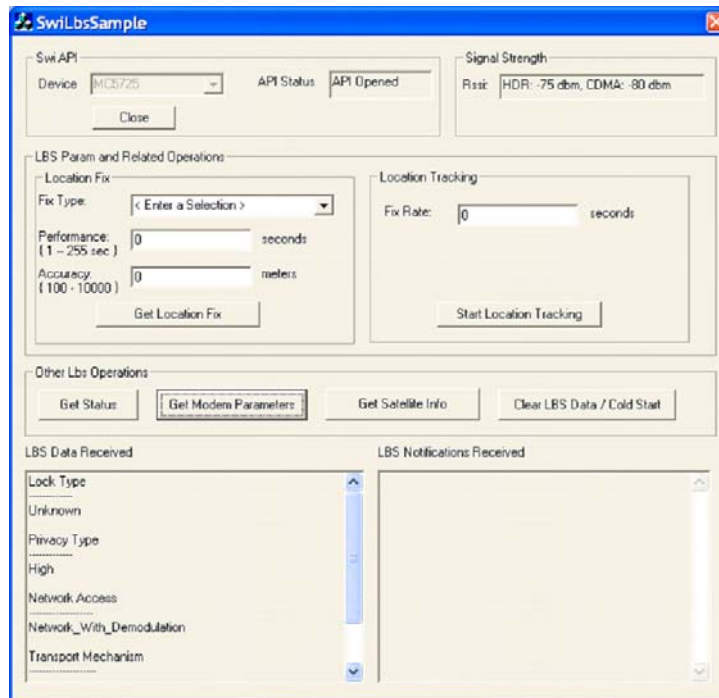


Figure 3-4: Requesting position determination parameters in the sample application

Using APIs to modify default position determination parameters

Note: If you are modifying more than one parameter, your application must wait for **notification** from the previous update before requesting another update. Otherwise, an error will occur.

Modifying the transport mechanism parameter: SwiSetLbsPaPtlmMode API

The following code sample demonstrates how to modify the transport mechanism parameter. Upon completion of modifications, the modem sends a SWI_NOTIFY_Lbs_PA_PtlmMode **notification**.

```
SWI_STRUCT_PDSM_PA_PTLM_Mode TransMech;
TransMech.sizeStruct = sizeof(SWI_STRUCT_PDSM_PA_PTLM_Mode);
TransMech.iTransportMech = SWI_PDSM_PD_TransportDataBurst;
m_rCode = SwiSetLbsPaPtlmMode(&TransMech,
TIMEOUT );
if (SWI_RCODE_OK != m_rCode)
{
```

```

        ProcessError(m_rCode);
    }

```

Modifying the privacy parameter: SwiSetLbsPaPrivacy API

The following code sample demonstrates how to modify the privacy parameter. Upon completion of modifications, the modem sends a SWI_NOTIFY_Lbs_PA_Privacy notification.

```

SWI_STRUCT_PDSM_PA_Privacy Privacy;
Privacy.sizeStruct = sizeof(SWI_STRUCT_PDSM_PA_Privacy);
Privacy.iPrivacy = SWI_PDSM_PA_Privacy_High;
m_rCode = SwiSetLbsPaPrivacy (&Privacy, TIMEOUT );
if (SWI_RCODE_OK != m_rCode)
{
    ProcessError(m_rCode);
}

```

Modifying the network access parameter: SwiSetLbsPaNetAccess API

The following code sample demonstrates how to modify the network access parameter. Upon completion of modifications, the modem sends a SWI_NOTIFY_Lbs_PA_Net_Access notification.

```

SWI_STRUCT_PDSM_PA_Net_Access NetAccess;
NetAccess.sizeStruct = sizeof(SWI_STRUCT_PDSM_PA_Net_Access);
NetAccess.iNetAccess = SWI_PDSM_PA_Network_Without_Demodulation;
m_rCode = SwiSetLbsPaNetAccess (&NetAccess, TIMEOUT );
if (SWI_RCODE_OK != m_rCode)
{
    ProcessError(m_rCode);
}

```

Error conditions for location fixes

Table 3-8 shows error conditions for location fixes.

Table 3-8: Error conditions for location fixes

| Condition | Description |
|--|--|
| SWI_PDSM_PD_ErrType_NoError | No error has occurred |
| SWI_PDSM_PD_ErrType_Internal_Modem_Error | An internal modem error has occurred |
| SWI_PDSM_PD_ErrType_Bad_Service_Type | An invalid (unsupported) service type has been specified |

Table 3-8: Error conditions for location fixes (Continued)

| Condition | Description |
|---|--|
| SWI_PDSM_PD_ErrType_Bad_Session_Type | An invalid session type has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Privacy | An invalid session privacy type has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Data_Download | An invalid download type has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Network_Access | An invalid network access type has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Operation_Mode | An invalid operation mode has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Number_of_Fixes | An invalid number of fixes has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Server_Info | Invalid server information has been specified |
| SWI_PDSM_PD_ErrType_Invalid_Timeout | An invalid timeout value has been specified |
| SWI_PDSM_PD_ErrType_Invalid_QoS_Parameter | An invalid quality of service (QoS) has been specified |
| SWI_PDSM_PD_ErrType_No_Session_Active | An attempt to end a session has been made when there is no active session |
| SWI_PDSM_PD_ErrType_Session_Already_Active | An attempt to initiate a session has been made when there is already an active session |
| SWI_PDSM_PD_ErrType_Session_Busy | A similar request is already in progress |
| SWI_PDSM_PD_ErrType_Phone_Is_Offline | The phone is in an offline state |
| SWI_PDSM_PD_ErrType_CDMA_Lock_Error | The device is CDMA locked |
| SWI_PDSM_PD_ErrType_GPS_Lock_Error | An invalid GPS Lock value has been specified |
| SWI_PDSM_PD_ErrType_Invalid_State | The specified state is invalid at the time of the request |
| SWI_PDSM_PD_ErrType_Connection_Failure | A connection failure occurred |
| SWI_PDSM_PD_ErrType_No_Buffers_Available | No buffers are available to queue the command |
| SWI_PDSM_PD_ErrType_Searcher_Error | An error in communicating with the searcher has occurred |
| SWI_PDSM_PD_ErrType_Cannot_Report_Now | Results cannot be reported at this time |
| SWI_PDSM_PD_Resource_Contention | A collision with a data call has occurred. |
| SWI_PDSM_PD_ErrType_Mode_Not_Supported | An invalid error type has been specified |
| SWI_PDSM_PD_Authentication_Failed | Authentication with the LPS has failed. |
| SWI_PDSM_PD_ErrType_Other_Error | An error other than stated above has occurred |
| SWI_PDSM_PD_ErrType_Fix_Rate_For_Tracking_Too_Large | An invalid fix rate has been specified for tracking |
| Unknown | An unknown error has occurred |

API FAQ

This section provides answers to common questions about location based services.

- Q: What are the **fix** type parameters and what do they mean?
- A: See [Table 3-3](#) on page 27 for parameters used for single location fixes, and [Table 3-4](#) on page 29 for parameters used for frequent tracking.
- Q: When should parameters or **fix** types be changed for frequent tracking?
- A: Changes to parameters or fix types should be based on application's requirements for operating LBS and current environmental conditions. Fix types (continuous or single) are application-dependent and the parameters depend on carrier support for LBS. See [Table 3-4](#) on page 29.
- Q: How is the data returned?
- A: The API facilitates data return in two ways: **notifications** and the Get method. Notifications indicate the data has been refreshed, while the Get method returns the last instance of the data. The two methods support synchronous (polling) or asynchronous applications.
- Q: What conditions would cause APIs to fail?
- A: The APIs are single-threaded. If another thread calls simultaneously, a failure could result.
- Q: Will the position tracking process time out?
- A: The **fix** acquisition process will time out if satellites cannot be located within the amount of time specified by the Performance parameter (from 0 to 255 seconds) for single fix requests. Tracking will not time out, but the number of fixes must be specified by the FixCount parameter.
- Q: Can I use the SwiSetLbsPdTrack API to get a single **fix**?
- A: Yes. You could accomplish this by setting the FixCount value to 1, but there is no advantage to using the position tracking API for this purpose. The SwiSetLbsPdGetPos API would be a better option.
- Q: Why are there two position determination APIs?
- A: The SwiSetLbsPdTrack API is for tracking, and the SwiSetLbsPdGetPos API is for single location **fixes**.

Q: Are there advantages or disadvantages to using the tracking API in favor of the single [fix](#) API?

A: There is no advantage in choosing one over the other, and the choice is based on host architecture preferences. Sierra Wireless supports both. Tracking engages the modem autonomously, but both APIs return the same data.

Q: What is [almanac data/ephemeris data](#)?

A: Almanac data consists of coarse orbital parameters. This information indicates to a GPS receiver where each GPS satellite should be at all times. Each satellite in the GPS transmits almanac data indicating its orbital information, and orbital information for every other satellite in the system. Ephemeris data is transmitted continuously by each satellite and provides precise information about the status of the satellite. This information is required to determine a geographic position.

Q: How is [almanac data/ephemeris data](#) downloaded?

A: See [Downloading ephemeris and almanac data: SwiSetLbsPdDownload API](#) on page 31.

Q: Do I need to command the modem to download assistance data from the network ([PDE](#)) when performing a [fix](#)?

A: The modem downloads this data automatically.

Q: Can I use the modem for LBS during a data session?

A: Yes.

Q: Does the modem cache [fix](#) data?

A: The modem caches position data only.

Q: How do I retrieve the last known location from the modem?

A: You can retrieve the last known location using the `SwiGetLbsPdData` function call. Note that, for the modem to return the last known location, prior to calling `SwiGetLbsPdData`, at least one GPS location [fix](#) should have occurred using either the `SwiSetLbsGetPos` or the `SwiSetLbsTrack` API function calls.

Q: How do I start a location [fix](#)?

A: You can start a location fix using the `SwiSetLbsGetPos` function call.

Q: Do carriers differ in their LBS implementations?

A: Yes. Carrier implementations and requirements do vary. Before testing LBS, it is recommended that you review their implementation and requirement documentation.

Configuring the modem and developer's kit for LBS

This section provides information on setting up the modem for live, over-the-air testing and general use.

Software components

These requirements apply to Watcher 7.4 or higher, or HyperTerminal. For information on other applications, refer to the manufacturers' documentation. For information on setting up Watcher to operate the modem in standalone mode, see [Setting up Watcher](#) on page 77. For information on setting up HyperTerminal, see [Setting up HyperTerminal for NMEA interface operation](#) on page 78.

Modem setup

To set up the modem:

1. Power up the modem.
2. Check the modem's lock setting using the AT!GPSLOCK command. (See [Opening and unlocking the modem: AT!GPSLOCK command](#) on page 55 for details.) Verify that the modem setting is 0: No GPS lock active.
3. Verify that the PDE TCP/IP address and port number are not set to 0. If these are set to 0, fixes requiring PDE access will fail. These include [AGPS](#), [MS-based](#), or the optimal trio (optimal speed, optimal accuracy, and optimal data).

Note: To retrieve the GPS IP address, use AT!GPSIPADDR. To retrieve the port ID, use AT!GPSPORTID.

4. Launch Watcher (or a similar application). (See [Setting up Watcher](#) on page 77 for information on Watcher setup.)
5. Access the LBS specific screen in the application.
6. Perform LBS fixes.

Note: Failure to verify that the modem lock setting is "0: No GPS lock active" is a common error, because some carriers require the default be set to 3, which enables LBS for [E911](#) only.

Using CnS commands

Control and status (CnS) commands are the basis of Sierra Wireless' binary host to modem control and status interface. The modem fully supports sending and receiving position determination session management (PDSM) messages. PDSM-related control and status interface messages were initially implemented for this modem only, but ultimately will be used for all Sierra Wireless modems that support the QUALCOMM PDSM interface.

A typical application scenario is that Sierra applications like Watcher use CnS to communicate LBS parameters and collect location information. These applications will display their own user interfaces to the user.

CnS supports a query-response type transaction as well as asynchronous [notifications](#). To efficiently perform all call processing-related transactions, the host system must be able to react to incoming notifications. Normal CnS procedure requires sending a notification request packet for each individual CnS object. However, because an application that will require one location fix notification, or one download data notification will typically need all of the notifications of that type, the CNS_PDSM_LOC_NOTIFY_STATUS and CNS_PDSM_DLOAD_NOTIFY_STATUS notifications have been created. The CNS_PDSM_PARAM_NOTIFY_STATUS notification is for an application that is interested in the value of one or more PDSM parameters. Sending a set request with one of these object IDs will turn on all the location, download data, or parameter-related notifications.

Refer to the Software Integration Guide (document number 2130759) for additional information on CnS commands.

Note: The following subsections address CDMA uses only.

CnS message interface

The secondary port of the modem supports the CnS message interface. This is intended to be the principal control mechanism of the modem. The CnS Reference (document number 2130754) provides a detailed description of this proprietary protocol.

The modem encapsulates CnS messages within a transport packet called HiP (Host Interface Protocol). To use this interface, your software must assemble and disassemble HiP and CnS packets.

Query, response, and notification

The CnS interface is the principal interface because it can support asynchronous [notifications](#) in addition to the traditional query and response interaction.

Notifications allow the modem to keep the host application informed of radio signal quality, connection status, and other aspects of radio modem operation, without the need for the host application to issue repeated queries.

Limitations

Although all modem feature commands and status information can be exchanged across the CnS message interface, the AT command interface is still used to create data connections.

Dialing, answering, and ending both data and fax connections require use of AT commands. The CnS interface is used to originate and answer voice calls.

For additional details on the limitations of the CnS message interface, refer to the CnS Reference.

Table 3-9 shows CnS commands used in PDSM transactions.

Table 3-9: CnS commands used in PDSM transactions

| Object | ID | Description | Set | Get | Notification |
|---|--------|--|-----|-----|--------------|
| Reserved | 0x0F00 | N/A | X | X | X |
| PD Download (page 51) | 0x0F01 | Requests download of ephemeris data and almanac data | ✓ | X | X |
| PD Get Position (page 49) | 0x0F02 | Requests a single location fix | ✓ | X | X |
| Get Status (page 46) | 0x0F03 | Provides status of LBS service | X | ✓ | X |
| PD Track (page 50) | 0x0F04 | Requests a location tracking session | ✓ | X | X |
| PD End Session (page 52, page 52) | 0x0F05 | Requests that an active session be terminated | ✓ | X | X |
| PA Parameters (page 53) | 0x0F06 | Requests the current PDSM default parameters | X | ✓ | X |
| Location notification status (page 46) | 0x0F07 | Enable, disable, or check current status of location notifications | ✓ | ✓ | X |
| Download notification status (page 46) | 0x0F08 | Enable, disable, or check current status of download notifications | ✓ | ✓ | X |
| Parameter notification status (page 46) | 0x0F09 | Enable, disable, or check current status of parameter notifications | ✓ | ✓ | X |
| Position Fix Data (page 49) | 0x0F0A | Location fix data | X | ✓ | ✓ |
| Position Fix Session Done (page 49) | 0x0F0B | Notifies registered clients that a location fix session has completed | X | X | ✓ |
| Position Fix Session End (page 49, page 52) | 0x0F0C | Notifies requesting client that the requested location fix session has ended with an error | X | X | ✓ |
| Position Fix Session Begin (page 49, page 50) | 0x0F0D | Notifies registered clients that a location fix session has started | X | X | ✓ |

Table 3-9: CnS commands used in PDSM transactions (Continued)

| Object | ID | Description | Set | Get | Notification |
|---|--------|--|-----|-----|--------------|
| PD Update Failed | 0x0F0E | Location fix session has failed | X | X | ✓ |
| Download Data Session Begin (page 51) | 0x0F0F | Notifies registered clients that a data download session has started | X | X | ✓ |
| Download Data (page 51) | 0x0F10 | Data download data available | X | X | ✓ |
| Download Data Session Done (page 51, page 52) | 0x0F11 | Notifies registered clients that a data download session has completed | X | X | ✓ |
| Download Data Session End (page 51, page 52) | 0x0F12 | Notifies requesting client that the requested data download session has ended with an error | X | X | ✓ |
| TCP/IP Address (page 53) | 0x0F13 | Default TCP/IP address | ✓ | ✓ | ✓ |
| GPS Lock (page 53) | 0x0F14 | GPS Lock value | ✓ | ✓ | ✓ |
| Transport Mechanism (page 53) | 0x0F15 | Default transport mechanism | ✓ | ✓ | ✓ |
| Port ID (page 53) | 0x0F16 | Default port ID | ✓ | ✓ | ✓ |
| Privacy Level (page 53) | 0x0F17 | Default privacy level | ✓ | ✓ | ✓ |
| Network Access Permissions (page 53) | 0x0F18 | Default network access permission | ✓ | ✓ | ✓ |
| Base Station information (page 52) | 0x0F19 | Current base station information | X | ✓ | ✓ |
| Satellite Information (page 52) | 0x0F1F | Satellites visible during the latest position fix session | X | ✓ | ✓ |
| Clear GPS Assistance Data (page 51) | 0x0F20 | Clears various location assistance parameters | ✓ | X | X |

Types of LBS clients

Registered clients are those interested in receiving **notification** of LBS events.

A **requesting** (or **active**) client is one that has initiated the current LBS session or event. A requesting client is typically also a registered client.

Enabling notifications: Location notification status and Download notification status objects

Because an application that requires one location [fix notification](#) or one download data notification will typically need all of the notifications of that type, it is recommended that you enable object 0x0F07, Location notification status, and object 0x0F08, Download notification status. If your application is interested in the value of, or change in, one or more LBS parameters, enable object 0x0F09, Parameter notification status.

Opening and unlocking the modem: GPS Lock object

Object 0x0F14, GPS Lock, is used to open and unlock the modem.

A request to open and unlock the modem follows these steps:

1. The host sends a set request to the modem to open and unlock the modem: CNS_PDSM_PA_GPS_LOCK.
2. The modem validates the request.
3. The modem sends a set response to the host: CNS_PDSM_PA_GPS_LOCK.
4. The modem sends a [notification](#) to the host: CNS_PDSM_PA_GPS_LOCK.

[Table 3-10](#) lists and describes GPS lock status settings.

Table 3-10: GPS lock status settings

| Parameter | Value | Description |
|-----------|-------|--|
| GPS Lock | 0 | Modem is unlocked |
| | 1 | Mobile-initiated sessions are not allowed |
| | 2 | Mobile-terminated sessions are not allowed |
| | 3 | E911 only allowed |

Getting the current status of location based services: PDSM Status object

Object 0x0F03, PDSM Status, is used to retrieve the current status of location based services.

Table 3-11 lists and describes current status information returned by the PDSM Status object.

Table 3-11: Current status information of Location Based Services

| Parameters | Length | Offset | Type | Description |
|------------|--------|--------|--------|--|
| 01 (Get) | 0 | -- | -- | None |
| 02 (Reply) | 82 | 0 | UINT16 | Object Version: Version of this object payload |
| | | 2 | CHAR | Last Fix Timestamp: 16-character string in the format YYYYMMDDdHHMMSS<null> where d = a week day: <ul style="list-style-type: none"> • 0 = Monday • 1 = Tuesday • 2 = Wednesday • 3 = Thursday • 4 = Friday • 5 = Saturday • 6 = Sunday |
| | | 18 | UINT16 | Last Fix Status: <ul style="list-style-type: none"> • 0: None • 1: Active • 2: Success • 3: Failed |
| | | 20 | UINT16 | Last Fix Error |
| | | 22 | CHAR | Last Download Timestamp: 16-character string in the format YYYYMMDDdHHMMSS<null> where d = a week day: <ul style="list-style-type: none"> • 0 = Monday • 1 = Tuesday • 2 = Wednesday • 3 = Thursday • 4 = Friday • 5 = Saturday • 6 = Sunday |
| | | 38 | UINT16 | Last Download Status: <ul style="list-style-type: none"> • 0: None • 1: Active • 2: Success • 3: Failed |
| | | 40 | UINT16 | Last Download Error |

Table 3-11: Current status information of Location Based Services (Continued)

| Parameters | Length | Offset | Type | Description |
|---|--------|--------|--------|--|
| | | 42 | CHAR | Last Fix Session Timestamp: 16-character string in the format YYYYMMDDdHHMMSS<null> where d = a week day: <ul style="list-style-type: none"> • 0 = Monday • 1 = Tuesday • 2 = Wednesday • 3 = Thursday • 4 = Friday • 5 = Saturday • 6 = Sunday |
| | | 58 | UINT16 | Last Fix Session Status: <ul style="list-style-type: none"> • 0: None • 1: Active • 2: Success • 3: Failed |
| | | 60 | UINT16 | Last Fix Session Error |
| | | 62 | CHAR | Last Download Session Timestamp: 16-character string in the format YYYYMMDDdHHMMSS<null> where d = a week day: <ul style="list-style-type: none"> • 0 = Monday • 1 = Tuesday • 2 = Wednesday • 3 = Thursday • 4 = Friday • 5 = Saturday • 6 = Sunday |
| | | 78 | UINT16 | Last Download Session Status: <ul style="list-style-type: none"> • 0: None • 1: Active • 2: Success • 3: Failed |
| | | 80 | UINT16 | Last Download Session Error |
| Legend <ul style="list-style-type: none"> • Length: Length of the message payload data (in bytes) • Offset: Number of bytes skipped from the first byte of the payload to reach the start of the referenced information | | | | |

Requesting the location of the modem

You can request the last known location or a single location [fix](#).

Requesting the last known location: Position Fix Data object

Object 0x0F0A, Position Fix Data, is used to request the last known location.

A request for the last known location follows these steps:

1. The host sends a get request to the modem for the last known location of the modem: CNS_PDSM_PD_DATA
2. The modem validates the request.
3. The modem provides the data if it is available using CNS_PDSM_PD_DATA. If data is not available, it returns a status code indicating that the location is not available.

Requesting a single location fix: PD Get Position object

Object 0x0F02, PD Get Position, is used to initiate a location [fix](#). A request for a single location fix follows these steps:

1. The host sends a set request to the modem: CNS_PDSM_PD_GET_POS
2. The modem validates the request.
3. If there is an error, the modem returns the PD error. If there is no error, the modem returns the set response: CNS_PDSM_PD_GET_POS
4. The modem notifies the host that the session has begun:
CNS_PDSM_PD_BEGIN
5. The modem sends data to the host when it is available:
CNS_PDSM_PD_DATA
If an error occurs, the modem will send the CNS_PDSM_PD_END [notification](#) and will not send data.
6. The modem notifies the host that the session has completed:
CNS_PDSM_PD_DONE

Note: This flow assumes the host has enabled notifications and that no contact with the network for assistance data is required. If assistance data is required and the host has enabled those notifications, the modem sends the host a CNS_PDSM_PD_DLOAD_BEGIN notification, one or more CNS_PDSM_PD_DLOAD_DATA notifications, and a CNS_PDSM_PD_DLOAD_DONE notification. If an error occurs, the modem sends the host a CNS_PDSM_PD_DLOAD_END notification.

Object 0x0F0D, Position Fix Session Begin, notifies [registered](#) host clients that a position determination [fix](#) session has begun. Object 0x0F0A, Position Fix Data, is used to return the results of the location fix. Object 0x0F0C, Position Fix Session

End, notifies host clients if the position determination fix session has ended in error. Object 0x0F0B, Position Fix Session Done, notifies registered host clients that a position determination fix session has completed.

Requesting frequent tracking: PD Track object

Object 0x0F04, PD Track, is used to start a tracking session. This object should be used when an application requires multiple location fixes over a small period of time (a fix between every 1 to 240 seconds, for example). If necessary, the modem downloads [ephemeris data](#) and [almanac data](#). (See [Downloading ephemeris and almanac data: PD Download object](#) on page 51 for details.)

A request for frequent modem tracking follows these steps:

1. The host sends a set request to the modem to initiate the tracking session:
CNS_PDSM_PD_TRACK
2. The modem validates the request.
3. If there are not errors in the request, the modem begins the tracking session and sends a set response to the host: CNS_PDSM_PD_TRACK
If there is an error, the modem returns the PD error type to the host.
4. The modem notifies the host that the tracking session has begun:
CNS_PDSM_PD_BEGIN
5. The modem notifies the host of the location: CNS_PDSM_PD_DATA
Notifications continue as fixes occur.
6. The modem notifies the host of the satellite information:
CNS_PDSM_SAT_INFO.
7. When the session has completed, the modem notifies the host:
CNS_PDSM_PD_END,
CNS_PDSM_PD_DONE

Note: This flow assumes the host has enabled notifications and that no contact with the network for assistance data is required.

Object 0x0F0D, Position Fix Session Begin, notifies [registered](#) host clients that a position determination fix session has begun.

Object 0x0F0A, Position Fix Data, is used to return the results of location fixes.

Object 0x0F1F, Satellite Information, is used to return the satellite information.

Object 0x0F0C, Position Fix Session End, notifies host clients if the position determination fix session has ended in error.

Object 0x0F0B, Position Fix Session Done, notifies [registered](#) host clients that a position determination fix session has completed.

Clear assistance data: Clear GPS Assistance Data object

Object 0x0F20, Clear GPS Assistance Data, is used to clear various assistance data in the modem, which forces a cold start. Individual items of the assistance data can be specified.

Downloading ephemeris and almanac data: PD Download object

Object 0x0F01, PD Download, requests the modem to perform a download of the [ephemeris data](#) and [almanac data](#). The host uses DloadOption to indicate whether the data download should occur just once, or periodically to keep the data warm.

A request to download ephemeris and almanac data follows these steps:

1. The host sends a set request to the modem to initiate data download:
CNS_PDSM_PD_DOWNLOAD
2. The modem validates the request.
3. The modem sends a set response to the host:
CNS_PDSM_PD_DOWNLOAD
4. The modem initiates the download session.
5. The modem notifies the host that the download session has begun:
CNS_PDSM_PD_DLOAD_BEGIN
6. The modem notifies the host when download data is available:
CNS_PDSM_PD_DLOAD_DATA
If DloadOption=PERIODIC, this event may be sent multiple times.
7. Once the download is complete, the modem notifies the host:
CNS_PDSM_PD_DLOAD_DONE

Note: This flow assumes the host has enabled notifications.

If an error causes the download session to terminate prematurely, the modem notifies the host:

CNS_PDSM_PD_DLOAD_END
CNS_PDSM_PD_DLOAD_DONE

Object 0x0F0F, Download Data Session Begin, notifies [registered](#) host clients that a download session of ephemeris and almanac data has begun.

Object 0x0F10, Download Data, notifies registered host clients that ephemeris and almanac data have been downloaded.

Object 0x0F12, Download Data Session End, notifies registered host clients if a position determination download data session has ended in error.

Object 0x0F11 Download Data Session Done, notifies registered host clients that a position determination download session has completed.

Query satellite information: Satellite information object

Object 0x0F1F, Satellite information, is used to retrieve information about satellites visible during the latest position fix session and satellites used in the latest position fix.

The CNS_PDSM_SAT_INFO notification informs the host of the satellite information.

Getting base station information for location processing: Base Station information object

Object 0x0F19, Base Station information, is used to request information about the Base Station currently being accessed.

Terminating a position determination session

You can terminate a location [fix](#) session or a download session.

Terminating a location fix session: End Session object

Object 0x0F05, PD End Session, is used to request the termination of a location fix session or a download data session.

A request to terminate a location fix session follows these steps:

1. The host sends a set request to the modem to terminate the fix session:
CNS_PDSM_PD_END_SESSION
2. The modem validates the request.
3. The modem terminates the session and notifies the host:
CNS_PDSM_PD_END
End status = Client ended
4. The modem notifies the host that the session has completed:
CNS_PDSM_PD_DONE

Note: This flow assumes the host has enabled notifications.

Object 0x0F0C, Position Fix Session End, notifies host clients if a position determination fix session has ended in error.

Object 0x0F0B, Position Fix Session Done, notifies [registered](#) host clients that a position determination fix session has completed.

Terminating a download session: End Session object

Object 0x0F05, PD End Session, is used to request the termination of a location [fix](#) session or a download data session.

A request to terminate a download session follows these steps:

1. The host sends a set request to the modem to terminate the download session: CNS_PDSM_PD_END_SESSION
2. The modem validates the request.
3. The modem sends a set response to the host:
CNS_PDSM_PD_END_SESSION
4. The modem terminates the session and notifies the host that the session was ended:
CNS_PDSM_PD_DLOAD_END
End status = Client ended
5. The modem notifies the host that the session has completed:
CNS_PDSM_PD_DLOAD_DONE

Note: This flow assumes the host has enabled notifications.

Object 0x0F12, Download Data Session End, notifies [registered](#) host clients if a position determination download data session has ended in error.

Object 0x0F11, Download Data Session Done, notifies registered host clients that a position determination download session has completed.

Accessing default position determination parameters: PA Parameter Settings object

Object 0x0F06, PA Parameter Settings, is used to retrieve default parameter settings used during LBS sessions. The entire set of position determination parameters (IP Address, Port ID, GPS Lock, Transport Mechanism, Privacy, Network Access) is returned.

A request to retrieve all default position determination parameters follows these steps:

1. The host sends a get request to the modem to request the default values of all parameters:
CNS_PDSM_PA_GET_PARAM
2. The modem validates the request.
3. The modem sends the host a get response containing the default values of all parameters:
CNS_PDSM_PA_GET_PARAM
4. The host inspects all values and requests changes if required.

Accessing individual default position determination parameters

Position determination parameters can also be accessed individually by specifying objects 0x0F13 TCP/IPAddress, 0x0F14 GPS Lock, 0x0F15 Transport Mechanism, 0x0F16 Port ID, 0x0F17 Privacy Level, and 0x0F18 Network Access Permissions.

Modifying default position determination parameters

Although all parameters can be requested together, the **CnS** interface doesn't provide a single interface to update more than one parameter at the same time. Instead, CnS objects corresponding to individual parameters must be used. In the event that more than one parameter must be updated, the host must wait for **notification** that the previous parameter has been updated before proceeding to update the next parameter. Otherwise, subsequent update requests will result in errors.

A request to update a default position determination parameter follows these steps:

1. The host sends the modem a set request that the default value of a specified parameter be changed to a new value:
CNS_PDSM_PA_XXX
where XXX is the parameter
2. The modem validates the request and returns a set response:
CNS_PDSM_PA_XXX
where XXX is the parameter
3. The modem notifies the host of the updated default value:
CNS_PDSM_PA_XXX
where XXX is the parameter

Note: This flow assumes the host has enabled notifications.

If there is an error, the modem returns the PA error type to the host.

Using AT commands

The modem's primary port supports the attention (or AT) command interface. This guide covers the AT commands used to make use of modem features.

Note: Serial connection to the modem is required to issue AT commands.

The CDMA AT Command Reference (document number 2130620) provides information about the connection characteristics in addition to a full reference for each command.

Many commands in this reference are restricted, and therefore, not generally accessible to end users. Access to the commands is unlocked by the OEM Access Lock using the !OEM command¹. This command sets the status of the lock for access to OEM restricted commands. Once the restricted command access has been unlocked, it remains unlocked until the modem is reset.

1. If you need information on this command, contact your account manager.

Query and response

The AT command interface is built on a query response model. The host issues AT commands and the modem responds. These are simple text-based commands for modem control and status.

A typical application scenario is a custom application using this interface to set LBS parameters and collect location information. These commands are more useful for “one shot” location lookups than for continuous monitoring, which is best served from the [NMEA](#) port.

When working with a wireless modem, there is often an interest in radio status, such as received signal strength or quality. In normal operation, this information is provided as periodic [notifications](#) by the modem using the CnS interface. The AT command interface cannot support notifications. Although the status information is available through the AT command interface, the host application must solicit this information when needed.

Enabling notifications

There are no notifications in the AT interface.

Opening and unlocking the modem: AT!GPSLOCK command

The AT!GPSLOCK command gets or sets the GPS lock for location processing.

Syntax:

AT!GPSLOCK?

or

AT!GPSLOCK=<lock>

[Table 3-12](#) lists and describes possible lock values.

Table 3-12: GPS lock settings

| Parameter | Value | Description |
|-----------|-------|--|
| GPS Lock | 0 | Modem is unlocked |
| | 1 | Mobile-initiated sessions are not allowed |
| | 2 | Mobile-terminated sessions are not allowed |
| | 3 | E911 only allowed |

Note: Successfully issuing the SET version of this command does not guarantee that the value has yet been updated in non-volatile (NV) memory. Rather, it indicates that the request to update the value is syntactically correct and that the request is being placed in the queue. It is up to the application that updated the value to verify the NV update by issuing a subsequent GET version of this command.

Examples:

```
AT!GPSLOCK=2
OK
```

or

```
AT!GPSLOCK?
1
OK
```

Getting the current status of location based services: AT!GPSSTATUS command

The AT!GPSSTATUS command queries the status of **fix** and download position location sessions.

Table 3-13 lists and describes location based services status values.

Table 3-13: Location based services status values

| Status | Meaning |
|---------|---|
| NONE | No session of this type has occurred since power up |
| ACTIVE | A session of this type is currently active |
| SUCCESS | Most recent session of this type succeeded |
| FAIL | Most recent session of this type failed |

The session status includes a timestamp indicating when the last session's status was determined. If a session is active, the system time when the session entered the ACTIVE state is displayed. If a session has a FAIL status, an error code is also displayed.

Example:

```
AT!GPSSTATUS
200505166000104 Last Fix Status = SUCCESS
200505166000215 Last DLoad Status = SUCCESS
200505166000104 Fix Session Status = ACTIVE
200505166000215 DLoad Session Status = SUCCESS
```

OK

or

AT!GPSSTATUS

200503094060302 Last Fix Status = FAIL FAILCODE = 12

0000000000000000 Last DLoad Status = NONE

200503094060302 Fix Session Status = FAIL FAILCODE = 12

0000000000000000 DLoad Session Status = NONE

Table 3-14 lists and describes location based services failure code values.

Table 3-14: Location based services failure codes

| Failure code | Meaning |
|--------------|--------------------------------|
| 0 | Phone is offline |
| 1 | No service |
| 2 | No PDE connection |
| 3 | No data available |
| 4 | Session manager busy |
| 5 | CDMA locked |
| 6 | GPS locked |
| 7 | PDE connection failed |
| 8 | GPS ended session due to error |
| 9 | User ended session |
| 10 | End key pressed from UI |
| 11 | Network session was ended |
| 12 | Timeout occurred |
| 13 | Privacy level error |
| 14 | Net access error |
| 15 | Error in fix |
| 16 | PDE rejected |
| 17 | Traffic channel exit |
| 18 | E911 override |
| 19 | Server error |
| 20 | Stale BS information |

Enabling the NMEA stream: AT!GPSNMEAENABLE command

The AT!GPSNMEAENABLE command enables or disables the output of the [NMEA](#) messages (stream) to the NMEA port.

Syntax:

```
AT!GPSNMEAENABLE?
```

or

```
AT!GPSNMEAENABLE=<nmea>
```

[Table 3-15](#) lists and describes possible NMEA values.

Table 3-15: NMEA settings

| Parameter | Value | Description |
|-------------|-------|-------------------------|
| NMEA stream | 0 | Disable the NMEA stream |
| | 1 | Enable the NMEA stream |

Note: Successfully issuing the SET version of this command does not guarantee that the value has yet been updated in non-volatile (NV) memory. Rather, it indicates that the request to update the value is syntactically correct and that the request is being placed in the queue. It is up to the application that updated the value to verify the NV update by issuing a subsequent GET version of this command.

Examples:

```
AT!GPSNMEAENABLE=1
```

```
OK
```

Requesting the location of the modem

You can request the last known location or a single location [fix](#).

Requesting the last known location: AT!GPSLOC command

The AT!GPSLOC command gets the last known location of the modem.

This command does not initiate a position location session; rather, it attempts to access the information from the most recent position location session. If the data is not available, the response will indicate that the location is unknown. Altitude and/or heading information will not be included if this data was not collected as part of the most recent location [fix](#). Therefore, either or both of these lines may not appear in the output. The VelVert value will be set to 0 for a 2D fix.

Examples:

```
AT!GPSLOC
```

```
Unknown
```

```
OK
```

or

```
AT!GPSLOC
```

```
Lat: 33 Deg 7 Min 21.00 Sec N (0x005E3712)
```

```
Lon: 117 Deg 16 Min 10.31 Sec W (0xFEB26EEF)
```

```
Time: 200603174135634
```

```
LocUncAngle: 0 LocUncA: 8 LocUncP: 9
```

```
3D Fix
```

```
Altitude: 43 LocUncVe: 11
```

```
Heading: 0 VelHoriz: 515 VelVert: 0
```

```
OK
```

or

```
AT!GPSLOC
```

```
Lat: 33 Deg 7 Min 21.00 Sec N (0x005E3712)
```

```
Lon: 117 Deg 16 Min 10.31 Sec W (0xFEB26EEF)
```

```
Time: 200603174135634
```

```
LocUncAngle: 0 LocUncA: 8 LocUncP: 9
```

```
2D Fix
```

```
Heading: 0 VelHoriz: 515 VelVert: 0
```

```
OK
```

Requesting a single location fix: AT!GPSFIX command

The AT!GPSFIX command initiates a location **fix** based on supplied parameters and returns the status of the position location session. If the modem is able to initiate the location fix, the OK message will be displayed. While the fix is in progress, the application may query the status of the session using the AT!GPSSTATUS command. If the session has completed successfully, AT!GPSLOC returns the location information. If the modem is unable to initiate the location fix, an error code will be returned.

Syntax:

```
AT!GPSFIX=<fixtype>,<maxtime>,<maxdist>
```

Table 3-16 lists and describes parameters for requesting a single location fix using an AT command.

Table 3-16: Parameters for single location fixes

| Parameter | Meaning |
|-----------|--|
| fixtype | 1 - Standalone |
| | 2 - MS-based only |
| | 3 - Assisted GPS (MS-assisted) only |
| | 4 - Optimum speed |
| | 5 - Optimum accuracy |
| | 6 - Optimum data |
| maxtime | Number of seconds allowed to capture satellite information Range: 0 - 255 |
| maxdist | Accuracy (in meters) the application prefers Range: 1 - 10,000 |

Examples:

```
AT!GPSFIX=2,30,200
```

OK

or

```
AT!GPSFIX=3,60,100
```

```
ERRCODE = 18
```

Requesting frequent tracking: AT!GPSTRACK command

The AT!GPSTRACK command initiates a tracking session. This command should be used when an application requires multiple location fixes over a small period of time (for example, a fix between every 1 to 240 seconds).

Using the fixcount and fixrate parameters, the application may specify how many fixes should be done, and at what rate they should occur. (The fixrate parameter is measured as the number of seconds between the start of one fix to the time that the subsequent fix is triggered.) If the modem is able to initiate the tracking session, an OK message will be displayed.

If the [ephemeris data](#) and [almanac data](#) requires updates, the time to first fix may require more time than subsequent fixes. Tracking applications should consider this possible initial delay and may want to behave proactively by initiating a data download session and/or a single location fix as part of their startup sequence.

Note: Almanac data is valid for up to 6 months, ephemeris data is valid for 30 to 120 minutes, and coarse location data is valid for 4 minutes.

While the session is in progress, the application may query its status using the AT!GPSSTATUS command. If the session has completed successfully, AT!GPSLOC returns the location information. If the modem is unable to initiate the tracking session, an error code will be returned. Once the status shows that the session is complete, the application should use the AT!GPSLOC command to obtain the result.

Syntax:

```
AT!GPSTRACK=<fixtype>,<maxtime>,<maxdist>,<fixcount>,<fixrate>
```

Table 3-17 lists and describes fix parameters.

Table 3-17: Fix parameters for frequent tracking

| Parameter | Meaning |
|-----------|--|
| fixtype | 1 - Standalone |
| | 2 - MS-based only |
| | 3 - Assisted GPS (MS-assisted) only |
| | 4 - Optimum speed |
| | 5 - Optimum accuracy |
| | 6 - Optimum data |
| maxtime | Number of seconds allowed to capture satellite information Range: 0 - 255 |
| maxdist | Accuracy (in meters) the application requires Range: 1 - 10,000 |
| fixcount | Number of fixes to perform Range: 1 - 999, or 1000 = continuous |
| fixrate | Number of seconds between each fix Range: 1 - 1,800 seconds |

Clear assistance data: AT!GPSCLRASSIST command

The AT!GPSCLRASSIST command clears various assistance data in the modem, which forces a cold start. Individual items of the assistance data can be specified.

Syntax:

```
!GPSCLRASSIST=<eph>,<alm>,<pos>,<learn>,<serv>,<time>,<throttle>
```

Example (typical settings for a cold start):

```
AT!GPSCLRASSIST=1,1,1,0,0,0,1
```

OK

Table 3-18 lists and describes fix parameters.

Table 3-18: Fix parameters for clearing assistance data

| Parameter | Value | Meaning |
|-----------|-------|--|
| eph | 0 | Ignore |
| | 1 | Clear epemeris assistance data |
| alm | 0 | Ignore |
| | 1 | Clear almanac assistance data |
| pos | 0 | Ignore |
| | 1 | Clear position assistance data |
| learn | 0 | Ignore |
| | 1 | Clear the Standalone Self Learning database |
| serv | 0 | Ignore |
| | 1 | Clear the Standalone Serving System database |
| time | 0 | Ignore |
| | 1 | Clear the Standalone Time Reference |
| throttle | 0 | Ignore |
| | 1 | Reset the MS-based throttle parameters |

Downloading ephemeris and almanac data: AT!GPSDLOAD command

The AT!GPSDLOAD command initiates a download of [ephemeris data](#) and [almanac data](#) and returns the status of the download session.

Syntax:

```
AT!GPSDLOAD=<freq>,<duration>
```

If the application requesting this data requires the use of the data over a period of time, the application should indicate that the download should be periodic.

Table 3-19 lists and describes download parameters.

Table 3-19: Download parameters

| Parameter | Meaning |
|-----------|--------------------|
| frequency | 1 - Periodic |
| | 2 - Single |
| duration | 1 to 65535 minutes |

Example:

```
AT!GPSDLOAD=2,1
```

```
OK
```

If the modem is unable to initiate the download operation, an error code will be returned.

Example:

```
AT!GPSDLOAD=1
```

```
ERRCODE = 17
```

Query satellite information: AT!GPSSATINFO command

The AT!GPSSATINFO command returns information on all satellites in view and those used in the last known, successful location fix.

Information is displayed for up to 12 satellites.

"*" indicates that this satellite was used in the fix calculation.

Meanings of abbreviations:

- SV: Satellite vehicle number (1-32)
- ELEV: Satellite elevation (0-90°)
- AZI: Satellite azimuth (0-360°)
- SNR: Signal to Noise Ratio, in dB (0-99)

Example:

```
AT!GPSSATINFO?
```

```
Satellites in view: 5
```

```
* SV:21 ELEV:56 AZI:323 SNR:40
```

```
* SV:24 ELEV:52 AZI: 43 SNR:37
```

```
* SV:18 ELEV:51 AZI:226 SNR:42
```

```
* SV:26 ELEV:35 AZI: 85 SNR:22
```

```
* SV:16 ELEV:23 AZI:298 SNR:32
```

```
OK
```

Getting base station information for location processing: AT!GPSBSINFO command

The AT!GPSBSINFO command gets base station information for location processing.

Information returned by this command includes:

- Stale indicator (0 = not stale, 1 = stale)
- Base station ID
- System ID
- Network ID
- Timestamp (time that system parameter message was received)
- Base station latitude
- Base station longitude

Note: The ability to access base station information is a compilation feature of the firmware. Therefore, it is possible that this feature is not supported in some releases of the product. In this event, a “Not supported” message is displayed.

Example:

```
AT!GPSBSINFO?
Stale: 0
Time: 200603174135938
BSID: 12      SID: 8      NID: 12
Lat: 33 Deg 7 Min 21.34 Sec N
Lon: 117 Deg 16 Min 32.10 Sec W
OK
```

Terminating a location fix or download session: AT!GPSEND command

The AT!GPSEND command ends an active position session (a position fix session or a download data session, both of which may be running concurrently).

Syntax:

AT!GPSEND=<sesstype>

Table 3-20 lists and describes session parameters.

Table 3-20: Session parameters

| Parameter | Meaning |
|-----------|--|
| sesstype | Type of session to abort <ul style="list-style-type: none"> • 0 = Position fix • 1 = Data download |

If the modem is unable to abort the specified session, or the session is not active, an error code will be returned.

Examples:

```
AT!GPSEND=0
```

```
OK
```

or

```
AT!GPSEND=1
```

```
ERRCODE = 12
```

Accessing default position determination parameters

Getting or setting the IP address: AT!GPSIPADDR command

The AT!GPSIPADDR command gets² or sets the IP address for location processing when TCP/IP is the transport mechanism.

Syntax:

```
AT!GPSIPADDR?
```

or

```
AT!GPSIPADDR=<address>
```

Note: Successfully issuing the SET version of this command does not guarantee that the value has yet been updated in NV. Rather, it indicates that the request to update the value is syntactically correct and that the request is being placed in the queue. It is up to the application that updated the value to verify the NV update by issuing a subsequent GET version of this command (after unlocking the modem²).

Examples:

```
AT!GPSIPADDR=196.168.1.15
```

```
OK
```

-
2. For a successful GET operation, you must first unlock the modem; if the modem has not been unlocked, the GET version returns 255.255.255.255 (block IP address read).

If you need information on the AT command to unlock the modem, please contact your account manager.

or

```
AT!GPSIPADDR?  
64.236.16.84  
OK
```

Getting or setting the port ID for location processing: AT!GPSPORTID command

The AT!GPSPORTID command gets³/sets the port ID for location processing TCP/IP is the transport mechanism.

Syntax:

```
AT!GPSPORTID?
```

or

```
AT!GPSPORTID=<port ID>
```

Note: Successfully issuing the SET version of this command does not guarantee that the value has yet been updated in NV. Rather, it indicates that the request to update the value is syntactically correct and that the request is being placed in the queue. It is up to the application that updated the value to verify the NV update by issuing a subsequent GET version of this command (after unlocking the modem³).

Examples:

```
AT!GPSPORTID=2301  
OK
```

or

```
AT!GPSPORTID?  
4911  
OK
```

-
3. For a successful GET operation, you must first unlock the modem. If you need information on the AT command to unlock the modem, please contact your account manager.

Placing the modem in diagnostic mode: AT!DIAG command

Hardware testing may require normal modem operations to be halted to prevent unintended changes in state. The AT!DIAG command places the modem in diagnostic mode. While in diagnostic mode, normal, automatic operations are suspended.

NMEA interface

The National Marine Electronics Association (NMEA) has developed standards for electrical signal requirements, data transmission protocol and time, and sentence formats used by GPS receivers to transmit location information in a read-only format. The NMEA 0183 (version 3.01) standard is supported by QUALCOMM's AMSS6800 and AMSS6085, and is therefore available to Sierra Wireless customers.

Once the modem is configured properly for location services, standard GPS applications like Microsoft Streets, CoPilot, or other NMEA-based applications can obtain location information from the NMEA COM port. The section titled [NMEA port configuration](#) on page 67 provides setup details.

NMEA port configuration

The NMEA 0183 (version 3.01) standard applies to 4800-baud serial data buses. Proper port configuration is shown in [Figure 3-5](#).

Note: Port selection may vary.

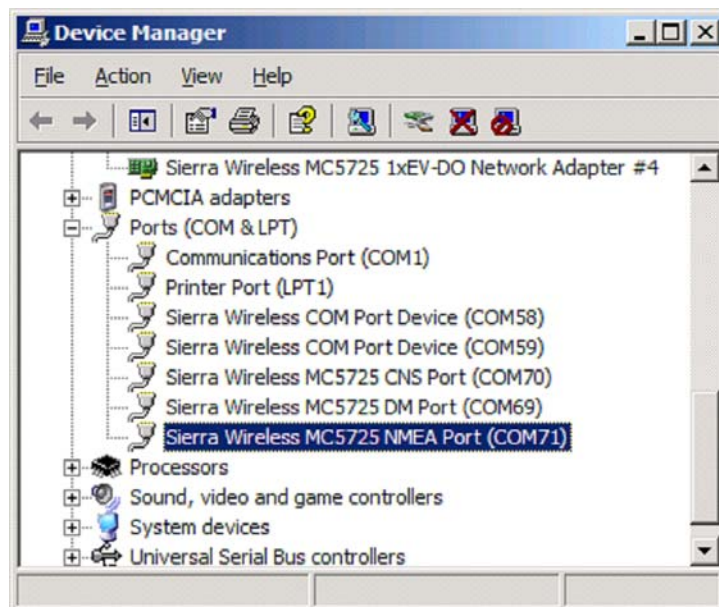


Figure 3-5: NMEA port configuration

Controlling data transmission

Connecting to the [NMEA](#) COM port should automatically enable data transmission. However, some carriers disable this automatic transmission. Therefore, depending on carrier configuration, connecting a device to a NMEA-enabled COM port may or may not enable NMEA-compliant GPS data transmission. Refer to carrier-specific documentation for requirements.

Note: The NMEA port is enabled if a modem's GPS capability is unlocked, but, for NMEA transmission to start automatically, the NV setting for auto tracking must be configured properly in the [PRI](#), and the USB driver must support vendor-specific commands for starting and stopping data transmission. Currently only the USB driver for the Windows CE platform supports these vendor-specific commands. Therefore, on other platforms, opening the NMEA port does not automatically start NMEA data transmission. In these cases, a single [fix](#) (which produces one set of NMEA messages) or a tracking session must be started, using either AT commands or [CnS](#) commands, to receive NMEA messages.

NMEA sentences

Support for sentence formats is carrier-dependent. Refer to carrier-specific documentation for requirements. This section describes current NMEA formats for devices implementing support for NMEA-183 data sentences.

NMEA consists of sentences specifying data format types. The first word of each sentence is a data type which defines the interpretation of the sentence based on the NMEA standard. Device or applications read the data by watching for the data sentence of interest and ignoring sentences that are not of interest. When data is sent in NMEA format, there is no indication as to whether the sentence is read correctly. The recipient determines the accuracy of the data by verifying the checksum. Inaccurate data is ignored and correct information is accepted and overwritten in memory.

Sentence one: \$GPGGA Global Positioning System Fix Data

This sentence provides 3D location and accuracy data. The following example shows comma-separated sentence one data:

```
$GPGGA,160813,3121.7662,N,06141.5628,W,1,04,1.4,181.3,M,
-24.1,M, ,*75
```

The elements of this sentence one example are shown in [Table 3-21](#).

Table 3-21: Sentence elements for sentence one

| Element | Description |
|---------------------------------|---|
| Sentence identifier: \$GPGGA | Global Positioning System Fix Data |
| Time: 160813, or 16:08:13 (UTC) | The Time element indicates the time (in Universal Coordinated Time) at which the fix was taken. |

Table 3-21: Sentence elements for sentence one (Continued)

| Element | Description |
|--|---|
| Latitude: 3121.7662,N, or 31d 21.7662' N | The first two digits are degrees, and the remaining digits are minutes. |
| Longitude: 06141.5628,W, or 61d 41. 5628' W | The first three digits are degrees, and the remaining digits are minutes. |
| Fix quality: 1 | <ul style="list-style-type: none"> 0 = Invalid 1 = GPS fix 2 = Coast Guard Maritime Differential GPS (DGPS) Service fix Fix quality is set to 2 for AGPS fixes because the PDE does the location computing and uses information from a differential GPS station. |
| Number of satellites: 04 | Indicates the number of satellites in view at the time the fix was taken |
| Horizontal dilution of precision (HDOP): 1.4 | HDOP is the predictable horizontal error based on satellite movement. The Returned data indicates the relative accuracy of the horizontal position. |
| Altitude: 181.3,M, or 181.3 | Measured in meters above mean sea level |
| Height of geoid above ellipsoid: -24.1,M, or -24.1 meters above mean sea level | Measured in meters above mean sea level |
| Checksum: *75 | Indicates accuracy of the data. Inaccurate data will be ignored. |

Sentence two: \$GPGSA, GPS dilution of precision and active satellites

This sentence provides GPS and dilution of precision (DOP) data. The following example shows comma-separated sentence two data:

```
$GPGSA,A,3,03,04,,10,13,,,24,,,,,1.5,1.2,1.9*75
```

The elements of this sentence two example are shown in [Table 3-22](#).

Table 3-22: Sentence elements for sentence two

| Element | Description |
|--|--|
| Sentence identifier: \$GPGSA | GPS dilution of precision and active satellites |
| Auto selection of fix type: A | <ul style="list-style-type: none"> A = automatic M = manual |
| Fix type: 3 | <ul style="list-style-type: none"> 1 = no fix 2 = 2D fix 3 = 3D fix |
| Satellite PRNs: 03,04,,10,13,,,24,,,,, | PRNs of the satellites used for the fix. There is space for 12 PRNs in the sentence. |

Table 3-22: Sentence elements for sentence two (Continued)

| Element | Description |
|---------------|---|
| PDOP: 1.5 | PDOP, or position dilution of precision, is the predictable error based on satellite movement. The returned data indicates the relative accuracy of the position. |
| HDOP: 1.2 | HDOP is the predictable horizontal error based on satellite movement. The returned data indicates the relative accuracy of the horizontal position. |
| VDOP: 1.9 | VDOP is the predictable vertical error based on satellite movement. The returned data indicates the relative accuracy of the vertical position. |
| Checksum: *75 | Indicates accuracy of the data. Inaccurate data will be ignored. |

Sentence three: \$GPGSV, satellites in view

This sentence provides information on satellites that might be in view (and the current ability to track this data), and is based on viewing mask and [almanac data](#). The following example shows comma-separated sentence three data: \$GPGSV,3,1,04,02,32,083,46,02,17,308,41,12,07,344,39,14,22,228,46*75

The elements of this sentence three example are shown in [Table 3-23](#).

Table 3-23: Sentence elements for sentence three

| Element | Description |
|--|--|
| Sentence identifier: \$GPGSV | Satellites in view |
| Number of sentences: 3 (All three sentences are not shown in this example.) | Number of sentences required to present full data in this cycle. A single GSV sentence can provide data for up to four satellites. |
| Sentence number: 1 (sentence 1 of 3) | Indicates the number of the current sentence in the cycle |
| Satellites in view: 04 | Indicates the number of satellites in view at the time of the fix |
| Satellite PRNs: 02 | The PRN(s) of satellites used for the fix |
| Elevation: 32 | Measured in degrees |
| Azimuth: 083 | Measured in degrees |
| Signal strength: 46 | Higher numbers represent higher the signal strengths. |
| Checksum: 75 | Indicates accuracy of the data. Inaccurate data will be ignored. |

Sentence four: \$GPRMC, recommended minimum specific GPS/transit data

This sentence provides recommended minimum data for position, velocity, and time (PVT) data. The following example shows comma-separated sentence four data: \$GPRMC,102417,A,2106.31,N,01031.29,E,029.5,079.5,230394,003.2,E*74

The elements of sentence four are shown in [Table 3-24](#).

Table 3-24: Sentence elements for sentence four

| Element | Description |
|---------------------------------|---|
| Sentence identifier: \$GPRMC | Recommended minimum data for position, velocity, and time data |
| Time: 102417, or 10:24:17 (UTC) | Time (in Universal Coordinated Time) at which the fix was taken. |
| Navigation receiver warning: A | <ul style="list-style-type: none"> A = valid position V = warning |
| Latitude: 2106.31,N | The first two digits are degrees, and the remaining digits are minutes. |
| Longitude: 01031.29,E | The first three digits are degrees, and the remaining digits are minutes. |
| Speed: 029.5 | Over ground, in knots |
| Track angle: 079.5 | In degrees true |
| Date: 130307, or March 13, 2007 | The Date element indicates the date (in Universal Coordinated Time) on which the fix was taken |
| Magnetic variation: 003.2,E | Course made good, true |
| Checksum: *74 | Indicates accuracy of the data. Inaccurate data will be ignored. |

Sentence five: \$GPVTG, track made good and ground speed

This sentence provides information on track made good and ground speed. The following example shows comma-separated sentence five data:

\$GPVTG,062.7,T,024.1,M,004.6,N,011.1,K*69

The elements of sentence five are shown in [Table 3-25](#).

Table 3-25: Sentence elements for sentence five

| Element | Description |
|-----------------------------------|----------------------------------|
| Sentence identifier: \$GPVTG | Track made good and ground speed |
| True track made good: 062.7,T | In degrees |
| Magnetic track made good: 024.1,M | (Some receivers may null out) |

Table 3-25: Sentence elements for sentence five (Continued)

| Element | Description |
|-----------------------|--|
| Ground speed: 004.6,N | In knots |
| Ground speed: 011.1,K | In kilometers per hour |
| Checksum: *69 | Indicates accuracy of the data. Inaccurate data will be ignored. |

>> 4: Testing

- Testing for accuracy
- Testing for yield
- Testing for latency
- Standard test equipment

Tests are run to measure **fix** accuracy, yield, and latency. Subsections in this chapter describe each of these tests in detail.

Note: The following tasks can be performed via either AT commands or the CnS interface. However, AT commands are easier to execute and do not require you to be a developer. Thus, this section mainly mentions the AT commands option.

To test location based services, the field engineer needs to be able to:

1. Request the most recent location data. See [Requesting the last known location: AT!GPSLOC command](#) on page 58 for details.
2. Initiate a location based session, while specifying the accuracy and timing requirements of that session. Accuracy is specified by the maxdist parameter, and timing is specified by maxtime. See [Requesting a single location fix: AT!GPSFIX command](#) on page 59 or [Requesting frequent tracking: AT!GPSTRACK command](#) on page 60 for details.
3. Terminate a position determination session. See [Terminating a location fix or download session: AT!GPSEND command](#) on page 64 for details.
4. Obtain status/results of the most recent location based session. See [Getting the current status of location based services: AT!GPSSTATUS command](#) on page 56 for details.

In addition, there are several modem settings that control how location based sessions are performed by the modem's software. When performing tests, these settings are specified in AT commands:

- GPS lock
The AT!GPSLOCK command gets/sets the GPS lock for location processing. For details, see [Opening and unlocking the modem: AT!GPSLOCK command](#) on page 55.
- IP address
The AT!GPSIPADDR command gets/sets the IP address for location processing when TCP/IP is the transport mechanism. For details, see [Getting or setting the IP address: AT!GPSIPADDR command](#) on page 65.
- Port ID
The AT!GPSPORTID command gets/sets the port ID for location processing TCP/IP is the transport mechanism. For details, see [Getting or setting the port ID for location processing: AT!GPSPORTID command](#) on page 66.

Testing for accuracy

Modems are tested to determine the accuracy of their **fix** results.

To test for accuracy:

1. Perform several hundred fixes with a simulation system or at a known (surveyed) location.
2. Compile the results and determine which meters of accuracy 68 percent and 95 percent of the fixes fall within. For example, a test report might indicate that 68 percent of 300 fixes were within 13 meters of truth, while 95 percent of these 300 fixes were within 18 meters of truth.
3. Repeat the same tests under various conditions (indoors, outdoors under open sky, and outdoors with obstructions) using different fix type (**Assisted GPS**, **MS-based**, and **Standalone**).

Testing for yield

A modem's yield is a percentage of how often it produces a **fix** versus how often it returns a timeout with an error.

To test for yield:

1. Perform several hundred fixes.
2. Compile the yield results and calculate a percentage to determine how many of the fixes were successful.
3. Repeat the same tests under various conditions (indoors, outdoors under open sky, and outdoors with obstructions) using different fix type (**Assisted GPS**, **MS-based**, and **Standalone**).

Note: A subset of the yield tests should be performed while moving.

Testing for latency

Modems are tested for latency, which is a measure of the average amount of time from cold starts (without assistance data) and warm starts (with assistance data) required to get a **fix** from start to finish under different conditions. From a cold start (when the modem has no assistance data), getting a fix takes much longer than when the modem has assistance data. Therefore, once the first fix is accomplished, subsequent fixes are faster when requested before the assistance data expires.

*Note: The **almanac data** is valid for up to 6 months, **ephemeris data** is valid for 30 to 120 minutes, and coarse location data is valid for 4 minutes.*

To test for latency:

1. Because fix type and test conditions affect latency, perform several hundred fixes under various conditions (indoors, outdoors under open sky, and

outdoors with obstructions) and using different fix type ([Assisted GPS](#), [MS-based](#), and [Standalone](#)).

2. Compile the yield results and calculate a percentage.

Note: The modem caches [almanac data](#) and [ephemeris data](#). Therefore, assistance data must be cleared to force a cold start. The AT command to clear the data is AT!GPSCLRASSIST ([page 61](#)); the [CnS](#) object is Clear GPS Assistance Data (0x0F20; CNS_PDSM_CLEAR_ASSISTANCE).

Standard test equipment

The standard reference device used by Sierra Wireless is an MC5725 modem and a laptop computer. The modem requires a GPS-capable antenna, preferably a tri-band cellular/PCS/GPS antenna. The antenna should be connected to the AUX RF port and a similar antenna on the main RF port.

Note: The antenna connected to the main RF port does not have to support GPS.


5: Modem Setup

- Setting up Watcher
- Setting up HyperTerminal for NMEA interface operation

Setting up Watcher

This section discusses configuring Watcher for use with a developer's kit. These instructions apply to Watcher version 7.4 or higher.

To set up Watcher:

1. Start Watcher.
2. From the Tools menu, select Display GPS.
3. Click anywhere in the GPS Monitor window, and press <Ctrl>D on your keyboard.
4. When prompted for the advanced user password, in the Advanced window enter `##debug` and click OK.
5. In the GPS Monitor window, click the Menu button  and select Settings.
6. If prompted for the service programming code, enter it and click OK.
7. In the GPS Settings window, click Location Fix.

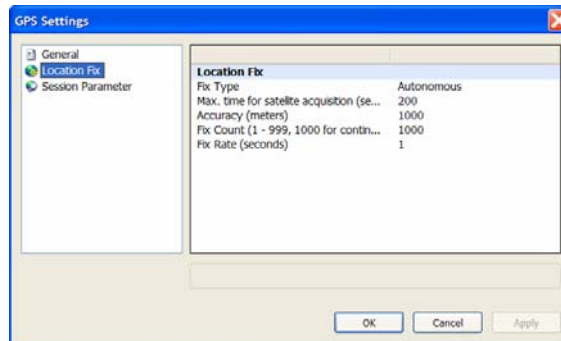


Figure 5-1: GPS Settings: Location Fix

8. Use the drop-down arrow beside the Fix Type field to select a fix type, and enter preferences for Max time for satellite acquisition, Accuracy, Fix Count, and Fix Rate.

9. Click Session Parameter.
The settings are displayed in the right part of the window.

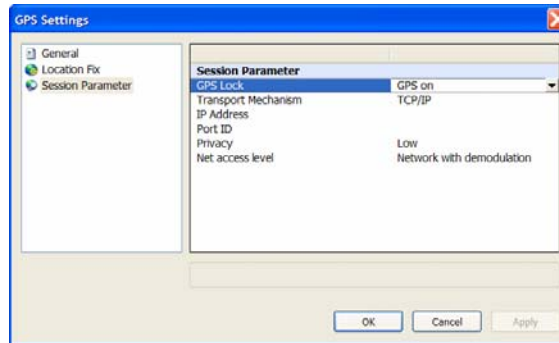



Figure 5-2: GPS Settings: Session Parameter

10. For GPS Lock, select No Lock Active.
11. Click OK.
GPS is now configured to run in the **Standalone** mode.
12. In the GPS Monitor window, click  to start a session.

Setting up HyperTerminal for NMEA interface operation

This section discusses configuring HyperTerminal for use with a developer's kit.

*Note: The NMEA port is enabled if a modem's GPS is unlocked, but for NMEA transmission to start automatically, the NV setting for auto tracking must be configured properly in the **PRI**, and the USB driver must support vendor-specific commands for starting and stopping data transmission. Currently only the USB driver developed for the Windows CE platform supports these vendor-specific commands. Therefore, on other platforms, opening the NMEA port does not start NMEA data transmission automatically. In these cases, a single **fix** (which produces one set of NMEA messages) or a tracking session must be started, using either AT commands or CnS commands, to receive NMEA messages.*

To set up HyperTerminal:

1. Select Device Manager (Start > Settings > Control Panel > System > Hardware > Device Manager) to open the Device Manager.

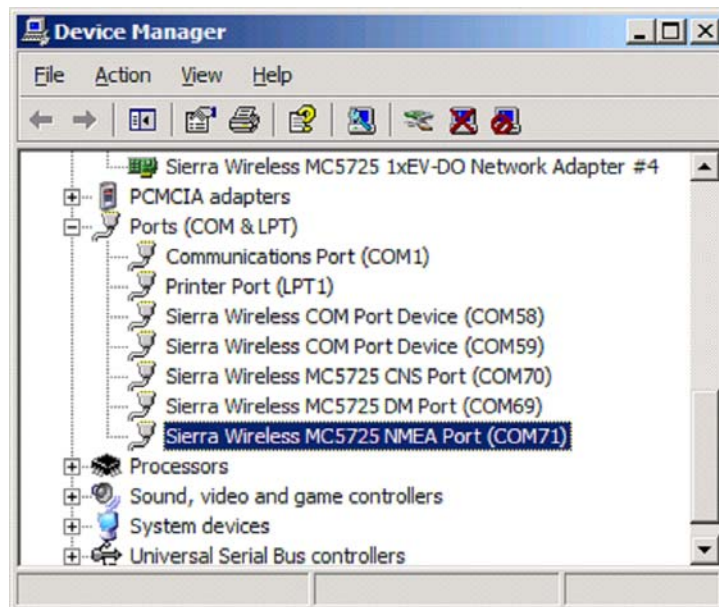


Figure 5-3: Device Manager

2. Note which of the ports is the NMEA COM port. This information will be necessary to set up HyperTerminal for use with attention commands.

Note: HyperTerminal, default settings are acceptable and do not need to be changed.

3. To configure HyperTerminal properties, click the Properties icon. The Properties window opens.

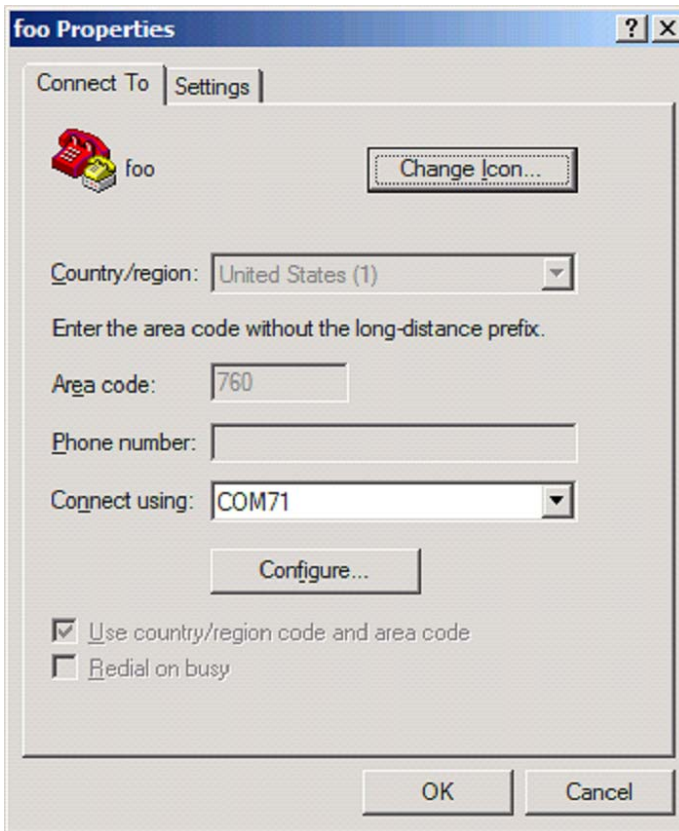


Figure 5-4: HyperTerminal Properties

4. Select a port.
5. Click OK.

Setting up HyperTerminal to run AT commands

To set up HyperTerminal to run AT (attention) commands:

1. Start HyperTerminal.
The Connection Description dialog opens.

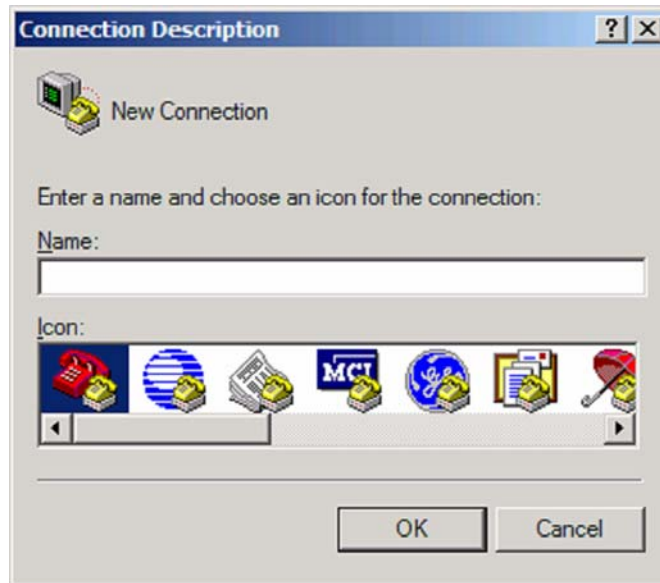


Figure 5-5: Connection Description dialog

2. Click Cancel.
3. Select Properties > Connect using.
The New Connection Properties dialog opens.

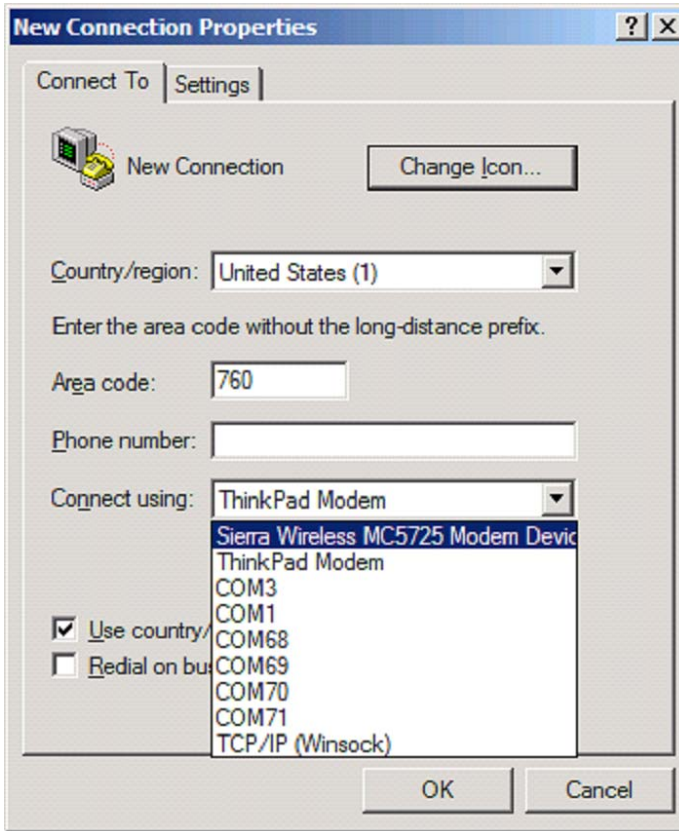


Figure 5-6: New Connection Properties dialog

4. Select the Sierra Wireless modem (in the example above, the MC5725 modem).
5. Click OK.
6. Type AT in the connection window and receive the OK response:
at
OK
7. To verify GPS lock settings using the AT!GPSLOCK command, type:
at!gpslock?
0

Note: For details, see *Opening and unlocking the modem: AT!GPSLOCK command* on page 55.

8. Start a **fix** by typing:

```
at!gpsfix=1,30,200
```

The fix is initiated.

See [Using AT commands](#) on page 54 for information on AT commands.

6: Modem Integration Notes

- Privacy control
- LBS usage interruptions
- Automatic and original equipment manufacturer-required actions
- Behavior during operation
- Data returned
- gpsOne

This chapter addresses privacy control, LBS usage interruptions, and firmware support and integration.

Privacy control

Customer satisfaction requires that a user-selectable privacy feature be included in all location-capable modems. Therefore, modems must be configurable to both disclose location information for emergency purposes (911-only mode), and not to disclose it for non-emergency purposes (location on or off mode). This can be accomplished by means at carriers' discretion, but should be simple and easy for the user to understand. This selection capability must be easily accessible during a call, and while not on a call.

In 911-only mode, the device provides location information using control plane operation during a 911 call. The location on or off mode requires that the modem not initiate a location session or report location information if the privacy setting is set to 911 only mode. If location information is requested while the device is in 911-only mode, the device will return an error. Devices must be set to 911-only mode when shipped and programmed, and users must have the option of password protecting privacy setting controls.

LBS usage interruptions

Devices that do not support simultaneous GPS provide a mechanism for handling calls received or placed while the LBS application is running.

Note: The modem supports simultaneous GPS.

Incoming call interruptions

If an incoming voice call arrives while the LBS application is running, a device that does not support simultaneous GPS alerts the user and displays a message asking whether the user wants to accept the call. If the user accepts the call, the device terminates all GPS activity and stop all access to the GPS engine. The device returns an error code for position requests while the call is in progress. When the LBS application is resumed, the device re-enables GPS functionality.

If the device does support simultaneous GPS, it continues GPS functionality while the application is suspended, or continues in the background if there is an existing or new position request. To support background LBS host applications on devices that do not support simultaneous GPS, the device detects that a voice call is in progress at the modem level instead of relying on the user interface.

Outgoing call interruptions

If a user places an outgoing voice call while the LBS application is running, a device that does not support simultaneous GPS terminates all GPS activity and blocks access to the GPS engine. The device returns an error code for any existing or new position requests. When the LBS application is resumed, the device re-enable GPS functionality. If the device does support simultaneous GPS, the functionality continues while the application is suspended, or in the background if there is an existing or new position request.

To support background LBS host applications on devices that do not support simultaneous GPS, the device detects that a voice call is in progress at the engine level instead of relying on the user interface.

Note: It is the application's responsibility to properly suspend or terminate any active data session.

Automatic and original equipment manufacturer-required actions

When using the [MS-based](#) mode, or an optimal mode that uses MS-based mode, the modem firmware automatically determines whether assistance data is required. If so, the modem connects to the network [PDE](#) and downloads it. Assistance data is considered good for 30-90 minutes.

Configuration

Firmware configuration is dependent on carrier requirements. Refer to carrier specifications for configuration requirements.

Behavior during operation

The modem requires a network connection for [Assisted GPS fixes](#), and may require one for [MS-based fixes](#). If the modem is not in a data connection, it will bring one up automatically to connect to the PDE to obtain assistance data. The modem provides fixes over the [NMEA](#) port and via [CnS](#) (provided that [CnS notifications](#) have been enabled).

Modem call flows and messages

Refer to the [CnS Reference](#) (document number 2130754) for detailed [CnS](#) message flows for location [fixes](#).

Using AT commands creates a call flow similar to that created when using [CnS](#) commands, but there are no notifications. The `AT!GPSSTATUS` command could be used instead to get fix status information. However, during tracking, the status alternates from `ACTIVE` to `SUCCESS` as fixes start and complete, therefore, a status could be missed during polling. Another option is to poll at a periodic rate using `AT!GPSLOC`.

Data returned

The AT command `AT!GPSLOC` and the [CnS](#) command `Position Fix Data (0x0F0A; CNS_PDSM_PD_DATA)` are used to return modem location information. After a successful [fix](#) attempt following modem power up, the AT command `AT!GPSLOC` and the [CnS](#) command `Position Fix Data (0x0F0A; CNS_PDSM_PD_DATA)` are used to return the last known location of the modem.

Refer to the [CnS Reference](#) (document number 2130754) for detailed information on the type and format of data returned for [CnS](#) commands, and to the [Extended AT Command Reference](#) (document number 2130621) for information on the type and format of data returned for AT commands.

gpsOne

The modem implements full support of QUALCOMM [gpsOne](#) functionality.

gpsOne, 7-th generation engine

The modem implements full support of QUALCOMM's seventh-generation [gpsOne](#) engine, including support for Standalone-GPS and Assisted-GPS modes, and [gpsOneXTRA Assistance™](#) technology delivering enhanced Standalone-GPS performance.

7: Resources

- Mobile local search
www.infospaceinc.com
- Child safety
www.sprint.com/familylocator
www.disneymobile.com
- Push-to-talk
www.claritycsi.com
- Mobile navigation
www.vznavigator.com
<http://mobile.garmin.com>
- Emergency response
www.squareloop.com
- National Marine Electronics Association (NMEA)
www.nmea.org
- CDMA Developers Group
www.cdg.org
- QUALCOMM
www.qualcomm.com
- gpsOne
www.cdmatech.com/products/gpsone.jsp
- E911
www.fcc.gov/pshs/services/911-services/enhanced911/Welcome.html

»» A: Glossary and Acronyms

A

| Term or abbreviation | Definition |
|----------------------|---|
| AFLT | Advanced Forward Link Trilateration—a location determination solution that uses CDMA pilot phase measurements in its final calculations. |
| AGPS | Assisted GPS—a location method characterized by a device's reception of GPS assistance data from a PDE to minimize time-to-first-fix, and to provide enhanced GPS sensitivity. AGPS is synonymous with mobile-assisted positioning. |
| almanac data | Almanac data consists of coarse orbital parameters. This information indicates to a GPS receiver where each GPS satellite should be at all times. Each satellite in the GPS transmits almanac data indicating its orbital information, and orbital information for every other satellite in the system. See also ephemeris data . |
| API | Application programming interface |
| CDMA | Code Division Multiple Access—a digital cellular technology for multiplexing, and a method for multiple access. Instead of assigning a specific frequency to each user, CDMA encodes individual conversations with a pseudo-random code sequence. |
| CnS | Control and status CnS commands are the basis of Sierra Wireless' binary host to modem control and status interface. CnS is a proprietary protocol for managing the control and status of the MC5728V Mini Card. |
| Control plane | A position determination architecture based on network-initiated location sessions using data burst messaging in accordance with TIA/EIA/IS-801-1. |
| DGPS | Coast Guard Maritime Differential GPS Service |
| DOP | Dilution of Precision—the predictable error of returned GPS data based on satellite movement. |
| E911 | Enhanced 911—a location technology mandated by the FCC to enable cellular phones to process emergency calls and to provide the geographic location of the device. |
| EOTD | Enhanced Observed Time Differential—one component of a terrestrial LBS method. It uses a triangulation method with multiple mobile stations to determine how far away a handset is from a tower by timing the signal. |
| ephemeris data | Continuously transmitted by each satellite, ephemeris data provides precise information about the status of the satellite. This information is required to determine a geographic position. See also almanac data . |
| FCC | Federal Communications Commission |
| FFA | Field Force Automation. FFA is characterized by the use of software to automate the business tasks of field services personnel. See also SFA . |
| fix | A single request to obtain position information. |

| Term or abbreviation | Definition |
|----------------------|---|
| GIS | Geographic Information Services—a computerized system for capturing and analyzing geographically referenced information. |
| GPS | Global Positioning System. Incorporating a constellation of at least 24 medium Earth orbit satellites, the GPS enables GPS receivers to determine location, speed and direction. |
| HDOP | Horizontal Dilution Of Precision—a predictable horizontal error based on satellite movement. |
| 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. |
| LBS | Location Based Services. Also known as position determination session management (PDSM), LBS is a GPS location feature supported by some carriers and networks. |
| LPS | Location Proxy Server—enables network-initiated applications. In a typical carrier architecture, the LPS contains a user privacy center to authenticate location requests. |
| MS | Mobile Station—a wireless terminal, such as a cell phone, pager, or radio modem, that is considered a station in the Domestic Public Cellular Radio Telecommunications Service. |
| Network fix | A location determination solution calculated by the device. Network solutions reside outside a carrier network and obtain location updates for devices operating on a carrier network. In this context, a network fix is synonymous with an MS-based fix . |
| NMEA | National Marine Electronics Association; has developed standards for electrical signal requirements, data transmission protocol and time, and sentence formats used by GPS receivers to transmit location information in a read-only format. |
| notification | A mechanism to send unsolicited data from either side (host; modem) of the interface; used when no reply or return data is required from the receiver (or conversations do not require stop-and-wait flow control). |
| NV | Non-Volatile (memory)—Random Access Memory that retains its contents even if the power is removed. |
| OEM | Original Equipment Manufacturer—a company that manufactures a product and sells it to a reseller. |
| OMA | Open Mobile Alliance |
| PDE | Position Determination Entity—a network element that provides GPS assistance data to the device and derives a location solution from measurements made by the device. |
| PDOP | Position Dilution Of Precision. See also DOP . |
| PDSM | Position Determination Session Management—a GPS feature supported by some carriers/networks. The CDMA network is used to assist the modem to acquire a GPS location fix . |
| POI | Point Of Interest |

| Term or abbreviation | Definition |
|----------------------|---|
| PRI | Product Release Information—a file that contains the settings used to configure modems for a particular service provider, customer, or purpose. |
| PRN | PseudoRandom Noise code of a satellite. |
| QoS | Quality Of Service |
| registered | registered (LBS client)—a client that is interested in receiving notification about LBS events. |
| requesting | requesting (LBS client)—a client that has initiated the current LBS session or event. A requesting client is typically also a registered client. |
| SFA | Sales Force Automation. SFA is characterized by the use of software to automate the business tasks of sales personnel. See also FFA . |
| SUPL | Secure User Plane Location—a standard developed by the Open Mobile Alliance (OMA) to enable LBS on any AGPS -capable phone on any mobile network. |
| TBT | Turn-By-Turn |
| TTFF | Time To First Fix |
| user plane | A position determination architecture based on mobile-initiated location sessions over a TCP/IP data connection. |
| VDOP | Vertical Dilution Of Precision—a predictable vertical error based on satellite movement. |
| WCDMA | Wideband Code Division Multiple Access. See also CDMA . |

Index

Symbols

IDIAG command, 67
IGPSBSINFO command, 64
IGPSCLRASSIST command, 61
IGPSDLOAD command, 62
IGPSEND command, 64
IGPSFIX command, 59
IGPSIPADDR command, 65
IGPSLOC command, 58
IGPSLOCK command, 58
IGPSNMEAENABLE command, 58
IGPSPORTID command, 66
IGPSSATINFO command, 63
IGPSSTATUS command, 56
IGPSTRACK command, 60
\$GPGGA (NMEA sentence one), 68
\$GPGSA (NMEA sentence two), 69
\$GPGSV (NMEA sentence three), 70
\$GPRMC (NMEA sentence four), 71
\$GPVTG (NMEA sentence five), 71

Numerics

0x0F01 (CnS object PD Download), 51
0x0F02 (CnS object PD Get Position), 49
0x0F03 (CnS object PDSM Status), 46
0x0F04 (CnS object PD Track), 50
0x0F05 (CnS object PD End Session), 52
0x0F06 (CnS object PA Parameter Settings), 53
0x0F07 (CnS object Location notification status), 46
0x0F08 (CnS object Download notification status), 46
0x0F09 (CnS object Parameter notification status), 46
0x0F0A (CnS object Position Fix Data), 49, 50, 87
0x0F0B (CnS object Position Fix Session Done), 50, 52
0x0F0C (CnS object Position Fix Session End), 49, 50, 52
0x0F0D (CnS object Position Fix Session Begin), 49, 50
0x0F0F (CnS object Download Data Session Begin), 51
0x0F10 (CnS object Download Data), 51
0x0F11 (CnS object Download Data Session Done), 51, 53
0x0F12 (CnS object Download Data Session End), 51, 53
0x0F13 (CnS object TCP/IPAddress), 53
0x0F14 (CnS object GPS Lock), 46, 53
0x0F15 (CnS object Transport Mechanism), 53
0x0F16 (CnS object Port ID), 53
0x0F18 (CnS object Network Access Permissions), 53
0x0F19 (CnS object Base Station information), 33, 52
0x0F1F (CnS object Satellite information), 52
0x0F20 (CnS object Clear GPS Assistance Data), 51, 75
0xF17 (CnS object Privacy Level), 53

A

Accuracy Optimal, 15
accuracy, testing for, 74
active client, 45
active satellites, 69
Advanced Forward Link Trilateration (AFLT), 15
AFLT (Advanced Forward Link Trilateration), 15
almanac data
 description, 41
 API, 31
 AT, 62
 clear, 30, 51, 61
 CnS, 51
antenna for test equipment, 75
API FAQ, 40
application
 enterprise served from network, 21
 laptop local, 20
 laptop served from a network, 20
applications, LBS, 19
assistance data, clear
 API, 30
 AT, 61
 CnS, 51
Assisted GPS fix, 14
AT commands
 overview, 54
 HyperTerminal, setting up, 81
azimuth, satellite
 AT, 63

B

base station information
 AT, 64
 CnS, 33, 52
BSA (Carrier Base Station Almanac), 17

C

call flows, 87
Carrier Base Station Almanac (BSA), 17
carrier differences in LBS implementation, 42
child safety, 89
client, LBS, types of, 45
CnS
 overview, 43
 objects used in LBS, list of, 44
cold start, force
 API, 30
 AT, 61
 CnS, 51

- configuration
 - firmware, 86
 - modem, 42
- continuous tracking, 14, 18, 29
- control plane, 17

D

- Data Optimal, 15
- data returned, 87
- data transmission, NMEA, controlling, 68
- default PDSM parameters
 - change (API), 37
 - change (AT), 65-??
 - change (CnS), 54
 - get (API), 35
 - get (AT), 65-??
 - get (CnS), 53
- device-initiated applications, 19
- !DIAG command, 67
- diagnostic mode of modem, 67
- dilution of precision (DOP), 69
- DOP (dilution of precision), 69
- Download notification status object, 46
- download session
 - terminate (API), 35
 - terminate (AT), 64
 - terminate (CnS), 52
- download session, last
 - status of (CnS), 48
 - timestamp of (AT), 56
 - timestamp of (CnS), 48
- download, last
 - status of (CnS), 47
 - timestamp of (AT), 56
 - timestamp of (CnS), 47
- downloading ephemeris and almanac data
 - API, 31
 - AT, 62
 - CnS, 51

E

- E911
 - description, 17
 - FCC web site, 89
- elevation, satellite
 - AT, 63
- emergency response, 89
- enabling notifications (CnS), 46
- End Session object, 52
- Enhanced 911 (E911), 17
- Enhanced observed time differential (EOTD), 14
- enterprise served from network application, 21
- EOTD (Enhanced observed time differential), 14

- ephemeris data
 - description, 41
 - API, 31
 - AT, 62
 - clear, 30, 51, 61
 - CnS, 51
- error codes
 - API, 38
 - AT, 57
- error conditions
 - API, 38
 - AT, 57

F

- failure codes
 - API, 38
 - AT, 57
- FAQ
 - API, 40
- FFA (field force automation), 19
- field force automation (FFA), 19
- finder applications, 19
- firmware configuration, 86
- fix
 - Assisted GPS, 14
 - control plane, 17
 - error conditions (API), 38
 - error conditions (AT), 57
 - initiation, 17
 - MS-based, 14
 - satellite information (API), 33
 - satellite information (AT), 63
 - satellite information (CnS), 52
 - single (API), 26
 - single (AT), 59
 - single (CnS), 49
 - standalone, 13
 - status of last fix (CnS), 47
 - status of last fix session (CnS), 48
 - terminate (API), 34
 - terminate (AT), 64
 - terminate (CnS), 52
 - timestamp of last fix (AT), 56
 - timestamp of last fix (CnS), 47
 - timestamp of last fix session (AT), 56
 - timestamp of last fix session (CnS), 48
 - user plane, 17
- fix modes, 13
- force a cold start
 - API, 30
 - AT, 61
 - CnS, 51
- frequent tracking
 - API, 28
 - AT, 60
 - CnS, 50
- frequently asked questions (FAQ)
 - API, 40

G

geo-coding, 20
 geo-data, 20
 geographic information services (GIS), 20
 geo-services, 20
 GIS (geographic information services), 20
 Global Positioning System Fix Data (NMEA sentence one), 68
 \$GPGGA (NMEA sentence one), 68
 \$GPGSA (NMEA sentence two), 69
 \$GPGSV (NMEA sentence three), 70
 \$GPRMC (NMEA sentence four), 71
 GPS dilution of precision and active satellites (NMEA sentence two), 69
 GPS dynamic mode. See Smart Mode
 GPS IP address
 API, 36
 AT, 65
 CnS, 53
 GPS Lock
 API, 36
 CnS, 53
 GPS Lock object, 46, 53
 GPS Network access permission
 API, 36, 38
 CnS, 53
 GPS PortID
 API, 36
 AT, 66
 CnS, 53
 GPS Privacy
 API, 36, 38
 CnS, 53
 GPS PTLM mode
 API, 36, 37
 CnS, 53
 IGPSSBSINFO command, 64
 GPS-capable antenna, 75
 IGPSSCLRASSIST command, 61
 IGPSSDLOAD command, 62
 IGPSSSEND command, 64
 IGPSSFIX command, 59
 IGPSSIPADDR command, 65
 IGPSSLOC command, 58
 IGPSSLOCK command, 55
 IGPSSNMEAENABLE command, 58
 gpsOne, 16, 87, 89
 IGPSSPORTID command, 66
 IGPSSSATINFO command, 63
 IGPSSSTATUS command, 56

IGPSTRACK command, 60
 \$GPVTG (NMEA sentence five), 71
 ground speed and track made good (NMEA sentence five), 71

H

HyperTerminal, setting up
 AT commands, 81
 NMEA, 78

I

incoming call interruptions, 86
 initiation of fixes, 17
 integration notes, 85
 Internet resources, 89
 interruptions, 86
 IP address, GPS
 API, 36
 AT, 65
 CnS, 53

L

laptop local application, 20
 laptop served from a network application, 20
 last download
 status of (CnS), 47
 timestamp of (AT), 56
 timestamp of (CnS), 47
 last download session
 status of (CnS), 48
 timestamp of (AT), 56
 timestamp of (CnS), 48
 last known location
 API, 26
 AT, 58
 CnS, 49
 latency, testing for, 74
 LBS
 applications, 19
 implementation, carrier differences in, 42
 NMEA, enable (AT), 58
 status (API), 46
 status (AT), 56
 usage interruptions, 85
 local search, mobile, 89

- location fix
 - error conditions (API), 38
 - error conditions (AT), 57
 - satellite information (API), 33
 - satellite information (AT), 63
 - satellite information (CnS), 52
 - single (API), 26
 - single (AT), 59
 - single (CnS), 49
 - status of last fix (CnS), 47
 - status of last fix session (CnS), 48
 - terminate (API), 34
 - terminate (AT), 64
 - terminate (CnS), 52
 - timestamp of last fix (AT), 56
 - timestamp of last fix (CnS), 47
 - timestamp of last fix session (AT), 56
 - timestamp of last fix session (CnS), 48
- Location Manager., 20
- Location notification status object, 46
- location proxy server (LPS), 19
- location, last known
 - API, 26
 - AT, 58
 - CnS, 49
- LPS (location proxy server), 19

M

- mapping applications, 19
- Microsoft Streets and Trips, 21
- mobile local search, 89
- mobile navigation, 89
- mobile-autonomous positioning, 13
- modem integration notes, 85
- modem setup, 42, 77
- modem, diagnostic mode, 67
- MS-based fix, 14
- MS-based Smart Mode
 - description, 15

N

- navigation applications, 19
- navigation, mobile, 89
- network access permission, GPS
 - API, 36, 38
 - CnS, 53
- Network Access Permissions object, 53
- network-initiated applications, 19
- NMEA
 - overview, 67
 - data transmission, controlling, 68
 - enable (AT), 58
 - HyperTerminal, setting up, 78
 - port configuration, 67
 - sentences, 68

- notification
 - status objects, 46
- notifications, enabling (CnS), 46

O

- OEM-required actions, 86
- open modem
 - API, 25
 - AT, 55
 - CnS, 46
- outgoing call interruptions, 86

P

- PA Parameter Settings object, 53
- PD Download object, 51
- PD End Session object, 52
- PD Get Position object, 49
- PD Track object, 50
- PDE
 - access frequency, restrictions, 31
 - description, 17
 - unavailable, reasons, 16
- PDSM
 - default parameters, change (API), 37
 - default parameters, change (AT), 65-??
 - default parameters, change (CnS), 54
 - default parameters, get (API), 35
 - default parameters, get (AT), 65-??
 - default parameters, get (CnS), 53
 - terminate session (API), 34
 - terminate session (CnS), 52
- PDSM Status object, 46
- POI (point of interest) applications, 19
- port configuration, NMEA, 67
- Port ID object, 53
- PortID, GPS
 - API, 36
 - AT, 66
 - CnS, 53
- Position determination entity (PDE)
 - description, 17
- Position Determination Session Management. See PDSM
- position determination session. See PDSM.
- Position Fix Data object, 49
- privacy control, 85
- Privacy Level object, 53
- privacy, GPS
 - API, 36, 38
 - CnS, 53
- PTLM mode, GPS
 - API, 36, 37
 - CnS, 53
- push-to-talk, 89
- PVT data, 71

R

recommended minimum specific GPS/transit data (NMEA sentence four), 71
 reference device, 75
 registered client, 45
 requesting client, 45
 resources, 89
 reverse geo-coding, 20
 route planning, 20

S

safety, child, 89
 sales force automation (SFA), 19
 satellite information
 API, 33
 AT, 63
 CnS, 52
 satellites in view
 API, 33
 AT, 63
 CnS, 52
 NMEA sentence three, 70
 satellites, active, 69
 search, mobile local, 89
 Secure User Plane Location (SUPL), 18
 Sentence five (\$GPVTG), 71
 Sentence four (\$GPRMC), 71
 Sentence one (\$GPGGA), 68
 Sentence three (\$GPGSV), 70
 Sentence two (\$GPGSA), 69
 sentences, NMEA, 68
 session, download, terminate
 API, 35
 AT, 64
 CnS, 52
 setup of modem, 42, 77
 SFA (sales force automation), 19
 Signal to Noise Ratio, satellite
 AT, 63
 single location fix
 API, 26
 AT, 59
 CnS, 49
 Smart Mode
 overview, 15
 MS-based, 15
 Standalone, 16
 SNR, satellite
 AT, 63
 software interface options, 23
 Speed Optimal, 15
 standalone fix, 13
 Standalone Smart Mode
 description, 16
 standard test equipment, 75
 status of last download (CnS), 47
 status of last download session (CnS), 48

status of last fix (CnS), 47
 status of last fix session (CnS), 48
 status of LBS
 API, 46
 AT, 56
 Streets and Trips, Microsoft, 21
 SUPL (Secure User Plane Location), 18
 SwiGetLbsPaGpsLock, 36
 SwiGetLbsPaIpAddr, 36
 SwiGetLbsPaNetAccess, 36
 SwiGetLbsPaParam, 35
 SwiGetLbsPaPortId, 36
 SwiGetLbsPaPrivacy, 36
 SwiGetLbsPaPtmMode, 36
 SwiGetLbsPdData, 26
 SwiGetLbsSatInfo, 33
 SwiSetLbsClearAssistance, 30
 SwiSetLbsPaGpsLock, 25
 SwiSetLbsPaNetAccess, 38
 SwiSetLbsPaPtmMode, 37
 SwiSetLbsPdDownload, 31
 SwiSetLbsPdEndSession, 34, 35
 SwiSetLbsPdGetPos, 26
 SwiSetLbsPdTrack, 28
 SwiSetLbsPaPrivacy, 38

T

TBT (turn-by-turn) applications, 19
 TCP/IPAddress object, 53
 terminate a download session
 API, 35
 AT, 64
 CnS, 52
 terminate location fix session
 API, 34
 AT, 64
 CnS, 52
 terminate PDSM session
 API, 34
 CnS, 52
 test equipment, standard, 75
 testing
 overview, 73
 accuracy, 74
 latency, 74
 standard test equipment, 75
 yield, 74
 timeout, tracking process, 40
 timestamp of last download
 AT, 56
 CnS, 47
 timestamp of last download session
 AT, 56
 CnS, 48
 timestamp of last fix
 AT, 56
 CnS, 47

timestamp of last fix session

AT, 56

CnS, 48

track made good and ground speed (NMEA sentence five), 71

tracking process, time out, 40

tracking, continuous, 29

tracking, frequent

API, 28

AT, 60

CnS, 50

Transport Mechanism object, 53

transport mode, GPS

API, 36, 37

CnS, 53

turn-by-turn (TBT) applications, 19

U

unlock modem

API, 25

AT, 55

CnS, 46

usage interruptions, LBS, 85

user plane, 18

V

vehicle tracking, 19

W

Watcher, 21, 77

Wireless enhanced 911, 17

Y

yield, testing for, 74