

# MT 3050



## AT Command Reference



# General

## TERMS OF USE OF NEW MATERIALS - PLEASE READ CAREFULLY

From time to time, Novatel Wireless, in its sole discretion, may make available for download on its website ([www.novatelwireless.com](http://www.novatelwireless.com)), or may transmit via mail or email, updates or upgrades to, or new releases of, the firmware, software or documentation for its products (collectively, 'New Materials'). Use of such New Materials is subject to the terms and conditions set forth below, and may be subject to additional terms and conditions as set forth in Novatel Wireless's Technical Support Policy (posted on its website) and/or any written agreement between the user and Novatel Wireless.

All New Materials are provided AS IS. Novatel Wireless makes no warranty or representation with respect to the merchantability, suitability, functionality, accuracy or completeness of any such New Materials. The user of such New Materials assumes all risk (known or unknown) of such use. Novatel Wireless reserves all rights in such New Materials. The user shall have only a revocable and limited license to use such New Materials in connection with the products for which they are intended. Distribution or modification of any New Materials without Novatel Wireless's consent is strictly prohibited.

IN NO EVENT WILL NOVATEL WIRELESS BE RESPONSIBLE FOR ANY INCIDENTAL, INDIRECT, CONSEQUENTIAL OR SPECIAL DAMAGES AS A RESULT OF THE USE OF ANY NEW MATERIALS. NOVATEL WIRELESS'S MAXIMUM LIABILITY FOR ANY CLAIM BASED ON THE NEW MATERIALS SHALL NOT EXCEED FIFTY U.S. DOLLARS (\$50).

## Version Verification

Please ensure you have the latest version of this document by downloading it from [www.novatelwireless.com](http://www.novatelwireless.com)

# Copyright

© 2013 Novatel Wireless. All rights reserved. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Novatel Wireless.

Novatel Wireless and the Novatel Wireless logo are either registered trademarks or trademarks of Novatel Wireless. in the United States.

251 Renner Pkwy

Richardson, TX 75080 USA

Phone: (972) 633-4400

Fax: (972) 633-4444

Email: [info@enfora.com](mailto:info@enfora.com)

[www.novatelwireless.com](http://www.novatelwireless.com)

# Table of Contents

---

<b>Introduction .....</b>	<b>1</b>
Document Scope .....	2
Platform Reference And Use .....	2
Command Syntax .....	3
Standard AT Commands .....	4
<b>General Commands .....</b>	<b>5</b>
AT\$MDN Mobile Directory Number .....	6
AT\$MSID Mobile Station Identification .....	7
AT\$NWGMR Manufacturer Revision .....	8
AT\$PKG Request Firmware Package .....	9
AT\$SPN Software Part Number .....	11
AT\$SRN Module Serial Number .....	12
AT+CGMI Request Manufacturer Identification .....	13
AT+CGMM Request Manufacturer Model Identification .....	14
AT+CGMR Request Revision Identification .....	15
AT+CGSN Product Serial Number .....	16
AT+GMI TA Manufacturer ID .....	17
AT+GMM TA Model ID .....	18
AT+GMR TA Revision Number .....	19
AT+GSN TA Serial Number .....	20
ATI Manufacturer Information About TA .....	21
<b>Network Commands .....</b>	<b>22</b>
AT\$AREG Auto Registration .....	23
AT\$CREG Network Registration Info .....	24
AT\$MDSTAT Query Modem Status .....	26

---

AT\$NETMON Monitor Network Availability .....	30
AT\$NWACTIVATION Provision MDN And MIN .....	32
AT\$OTASP OTA Service Provisioning Status .....	33
AT\$ATPASSWD Set Authorization For AT Commands .....	35
AT\$QCDNSP DNS Primary IP Address .....	37
AT\$QCDNSS Secondary DNS IP Address .....	38
AT\$QCMIP Mobile IP .....	39
AT\$QCMIPPEP Enable Or Disable Current MIP Profile .....	40
AT\$QCMIPGETP Get Profile Information .....	41
AT\$QCMIPMASS MN-AAA Shared Secrets .....	42
AT\$QCMIPMHSS MM-HA Shared Secrets .....	43
AT\$QCMIPNAI Network Access Identifier .....	44
AT\$SPC Service Provisioning Code .....	45
AT+CNUM Subscriber Number .....	46
AT+CSS Serving System .....	47
AT+CSQ Signal Quality And Bit Error Rate .....	49
<b>META Commands .....</b>	<b>50</b>
AT+CEER Extended Error Reporting .....	51
AT\$RESET Reset Modem .....	52
AT\$V Display Smart Agent Configuration Settings .....	53
AT&F Set All TA Parameters To Factory Defined Configuration .....	54
AT&V Display Current Profile .....	55
AT&W Save Current Settings .....	56
ATE Command Echo Mode .....	57
ATQ Result Code Suppression .....	58
ATS3 Command Line Termination Character .....	59
ATS4 Response Formatting Character .....	60
ATS5 Editing Character .....	61

---

ATV Response Format .....	62
ATX CONNECT Result .....	63
<b>SMS Commands .....</b>	<b>64</b>
AT\$QCMSS Send Message From Storage .....	65
AT\$QCMGD Delete Message .....	66
AT\$QCMGF SMS Format .....	67
AT\$QCMGL List Messages .....	68
AT\$QCMGR Read Message .....	70
AT\$QCMGS Send Message .....	71
AT\$QCMGW Write Message To Memory .....	72
AT\$QCNMI New Message Indications To TE .....	73
AT\$QCPMS Preferred Message Storage .....	75
AT\$QCSMP Set Text Mode Parameters .....	77
AT\$SMSURC Enable URC To Be Sent By SMS .....	78
AT+CMGF SMS Format .....	79
AT+CMGD Delete Message .....	80
AT+CMGL List Messages .....	81
AT+CMGR Read Message .....	83
AT+CMGS Send Message .....	84
AT+CMGW Write Message To Memory .....	85
AT+CMSS Send Message From Storage .....	86
AT+CNMI New Message Indication To TE .....	87
AT+CPMS Preferred Message Storage .....	90
<b>API Commands .....</b>	<b>92</b>
TCP API Commands .....	93
AT\$ACKTM Acknowledgment Message Period & Retry Number .....	93
AT\$TCPAPI TCP API Control .....	95
AT\$TCPIDLETO TCP API Idle Timeout .....	97

---

AT\$TCPRESTR TCP API Restart .....	98
AT\$TCPRETRYTO TCP API Retry Timeout .....	99
AT\$TCPSRC TCP API Source Ports .....	100
AT\$TCPSTATS TCP API Statistics .....	101
AT\$TCPURC Enable URC Over TCP API .....	103
<b>Message Commands .....</b>	<b>104</b>
AT\$APIOPT Enable API Optional Header Fields .....	105
AT\$APIPWD API Password .....	109
AT\$CHKIN Configure Device Check-in .....	110
AT\$FID Get Factory ID .....	112
AT\$FRIEND Modem Friends .....	113
AT\$IDENT Modem Identification Used In API Optional Header .....	116
AT\$MDMID Modem ID .....	117
AT\$MLQSZ Message Log Queue Size .....	118
AT\$MSGLOGCL Message Log Clear .....	122
AT\$MSGLOGDMP Dump Unsent Messages To Serial Port .....	123
AT\$MSGLOGEN Message Log Enable .....	125
AT\$MSGLOGRD Message Log Read Data .....	126
AT\$MSGSEND Message Send .....	129
AT\$SMSDA Destination Address For SMS Messages .....	131
AT\$SMSDAEN Enable/Disable AT Commands Over SMS .....	133
AT\$UDPMMSG Send And Receive UDP Messages .....	134
AT\$WAKEUP Modem To Server Wakeup/Keep Alive .....	136
AT%SNCNT Query (or Reset) The Byte Counters .....	138
<b>Event Commands .....</b>	<b>139</b>
AT\$ETSAV# Event Timer Save Configuration .....	140
AT\$EVCID User Defined Incoming Call Number Event .....	141
AT\$EVDEL Delete Event .....	143

AT\$EVDELA Delete Event (All) .....	144
AT\$EVDELR Delete A Range Of Event Groups .....	145
AT\$EVENT User Defined Input/Output .....	146
AT\$EVGQRY Event Group Query .....	148
AT\$EVNTRY Event Count Query .....	149
AT\$EVTEST Generate Test Input Event .....	150
AT\$EVTIM# User Defined Input Event Timers .....	151
AT\$EVTIMQRY Event Counter .....	152
AT\$EVTOFF Event Engine Disable .....	153
AT\$EVTQRY Query The State Or Value Of The Specified Input Event .....	154
AT\$STOATEV Store AT Command Events .....	155
AT\$USRVAL User Value .....	157
<b>RTC Commands .....</b>	<b>158</b>
AT\$RTCALRM Real Time Clock Alarm .....	159
AT\$RTCCLRA Real Time Clock Clear Alarm .....	161
AT\$RTCRSET RTC Report Reset State .....	162
AT\$RTCTIME Real Time Clock Time .....	164
AT\$RTCWAKE Real Time Alarm Wake .....	166
<b>FOTA Commands .....</b>	<b>167</b>
AT\$FOTACFG Configure Firmware Upgrade .....	168
AT\$FOTAGET Get Firmware Upgrade OTA .....	171
AT\$FOTAUPG Upgrade Current Firmware .....	173
<b>FTP Commands .....</b>	<b>174</b>
AT\$ATEXEC Execute AT Command File .....	175
AT\$FFS Generic Flash File System Access .....	177
AT\$FTPABORT Aborts Current Data Operation With FTP Server .....	181
AT\$FTPBLK Change Current FTP BLock Size .....	182



---

AT\$FTPCFG Configure FTP Parameters .....	183
AT\$FTPCHDIR Change Current Working Directory .....	184
AT\$FTPCLOSE Closes The Connection To The FTP Server .....	185
AT\$FTPDIR Directory Listing .....	186
AT\$FTPGET Retrieve A Remote File .....	187
AT\$FTPOPEN Opens FTP Connection .....	188
AT\$FTPPUT Store A Local File On The FTP Server .....	189
AT\$FTPRST Restart The Last FTP GET Operation .....	190
<b>IP Router Commands .....</b>	<b>191</b>
AT\$FLFILT Filter ICMP Echo Request .....	192
AT\$NETIP Display Network Assigned IP & DNS .....	193
AT\$PING Send ICMP Echo Request .....	194
<b>GPS Commands .....</b>	<b>195</b>
AT\$ALTOSI Set And Query GPS Overspeed Interval .....	196
AT\$EXCACC Excessive Acceleration .....	198
AT\$EXCDEC Excessive Deceleration .....	199
AT\$GEOFNC Geo Fencing A Circular Area .....	200
AT\$GFDBNC Set Geofence Debounce Count .....	201
AT\$GFDEL Delete A Range Of Geo-Fences .....	202
AT\$GFIDX# Geo Fencing Index .....	203
AT\$GPSCMD GPS Command .....	204
AT\$GPSFD Restore GPS Filter Defaults .....	205
AT\$GPSFLT GPS Speed Filter .....	206
AT\$GPSLCL Configure Sending Of GPS Message To The Serial Port .....	211
AT\$GPSLNA Enable/Disable Internal LNA .....	213
AT\$GPSOSI Set And Query GPS Overspeed Interval .....	214
AT\$GPSQUAL GPS Quality Filters .....	216
AT\$GPSRD Read Current GPS ASCII Data .....	218

---

AT\$GPSRST Set And Query GPS Reset Interval .....	220
AT\$GPSVAL GPS Speed Validation .....	222
AT\$ODOMETER GPS Trip Odometer .....	223
AT\$PLYCLR Clear Polygonal Geofence .....	224
AT\$PLYFN# Geo Fencing A Polygonal Area .....	225
AT\$PWRSV Enable Power Save Mode .....	228
AT\$TODOM Virtual Trip Odometer .....	229
<b>Battery Commands .....</b>	<b>231</b>
AT\$OBDBAT Enable/Disable Backup Battery .....	232
AT\$OBDBBT Backup Battery Timer .....	234
AT\$OBDBQS Backup Battery Quick Shutdown .....	235
<b>OBD Commands .....</b>	<b>236</b>
AT\$OBDAKL Acceleration Thresholds .....	237
AT\$OBDAKLN Set Down Accelerometer Threshold .....	238
AT\$OBDAKL Auto Discovery Timer .....	239
AT\$OBDAKL Set Left Turn Accelerometer Threshold .....	240
AT\$OBDAKL Set Any Motion Parameters .....	241
AT\$OBDAKL Set Right Turn Threshold .....	243
AT\$OBDAKL OBD2 Set Up Accelerometer Threshold .....	244
AT\$OBDAKL Clear The OBD2 User Trip Odometer .....	245
AT\$OBDAKL Deceleration Thresholds .....	246
AT\$OBDAKL Discovery Configuration .....	247
AT\$OBDAKL Engine Coolant Temperature Threshold .....	248
AT\$OBDAKL Excessive Engine Speed Thresholds .....	249
AT\$OBDAKL Set Thresholds To Factory Defaults .....	250
AT\$OBDAKL Use OBD Speed In Place Of GPS Speed .....	251
AT\$OBDAKL Idle Speed Threshold .....	252
AT\$OBDAKL OBD2 Ignition Off Configuration .....	253

AT\$OBDLBL Low Battery Level Threshold .....	254
AT\$OBDLED OBD2 LED Timer .....	255
AT\$OBDLFL Low Fuel Level Threshold .....	256
AT\$OBDSAV Save OBDII Thresholds To Memory .....	257
AT\$OBDSPD Vehicle Speed Thresholds .....	258
AT\$OBDTYP Protocol Type .....	259
AT\$OBDVER Auxiliary Processor .....	261
<b>Miscellaneous Commands .....</b>	<b>262</b>
AT\$IGNDBNC Ignition Debounce .....	263
AT\$HBRST Automatic Modem Reset .....	264
AT\$OFF Power Off Command .....	266
AT%SLEEP Select Level Of Sleep Mode .....	267
<b>Event Tables .....</b>	<b>269</b>
Event Engine .....	270
Event Category .....	270
Event Type .....	270
Input Event Table .....	273
Output Event Table .....	280
Bit Field Tables .....	284
Bit-Field Table Selection .....	284
Bit-Field Table 0 - Legacy (0,0) .....	284
Bit-Field Table 2 - (1,0) .....	288
User Variable Index Table .....	293
GPIO Flash Table .....	302
OBD Bit Field Tables .....	302
OBD BIT-FIELD TABLE SELECTION .....	302
BIT-FIELD TABLE 0 - OBD (0,0) .....	302
<b>Appendix - Result Codes .....</b>	<b>306</b>

---

Appendix A – Result Codes .....	307
Result Codes .....	307
UNSOLICITED RESULT Codes .....	307
SMS Unsolicited Result Codes .....	308
<b>Appendix - Error Codes .....</b>	<b>309</b>
Appendix B – Error Codes .....	310
General Error Codes .....	310
SMS Error Codes (+CMS) .....	311
Release Causes For Extended Error Reporting (+CEER) .....	314
Extended Error Codes .....	316
<b>Index .....</b>	<b>317</b>

# Introduction

---

# Document Scope

The following documentation pertains to the AT Command Set to be used in conjunction with the Novatel Wireless Device identified in the title of this document.

## Platform Reference And Use

The device will be referred to using various terms, to include: MS (Mobile Station), TA (Terminal Adapter), DCE (Data Communication Equipment), or ME (Mobile Equipment).

The device can be controlled via the use of a DTE (Data Terminal Equipment) platform by issuing the AT commands via a serial interface.

# Command Syntax

The attention or “AT” prefix is required prior to entering any command. All commands require a carriage return or <CR> following the entry of the desired command. All command responses are encapsulated by a carriage return and line feed or <CR><LF>. The ASCII display of these characters is suppressed with only the modem response being presented.

In addition to terminating AT commands, the carriage return <CR> is also used to abort commands that are executing.

Most AT commands complete immediately so there is no opportunity to abort them, for instance ATI. However, Some commands like AT+COPS or AT+CFUN can actually take several seconds to complete. The AT command interface is said to be in execution mode when a command is running and has not returned a result code (OK/ERROR). A second <CR> entered while the AT command interface is in execution mode will abort the command and return the interface to command mode.

Some AT commands require additional input, for instance AT+CMGS. After terminating the AT+CMGS command with a <CR> the AT command interface enters line edit mode. While in line edit mode all characters are accepted except CNTL-Z. CNTL-Z terminates line edit mode and the AT command interface enters execution mode. Like before, at this point another <CR> will abort the command.

AT message concatenation can be done using the ; <semicolon> between commands.

The following examples demonstrate the potential usage of AT commands presented:

Type	Example	Description
Syntax Query	AT+GXXX=?	When entered will return the command format and value ranges.
Read Syntax	AT+GXXX?	When entered will return the current value assigned to the command.
Write Syntax	AT+GXXX-X=<value>,<value>,	When entered will set the command to specified value(s).
Execute Syntax	AT+GXXX	When entered will execute the specified command.
Command Con- catenation	AT+CRC=1;S0=1	When entered it will execute both the CRC and S0 command.

# Standard AT Commands

The following is the format in which all commands will be presented.

ATx(Command)	Xxxxx (Command Description)
Command Function	(Description of the command function)
Syntax Query	ATx=?
Syntax	ATx: (parameter1 name 1 - 15), (parameter2 name 1-10),...
Write Syntax	ATx=<value>,<value>[,<optional value>],...
Write Response	OK or ERROR
Read Syntax	ATx?
Read Response	<value>,<value>,...
Execute Syntax	ATx
Execute Response	OK, ERROR, or <value>
Unsolicited Response	
Parameter Values	
<Value 1>	ATx: (1-15),(1-10)
<Value 2>	
Notes	(Additional command notes)
Examples	



Where applicable, the <value> responses provided for the READ and EXECUTE formats are modem default values. All efforts will be made by Novatel Wireless. to keep these values current in the documentation but will not be responsible for any differences that may occur as a result subsequent software builds and version enhancements.



Do not use tab characters in the custom AT command scripts.



# General Commands

---

# AT\$MDN Mobile Directory Number

AT\$MDN	Mobile Directory Number
Command Function	The command queries and sets the mobile directory number.
Syntax Query	AT\$MDN=?
Syntax	OK
Write Syntax	AT\$MDN=<mdn number>
Write Response	Device is SPC locked
Read Syntax	AT\$MDN?
Read Response	<mdn number>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<mdn number>	10 digit number, generally the phone number of the device
Notes	
Examples	

# AT\$MSID Mobile Station Identification

AT\$MSID	Mobile Station Identification
Command Function	The command queries the mobile station identification. This is also known as the MIN, mobile identification number.
Syntax Query	AT\$MSID=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$MSID?
Read Response	<msid number>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<msid number>	10 digit number
Notes	MSID is not the same as MDN. MDN is the phone number. MSID/MIN is an identification number defined by the network. This allows a device to keep the same phone number even if changing service providers.
Examples	

# AT\$NWGMR Manufacturer Revision

AT\$NWGMR	Manufacturer Revision
Command Function	This AT command displays software and hardware revision information.
Syntax Query	AT\$NWGMR=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$NWGMR
Execute Response	Chipset: <chipset> Radio: <radio_vers> Smart Agent: <smart_agent_vers> HW Rev: <hw_rev> PRI Info: <pri_pn> PRI Version: <pri_vers> OK
Unsolicited Response	N/A
Parameter Values	
<chipset>	Type of radio chipset.
<radio_vers>	Internal protocol stack software version.
<smart_agent_vers>	Smart Agent™ M2M feature set software version.
<hw_rev>	Internal 5-bit modem hardware revision.
Notes	N/A

# AT\$PKG Request Firmware Package

AT\$PKG	Request Firmware Package
Command Function	This command is used to obtain the firmware package version.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$PKG
Execute Response	<baseline release>-<release offset> OK
Unsolicited Response	N/A
Parameter Values	
<baseline release>	1 to 99 - number associated with official production release
<release offset>	<p>A = Controlled Availability : The 'A' release is a prerelease of the next production release and focuses on new features. These releases are rarely placed in manufacturing for shipment and the expectation is that customers move to the production release when available.</p> <p>C = Candidate (internal only) : This label is used to differentiate releases during the system test phase. Internal Release (IR) is by definition, "C1". It's possible that a 'C' release will be sent to a customer for the same reasons as the Controlled Availability release. This release type will not be available for order from manufacturing.</p> <p>D = Maintenance : A 'D' release includes the previous production release plus very specific bug fixes (no new features). Maintenance releases can be, and usually are, released to production. Not all products will necessarily have a maintenance release. For instance, if a Nitro has a GPS issue that requires a maintenance release, the OEM modules will not have a release.</p> <p>E = Engineering (internal only) : Although these are internal releases, they can be sent to specific customers to verify either a</p>

AT\$PKG	Request Firmware Package
	bug fix or new feature. These releases cannot be used by customers as a production release. This release type will not be available for order from manufacturing.
Notes	Return value is manufacturer specific.
Examples	<p>AT\$PKG?</p> <p>48 This is a baseline production release</p> <p>AT\$PKG?</p> <p>48-D1 This is the baseline production release with additional bug fixes</p>

# AT\$SPN Software Part Number

AT\$SPN	Software Part Number
Command Function	This command will return the software part number.
Syntax Query	AT\$SPN=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$SPN?
Read Response	\$SPN: SWxxx
Execute Syntax	AT\$SPN
Execute Response	\$SPN: SWxxx OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	

# AT\$SRN Module Serial Number

AT\$SRN	Module Serial Number
Command Function	This command will return the serial number of the module.
Syntax Query	AT\$SRN=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$SRN
Execute Response	\$SRN: xxxxxxxxxxxxxxxx
Unsolicited Response	N/A
Parameter Values	N/A
Notes	Returned values are unique for each module



# AT+CGMI Request Manufacturer Identification

AT+CGMI	Request Manufacturer Identification
Command Function	This command is used to obtain the manufacturer identification information.
Syntax Query	AT+CGMI=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CGMI
Execute Response	Novatel Wireless OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	Return value is manufacturer specific.

# AT+CGMM Request Manufacturer Model Identification

AT+CGMM	Request Manufacturer Model Identification
Command Function	This command is used to obtain the manufacturer model identification information.
Syntax Query	AT+CGMM=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CGMM
Execute Response	<model> OK
Unsolicited Response	N/A
Parameter Values	N/A
<model>	Text string containing the model identification information.
Notes	<model> value is manufacturer and model specific.

# AT+CGMR Request Revision Identification

AT+CGMR	Request Revision Identification
Command Function	This command is used to obtain the manufacturer embedded firmware revision information.
Syntax Query	AT+CGMR=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CGMR
Execute Response	AT+CGMR Radio: <radio_vers> Smart Agent: <smart_agent_vers> OK
Unsolicited Response	N/A
Parameter Values	
<radio_vers>	Radio version
<smart_agent_vers>	Smart Agent version
Notes	Return value is manufacturer specific.

# AT+CGSN Product Serial Number

AT+CGSN	Product Serial Number
Command Function	This command is used to read the MEID (Mobile Equipment Identifier) and serial number. The decimal value is displayed on the first line, and the hexadecimal value is displayed on the second line.
Syntax Query	AT+CGSN=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CGSN
Execute Response	<MEID decimal>:<ESN decimal> <MEID hexadecimal>:<ESN hexadecimal> OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	Return value is manufacturer specific. The TA returns the MEID (Mobile Equipment Identifier and ESN).

# AT+GMI TA Manufacturer ID

AT+GMI	TA Manufacturer ID
Command Function	TA returns information about the manufacturer.
Syntax Query	AT+GMI=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+GMI
Execute Response	Novatel Wireless, A Novatel Wireless Company OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A

# AT+GMM TA Model ID

AT+GMM	TA Model ID
Command Function	TA returns manufacturer model identification.
Syntax Query	AT+GMM=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+GMM
Execute Response	Model ID e.g. Enabler HS 3001V or MT 4100 OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A

# AT+GMR TA Revision Number

AT+GMR	TA Revision Number
Command Function	Returns software revision information.
Syntax Query	AT+GMR=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+GMR
Execute Response	Radio and Smart Agent versions, e.g. Radio: 10.2 Smart Agent: 11.1 OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A

# AT+GSN TA Serial Number

AT+GSN	TA Serial Number
Command Function	This command is used to read the MEID (Mobile Equipment Identifier) and serial number. The decimal value is displayed on the first line, and the hexadecimal value is displayed on the second line.
Syntax Query	AT+GSN=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+GSN
Execute Response	<MEID decimal>:<ESN decimal> <MEID hexadecimal>:<ESN hexadecimal> OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	Return value is manufacturer specific. The TA returns the MEID (Mobile Equipment Identifier) and ESN.
Examples	AT+GSN 45317471580868859:2149643391 0xa1000013b250fb:0x8020f47f OK



# ATI Manufacturer Information About TA

ATI	Manufacturer Information About TA
Command Function	List manufacturer.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	ATI
Execute Response	Novatel Wireless, A Novatel Wireless Company OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A

# Network Commands

---

# AT\$AREG Auto Registration

<b>AT\$AREG</b>	<b>Auto Registration</b>
<b>Command Function</b>	This command sets the auto registration state of the modem
<b>Syntax Query</b>	AT\$AREG=?
<b>Syntax</b>	\$AREG: (0-2) OK
<b>Write Syntax</b>	AT\$AREG=<state>
<b>Write Response</b>	OK
<b>Read Syntax</b>	AT\$AREG?
<b>Read Response</b>	\$AREG: <state> OK
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;state&gt;</b>	0 = Autoreg off 1 = Voice/SMS Autoreg on 2 = Auto Voice/SMS/Data Activation on Power up. (for \$hostif=1 and 2, MT will perform Network activation and go into PAD data mode. For \$hostif=0 MT will perform Network activation, but remain in AT command mode)
<b>Notes</b>	This command sets Network registration state. When set to 1 or 2, upon power on, the modem will automatically register on the network. This command also controls Network registration dynamically, this is the registration/deregistration occurs immediately.

# AT\$CREG Network Registration Info

AT\$CREG	Network Registration Info
Command Function	Displays current network registration status. Write command enables/disables the unsolicited response code.
Syntax Query	AT\$CREG=?
Syntax	\$CREG: (0-1) OK
Write Syntax	AT\$CREG=<enableFlag>
Write Response	OK
Read Syntax	AT\$CREG?
Read Response	\$CREG: <enableFlag> OK
Execute Syntax	AT\$CREG
Execute Response	\$CREG: <status>,<lac>,<cid>,<mip> OK
Unsolicited Response	\$CREG: <status>,<lac>,<cid>,<mip>
Parameter Values	
<enableFlag>	0 = disable network registration unsolicited result code 1 = enable network registration unsolicited result code
<status>	\$CREG status is synthesized from CDMA information and approximates the GSM +CREG:  0 = airplane mode 1 = registered on home network 2 = registration in progress 3 = no service 4 = unknown 5 = registered roaming
<lac>	Location Area Code (GSM) or Network ID (CDMA)

AT\$CREG	Network Registration Info
<cid>	Cell ID (GSM) or Base Station ID (CDMA)
<mip>	MobileIP support 0 = SimpleIP is supported and/or MobileIP not supported 1 = MobileIP supported
Notes	N/A

# AT\$MDSTAT Query Modem Status

AT\$MDSTAT	Query Modem Status
Command Function	This command allows the user to display multiple modem status parameters with a single command.
Syntax Query	AT\$MDSTAT=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$MDSTAT?
Read Response	\$DEVTYP: 29,1 \$PKG: 1 Enabler HS 3001V NET_ID: 7 BASE_ID: 549 \$CREG: 1 \$CGREG: 1 CODE: 1 FREQ: 150 PN: 348 +CSS: ?,Z,999999,0 +CSQ: 31, 99 \$NETIP: "0.0.0.0","0.0.0.0","0.0.0.0" \$TCPERR: 0,0,0 OK
Execute Syntax	AT\$MDSTAT
Execute Response	Same as read response
Unsolicited Response	N/A
Parameter Values	
<\$DEVTYP>	see AT\$DEVTYP? command
<\$PKG>	see AT\$PKG command
<NET_ID>	Serving cell Network Id is a 16-bit integer.
<BASE_ID>	Serving cell Base Station Id is a 16-bit integer.
<\$CREG>	\$CREG is synthesized from CDMA information and approximates the GSM +CREG: 0 = airplane mode 1 = registered on home network 2 = registration in progress

AT\$MDSTAT	Query Modem Status
	3 = no service 5 = registered roaming 6 = unknown
<CODE>	Serving cell code, also known as “channel” is a 16-bit integer.
<FREQ>	Serving cell frequency is a 16-bit integer.
<PN>	Serving cell pseudo-random offset, range 0-511.
<+CSS>	see AT+CSS command

AT\$MDSTAT	Query Modem Status
<\$NETIP>	see AT\$NETIP? command
<\$TCPERR>	<p>Contains 3 parameters</p> <p><b>First parameter: TCP ERROR values</b></p> <p>0 = no error stored  1 = normal closure  2 = unexpected RST  3 = unexpected SYN  4 = received FIN  5 = timeout establishing connection  6 = failure establishing connection  7 = timeout in close  8 = timeout on established connection  9 = RST on established connection  10 = ICMP rec'd: no route to host  11 = ICMP rec'd: connection refused  12 = ICMP rec'd: frag req, DF set  13 = no route found for destination  14 = interface down or other failure  15 = internal stack failure</p> <p><b>Second parameter: PAD disconnect values</b></p> <p>0 = no reason stored  1 = server timeout  2 = PPP caused  3 = host caused  4 = connection timeout  5 = caused by AT command</p> <p><b>Third parameter: TCPAPI disconnect values</b></p> <p>0 = no reason stored  4 = connection timeout  5 = caused by AT command</p>



AT\$MDSTAT	Query Modem Status
	6 = bad message format 7 = connect fail
Notes	

# AT\$NETMON Monitor Network Availability

AT\$NETMON	Monitor Network Availability
Command Function	This command allows the modem to take aggressive network recovery action based upon the results of continuous network monitoring.
Syntax Query	AT\$NETMON=?
Syntax	\$NETMON: (0,5-1440),(0-10),(0-255),(0-3) OK
Write Syntax	AT\$NETMON= <net_unavail_min>,<reset_cnt>,<ping check>,<rst timers>
Write Response	OK
Read Syntax	AT\$NETMON?
Read Response	AT\$NETMON: "<net_unavail_min >,<reset_cnt>,<ping check>,<rst timers>"
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<net_unavail_min>	Number of minutes the network must remain unavailable before current Network Activation is released, and a new Network Activation is attempted. Network availability is valid Network IP (AT\$NETIP).  A value of zero means the Network Activation will never be released via AT\$NETMON.
<reset_cnt>	The number of times the modem will attempt to recover by using a network stack deactivation and re-activation process before using a modem reset to recover.  If the value is 0, then never perform a modem reset.
<ping check>	Number of minutes between modem-initiated ping checks. If no network data has been received within <ping check> minutes, the modem will initiate pings (up to 4 ICMP messages are generated) to the 1st server on the \$FRIEND list.

AT\$NETMON	Monitor Network Availability
	<p>If no ping response is received to any of the 4 ICMP messages, the modem will initiate pings to the next server in the list. If no ping response is returned from any of the \$FRIEND servers, a new IP is obtained via a modem-initiated network de-activation / activation sequence.</p> <p>A value of zero indicates that the modem will never initiate a ping check.</p>
<rst timers>	<p>0 - Reset network monitoring timers upon any activity on the serial port AND use legacy IP reselect for &lt;ping check&gt; recovery</p> <p>1 - Do not reset the network monitoring timers if there is activity on the serial ports</p> <p>2 - Use modem reset for &lt;ping check&gt; recovery</p> <p>3 - Do not reset the network monitoring timers if there is activity on the serial ports AND use modem reset for &lt;ping check&gt; recovery</p>
Notes	<p>This command allows the &lt;net_unavail_min&gt; parameter to be set as low as 5 minutes for backwards compatibility with legacy scripts. However, due to certification requirements, the soonest a \$NETMON recovery action can be performed is 30 minutes</p>

# AT\$NWACTIVATION Provision MDN And MIN

AT\$NWACTIVATION	Provision MDN and MIN
Command Function	This command is used to activate a modem on the Sprint network using AT commands instead of over-the-air provisioning. The MDN and MIN may be queried and set.
Syntax Query	AT\$NWACTIVATION=?
Syntax	\$NWACTIVATION: (ACTIVATION CODE:[xxxxxxx] MDN:[XXXXXXXXXXXX] MIN:[XXXXXXXXXXXX]) or ERROR
Write Syntax	AT\$NWACTIVATION=<activationCode>,<mdn>,<min>
Write Response	OK or ERROR
Read Syntax	AT\$NWACTIVATION?
Read Response	\$NWACTIVATION: <mdn>, <min> OK or ERROR
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	
Parameter Values	
<activationCode>	6 decimal digit number, service provisioning code or MSL.
<mdn>	10 decimal digit number, MDN (phone number).
<min>	10 decimal digit number, MIN.
Notes	After the command completes (OK), the modem should be reset manually.  On some Network ERROR is the expected response to AT\$NWACTIVATION? or AT\$NWACTIVATION=?
Example:	

# AT\$OTASP OTA Service Provisioning Status

AT\$OTASP	OTA Service Provisioning Status
Command Function	The command queries and clears the OTASP status. Also, when clearing the status, the MDN and MSID are set to default settings. Once power is cycled after clearing the status, the device will not be able to make a voice call until an OTASP is initiated.
Syntax Query	AT\$OTASP=?
Syntax	OTASP: (1-2) OK
Write Syntax	AT\$OTASP=<OTASP_cmd>
Write Response	OK
Read Syntax	AT\$OTASP?
Read Response	\$OTASP: <OTASP_status>,<DMU_Status> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	<p>All internal OTASP related state changes in the OTASP process will be displayed to the serial console in the form:</p> <p>\$OTASP: &lt;state&gt;</p> <p>Where &lt;state&gt; is one of these values:</p> <ul style="list-style-type: none"> <li>-1: None</li> <li>0: SPL unlocked</li> <li>1: SPC retries exceeded</li> <li>2: AKEY exchanged</li> <li>3: SSD updated</li> <li>4: NAM downloaded</li> <li>5: MDN downloaded</li> <li>6: IMSI downloaded</li> <li>7: PRL downloaded</li> <li>8: Committed</li> </ul>

AT\$OTASP	OTA Service Provisioning Status
	9: OTAPA started 10: OTAPA stopped 11: OTAPA aborted 12: Reserved
Parameter Values	
<OTASP_cmd>	0 = No completed DMU since initial powering on or since last cleared 1 - Clear OTASP status 2 - Clear OTASP and DMU_statuses
<OTASP_status>	0 = Error during OTASP commit 1 = Success 255 = No completed OTASP request since powering on or since last cleared
<DMU_status>	0 = Has not successfully completed DMU 1 = DMU completed successfully
Notes	
Example:	

# AT\$ATPASSWD Set Authorization For AT Commands

\$ATPASSWD	Set Authorization for AT commands
Command Function	This command allows the user to enable or disable authorization for AT commands for the serial, SMS and API. It also sets the password required to run this command
Syntax Query	AT\$ATPASSWD=?
Syntax	\$ATPASSWD: "PASSWORD",mask "passwd" OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$ATPASSWD= <"oldpasswd","newpasswd">
Execute Response	\$ATPASSWD=<"passwd",mask>
Unsolicited Response	N/A
Parameter Values	
<oldpasswd>	Specified when the password is being changed.
<newpasswd>	This is the value of the new password and is specified only when the password is being changed. It must be no more than eight characters in length and must be enclosed in double quotes.

\$ATPASSWD	Set Authorization for AT commands									
<passwd>	Specified when changing the AT command authorization mask.									
<mask>	<div>Bit mask specifying which interfaces will be authorized to enter AT commands. Each bit specifies one interface as enumerated in the table below.</div> <table><tr><th>Bit value</th><th>Interface</th></tr><tr><td>1</td><td>Serial Port</td></tr><tr><td>2</td><td>SMS</td></tr><tr><td>4</td><td>API</td></tr></table> <div>To select multiple items to authorize, add the bit values of each interface to be authorized. To authorize API and SMS only, the mask value is 6 (4 + 2). AT commands entered over the serial port will not execute and will reply with ERROR.</div>		Bit value	Interface	1	Serial Port	2	SMS	4	API
Bit value	Interface									
1	Serial Port									
2	SMS									
4	API									
Notes	N/A									



# AT\$QCDNSP DNS Primary IP Address

AT\$QCDNSP	DNS Primary IP Address
Command Function	This command sets the default primary IP address used for Domain Name Services (DNS). This IP address is used only if no DNS server address is received over the air during network data activation. The value is stored in NVRAM.
Syntax Query	AT\$QCDNSP=?
Syntax	\$QCDNSP: OK
Write Syntax	AT\$QCDNSP=<IP address>
Write Response	OK
Read Syntax	AT\$QCDNSP?
Read Response	\$QCDNSP: <IP address> OK
Execute Syntax	AT\$QCDNSP
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<Param1>	
<Param2>	
<IP address>	IP address in dotted decimal notation Note: valid values are 0-254. The value 255 is not allowed.
Notes	
Examples	AT\$QCDNSP? \$QCDNSP: 151.144.244.200 OK

# AT\$QCDNSS Secondary DNS IP Address

AT\$QCDNSS	Secondary DNS IP Address
Command Function	This command is used to set the default secondary IP address used for Domain Name Services (DNS).
Syntax Query	AT\$QCDNSS=?
Syntax	\$QCDNSS: OK
Write Syntax	AT\$QCDNSS=<IP address>
Write Response	OK
Read Syntax	AT\$QCDNSS?
Read Response	\$QCDNSS: <IP address> OK
Execute Syntax	AT\$QCDNSS
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<IP address>	IP address in dotted decimal notation Note: valid values are 0-254. The value 255 is not allowed.
Notes	
Examples	

# AT\$QCMIP Mobile IP

AT\$QCMIP	Mobile IP
Command Function	Enable/disable Mobile IP
Syntax Query	AT\$QCMIP=?
Syntax	\$QCMIP: (0-2) OK
Write Syntax	AT\$QCMIP=<MIP_enable>
Write Response	OK
Read Syntax	AT\$QCMIP?
Read Response	\$QCMIP: <MIP_enable> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<MIP_enable>	<p>0 = Mobile IP disabled Simple IP only</p> <p>1 = Mobile IP preferred In the initial MIP registration, if the network does not support Mobile IP, the mobile automatically reverts to Simple IP (forces a PPP renegotiation by sending an LCP C-Req). However, if a Mobile IP session is registered, and the mobile enters a network that does not support Mobile IP, it will drop the session and inform the upper layers of the failure (e.g., by dropping DCD to a laptop).</p> <p>2 = Mobile IP only The mobile will make data calls only when Mobile IP is supported in the network. During an MIP session, if the mobile hands off to a network that does not support MIP, the mobile will drop the session and inform the upper layers of the failure, e.g., by dropping DCD to a laptop. This value is stored in NV.</p> <p>The default value is 0.</p>
Notes	N/A

# AT\$QCMIPeP Enable Or Disable Current MIP Profile

AT\$QCMIPeP	Enable or Disable Current MIP Profile
Command Function	This command is used to enable or disable current MIP profile.
Syntax Query	AT\$QCMIPeP=?
Syntax	\$QCMIPeP: (0-1)
Write Syntax	AT\$QCMIPeP=<N>
Write Response	OK
Read Syntax	AT\$QCMIPeP?
Read Response	\$QCMIPeP:<N>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<N>	0 = Disable active profile. 1 = Enable active profile.
Notes	
Examples	

# AT\$QCMIPGETP Get Profile Information

<b>AT\$QCMIPGETP</b>	<b>Get Profile Information</b>
<b>Command Function</b>	This command is used to Get Profile Information.
<b>Syntax Query</b>	AT\$QCMIPGETP=?
<b>Syntax</b>	\$QCMIPGETP: (0-5)
<b>Write Syntax</b>	AT\$QCMIPGETP=<N>
<b>Write Response</b>	Profile:<N> <enableFlag> NAI:<nai_value> Home Addr:<home_addr> Primary HA:<primary_ha> Secondary HA:<secondary_ha> MN-AAA SPI:<mn_aaa> MN-HA SPI:<mn_ha> Rev Tun:<reverse_tunnel> MN-AAA SS:<aaa_shared_secret set or unset> MN-HA SS:<ha_shared_secret set or unset> OK
<b>Read Syntax</b>	AT\$QCMIPGETP?
<b>Read Response</b>	\$QCMIPGETP:<N>
<b>Execute Syntax</b>	AT\$QCMIPGETP
<b>Execute Response</b>	<Output in same format as write response for current profile>
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;N&gt;</b>	Profile number, 0-5.
<b>Notes</b>	If no profile number is entered, all information corresponding to the currently active profile is returned. If there is no profile associated with the specified number, an error is returned.
<b>Examples</b>	

# AT\$QCMIPMASS MN-AAA Shared Secrets

AT\$QCMIPMASS	MN-AAA Shared Secrets
Command Function	This command is used to sets the MN-AAA shared secrets for the currently active MIP profile.
Syntax Query	AT\$QCMIPMASS=?
Syntax	\$QCMIPMASS: (20,21,23-7E),(0-1)
Write Syntax	AT\$QCMIPMASS=<STR>,<STORE>
Write Response	OK
Read Syntax	AT\$QCMIPMASS?
Read Response	\$QCMIPMASS:<Set or Unset>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<STR>	Shared secret data string
<STORE>	0 = Store to NV memory 1 = Do not store to NV memory
Notes	If the value provisioned is not committed to NV, the temporary values will be deleted at the end of the following call or if \$QCMIPP is called.
Examples	at\$qcmipmass="secret",1

# AT\$QCMIPMHSS MM-HA Shared Secrets

AT\$QCMIPMHSS	MM-HA Shared Secrets
Command Function	This command is used to set then MM-HA shared secrets for the currently active MIP profile.
Syntax Query	AT\$QCMIPMHSS=?
Syntax	\$QCMIPMHSS: (20,21,23-7E),(0-1)
Write Syntax	AT\$QCMIPMHSS= <STR> ,<STORE>
Write Response	OK
Read Syntax	AT\$QCMIPMHSS?
Read Response	\$QCMIPMHSS: <Set or Unset> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<STR>	Shared secret data string
<STORE>	0 = Store to NV memory 1 = Do not store to NV memory
Notes	Double quotes are only required if the string contains a comma. If the value provisioned is not committed to NV, the temporary values will be deleted at the end of the following call or if \$QCMIPP is called.
Examples	at\$qcmipmhss="secret",1

# AT\$QCMIPNAI Network Access Identifier

AT\$QCMIPNAI	Network Access Identifier
Command Function	This command is used to set the network access identifier.
Syntax Query	AT\$QCMIPNAI=?
Syntax	\$QCMIPNAI: (20,21,23-7E),(0-1)
Write Syntax	AT\$QCMIPNAI= <NAI>,<STORE>
Write Response	OK
Read Syntax	AT\$QCMIPNAI?
Read Response	\$QCMIPNAI:<Set or Unset> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<NAI>	Network access identifierstring
<STORE>	0 = Store to NV memory 1 = Do not store to NV memory?
Notes	<p>The first parameter of \$QCMIPNAI=? returns (20,21,23-7E) which is the range of printable ASCII characters allowed in the string.</p> <p>Double quotes are only required if the string contains a comma.</p> <p>If the value provisioned is not committed to NV, the temporary values will be deleted at the end of the following call or if \$QCMIPP is called.</p>
Examples	at\$qcmipnai="nai@network.com",1



# AT\$SPC Service Provisioning Code

AT\$SPC	Service Provisioning Code
Command Function	The command is used to enter the service provisioning code for the modem. The SPC must be entered to unlock certain other commands which update shared secret authentication information.
Syntax Query	N/A
Syntax	N/A
Write Syntax	AT\$SPC=<SPC>
Write Response	SPC Correct - Device SPC Unlocked -or- SPC doesn't match
Read Syntax	AT\$SPC?
Read Response	OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	
Parameter Values	
<SPC>	6 decimal digit number, service provisioning code. Generally provided by the carrier before activating the device.
Notes	
Example:	

# AT+CNUM Subscriber Number

AT+CNUM	Subscriber Number
Command Function	This command is used to obtain the MSISDNs related to the subscriber.
Syntax Query	AT+CNUM=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CNUM
Execute Response	+CNUM: "Line1", "xxx xxx xxxx", 145 OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	

# AT+CSS Serving System

AT+CSS	Serving System
Command Function	This command returns the serving system.
Syntax Query	AT+CSS=?
Response	<Band_Class>,<Band>,<SID>, <P_REV_IN_USE> OK
Write Syntax	N/A
Response	N/A
Read Syntax	AT+CSS?
Response	<Band_Class>,<Band>,<SID>, <P_REV_IN_USE> OK
Execute Syntax	N/A
Response	N/A
Parameter Values	
<Band_Class>	0 = The current band class is unsupported by this command. 1 = Band Class 0 (800 MHz Cellular) 2 = Band Class 1 (1900 MHz PCS)
<Band>	CA = MS is registered with cellular A-band system CB = MS is registered with cellular B-band system PA = MS is registered with PCS A-band system PB = MS is registered with PCS B-band system PC = MS is registered with PCS C-band system PD = MS is registered with PCS D-band system PE = MS is registered with PCS E-band system PF = MS is registered with PCS F-band system Z = The mobile station is not registered.
<SID>	0-16383 = The mobile station is registered with the system indicated. 99999 = The mobile station is not registered.

AT+CSS	Serving System
<P_REV_IN_USE>	0 = For unrecognized systems or non-CDMA systems 1 = IS-95 or J-STD-008 2 = IS-95-A 3 = IS-95-A + TSB74 4 = TIA/EIA-95-B 5 = TIA/EIA-95-B 6 = IS-2000 7 = IS-2000-A 8 = IS-2000-B 9 = IS-2000-C 10 = IS-2000-C
Notes	
Examples	

# AT+CSQ Signal Quality And Bit Error Rate

AT+CSQ	Signal Quality and Bit Error Rate
Command Function	Execution command returns received signal strength indication <rssi> and channel bit error rate <ber> from the ME.
Syntax Query	AT+CSQ=?
Syntax	+CSQ: (0-31,99),(0-7,99) OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT+CSQ?
Read Response	+CSQ: <rssi>,<ber> OK
Execute Syntax	AT+CSQ
Execute Response	+CSQ: <rssi>,<ber> OK
Unsolicited Response	NA
Parameter Values	
<rssi>	0 = -113 dBm or less 1 = -111 dBm 2-30 = -109 through -53 dBm 31 = -51 dBm or greater 99 = not known or not detectable
<ber>	The value '99' will always be returned which means "not known."
Notes	N/A

# META Commands

---

# AT+CEER Extended Error Reporting

AT+CEER	Extended Error Reporting
Command Function	This command is used to control the display of extended result codes for last unsuccessful call setup, in-call modification, last call release, last short message, or last network data session.
Syntax Query	AT+CEER=?
Syntax	Error
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT+CEER?
Read Response	+CEER: <description>
Execute Syntax	AT+CEER
Execute Response	+CEER: <description> OK
Unsolicited Response	N/A
Parameter Values	
<description>	See the error codes listed in Appendix B
Notes	N/A

# AT\$RESET Reset Modem

AT\$RESET	Reset Modem
Command Function	This command is used to perform a modem reset.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$RESET
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
Notes	Execution of this command will perform a full reset of the software stack. If the modem is currently registered onto the network, the modem will perform a detach before performing the stack reset.



# AT\$V Display Smart Agent Configuration Settings

<b>AT\$V</b>	<b>Display Smart Agent Configuration Settings</b>
<b>Command Function</b>	This command allows the user to view the Smart Agent configuration settings. The settings for a single subsystem or for the entire Smart Agent application may be displayed.
<b>Syntax Query</b>	AT\$V=?
<b>Syntax</b>	(ALL, SYS, DATA, EVENT, IO, VOICE[, GFMI][, GPS][, OBD][, UAPP]) OK
<b>Write Syntax</b>	AT\$V=<subsystem>
<b>Write Response</b>	<subsystem configuration values> OK
<b>Read Syntax</b>	N/A
<b>Read Response</b>	N/A
<b>Execute Syntax</b>	AT\$V
<b>Execute Response</b>	<configuration values for all subsystems> OK
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;subsystem&gt;</b>	Select a single subsystem to limit the response or select "ALL" to view the entire Smart Agent configuration. Select one of the following subsystem values: ALL, SYS, DATA, EVENT, IO, VOICE, GFMI, GPS, UAPP.
<b>Notes</b>	Some subsystems do not apply to all products. i.e. some products do not support Voice, GFMI and/or GPS.

# AT&F Set All TA Parameters To Factory Defined Configuration

AT&F	Set All TA Parameters to Factory Defined Configuration
Command Function	Set All TA Parameters to Factory Defined Configuration
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT&F
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	



For devices that support the AT\$CMDPORT command: AT\$CMDPORT is not effected by AT&F. Check AT\$CMDPORT after running AT&F

---

# AT&V Display Current Profile

AT&V	Display Current Profile
Command Function	This command allows the user to view the settings in the current profile.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT&V
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A



Only the industry standard AT command settings are displayed in the AT&V output. Use AT\$V to view the Smart Agent command configuration.

# AT&W Save Current Settings

AT&W	Save Current Settings
Command Function	This command allows the user to save the current settings in memory.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT&W
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	<p>To ensure successful completion of the command, do not issue additional commands until 'OK' is returned.</p> <p>AT&amp;W does not save commands that are implemented on the aux processor (OBD and accelerometer commands, essentially any command prefixed with "OBD"). These settings are only persistent if AT\$OBDSAV is sent. If AT\$OBDSAV is not sent, any changes made to OBD or accelerometer commands are maintained through a reset (AT\$RESET), but are lost if the power is cycled.</p>



Users should avoid sending AT&W immediately before a modem reset. A minimum of a few seconds should be allowed between issuing the command and a modem reset.

---

# ATE Command Echo Mode

ATE	Command Echo Mode
Command Function	Determines whether the TA echoes characters typed locally.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATE<value>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<value>	0 = Do not echo characters locally 1 = Echo characters locally
Notes	ATE1 cannot be saved using AT&W

# ATQ Result Code Suppression

ATQ	Result Code Suppression
Command Function	Determines whether or not the TA transmits any result code to the TE.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATQ<value>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<value>	0 =DCE transmits result codes 1 =Result codes are suppressed and not transmitted
Notes	The parameter for this command is persistent via AT&W.

# ATS3 Command Line Termination Character

ATS3	Command Line Termination Character
Command Function	Determines the character recognized by the TA to terminate an incoming command line.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATS3=<n>
Write Response	OK
Read Syntax	ATS3?
Read Response	013 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
Notes	The parameter for this command is persistent via AT&W.

# ATS4 Response Formatting Character

ATS4	Response Formatting Character
Command Function	Determines the character generated by the TA for result code and information text.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATS4=<n>
Write Response	OK
Read Syntax	ATS4?
Read Response	010 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
Notes	The parameter for this command is persistent via AT&W.



# ATS5 Editing Character

ATS5	Editing Character
Command Function	Determines the character recognized by the TA as a request to delete the preceding character form the command line.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATS5=<n>
Write Response	OK
Read Syntax	ATS5?
Read Response	008 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
Notes	The parameter for this command is persistent via AT&W.

# ATV Response Format

ATV	Response Format
Command Function	Determines the DCE response format, with or without header character, and the use of numerical results code.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATV<value>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<value>	0 = DCE transmits limited headers and trailers and numeric result codes 1 = DCE transmits full headers and trailers and verbose response text
Notes	N/A

# ATX CONNECT Result

ATX	CONNECT Result
Command Function	Determines whether or not the TA transmits particular result codes.
Syntax Query	N/A
Syntax	N/A
Write Syntax	ATX<value>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<value>	0 = Short result code format 1 = Long result code format
Notes	For UDP and TCP PAD operation, setting of ATX1 will display the network assigned IP after the CONNECT or LISTEN message.

# SMS Commands

---

# AT\$QCMSS Send Message From Storage

AT\$QCMSS	Send Message from Storage
Command Function	Sends message (with location value) from preferred message storage.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$QCMSS=<index>
Execute Response	\$QCMSS: <mr> OK
Unsolicited Response	N/A
Parameter Values	
<index>	Integer value of location number supported by associated memory
<mr>	Message reference
Notes	<p>The above is for text mode only.</p> <p>The following limitations on SMS functionality currently apply.</p> <ol style="list-style-type: none"> <li>1. Concatenated SMS transmit is not supported.</li> <li>2. Broadcast SMS is not supported.</li> </ol>

# AT\$QCMGD Delete Message

AT\$QCMGD	Delete Message
Command Function	Deletes message from preferred storage location.
Syntax Query	AT\$QCMGD=?
Syntax	\$QCMGD:(<indexList>),(0-4)
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$QCMGD=<index>[,<status>]
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<index>	Integer value of memory location.
<status>	<p>0 = Delete the messages specified by the index</p> <p>1 = Ignore the index and delete all the read messages</p> <p>2 = Ignore the index and delete all the read and sent messages</p> <p>3 = Ignore the index and delete all the read, sent and unsent messages.</p> <p>4 = Ignore the index and delete all the messages whatever the status is</p>
Notes	<p>If there is no message stored in the selected index, an error will be returned.</p> <p>To delete a single message, use the execute command with the single index parameter and do not include the &lt;status&gt; parameter.</p>
Examples	<p>To delete message 1: AT\$QCMGD=1</p> <p>To delete all messages: AT\$QCMGD=,4</p>

# AT\$QCMGF SMS Format

AT\$QCMGF	SMS Format
Command Function	Set command tells the TA, which input and output format of messages to use. <mode> indicates the format of messages used with send, list, read and write commands and unsolicited result codes resulting from received messages.
Syntax Query	AT\$QCMGF=?
Syntax	AT\$QCMGF: (1) OK
Write Syntax	AT\$CMGF=<mode>
Write Response	OK
Read Syntax	AT\$QCMGF?
Read Response	\$QCMGF: 1 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<mode>	1 = Text mode
Notes	

# AT\$QCMGL List Messages

AT\$QCMGL	List Messages
Command Function	List messages from storage.
Syntax Query	AT\$QCMGL=?
Syntax	\$QCMGL: ("REC UNREAD","REC READ","STO UNSENT","STO SENT","ALL") OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$QCMGL =<stat>
Execute Response	\$QCMGL: <index>,<stat>,<da/oa>, [<alpha>,<scts>,<tooa/toda>,<length>] <CR><LF> data OK
Unsolicited Response	N/A
Parameter Values	See Notes
<index>	Memory location integer
<stat>	Status of message "REC UNREAD" "REC READ" "STO UNREAD" "STO READ" "ALL"
<do/oa>	destination address
<alpha>	alphanumeric representation of <da> or <oa> corresponding to the entry found in MT phonebook
<scts>	Service center time stamp



<b>AT\$QCMGL</b>	<b>List Messages</b>
<b>&lt;tooa/toda&gt;</b>	Address Type-of-Address octet in integer format
<b>&lt;length&gt;</b>	Length of message in octets
<b>Notes</b>	<p>Above settings for &lt;stat&gt; assume AT\$QCMGF=1 (text mode).</p> <p>Parameters in [] may or may not be reported dependent upon the setting of AT\$QCMGF.</p> <p>0 = "Rec Unread"</p> <p>1 = "Rec Read"</p> <p>2 = "Sto Unsent"</p> <p>3 = "Sto Sent"</p> <p>4 = "ALL"</p>

# AT\$QCMGR Read Message

AT\$QCMGR	Read Message
Command Function	Read stored messages.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$QCMGR=<index>
Execute Response	\$QCMGR: <stat>,<oa>,<scts>,<tooa>,<fo>,<pid>,<sca>,<tosca>,<length>]<CR><LF><data> OK
Unsolicited Response	N/A
Parameter Values	
<stat>	Status of message (Rec Read, Rec Unread, Sto Unsent, Sto Sent)
<oa>	Originating address
<scts>	Service center time stamp
<tooa>	Originating address - type of address
<fo>	First octet
<pid>	Protocol identifier
<sca>	Service center address
<tosca>	Type of address
<length>	Length of message in octets
Notes	The above parameters are for text mode.

# AT\$QCMGS Send Message

<b>AT\$QCMGS</b>	<b>Send Message</b>
<b>Command Function</b>	Sends message from the TE to the network.
<b>Syntax Query</b>	N/A
<b>Syntax</b>	N/A
<b>Write Syntax</b>	N/A
<b>Write Response</b>	N/A
<b>Read Syntax</b>	N/A
<b>Read Response</b>	N/A
<b>Execute Syntax</b>	AT\$QCMGS="<da>",<[<tda>]
<b>Execute Response</b>	Enter text <cntl Z> \$QCMGS <mr> OK
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;da&gt;</b>	Destination address
<b>&lt;mr&gt;</b>	Message reference
<b>Notes</b>	<p>The example provided is for text mode (AT\$CMGF=1).</p> <p>The following limitations on SMS functionality currently apply.</p> <ol style="list-style-type: none"> <li>1. Concatenated SMS transmit is not supported.</li> <li>2. Broadcast SMS is not supported.</li> </ol>

# AT\$QCMGW Write Message To Memory

AT\$QCMGW	Write Message to Memory
Command Function	Writes message to preferred storage location.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$QCMGW=<"da"><CR><LF>
Execute Response	Text is entered<ctrlZ> \$CMGW: <index> OK
Unsolicited Response	N/A
Parameter Values	
<da>	Destination Address
<index>	Integer value of memory location of the stored message
Notes	The above is for text mode only.

# AT\$QCNMI New Message Indications To TE

AT\$QCNMI	New Message Indications to TE
Command Function	New Message Indications to TE
Syntax Query	\$QCNMI=?
Response	\$QCNMI: (0,1,2),(0,1,2,3),(0,1)
Write Syntax	\$QCNMI=[<mode>[,<mt>[,<bfr>]]]
Response	+CMS ERROR: <err>
Read Syntax	\$QCNMI?
Response	\$QCNMI: <mode>,<mt>,<bfr>
Execute Syntax	N/A
Response	N/A
Parameter Values	
<mode>	<p>0 = Buffer unsolicited result codes in the TA</p> <p>1 = Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved</p> <p>2 = Buffer unsolicited result codes in the TA when TA-TE link is reserved and flush them to the TE after reservation</p>
<mt>	Receiving procedure for different message data coding schemes (refer to GSM 03.38 [2])
<mt> 0	<p>no class: as in GSM 03.38, but use &lt;mem3&gt; as preferred memory</p> <p>class 0: as in GSM 03.38, but use &lt;mem3&gt; as preferred memory if message is tried to be stored</p> <p>class 1: as in GSM 03.38, but use &lt;mem3&gt; as preferred memory</p> <p>class 2: as in GSM 03.38</p> <p>class 3: as in GSM 03.38, but use &lt;mem3&gt; as preferred memory</p> <p>message waiting indication group (discard message): as in GSM 03.38, but use &lt;mem3&gt; as preferred memory if message is tried to be stored</p> <p>message waiting indication group (store message): as in GSM 03.38, but use &lt;mem3&gt; as preferred memory</p>

AT\$QCNMI	New Message Indications to TE
<mt>1	as <mt>=0 but send indication if message stored successfully
<mt>2	no class: route message to TE class 0: as in GSM 03.38, but also route message to TE and do not try to store it in memory class 1: route message to TE class 2: as <mt>=1 class 3: route message to TE message waiting indication group (discard message): as in GSM 03.38, but also route message to TE and do not try to store it in memory message waiting indication group (store message): as <mt>=1
<mt>3	class 3: route message to TE others: as <mt>=1
<bfr>	0 = TA buffer of unsolicited result codes defined within this command is flushed to the TE when <mode> 1...2 is entered. 1 = TA buffer of unsolicited result codes defined within this command is cleared when <mode> 1...2 is entered.
Notes	
Examples	

# AT\$QCPMS Preferred Message Storage

AT\$QCPMS	Preferred Message Storage
Command Function	Set command selects memory storages <mem1>, <mem2> and <mem3> to be used for reading, writing, etc.
Syntax Query	AT\$QCPMS=?
Syntax	\$QCPMS: "ME","MT" OK
Write Syntax	AT\$QCPMS=<mem1>,<mem2>,<mem3>
Write Response	OK
Read Syntax	AT\$QCPMS?
Read Response	\$QCPMS: <mem1>,<usedSpace1>,<maxSpace1>,<mem2>,<usedSpace2>,<maxSpace2>,<mem3>,<usedSpace3>,<maxSpace3> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<mem1>	String type; memory from which messages are read and deleted (commands List Messages \$QCMGL, Read Message \$QCMGR and Delete Message \$QCMGD)  Defined values: "ME" = ME message storage "MT" = Any of the message storage associated with the device

AT\$QCPMS	Preferred Message Storage
<mem2>	String type; memory to which writing and sending operations are made (commands Send Message from Storage \$QCMSS and Write Message to Memory \$QCMGW) refer to <mem1> for defined values
<mem3>	String type; memory to which received messages are preferred to be stored (unless class of message defines a specific storage location; refer to command New Message Indications \$QCNMI) refer to <mem1> for defined values
Notes	



# AT\$QCSMP Set Text Mode Parameters

AT\$QCSMP	Set Text Mode Parameters
Command Function	Selects additional values needed when the SMS is sent to the network or placed in storage.
Syntax Query	AT\$QCSMP=?
Syntax	OK
Write Syntax	AT\$QCSMP=<tid>,<vpf>,<vp>,<ddtf>,<ddt>
Write Response	OK
Read Syntax	AT\$QCSMP?
Read Response	\$QCSMP: 4095, 0, "string", 0, 0 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<tid>	Teleservice ID 4095 - 4102
<vpf>	Validity Period Format 0 = Absolute 1 = Relative
<vp>	Validity Period string[22]
<ddtf>	Deferred Delivery Time Format 0 Absolute, 1 Relative
<ddt>	Deferred Delivery Time string[22]
Notes	N/A

# AT\$SMSURC Enable URC To Be Sent By SMS

AT\$SMSURC	Enable URC to be sent by SMS
Command Function	This command enables some URCs to be sent via SMS.
Syntax Query	AT\$SMSURC=?
Syntax	\$SMSURC: (0-1),(0-1) OK
Write Syntax	AT\$SMSURC=<urc1>,<urc2>
Write Response	OK
Read Syntax	AT\$SMSURC?
Read Response	\$SMSURC: OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<urc1>	0 = Do not send URCs from \$FTPOPEN, \$FTPCLOSE, \$FTPGET, and \$FTPPUT via SMS if the command was sent via SMS 1 = Send URCs from \$FTPOPEN, \$FTPCLOSE, \$FTPGET, and \$FTPPUT via SMS if the command was sent via SMS
<urc2>	0 = Do not send URCs from \$ATEXEC via SMS if the command was sent via SMS 1 = Send URCs from \$ATEXEC via SMS if the command was sent via SMS
Notes	N/A

# AT+CMGF SMS Format

AT+CMGF	SMS Format
Command Function	Set command tells the TA, which input and output format of messages to use. <mode> indicates the format of messages used with send, list, read and write commands and unsolicited result codes resulting from received messages.
Syntax Query	AT+CMGF=?
Syntax	+CMGF: (1) OK
Write Syntax	AT+CMGF=<mode>
Write Response	OK
Read Syntax	AT+CMGF?
Read Response	+CMGF: 1 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<mode>	1 = Text mode
Notes	

# AT+CMGD Delete Message

AT+CMGD	Delete Message
Command Function	Deletes message from preferred storage location.
Syntax Query	AT+CMGD=?
Syntax	+CMGD: (<indexList>),(0-4)
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CMGD=<index>[,<status>]
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<index>	Integer value of memory location.
<status>	<p>0 = Delete the messages specified by the index</p> <p>1 = Ignore the index and delete all the read messages</p> <p>2 = Ignore the index and delete all the read and sent messages</p> <p>3 = Ignore the index and delete all the read, sent and unsent messages.</p> <p>4 = Ignore the index and delete all the messages whatever the status is</p>
<indexList>	<p>List of the indexes for the messages in memory.</p> <p>For example: (0,1,2) for memory which contains 3 messages.</p> <p>If no messages are available for deletion, the &lt;indexList&gt; will be ().</p>
Notes	<p>If there is no message stored in the selected index, an error will be returned.</p> <p>To delete a single message, use the execute command with the single index parameter and do not include the &lt;status&gt; parameter.</p>
Examples	<p>To delete message 1: AT+CMGD=1</p> <p>To delete all messages: AT+CMGD=,4</p>

# AT+CMGL List Messages

AT+CMGL	List Messages
Command Function	List messages from storage.
Syntax Query	AT+CMGL=?
Syntax	+CMGL: ("REC UNREAD","REC READ","STO UNSENT","STO SENT","ALL") OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CMGL =<stat>
Execute Response	+CMGL: <index>,<stat>,<da/oa>, [<alpha>,<scts>,<tooa/toda>,<length>] <CR><LF> data OK
Unsolicited Response	N/A
Parameter Values	See Notes
<index>	Memory location integer
<stat>	Status of message "REC UNREAD" "REC READ" "STO UNREAD" "STO READ" "ALL"
<do/oa>	destination address
<alpha>	alphanumeric representation of <da> or <oa> corresponding to the entry found in MT phonebook
<scts>	Service center time stamp

AT+CMGL	List Messages
<tooa/toda>	Address Type-of-Address octet in integer format
<length>	Length of message in octets
Notes	<p>Above settings for &lt;stat&gt; assume AT+CMGF=1 (text mode).</p> <p>Parameters in [] may or may not be reported dependent upon the setting of AT+CMGF.</p> <p>0 = "Rec Unread"</p> <p>1 = "Rec Read"</p> <p>2 = "Sto Unsent"</p> <p>3 = "Sto Sent"</p> <p>4 = "ALL"</p>

# AT+CMGR Read Message

AT+CMGR	Read Message
Command Function	Read stored messages.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CMGR=<index>
Execute Response	+CMGR: <stat>,<oa>,<scts>,[<tooa>,<fo>,<pid>,<sca>,<tosca>,<length>]<CR><LF><data> OK
Unsolicited Response	N/A
Parameter Values	
<stat>	Status of message (Rec Read, Rec Unread, Sto Unsent, Sto Sent)
<oa>	Originating address
<scts>	Service center time stamp
<tooa>	Originating address - type of address
<fo>	First octet
<pid>	Protocol identifier
<sca>	Service center address
<tosca>	Type of address
<length>	Length of message in octets
Notes	The above parameters are for text mode.

# AT+CMGS Send Message

AT+CMGS	Send Message
Command Function	Sends message from the TE to the network.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CMGS="<da>",[<toda>]
Execute Response	Enter text <cntl Z> +CMGS <mr> OK
Unsolicited Response	N/A
Parameter Values	
<da>	Destination address
<mr>	Message reference
Notes	The example provided is for text mode (AT+CMGF=1).



# AT+CMGW Write Message To Memory

AT+CMGW	Write Message to Memory
Command Function	Writes message to preferred storage location.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CMGW=<"da"><CR><LF>
Execute Response	Text is entered<ctrlZ> +CMGW: <index> OK
Unsolicited Response	N/A
Parameter Values	
<da>	Destination Address
<index>	Integer value of memory location of the stored message
Notes	The above is for text mode only.

# AT+CMSS Send Message From Storage

AT+CMSS	Send Message from Storage
Command Function	Sends message (with location value) from preferred message storage.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT+CMSS=<index>
Execute Response	+CMSS: <mr> OK
Unsolicited Response	N/A
Parameter Values	
<index>	Integer value of location number supported by associated memory
<mr>	Message reference
Notes	The above is for text mode only.

# AT+CNMI New Message Indication To TE

AT+CNMI	New Message Indication to TE
Command Function	Selects how incoming messages from the network are indicated to the TE when the TE is active.
Syntax Query	AT+CNMI=?
Syntax	+CNMI: (0-2),(0-3),(0),(0),(0,1) OK
Write Syntax	AT+CNMI=<mode>,<mt>,<bm>,<ds>,<bfr>
Write Response	OK
Read Syntax	AT+CNMI?
Read Response	+CNMI: 0,0,0,0,0 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<mode>	0 = Buffer unsolicited result codes in the TA 1 = Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved 2 = Buffer unsolicited result codes in the TA when TA-TE link is reserved and flush them to the TE after reservation
<mt>	Receiving procedure for different message data coding schemes (refer GSM 03.38 [2])
<mt> 0	no class: as in GSM 03.38, but use <mem3> as preferred memory class 0: as in GSM 03.38, but use <mem3> as preferred memory if message is tried to be stored class 1: as in GSM 03.38, but use <mem3> as preferred memory class 2: as in GSM 03.38 class 3: as in GSM 03.38, but use <mem3> as preferred memory message waiting indication group (discard message): as in GSM 03.38, but use <mem3> as preferred memory if message is tried to

AT+CNMI	New Message Indication to TE
	<p>be stored</p> <p>message waiting indication group (store message): as in GSM 03.38, but use &lt;mem3&gt; as preferred memory</p>
<mt>1	as <mt>=0 but send indication if message stored successfully
<mt>2	<p>no class: route message to TE</p> <p>class 0: as in GSM 03.38, but also route message to TE and do not try to store it in memory</p> <p>class 1: route message to TE</p> <p>class 2: as &lt;mt&gt;=1</p> <p>class 3: route message to TE</p> <p>message waiting indication group (discard message): as in GSM 03.38, but also route message to TE and do not try to store it in memory</p> <p>message waiting indication group (store message): as &lt;mt&gt;=1</p>
<mt>3	<p>class 3: route message to TE</p> <p>others: as &lt;mt&gt;=1</p>
<bm>	<p>0 &lt;bm&gt; No CBM indications are routed to the TE</p> <p>If CBM is stored into ME/TA, indication of the memory location is routed to the TE using unsolicited result code:</p> <p>+CBMI: &lt;mem&gt;,&lt;index&gt;</p> <p>1 New CBMs are routed directly to the TE using unsolicited result code</p> <p>2 Class 3 CBMs are routed directly to TE using unsolicited result codes defined in &lt;bm&gt;=2. If CBM storage is supported, messages of other classes result in indication as defined in &lt;bm&gt;=1</p>

AT+CNMI	New Message Indication to TE
<ds>	<p>0 = No SMS-STATUS_REPORTs are routed to the TE</p> <p>1 = SMS-STATUS-REPORTs are routed to the TE using unsolicited result code.</p>
<bfr>	<p>0 =TA buffer of unsolicited result codes defined within this command is flushed to the TE when &lt;mode&gt; 1...2 is entered.</p> <p>1 = TA buffer of unsolicited result codes defined within this command is cleared when &lt;mode&gt; 1...2 is entered.</p>
Notes	For CDMA Version: The 3rd and 4th params are ignored and are not range checked.

# AT+CPMS Preferred Message Storage

AT+CPMS	Preferred Message Storage
Command Function	Set command selects memory storages <mem1>, <mem2> and <mem3> to be used for reading, writing, etc.
Syntax Query	AT+CPMS=?
Syntax	+CPMS: "ME","MT" OK
Write Syntax	AT+CPMS=<mem1>,<mem2>,<mem3>
Write Response	OK
Read Syntax	AT+CPMS?
Read Response	+CPMS: <mem1>,<usedSpace1>,<maxSpace1>,<mem2>,<usedSpace2>,<maxSpace2>,<mem3>,<usedSpace3>,<maxSpace3> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<mem1>	String type; memory from which messages are read and deleted (commands List Messages +CMGL, Read Message +CMGR and Delete Message +CMGD) defined values: "ME" = ME message storage "MT" = Any of the message storage associated with the device

AT+CPMS	Preferred Message Storage
<mem2>	String type; memory to which writing and sending operations are made (commands Send Message from Storage +CMSS and Write Message to Memory +CMGW) refer to <mem1> for defined values
<mem3>	String type; memory to which received messages are preferred to be stored (unless class of message defines a specific storage location; refer to command New Message Indications +CNMI) refer to <mem1> for defined values
Notes	

# API Commands

---



# TCP API Commands

## AT\$ACKTM Acknowledgment Message Period & Retry Number

AT\$ACKTM	Acknowledgment Message Period & Retry Number
Command Function	This command allows the user to configure the modem msg acknowledge behavior. If server acknowledgement is selected for a message, the message will be re-sent every <retry period> number of seconds until the acknowledge message sequence is complete, or until an acknowledge message is received from a server. An acknowledge message sequence consists of sending <max retry> messages to each server friend in sequence (i.e. server 2 is contacted after all retries for server 1 is complete) and is complete when each server friend has received <max retry> messages, or upon receipt of an acknowledge message from a server.
Syntax Query	AT\$ACKTM=?
Syntax	\$ACKTM: (1-255),(0-3600),(0-1) OK
Write Syntax	AT\$ACKTM=<max retry>,<retry period>,<IP reselect>
Write Response	OK
Read Syntax	AT\$ACKTM?
Read Response	\$ACKTM: <max retry>,<retry period >,<IP reselect>  OK
Execute Syntax	N/A
Execute Response	
Unsolicited Response	N/A
Parameter Values	
<max retry>	<p>The maximum number of times an acknowledge message is re-sent to a single friend server. After all retries to the friend server are exhausted, the modem will move on to the next friend server if one exists. If there are no more friend servers available, the message will be discarded.</p> <p>In the case of the default acknowledge wakeup message: The maximum number of wakeup messages the modem will send to each server friend upon receipt of a new IP, or upon expiration of</p>



# AT\$TCPAPI TCP API Control

AT\$TCPAPI	TCP API Control
Command Function	This command allows the user to initiate and terminate and query the status of the TCP API connection. Please note that the TCP API can only be used over the air.
Syntax Query	AT\$TCPAPI=?
Syntax	\$TCPAPI: (0-1) OK
Write Syntax	AT\$TCPAPI=<Status>
Write Response	OK
Read Syntax	AT\$TCPAPI?
Read Response	\$TCPAPI: <Status> (M-<Mgr Task>,R-<Rec Task>,T-<Trans Task>,<Idx <Friend Index>)
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<Status>	TCP API connection status 0=Disabled 1=Enabled
<Mgr Task>	TCP API Manager Task 0 = None 1 = Init 2 = Idle 3 = Connecting 4 = Connected 5 = Disconnecting
<Rec Task>	TCP API Receive Task 0 = None 1 = Init 2 = Idle

AT\$TCPAPI	TCP API Control
	3 = Connected 4 = Waiting for Header 5 = Waiting for Frame 6 = Processing API Command
<Trans Task>	TCP API Transmit Task 0 = None 1 = Init 2 = Idle 3 = Connected 4 = Sending
<Friend Index>	Friend Index (1 - 10, 255) Note: 255 = not connected
Notes	N/A

# AT\$TCPIDLETO TCP API Idle Timeout

<b>AT\$TCPIDLETO</b>	<b>TCP API Idle Timeout</b>
<b>Command Function</b>	Specifies the number of seconds without data traffic, in either direction, before closing the connection.
<b>Syntax Query</b>	AT\$TCPIDLETO=?
<b>Syntax</b>	\$TCPIDLETO: (10-86400) OK
<b>Write Syntax</b>	AT\$TCPIDLETO=<Timeout>
<b>Write Response</b>	OK
<b>Read Syntax</b>	AT\$TCPIDLETO?
<b>Read Response</b>	\$TCPIDLETO: <Timeout>
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;Timeout&gt;</b>	TCP API idle timeout value in seconds
<b>Notes</b>	After closing the connection, the device will attempt to reconnect using the FRIEND list.

## AT\$TCPRESTR TCP API Restart

AT\$TCPRESTR	TCP API Restart
Command Function	If a connection exists, it is dropped and a new connection is attempted starting at the beginning of the Friend list.
Syntax Query	AT\$TCPRESTR=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$TCPRESTR
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A

# AT\$TCPRETRYTO TCP API Retry Timeout

<b>AT\$TCPRETRYTO</b>	<b>TCP API Retry Timeout</b>
<b>Command Function</b>	Specifies the number of seconds without receiving a TCP level ACK that will cause the connection to be closed.
<b>Syntax Query</b>	AT\$TCPRETRYTO=?
<b>Syntax</b>	\$TCPRETRYTO: (120-65535) OK
<b>Write Syntax</b>	AT\$TCPRETRYTO=<Timeout>
<b>Write Response</b>	OK
<b>Read Syntax</b>	AT\$TCPRETRYTO?
<b>Read Response</b>	\$TCPRETRYTO: <Timeout>
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;Timeout&gt;</b>	TCP API retry timeout value in seconds
<b>Notes</b>	<p>After closing the connection, the device will attempt to reconnect using the FRIEND list. The purpose of this command is to provide an abort to the TCP stack level retries.</p> <p>Currently, the number of retries is 10 and the amount of time varies based on calculated round trip time. The minimum time allowed is 120 seconds.</p> <p>Attempts to set the retry timeout to a value less than 120 or more than 65535 will result in an error.</p>

# AT\$TCPSRC TCP API Source Ports

AT\$TCPSRC	TCP API Source Ports
Command Function	Specifies the TCP API source port range used when making a TCPAPI connection.
Syntax Query	AT\$TCPSRC=?
Syntax	\$TCPSRC: (1024-65535),(1024-65535) OK
Write Syntax	AT\$TCPSRC=<Start Port Number>,[<End Port Number>]
Write Response	OK
Read Syntax	AT\$TCPSRC?
Read Response	\$TCPSRC: <Start Port Number>,<End Port Number>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<Start Port Number>	TCP API starting port number
<End Port Number>	TCP API ending port number
Notes	<p>Each connection attempt uses the next port number in sequence until the end port is passed. When this happens the port is set to the start port number.</p> <p>This current port number in use is retained over a power cycle.</p> <p>If only the start port number is provided, the end port number will be start port number + 49 (range of 50).</p> <p>The default start port number is 1024. The modem will increment the port number with each connection attempt until it reaches the default end port number of 65535. Once the modem reaches the end port number it will begin incrementing the ports again using the start port number of 1024.</p>



# AT\$TCPSTATS TCP API Statistics

AT\$TCPSTATS	TCP API Statistics
Command Function	Displays bytes transmitted and received since last reset or last AT\$TCPSTATS=0 command.
Syntax Query	AT\$TCPSTATS=?
Syntax	\$TCPSTATS: (0) OK
Write Syntax	AT\$TCPSTATS=<Clear>
Write Response	OK
Read Syntax	AT\$TCPSTATS?
Read Response	\$TCPSTATS: Rx <Rx Bytes>,Tx <Tx Bytes>,M <Mode Changes>,D <Data Deactivates>,R <Restarts>,C <Connection Timeouts>,I <Idle Timeouts>,S <Socket Errors>,Rt<Retry Timeouts>,Rc<Reconnect Timeouts>, Dc<Disconnect Timeouts>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<clear>	0 = clear TCPAPI statistics
<Rx Bytes>	TCP API bytes received
<Tx Bytes>	TCP API bytes transmitted
<Mode Changes>	Counts the number of times the TCPAPI connection was closed due to configuration changes like AT\$TCPAPI, AT\$TCPSRC, and AT\$FRIEND.
<Data Deactivates>	Counts the number of times the TCPAPI connection was closed due to the loss of the network data connection.
<Restarts>	TCP API restarts (AT\$TCPRESTR)
<Connection Timeouts>	TCP API connection timeouts
<Idle Timeouts>	TCP API idle timeous.
<Socket Errors>	TCP API socket errors. These inclise connection refuses and socket read/write errors possibly caused by the TCP server friend.

AT\$TCPSTATS	TCP API Statistics
<Retry Timeouts>	The number of times a connection was closed due to Retry timer expirations.
<Reconnect Timeouts>	The number of times the reconnect timer expired. The reconnect timer restarts a TCPAPI connection when not connected and TCPAPI mode is enabled.
<Disconnect Timeouts>	The number of times the Disconnect timer expired while closing a TCPAPI connection for any reason.
Notes	AT\$TCPSTATS=0 will clear all TCP API statistics.

# AT\$TCPURC Enable URC Over TCP API

AT\$TCPURC	Enable URC Over TCP API
Command Function	This command is used to enable and disable sending Unsolicited Response Codes over TCPAPI.
Syntax Query	AT\$TCPURC=?
Response	\$TCPURC: (0-1)
OK	
Write Syntax	AT\$TCPURC=<MODE>
Response	OK
Read Syntax	AT\$TCPURC?
Response	\$TCPURC: <MODE>
OK	
Execute Syntax	N/A
Response	N/A
Parameter Values	
<MODE>	0 = Disabled 1 = Enabled
Notes	

# Message Commands

---

# AT\$APIOPT Enable API Optional Header Fields

AT\$APIOPT	Enable API Optional Header Fields
Command Function	This command allows the user to enable specific Optional Header Fields to be included in the UDPAPI and TCPAPIs' API Optional Header.
Syntax Query	AT\$APIOPT=?
Syntax	\$APIOPT: (0-1),(0-1),(0-4),(0-1),(0-1),(0-1),(0-2),(0-1) OK
Write Syntax	AT\$APIOPT=<MDMID>,<Msg Event Format>,<Event Seq Num>,<HdrDisable>,<Output Event Type>,<HexModemID>,<SendParam3>
Write Response	OK
Read Syntax	AT\$APIOPT?
Read Response	\$APIOPT=<MDMID>,<Msg Event Format>,<Event Seq Num>,<HdrDisable>,<Output Event Type>,<HexModemID>,<DriverId>,<SendParam3>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<MDMID>	0 = Disable sending of MDMID value in TCPAPI or UDPAPI Header 1 = Enable sending of MDMID value in TCPAPI or UDPAPI Header
<Msg Event Format>	0 = Disable sending of Output Message Event Format value in TCPAPI or UDPAPI Header 1 = Enable sending of Output Message Event Format in TCPAPI or UDPAPI Header
<Event Seq Num>	0 = Disable sending of Event Sequence Number value in TCPAPI or UDPAPI Header 1 = Enable sending of the least significant byte of the Event Sequence Number in TCPAPI or UDPAPI Header 2 = Enable sending of the two least significant bytes of the

AT\$APIOPT	Enable API Optional Header Fields
	<p>Event Sequence Number in TCPAPI or UDPAPI Header</p> <p>3 = Enable sending of the three least significant bytes of the Event Sequence Number in TCPAPI or UDPAPI Header</p> <p>4 = Enable sending of the full four bytes of the Event Sequence Number in TCPAPI or UDPAPI Header</p>
<HdrDisable>	<p>0 = UDPAPI header is enabled (default)</p> <p>1 = UDPAPI header is disabled for UDP messages sent via event engine. This flag has no effect on commands sent from the server or the ack sequence. This feature is intended to be used with UDP messages when the size of the data packet is critical due to the rapid rate at which messages are sent over an extended period of time.</p>
<Output Event Type>	<p>0 = Disable sending of Output Event number in TCPAPI or UDPAPI Header</p> <p>1 = Enable sending of Output Event number in TCPAPI or UDPAPI Header</p>
<HexModemID>	<p>0 = Do not include the hex modem ID in the optional header</p> <p>1 = Include the hex modem ID in the optional header</p> <p>Note: The optional header type for hex modem ID is 7.</p>
<DriverId>	<p>0 = Do not include &lt;DriverId&gt; ibutton serial number</p> <p>1 = 4 bytes of &lt;DriverId&gt;, 6 bytes total - 06 08 aa bb cc dd, where aa bb cc dd is the least significant 4 bytes of the ibutton serial number</p> <p>2 = 8 bytes of &lt;DriverId&gt;, 10 bytes total - 0A 08 aa bb cc dd ee ff gg hh, with the full 8 bytes of the ibutton serial number.</p>

AT\$APIOPT	Enable API Optional Header Fields
<SendParam3>	<p>0 = Disable sending of Parameter 3 data in the optional header</p> <p>1 = Enable sending of Parameter 3 data in the optional header</p>
Notes	<p>When a message is formatted for transmission via the event engine (UDP, TCP, SMS, or serial), we call the function to retrieve/increment the sequence number. The sequence number is passed to the task that actually makes the decision about what to do with the message. If the message is SMS or serial, the sequence number doesn't actually get used in the end, but does get incremented.</p> <p>To support the hex modem ID feature, each MDMID character must be a decimal character between '0' and '9'. If the MDMID contains a character outside of this range, the hex MDMID will be the NULL string. If all MDMID characters are decimal characters, then the decimal MDMID is converted into the hexadecimal equivalent value.</p>
Examples	<p>This example will disable the sending of the MDMID, enable the Msg Event Format and enable the full Event Sequence Number.</p> <p>AT\$APIOPT=0,1,4</p> <p>This example will disable the sending of the MDMID, enable the Msg Event Format, enable the full Event Sequence Number, and enable sending of Parameter 3 data in the optional header</p> <p>AT\$APIOPT=0,1,4,0,0,0,0,1</p> <p>This example will enable the sending of the MDMID and the least two significant bytes of the Event Sequence Number. The state of the Enable Msg Event Format will remain unchanged</p> <p>AT\$APIOPT=1,,2</p> <p>This example will enable the sending of the MDMID and disable the Msg Event Format. The state of the Event Seq Num will remain unchanged</p> <p>AT\$APIOPT=1,0</p> <p>Hex Modem ID Examples:</p> <p>AT\$APIOPT=0,,,,,1</p> <p>MDMID - "001036000211071"</p>

AT\$APIOPT	Enable API Optional Header Fields
	{07}{07}{F1}{36}{6C}{B0}{7F}  MDMID - "99" {03}{07}{63}  MDMID - "999999999999999999A" {03}{07}{00}



# AT\$APIPWD API Password

AT\$APIPWD	API Password
<b>Command Function</b>	This command allows the user to query/set the API password. Any IP address will be able to access the device if the password is not set (null). If the password is set, control is given to the last friend IP that supplied the correct password.
<b>Syntax Query</b>	AT\$APIPWD=?
<b>Syntax</b>	\$APIPWD: ("PASSWORD") OK
<b>Write Syntax</b>	AT\$APIPWD="<API password>"
<b>Write Response</b>	OK
<b>Read Syntax</b>	AT\$APIPWD?
<b>Read Response</b>	\$APIPWD: "<API password>"
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;API password&gt;</b>	Maximum of 8 characters. A NULL password indicates ALL remote users are allowed API access.
<b>Notes</b>	If APIPWD is set, the modem will require the correct password from a "friendly IP" (set by AT\$FRIEND command) before accepting any remote requests. Remote requests are only accepted from the last "friendly IP" that sent the correct password. If APIPWD is null, then all remote requests, from any IP address, will be accepted. The password is a maximum of 8 characters.

# AT\$CHKIN Configure Device Check-in

AT\$CHKIN	Configure Device Check-in
Command Function	This command adds event group 8 to the event table and configures the device check-in feature. The feature is used to send a wakeup type message to an alternate server (server other than the one the device normally reports). This alternate server is listed in the \$FRIEND list with a usage=4.
Syntax Query	AT\$CHKIN=?
Syntax	\$CHKIN: (0-1),(0-43200) OK
Write Syntax	AT\$CHKIN=<enable><periodic_interval>
Write Response	OK
Read Syntax	AT\$CHKIN?
Read Response	\$CHKIN: 1,43200 OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<enable>	0 = remove group 8 from the event table. 1 = add event group 8 to the event table.

AT\$CHKIN	Configure Device Check-in
<periodic_interval>	<p>Sets the periodic reporting interval (in minutes)</p> <p>Range: 0-43200</p> <p>Default: 43200</p>
Notes	<p>Input event 200 is used for the periodic check-in timer.</p> <p>Output event 152 is used for sending UDP messages to the check-in servers specified in the \$FRIEND list.</p> <p>NOTE: Setting AT\$CHKIN=1 will overwrite any existing event group 8.</p>
Example:	<p>AT\$CHKIN=1,x adds event group 8 to the event table.</p> <p>AT\$CHKIN=0,x removes event group 8 from the event table.</p>

# AT\$FID Get Factory ID

AT\$FID	Get Factory ID
Command Function	This command gets theFactory ID
Syntax Query	AT\$FID=?
Syntax	\$FID: ("FACTORY ID") OK
Write Syntax	AT\$FID ="<Factory ID>"
Write Response	OK
Read Syntax	AT\$FID?
Read Response	\$FID: "<Factory ID>"
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<Factory ID>	0-20 character string in ASCII format.
Notes	

# AT\$FRIEND Modem Friends

AT\$FRIEND	Modem Friends
<b>Command Function</b>	This command allows the user to configure the modem friend/server list. A friend is always allowed remote API access. Friend servers can be configured to receive WAKEUP messages whenever the modem receives a new IP, or after a certain period has elapsed. (see AT\$WAKEUP)
<b>Syntax Query</b>	AT\$FRIEND=?
<b>Syntax</b>	\$FRIEND: (1-10),(0-2),"(0-255).(0-255).(0-255).(0-255)" ,(0-65535), (1,2,4) OK
<b>Write Syntax</b>	AT\$FRIEND =<friend number>,<server indication>,"<friend IP> or <DNS name>",<destination port>,<usage>
<b>Write Response</b>	OK
<b>Read Syntax</b>	AT\$FRIEND?
<b>Read Response</b>	\$FRIEND: =01, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =02, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =03, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =04, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =05, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =06, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =07, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =08, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =09, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage> \$FRIEND: =10, <server indication>,"<friend IP> or <DNS name>",<destination port>,<usage>

<b>AT\$FRIEND</b>	<b>Modem Friends</b>
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;friend number&gt;</b>	friend identification (1-10).
<b>&lt;server indication&gt;</b>	<p>0 = Server is disabled.</p> <p>1 = Server is enabled.</p> <p>2 = Modem will accept and respond to incoming AT commands on this port as long as the source IP address matches the IP entry. A value of 255 can be entered for any octet of the IP address and will be treated as a wildcard that will match any value for the corresponding octet of the incoming IP address. Since this entry does not represent a specific IP address, no attempt to sent wakeup/event messages to this friend entry will be made.</p>
<b>&lt;friend IP&gt;</b>	friend IP value (including wildcard values of 255 if <server indication> = 2).
<b>OR</b>	
<b>&lt;DNS name&gt;</b>	friend DNS name
<b>&lt;destination port&gt;</b>	<p>Friend destination port.</p> <p>Note: If &lt;usage&gt; = 4, this parameter is used to configure the UDP port for the AT\$CHKIN command.</p>

<b>AT\$FRIEND</b>	<b>Modem Friends</b>
<b>&lt;usage&gt;</b>	1 = TCPAPI 2 = UDPAPI (default) 4 = Alternate server for UDPAPI check-in message (see AT\$CHKIN)
<b>Notes</b>	<p>The AT&amp;F command does not modify the server address. It sets the usage parameter to the default value for all active servers. It clears the server indication and destination port fields.</p> <p>If the server indication is disabled for a friend server, then the UDP/TCP socket is not opened for communication. This action prevents the server from sending and receiving messages from the modem.</p>
<b>Example</b>	AT\$FRIEND=9,2,"193.45.255.255",1735,2 <p>Would accept and respond to UDP AT commands on port 1735 from any IP address between 193.45.0.0 to 193.45.254.254. Wakeup and other outgoing messages would not be sent to these addresses (unless one of the addresses had a separate entry in the friends list).</p>

# AT\$IDENT Modem Identification Used In API Optional Header

<b>AT\$IDENT</b>	<b>Modem identification used in API optional header</b>
<b>Command Function</b>	This command is used to specify which form of modem identification is used in the API optional header.
<b>Syntax Query</b>	AT\$IDENT=?
<b>Syntax</b>	\$IDENT: (0-4),(0-9) OK
<b>Write Syntax</b>	AT\$IDENT =<IDENTIFIER>,<PREPEND DIGIT>
<b>Write Response</b>	OK or ERROR
<b>Read Syntax</b>	AT\$IDENT?
<b>Read Response</b>	\$IDENT: <IDENTIFIER>,<PREPEND DIGIT>,<OTA IDENTIFIER>
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;IDENTIFIER&gt;</b>	API optional header modem identification 0 = MDMID 1 = MEID 2 = pESN 3 = FID 4 = binary ESN with prepended digit
<b>&lt;PREPEND DIGIT&gt;</b>	The digit that is prepended to the binary ESN. This is only valid with an <IDENTIFIER> value of 4.
<b>&lt;OTA IDENTIFIER&gt;</b>	The identification data sent in the API optional header
<b>Notes</b>	When using <IDENTIFIER> = 4 a new tag is used in the optional header. The new tag is: APIOPTHDR_HESN_TYPE = 10.
<b>Examples</b>	



# AT\$MDMID Modem ID

AT\$MDMID	Modem ID
Command Function	This command allows the user to query/set the modem ID. The modem ID is copied into each wakeup message sent from the modem. (see AT\$WAKEUP)
Syntax Query	AT\$MDMID=?
Syntax	\$MDMID: ("MODEM ID") OK
Write Syntax	AT\$MDMID = "<modem ID>"
Write Response	OK
Read Syntax	AT\$MDMID?
Read Response	\$MDMID: "<modem ID>"
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<modem ID >	0-20 character string in ASCII format.
Notes	The modem's Mobile Equipment Identifier (MEID) is used as the default modem ID.

# AT\$MLQSZ Message Log Queue Size

AT\$MLQSZ	Message Log Queue Size
Command Function	The \$MLQSZ allows the user to restrict the number of unsent messages maintained in the buffers for the message log queues.
Syntax Query	AT\$MLQSZ=?
Syntax	\$MLQSZ:(0-3000),(0-3000),(0-3000),(0-3000),(0-3000),(0-3000) OK
Write Syntax	AT\$MLQSZ=<UdpQsize>,<SmsBckupQsize>,<SmsQsize>,<TcpQsize>,<UdpAckQsize>,<UdpAckSmsBckupQsize>
Write Response	OK
Read Syntax	AT\$MLQSZ?
Read Response	\$MLQSZ: <UdpQsize>,<SmsBckupQsize>,<SmsQsize>,<TcpQsize>,<UdpAckQsize>,<UdpAckSmsBckupQsize> OK
Execute Syntax	N/A
Execute Response	
Unsolicited Response	N/A
Parameter Values	
<UdpQsize>	Maximum # of unsent messages stored in message log queue for UDP messages. Valid range is 0-3000. The default is 0, which is treated as a special value indicating no message size restriction. This is the equivalent of restricting the size of this queue to 3000 messages.
<SmsBckupQsize>	Maximum # of unsent messages stored in message log queue for UDP messages where SMS messaging is allowed as a backup when UDP messaging is unavailable. Valid range is 0-3000. The default is 0, which is treated as a special value indicating no message size restriction. This is the equivalent of restricting the size of this queue to 3000 messages.
<SmsQsize>	Maximum # of unsent messages stored in message log queue for SMS messages. Valid range is 0-3000. The default is 0,

AT\$MLQSZ	Message Log Queue Size
	which is treated as a special value indicating no message size restriction. This is the equivalent of restricting the size of this queue to 3000 messages.
<TcpQsize>	Maximum # of unsent messages stored in message log queue for TCP messages. Valid range is 0-3000. The default is 0, which is treated as a special value indicating no message size restriction. This is the equivalent of restricting the size of this queue to 3000 messages.
<UdpAckQsize>	Maximum # of unsent messages stored in message log queue for UDP with Acknowledgement messages. Valid range is 0-3000. The default is 0, which is treated as a special value indicating no message size restriction. This is the equivalent of restricting the size of this queue to 3000 messages.

AT\$MLQSZ	Message Log Queue Size
<UdpAckSmsBckupQsize>	<p>Maximum # of unsent messages stored in message log queue for UDP with Acknowledgement messages where SMS messaging is allowed as a backup when UDP messageing is unavailable. Valid range is 0-3000. The default is 0, which is treated as a special value indicating no message size restriction. This is the equivalent of restricting the size of this queue to 3000 messages.</p>
Notes	<p>Over-the-air messages sent via the event engine are stored in flash and indexed via one of the message log queues. Due to indexing restrictions, no queue can contain more than 3000 unsent messages.</p> <p>For performance reasons, messages formatted via the event engine re stored in an array of 256 1000-byte flash files. Indexing for these messages is maintained in separate flash files (one for each queue).</p> <p>Normally, the number of unsent messages that can be stored in flash is a function of the size of the messages. For example, when storing messages containing only NMEA data, the message log files will be completely filled after about 2500 messages. Binary messages are typically smaller and take up less space in the message log files, but are still subject to the 3000 message maximum per queue.</p> <p>When the message log files are full OR when an individual queue exceeds 3000 messages, the oldest log file is deleted so that newer messages can be stored in a circular fashion using a new file in the place made available by deleting the oldest file. For some applications, the message buffers can contain far more unsent messages than is desired.</p> <p>The device only saves messages in the message log if the STORE_MSG (PARAM2 bit 6) is set. Wakeup messages and other messages which do not have the STORE_MSG bit set will not be stored in the message log. This is a change from the behavior of other Novatel Wireless devices. (GSM/GPRS devices will save some messages in the message log depending upon the network status even though the STORE_MSG bit is not set.)</p> <p>The purpose of this command is to mark older messages as 'sent' so that the number of messages in the queue never exceeds the threshold set by the user.</p> <p>Please note that the values specified in this command are not reflected in the value calculated for input event 61 (memory full percentage). That percentage represents the worst case (largest percentage) of either the % of the size of all unsent messages with respect to the size of all 256 log files OR the %</p>

AT\$MLQSZ	Message Log Queue Size
	<p>of unsent messages with respect to the 3000 messages allowed for an individual queue.</p> <p>This command is used in conjunction with MSGLOGDMP and allows the user to restrict the number of unsent messages maintained in the buffers for the message log queues. To dump the message the user needs to turn off the GPS receiver with the \$GPSCMD for those products which support GPS.</p>

# AT\$MSGLOGCL Message Log Clear

AT\$MSGLOGCL	Message Log Clear
Command Function	The \$MSGLOGCL command erases the log file.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$MSGLOGCL
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	None
Notes	If AT&F is performed, the message log buffer will be cleared.

# AT\$MSGLOGDMP Dump Unsent Messages To Serial Port

AT\$MSGLOGDMP	Dump Unsent Messages to Serial Port
Command Function	This command allows the user to dump the contents of the unsent messages to the serial port. This command is non-destructive in that it does not actually remove the messages from the queue
Syntax Query	AT\$MSGLOGDMP=?
Syntax	\$MSGLOGDMP: (0-5),(0-1),(1-83),(0-1) OK
Write Syntax	AT\$MSGLOGDMP=<queue>,<format>,<bytes_per_line>,<display_all>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
<queue>	<p>0 = event data that was configured to be sent to a remote server via UDP API only.</p> <p>1 = event data that was configured to be sent to a remote server via UDP API primarily but also use SMS as backup method if UDP is not available.</p> <p>2 = event data that was configured to be sent to a remote server via SMS only.</p> <p>3 = event data that was configured to be sent to a remote server via TCPAPI only.</p> <p>4 = event data that was configured to be sent to a remote server via UDP API with Acknowledgement messages.</p> <p>5 = event data that was configured to be sent to a remote server via UDP API with Acknowledgement messages primarily but also use SMS as backup method if UDP is not available</p>
<format>	0 = ASCII format (if message contains a byte that is not a printable ASCII character, it will be displayed as '?')

AT\$MSGLOGDMP	Dump Unsent Messages to Serial Port
	1 = hex format (Each byte in message is displayed as a two-digit hex character representing the value of the byte with spaces between each byte. Maximum of 16 bytes per line.)
<bytes_per_line>	1-83 (default = 16) number of bytes displayed per line for binary data (each byte is represented as a two-digit hex value followed by a space)
<display_all>	0 = display unsent messages only (default) 1 = display all messages (unsent and sent) from the indicated queue
Notes	<p>This command was developed primarily as a troubleshooting utility to help debug problems related to handling unsent messages in flash.</p> <p>However, it has also been useful in collecting GPS data when other memory storage is not available and it is not possible to connect the unit to a laptop. With this utility, you can put multiple devices in a vehicle for a drive test to collect NMEA data and dump the data for analysis easily when you return.</p> <p>GPS must be off to forward the data (on products that support GPS).</p> <p>When using larger bytes per line values (~30 and above), the user may need to issue the AT\$EVTOFF=1 command to successfully dump data via this command (AT\$MSGLOGDMP).</p>



# AT\$MSGLOGEN Message Log Enable

AT\$MSGLOGEN	Message Log Enable
Command Function	The \$MSGLOGEN command has been created to enable or disable saving Event data generated via the event engine in modem's memory
Syntax Query	AT\$MSGLOGEN=?
Syntax	\$MSGLOGEN: (0-1) OK
Write Syntax	AT\$MSGLOGEN=<setting>
Write Response	OK
Read Syntax	AT\$MSGLOGEN?
Read Response	\$MSGLOGEN: <setting>
Execute Syntax	
Execute Response	
Unsolicited Response	N/A
Parameter Values	
<setting>	0 - 1 (possible valid values)  0 = Disable message logging (default). Event data is sent to the remote server upon occurrence.  1 = Enable message logging. Event data has to be read via AT\$MSGLOGRD command or when AT\$MSGLOGEN=0 is sent.
Notes	If AT\$MSGLOGEN command was enabled and any unsent messages exist in memory, then the unsent data will be sent to the remote server when data logging is disabled.

# AT\$MSGLOGRD Message Log Read Data

AT\$MSGLOGRD	Message Log Read Data
Command Function	The \$MSGLOGRD command is used to query the status of the four event-engine message logs. Using the queue status this command can be used to resend messages that have already been sent and are still in the log.
Syntax Query	AT\$MSGLOGRD=?
Syntax	\$MSGLOGRD: (0-5),(0-2999),(0-2999) OK
Write Syntax	\$MSGLOGRD=<queue>,<number of messages>,<starting index>
Write Response	OK
Read Syntax	AT\$MSGLOGRD?
Read Response	<p>\$MSGLOGRD: 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0</p> <p>OK</p> <p>AT\$MSGLOGRD? command returns 108 values:</p> <p>\$MSGLOGRD: udp1,udp2,udpsms1,udpsms2,sms1,sms2,tcp1,tcp2,udpak1,udpack2,udpacksms1,udpacksms2</p> <p>First pair of numbers: unsent UDP messages, total number of UDP messages in the UDP message log.</p> <p>Second pair of numbers: unsent UDP messages with SMS as backup, total number of UDP messages with SMS as backup in the UDP/SMS log.</p> <p>Third Pair of numbers: unsent SMS messages, total number of SMS messages in the SMS log.</p> <p>Fourth Pair of messages: unsent messages via TCP, total number of messages in the TCP log.</p> <p>Fifth pair of messages: UDP with Acknowledgement messages.</p> <p>Sixth pair of messages: UDP with Acknowledgement messages where SMS messaging is allowed as a backup when UDP messaging is unavailable.</p> <p>Each value is comma (,) delimited.</p> <p>The first value of any queue represents “Total Number of Unsent Messages”. This value can be used as the &lt;number of messages&gt; field while resending messages.</p>

AT\$MSGLOGRD	Message Log Read Data
	The second value of any queue represents: "Total Number of Messages Stored for that Queue". Subtract the "Total Number of Unsent Messages" from the "Total Number of Messages Stored for that Queue" and use that as the <starting index> to resend all of the messages still in the log.
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<queue>	<p>0 = event data that was configured to be sent to a remote server via UDP API only.</p> <p>1 = event data that was configured to be sent to a remote server via UDP API primarily but also use SMS as backup method if UDP is not available</p> <p>2 = event data that was configured to be sent to a remote server via SMS only</p> <p>3 = event data that was configured to be sent to a remote server via TCP API only</p> <p>4 = event data that was configured to be sent to a remote server via UDP API with Acknowledgement messages</p> <p>5 = event data that was configured to be sent to a remote server via UDP API with Acknowledgement messages primarily but also use SMS as backup method if UDP is not available</p>

AT\$MSGLOGRD	Message Log Read Data
<number of messages>	<p>x</p> <p>x = total number of messages one desires to read from the memory. A user can choose to read 1 message in which case x = 1 or the user can read all messages in which case x = the total number of messages available in the log (use the \$MSGLOGRD? command to determine the total number of messages available for the log of interest).</p> <p>Note: x cannot be greater than the total number of stored messages in the queue.</p>
<starting index>	<p>y</p> <p>y = starting index number of messages that are stored in the memory.</p> <p>NOTE: y cannot be greater than total number of stored messages for the log.</p>
Notes	<p>If AT&amp;F is performed, the message log buffer will be cleared.</p> <p>The message logs are cleared when the modem id is changed (AT\$MDMID).</p> <p>The message logs work on a FIFO basis. I.e. when a log is full the oldest messages are deleted to make room for new messages.</p>

# AT\$MSGSEND Message Send

AT\$MSGSEND	Message Send
Command Function	The \$MSGSEND command has been created to allow sending of data from one mode to another.
Syntax Query	AT\$MSGSEND=?
Syntax	AT\$MSGSEND: (0-5),("DATA") OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$MSGSEND=<destination>,<"data">
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	

AT\$MSGSEND	Message Send
<destination>	<p>0 – 5 (possible valid values)</p> <p>0 = &lt;"data"&gt; is sent out the USB port</p> <p>1 = &lt;"data"&gt; is sent to all SMS addresses listed in AT\$SMSDA command.</p> <p>2 = &lt;"data"&gt; is sent via UDP API to IP address and port number of the first UDP server listed in the AT\$FRIEND command</p> <p>3 = &lt;"data"&gt; is sent via PAD to IP address and Port number listed in the AT\$PADDEST command</p> <p>4 = &lt;"data"&gt; is sent via TCP API to IP address and port number of the first TCP server listed in the AT\$FRIEND command</p> <p>5 = &lt;"data"&gt; is sent out the USB port</p>
<"data">	a maximum of 50 bytes ASCII characters
Notes	<p>Use the AT\$CMDPORT command to select the primary UART serial port and/or the USB AT command port.</p> <p>Messages sent to OTA destinations (options 1, 2, 3, and 4) are transmitted immediately. If network coverage is not available, then the message is silently discarded. The message is not stored for later transmission when the modem returns to good network coverage.</p> <p>AT\$MSGSEND command can be sent to the device via SMS, UDP-API, or serial port.</p> <p>AT\$MSGSEND=4 checks for TCPAPI usage in the first \$FRIEND entry. AT\$MSGSEND=2 checks for UDPAPI usage in the first \$FRIEND entry. ERROR will be reported if invalid usage is selected.</p> <p>AT\$MSGSEND=4 checks if TCPAPI is enabled (AT\$TCPAPI=1) before attempting to send message. Will report error if not enabled.</p> <p>A carriage return can be added to a \$MSGSEND message by inserting "\OD" in to the text portion of the message</p> <p>For example:  AT\$MSGSEND=0,"Hello \OD"  OK  Hello</p> <p>To add a carriage return to a \$STOATEV use the following format:  AT\$STOATEV=1,AT\$MSGSEND=0,"Hello \OD"</p>

# AT\$SMSDA Destination Address For SMS Messages

AT\$SMSDA	Destination Address for SMS Messages
Command Function	This command allows a user to configure the phone number or email address for the sending of event data. It is also used in limiting the originating address for sending AT commands over SMS.
Syntax Query	AT\$SMSDA=?
Syntax	\$SMSDA: (1-5),"+1234...", "+123.."
Write Syntax	AT\$SMSDA=<index>,<dest addr>,<gateway number>
Write Response	OK
Read Syntax	AT\$SMSDA?
Read Response	\$SMSDA: 1,"<dest addr>","<gateway number>", \$SMSDA: 2,"<dest addr>","<gateway number>", \$SMSDA: 3,"<dest addr>","<gateway number>", \$SMSDA: 4,"<dest addr>","<gateway number>", \$SMSDA: 5,"<dest addr>","<gateway number>", OK
Execute Syntax Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<index>	1 - 5 defines the index number for destination address

AT\$SMSDA	Destination Address for SMS Messages
<dest addr>	38 characters or less phone number or email address (Phone numbers must match exactly as they appear in the AT+CGML="ALL" command response.)
<gateway>	7 characters or less gateway number for email address
Notes	<p>The ability to limit AT Commands over SMS is now also controlled by AT#SMSDAEN.</p> <p>If AT\$SMSDAEN=1, (default) and if there are no addresses populated in AT\$SMSDA, then all SMS with the correct AT command format addressed to the MSISDN of the device, and with the correct MDMID (if populated) will be allowed.</p> <p>To disable all AT Commands over SMS, set AT\$SMSDAEN=0.</p> <p>The gateway number is provided by the Network Provider and is only used for sending email over SMS. It is not required if you are sending SMS to a phone number.</p> <p>An AT\$EVENT command has to be set to send a message over SMS.</p>



# AT\$SMSDAEN Enable/Disable AT Commands Over SMS

AT\$SMSDAEN	Enable/Disable AT Commands over SMS
Command Function	This command allows the user to enable or disable the ability to allow AT commands over SMS
Syntax Query	AT\$SMSDAEN=?
Syntax	\$SMSDAEN: (0-1) OK
Write Syntax	AT\$SMSDAEN =<mode>
Write Response	OK
Read Syntax	AT\$SMSDAEN?
Read Response	AT\$SMSDAEN: <mode> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
< mode >	0 = Disables ability to send AT commands over SMS 1 = Enables the ability to send AT commands over SMS
Notes	<p>AT\$SMSDAEN is set to 1 by default. This allows SMS over AT commands to be sent to the modem. IF there is no entries in AT\$SMSDA, then all SMS with the correct AT command format, and addressed to the MSISDN of the device, and with the correct MDMID (if populated) will be allowed.</p> <p>AT\$SMSDAEN=0 will completely disable all AT COMMAND over SMS. Any SMS received, will be treated as normal SMS.</p> <p>To limit AT commands Over SMS to originating from a given address, ensure AT\$SMSDAEN=1 and populate the desired addresses in AT\$SMSDA.</p>
Example:	See App note GSM0308AN001- Sending AT Commands over SMS

# AT\$UDPMMSG Send And Receive UDP Messages

AT\$UDPMMSG	Send and Receive UDP Messages
Command Function	<p>This command allows the user to send UDP/IP data packets while in AT command mode. The destination IP address and port number are set by the AT\$FRIEND command. The modem must have a network data connection established (\$areg=2 command setting). Incoming messages addressed to the modem's IP which have a valid header and are from an acceptable IP address (valid friend) will be displayed on the serial port with the unsolicited response \$UDPMMSG: followed by the message.</p>
Syntax Query	AT\$UDPMMSG=?
Syntax	<p>\$UDPMMSG: (0-1),(0-2),("data"),(0-1)</p> <p>OK</p>
Write Syntax	<p>AT\$UDPMMSG=&lt;format&gt;,&lt;type&gt;,&lt;data&gt;,&lt;flash&gt;</p> <p>&lt;cr&gt;</p>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<format>	<p>0 = &lt;data&gt; is an ASCII string (i.e.: "is this is my data")</p> <p>1 = &lt;data&gt; is an ASCII-Hex bytes (i.e.: 050a25)</p>
<type>	<p>0 = message will only be sent to the first UDP server listed in AT\$FRIEND</p> <p>1 = message will be sent via the ACK method (controlled by \$ACKTM command) to the UDP server(s) listed in AT\$FRIEND</p> <p>2 = message will be sent to all UDP servers listed in AT\$FRIEND</p>

AT\$UDPMSG	Send and Receive UDP Messages
<data>	"ABCD" (Data to be transmitted in quotes) (NOTE: HEX format data shall always be entered as two ASCII characters per byte. ex: 0x5 should be entered as 05)
<flash>	<p>This optional parameter, when enabled, allows a user to store the message in the device's FLASH memory when the device is out of network coverage or not registered. Stored messages will be transmitted when the device reestablishes network data coverage.</p> <p>0 = Do not store messages in FLASH memory. 1 = Store messages in FLASH memory.</p>
Notes	<p>Data received from OTA shall be sent to the modem's serial port as: \$UDPMSG: &lt;text&gt; (ASCII or Binary data) (NOTE: Binary message will be displayed as two ASCII Hex characters)</p> <p>&lt;data&gt; field from the at\$udpmsg command will be sent to UDP servers listed in AT\$FRIEND.</p> <p>&lt;data&gt; sent or received OTA shall be appended with a 4-byte UDP-API header as follows:</p> <p>Bytes 0 - 1: First 2 bytes of &lt;data&gt; field  Byte 2: 0x06 for ASCII data type or 0x07 for Binary data type  Byte 3: reserved  Byte 4 - n: &lt;data&gt; minus the first two bytes</p> <p>* A minimum of 2 and maximum of 244 ASCII characters are support. For HEX, a minimum of 2 and maximum of 122 bytes are supported.</p>

# AT\$WAKEUP Modem To Server Wakeup/Keep Alive

AT\$WAKEUP	Modem to Server Wakeup/Keep Alive
Command Function	This command allows the user to configure the modem wakeup/keep alive parameters. These parameters control how the modem initiates contact with its server friends. Parameters can be selected so that a wakeup message sequence is executed every time the modem receives a new IP, and/or after a requested period has passed since the previous wakeup sequence has completed. A wakeup message sequence consists of sending <max retry> messages to each server friend in sequence (i.e. server 2 is contacted after all retries for server 1 is complete) and is complete when each server friend has received <max retry> messages, or upon receipt of an acknowledge message from a server.
Syntax Query	AT\$WAKEUP=?
Syntax	\$WAKEUP: (0-2),(0-10080) OK
Write Syntax	AT\$WAKEUP=<wakeup mode>,<retry period>
Write Response	OK
Read Syntax	AT\$WAKEUP?
Read Response	\$WAKEUP: <wakeup mode>,<retry period>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	

AT\$WAKEUP	Modem to Server Wakeup/Keep Alive
<wakeup mode>	<p>0 = No wakeup messages sent</p> <p>1 = Send one message upon receipt of new IP and every &lt;retry period&gt; minutes</p> <p>2 = send acknowledgement message using at\$acktm parameters upon receipt of new IP and every &lt;retry period&gt; minutes message</p>
<retry period >	The number of minutes for keep alive period. Zero indicates no retries.
Notes	<p>When this command is used, it will generate event group 0 events in the event table when the AT\$EVENT? command is issued.</p> <p>The &lt;retry period&gt; parameter of this command populates the event timer value when the AT\$EVTIM4? command is issued. The AT\$EVTIM value will be in seconds. The parameter will also generate additional event group 0 entries.</p> <p>If AT\$EVDEL=0 is issued or any entry for group 0 is deleted, this command MUST be re-entered for proper functionality. If a read command is issued, it will not reflect the true state of the AT\$WAKEUP setting.</p> <p>Wakeup messages are sent to the UDP servers specified in AT\$FRIEND.</p>

# AT%SNCNT Query (or Reset) The Byte Counters

AT%SNCNT	Query (or Reset) the Byte Counters.
Command Function	Returns (or resets) the byte counts of the current connection.
Syntax Query	AT%SNCNT=?
Syntax	OK %SNCNT: (0)
Write Syntax	%SNCNT=<rst>
Write Response	OK
Read Syntax	AT%SNCNT?
Read Response	OK %SNCNT: <upo>,<dno>,<upp>,<dnp><CR><LF>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<rst>	resets the counters if rst = 0
<upo>	uplink octets count.
<dno>	downlink octets count.
<upp>	uplink packets count.
<dnp>	downlink packets count.
Notes	When AT%SNCNT? Is sent, the modem will immediately respond with OK, signifying the modem is ready to receive AT commands. If there is a current context activation, an unsolicited response will be returned to the serial port with the counter data.

# Event Commands

---

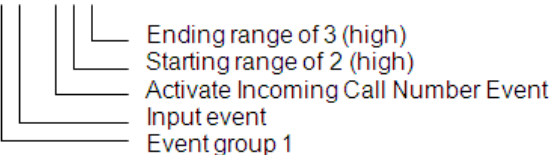
# AT\$ETSAV# Event Timer Save Configuration

AT\$ETSAV#	Event Timer Save Configuration
Command Function	<p>These commands allow the user to set/query a flag that is used to determine whether the event timer value will be persistent through a reset. If the flag is set for the timer, the timer count will be saved roughly once a minute and the saved value will be used as the starting value for the timer following a reset. This is intended to support long-range timers (for example, 3 hours) where resolution of +/-1 minute would be acceptable.</p> <p>If the flag is clear, the timer always starts at zero following an reset.</p> <p>The # sign represents timers 1 through 10 (i.e. ETSAV1).</p>
Syntax Query	AT\$ETSAV#=?
Syntax	\$ETSAV#:(0-1) OK
Write Syntax	AT\$ETSAV#=<flag>
Write Response	OK
Read Syntax	AT\$ETSAV#?
Read Response	\$ETSAV#: <flag>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<flag>	1 =save this event timer value roughly once a minute to flash 0 = restart this event timer at zero following reset (default)
Notes	AT&W must be issued after flag is set to ensure flag retains its status through a reset.
Example	N/A



# AT\$EVCID User Defined Incoming Call Number Event

AT\$EVCID	User defined incoming call number event
Command Function	This command allows the user to define up to 5 separate incoming call number user input events
Command Functional	Novatel Wireless Specific
Group	
Syntax Query	AT\$EVCID=?
Syntax	\$EVCID: (1-5),(0-2),"<ph_num>" OK
Write Syntax	AT\$EVCID=<entry>,<mode>[,<"number">]
Write Response	OK
Read Syntax	AT\$EVCID?
Read Response	\$EVCID: <entry>,<mode>,<"number">
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<entry>	1-5 = Selects which CID entry to modify
<mode>	0 = Disable event generation for incoming call number 1 = Enable event generation for incoming call number and suppress ring indication and respond to network with busy signal. 2 = Enable event generation for incoming call number and do not suppress ring indication.

AT\$EVCID	User defined incoming call number event
<number>	string type;Character string [-]<0..9,+,>?. Where <?> is a single character wildcard. If number starts with '~' it will match to any incoming call number with 0 or more digits preceding the remaining digits in the string. This is useful for matching to local, national and international ISDN telephony numbering plans.
Notes	In the event the incoming call number matches more than one incoming call number selection, the mode selection will be based on priority order. The priority order will be for entries 1 through 5 with entry 1 having the highest priority.
Example:	<p>These commands will cause the example in AT\$EVENT to trigger for incoming call numbers matching event call id 2 or event call id 3.</p> <pre>AT\$EVCID=2,1,"123456789?" // Define incoming call number                              with the last digit a wildcard AT\$EVCID=3,1,"~123456789" // Define incoming call number to allow                              For local and international prefixes</pre> <pre>AT\$EVENT=1,1,65,2,3</pre>  <p>Ending range of 3 (high)  Starting range of 2 (high)  Activate Incoming Call Number Event  Input event  Event group 1</p>

# AT\$EVDEL Delete Event

AT\$EVDEL	Delete Event
Command Function	This command allows the user to delete items from the user generated event table. Entering only the group number will delete the whole group.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$EVDEL=<group><letter ID>
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<group>	event list group number
<letter ID>	letter indicating which element of the group (optional)
Notes	
Example:	AT\$EVDEL=1 Will delete all entries event group 1 AT\$EVDEL=1b Will delete only the second entry in event group 1

# AT\$EVDELA Delete Event (All)

AT\$EVDELA	Delete Event (All)
Command Function	This command allows the user to delete all events from the event table.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$EVDELA
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	N/A



This command deletes ALL events from the event table including the default events that control the flashing of the status LEDs. If the default events are deleted, they must be manually recreated or use AT&F to restore the factory configuration.

---

# AT\$EVDEL R Delete A Range Of Event Groups

AT\$EVDEL R	Delete a Range of Event Groups
Command Function	This command deletes a range of event groups.
Syntax Query	AT\$EVDEL R=?
Syntax	\$EVDEL R: (0-255),(0-255) OK
Write Syntax	AT\$EVDEL R=<start>,<stop>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<start>	First group index in range to be deleted
<stop>	Last group index in range to be deleted.
Notes	N/A

# AT\$EVENT User Defined Input/Output

AT\$EVENT	User Defined Input/Output																																																						
Command Function	This command allows the user to customize the modem’s input and output capabilities. Any combination of input events can be monitored to trigger any combination of output events.																																																						
Syntax Query	AT\$EVENT=?																																																						
Syntax	\$EVENT: (0-255),(0-3),(0-255),(-2147483647-2147483647),(-2147483647-2147483647),(-2147483647-2147483647)																																																						
Write Syntax	AT\$EVENT=<event group>,<event type>,<event category>,<parm1>,<parm2>,<parm3>																																																						
Write Response	OK																																																						
Read Syntax	AT\$EVENT?																																																						
Read Response	<div>\$EVENT:&lt;evgp&gt;,&lt;evtyp&gt;,&lt;evcat&gt;,&lt;p1&gt;,&lt;p2&gt;,&lt;p3&gt;</div> <table><thead><tr><th>evgp</th><th>evtyp</th><th>evcat</th><th>p1</th><th>p2</th><th>p3</th></tr></thead><tbody><tr><td>1A</td><td>0</td><td>9</td><td>2</td><td>4</td><td></td></tr><tr><td>1B</td><td>3</td><td>37</td><td>1</td><td>0</td><td>0</td></tr><tr><td>2A</td><td>0</td><td>9</td><td>5</td><td>5</td><td></td></tr><tr><td>2B</td><td>3</td><td>21</td><td>0</td><td>0</td><td>0</td></tr><tr><td>3A</td><td>0</td><td>9</td><td>0</td><td>0</td><td></td></tr><tr><td>3B</td><td>3</td><td>13</td><td>0</td><td>0</td><td>0</td></tr><tr><td>4A</td><td>0</td><td>9</td><td>1</td><td>1</td><td></td></tr><tr><td>4B</td><td>3</td><td>21</td><td>0</td><td>0</td><td>0</td></tr></tbody></table>	evgp	evtyp	evcat	p1	p2	p3	1A	0	9	2	4		1B	3	37	1	0	0	2A	0	9	5	5		2B	3	21	0	0	0	3A	0	9	0	0		3B	3	13	0	0	0	4A	0	9	1	1		4B	3	21	0	0	0
evgp	evtyp	evcat	p1	p2	p3																																																		
1A	0	9	2	4																																																			
1B	3	37	1	0	0																																																		
2A	0	9	5	5																																																			
2B	3	21	0	0	0																																																		
3A	0	9	0	0																																																			
3B	3	13	0	0	0																																																		
4A	0	9	1	1																																																			
4B	3	21	0	0	0																																																		
Execute Syntax	N/A																																																						
Execute Response	N/A																																																						
Unsolicited Response	N/A																																																						
Parameter Values																																																							
<event group>	<p>This parameter defines the group number of a group of events and the order they are executed. Events are grouped together to control execution sequence. A group number has to have at least one input event and one output event. Multiple input events within a group number would be treated as a logical AND condition. Multiple output events within a group number would be executed individually in a sequential manner.</p> <p>Valid values for group number are: 0 thru 255.</p> <p>For additional details see the Event Tables section of this manual.</p>																																																						

AT\$EVENT	User Defined Input/Output
<event type>	<p>This parameter defines the type of event: Input or Output. An Input event can be defined as: Transition, Occurrence, or Input. The output event is executed when input event conditions are met.</p> <p>For additional details see the Event Tables section of this manual.</p>
<event category> , <parm1>, <parm2>, <parm3>	<p>These parameters define the actual Input or Output Event number and their valid range for &lt;parm1&gt; and &lt;parm2&gt;.</p> <p>The &lt;parm3&gt; parameter is only used for output events 40, 41, 42 and 52; it is ignored if entered for an input event or any other output events.</p> <p>For additional details see the Event Tables section of this manual.</p>
<b>Notes</b>	<p>A maximum of 600 events (input and output) are supported.</p> <p>User variable values are automatically saved to flash and explicit user intervention to save the values is not required. AT&amp;W does not save user variables to flash. At the end of each event engine cycle, if any user variable has been changed the user variable values are automatically saved to flash at that time (so you can make multiple changes to user variables inside event groups without worrying about writing to flash too often). Changing a user variable via \$EVTEST also causes the user variable values to be saved to flash.</p>

# AT\$EVGQRY Event Group Query

AT\$EVGQRY	Event Group Query
Command Function	Allows user to query the contents of a single event group (same output format as \$EVENT?, but limited to a single event group).
Syntax Query	AT\$EVGQRY=?
Syntax	AT\$EVGQRY: (0-255) OK
Write Syntax	AT\$EVGQRY=<event group>
Write Response	<pre> \$EVGQRY:  evgp  evtyp  evcat  p1  p2  p3            1A    0      27    1   1   0            1B    3      22    0   0   0  OK </pre>
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<event group>	<p>This parameter defines the group number of a group of events and the order they are executed. Events are grouped together to control execution sequence. A group number has to have at least one input event and one output event. Multiple input events within a group number would be treated as a logical AND condition. Multiple output events within a group number would be executed individually in a sequential manner.</p> <p>Valid values for group number are: 0 thru 255.</p>
Notes	



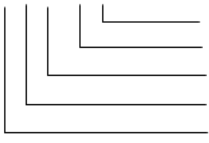
# AT\$EVNTRY Event Count Query

AT\$EVNTRY	Event Count Query
Command Function	This command queries how many events have been used and how many are left.
Syntax Query	AT\$EVNTRY=?
Syntax	ERROR
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$EVNTRY?
Read Response	\$EVNTRY: <used>,<left>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<used>	Number of events that have been used
<left>	Number of events available for new entries
Notes	

# AT\$EVTEST Generate Test Input Event

AT\$EVTEST	Generate Test Input Event
Command Function	This command allows the user to generate any input event. This is useful for testing the user event table.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$EVTEST=<event>,<state>
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<event>	input event number
<state>	input event test state
Notes	User variable values are automatically saved to flash and explicit user intervention to save the values is not required. AT&W does not save user variables to flash. At the end of each event engine cycle, if any user variable has been changed the user variable values are automatically saved to flash at that time (so you can make multiple changes to user variables inside event groups without worrying about writing to flash too often). Changing a user variable via \$EVTEST also causes the user variable values to be saved to flash.

# AT\$EVTIM# User Defined Input Event Timers

AT\$EVTIM#	User Defined Input Event Timers
Command Function	This command allows the user to define up to 10 separate periodic input events in 1 second increments. The # sign represents timers 1 through 10 (i.e. AT\$EVTIM1).
Syntax Query	AT\$EVTIM#=?
Syntax	\$EVTIM#: (0-604800) OK
Write Syntax	AT\$EVTIM#=<rate>
Write Response	OK
Read Syntax	AT\$EVTIM#?
Read Response	\$EVTIM#: <rate>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<rate>	number of seconds between each generated input event.
Notes	
Example	<p>These commands will cause the example in AT\$EVENT to trigger every 60 seconds.</p> <p>AT\$EVTIM1=60</p> <p>AT\$EVENT=1,1,12,1,1</p>  <p>Ending range of 1 (high) Starting range of 1 (high) Activate event timer 1 Input event Eventgroup 1</p>

# AT\$EVTIMQRY Event Counter

AT\$EVTIMQRY	Event Counter
Command Function	This command shows the current count for the event counter of the timer specified indicated by the argument.
Syntax Query	AT\$EVTIMQRY=?
Syntax	\$EVTIMQRY: (1-10) OK
Write Syntax	AT\$EVTIMQRY=<timer_index>
Write Response	\$EVTIMQRY:<timer_index>=<count>
	OK
Read Syntax	AT\$EVTIMQRY?
Read Response	ERROR
Execute Syntax	AT\$EVTIMQRY=10
Execute Response	\$EVTIMQRY: 10=0 OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	AT\$EVTIMQRY=x will respond with ERROR if X timer has not been set (0).

# AT\$EVTOFF Event Engine Disable

AT\$EVTOFF	Event Engine Disable
Command Function	The \$EVTOFF command gives the user the ability to disable the event engine. For example, this would be a good command to run prior to a software upgrade.
Syntax Query	AT\$EVTOFF=?
Syntax	\$EVTOFF:(0-1) OK
Write Syntax	AT\$EVTOFF=<state>
Write Response	OK
Read Syntax	AT\$EVTOFF?
Read Response	\$EVTOFF: <state>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<state>	0 = event engine enabled (default) 1 = event engine disabled
Notes	AT\$EVTOFF state is saved with AT&W
Example	N/A

# AT\$EVTQRY Query The State Or Value Of The Specified Input Event

AT\$EVTQRY	Query the State or Value of the Specified Input Event
Command Function	This command allows the user to query the state or value of the input event number
Syntax Query	AT\$EVTQRY=?
Syntax	\$EVTQRY: (0 - <max input event>) OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$EVTQRY=<input event>
Execute Response	\$EVTQRY: <input event> = <state> (bInit=<updateFlag>)
Unsolicited Response	N/A
Parameter Values	
<max input event>	Maximum input event # supported by the current firmware build (see \$EVENT for the current maximum input event value).
<input event>	Range: 0 to <max input event> Selects which input event to query
Notes	
Example:	AT\$EVTQRY=29 /* query input event 29 (Invalid GPS) \$EVTQRY: 29=1436 /* 1436 seconds of consecutive invalid GPS data OK

# AT\$STOATEV Store AT Command Events

AT\$STOATEV	Store AT Command Events
Command Function	This commands allows the user to store AT command output events. The AT command is executed upon the triggering of the associated input event.
Syntax Query	AT\$STOATEV=?
Syntax	\$\$STOATEV: (1-35),<AT command> OK
Write Syntax	AT\$STOATEV = <1-35>, <AT command>
Write Response	OK
Read Syntax	AT\$ STOATEV?
Read Response	\$STOATEV: AT Event# AT Cnds 1 * 2 * ... 35 * OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	

<b>AT\$STOATEV</b>	<b>Store AT Command Events</b>
<b>&lt;1-35 &gt;</b>	AT event index.
<b>&lt;AT command&gt;</b>	<p>AT command associated with the AT event index. The AT command is not checked for validity.</p> <p>This parameter is limited to a maximum of 141 characters.</p>
<b>Notes</b>	<p>This command is used in conjunction with the Dynamic Input Output event (AT\$EVENT). The output event associated with this command is event 44. When output event 44 is defined in the event table, Parm1 defines which index to refer to.</p> <p>The AT command associated with the index is executed. The use of Dynamic Event Scripting using AT\$EVENT or AT\$EVDEL as a stored AT Command Event can lead to unpredictable operation and is not recommended. When storing command to dial a voice call, a “v” replaces the “;” at the end of the dial string..(i.e., atd17195551212v)</p> <p>* Only populated event index lines are displayed. Note that this behavior is a change from the operation of this command on previous Novatel Wireless devices.</p>



# AT\$USRVAL User Value

AT\$USRVAL	User Value
Command Function	Allows the user to store a value in flash memory which can later be retrieved.
Syntax Query	AT\$USRVAL=?
Syntax	\$USRVAL: (0-FFFFFFFF) OK
Write Syntax	AT\$USRVAL=<hex value>
Write Response	OK
Read Syntax	AT\$USRVAL?
Read Response	\$USRVAL:(hex value) OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<hexval>	(0-FFFFFFFF)
Notes:	N/A

# RTC Commands

---

# AT\$RTCALRM Real Time Clock Alarm

AT\$RTCALRM	Real Time Clock Alarm
Command Function	This command handles the setting and querying of the RTC alarm registers. When the alarm feature has been enabled the \$EVENT engine will be invoked upon the going off. If the \$RTCWAKE call is invoked following the alarm feature setup the modem will power back up automatically upon the alarm going off. The actions of these two features are mutually exclusive of each other, so one or the other will occur but not both.
Syntax Query	AT\$RTCALRM=?
Syntax	\$RTCALRM: (0-99),(1-12),(1-31),(0-23),(0-59),(0-59),(0-527040) OK
Write Syntax	AT\$RTCALRM= <rtc_year>,<rtc_month>,<rtc_day>,<rtc_hour>,<rtc_min>,<rtc_sec>,<rtc_alarmTimeinMinutes>
Write Response	OK
Read Syntax	AT\$RTCALRM?
Read Response	\$RTCALRM: <rtc_enabled>,<rtc_year>,<rtc_month>,<rtc_day>,<rtc_hour>,<rtc_min>,<rtc_sec>,<rtc_alarmTimeinMinutes>” OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	Parameters are positional dependent, any parameter may be omitted with the use of the comma (’,’) as a place holder on command line. If a parameter is omitted then the current value in the hardware is used.
< rtc_enabled >	Indicates if alarm is enabled or not. 1= Enabled 0= Disabled
< rtc_year >	The year on which the alarm is being set to trigger. The RTC supports years 2000-2099. The data is entered as a two digit value 0..99.
<rtc_month>	The month on which the alarm is being set to trigger. Values range from 1..12.

<b>AT\$RTCALRM</b>	<b>Real Time Clock Alarm</b>
<b>&lt;rtc_day&gt;</b>	The day on which the alarm is being set to trigger. Values range from 1..31.
<b>&lt;rtc_hour&gt;</b>	The hour on which the alarm is being set to trigger. Values range from 0..24 for 24-Hour mode settings. NOTE: only 24-Hour mode currently supported.
<b>&lt;rtc_min&gt;</b>	The minute on which the alarm is being set to trigger. Values range from 0..59.
<b>&lt;rtc_sec&gt;</b>	The second on which the alarm is being set to trigger. Values range from 0..59.
<b>&lt;rtc_alarmTimeinMinutes&gt;</b>	Periodic Alarm time in minutes. RTC Alarm will be reset at a period specified by this parameter.
<b>Notes</b>	<p>This command is used to set the Alarm time for the RTC. Currently all time is based on 24-Hour time format. The alarm may be cleared using the command AT\$RTCCLRA. This call in conjunction with the use of either the \$EVENT engine or the \$RTCWAKE command the user has a rich feature set of driving other events or waking the system up at a pre-determined time in the future. No checks are made for alarm time not being later than current time.</p> <p>AT\$RTCALRM will not trigger if the alarm time occurs while the unit is resetting.</p>
<b>Examples</b>	<p>Following sets and alarm for 2003, October, 13th at 17:00 Hours at\$rtcalrm=3,10,13,17,0,0 OK</p> <p>Following queries the alarm for current time, and shows that the alarm being; Enabled, for 2003, October 13th at 17:00 hours. at\$rtcalrm? \$RTCALRM: 01, 03, 10, 13, 17, 00, 00 OK</p> <p>Following call unsets alarm followed by displaying alarm time information. at\$rtcclra OK at\$rtcalrm? \$RTCALRM: 00, 03, 10, 13, 17, 00, 00 OK</p>

# AT\$RTCCLRA Real Time Clock Clear Alarm

AT\$RTCCLRA	Real Time Clock Clear Alarm
Command Function	This command allows the modem to clear/disable the active RTC alarm. The alarm interrupt enable is cleared but alarm time not altered.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$RTCCLRA
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	None
Notes	This command will disable the RTC alarm while leaving the value of the last alarm time setting alone.

# AT\$RTCRSET RTC Report Reset State

AT\$RTCRSET	RTC Report Reset State
Command Function	This command reports the reset state of the RTC following a power cycle. The command reports TRUE only if a reset occurred since last power up and last call to check it. So multiple calls will report the current status only.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$RTCRSET?
Execute Response	\$RTCRSET : <reset state> OK
Unsolicited Response	N/A
Parameter Values	

AT\$RTCRSET	RTC Report Reset State
<reset state>	1 - indicates that a RTC reset occurred, 0 - indicates that a RTC reset did NOT occur
Notes	This command returns the current reset state of the RTC since power-up. If multiple calls are made only current reset state is returned.
Examples	<p>Following example shows the check for the RTC being reset since last check of reset and since last power up, with a response of True.</p> <pre>AT\$rtcrset? \$RTCRSET: 1 OK</pre> <p>Following example shows the check for the RTC being reset since last check of reset and since last power up, with a response of False.</p> <pre>AT\$rtcrset? \$RTCRSET: 0 OK</pre>

# AT\$RTCTIME Real Time Clock Time

AT\$RTCTIME	Real Time Clock Time
Command Function	This command handles the setting and querying of the RTC time registers.
Syntax Query	AT\$RTCTIME=?
Syntax	\$RTCTIME: (0-6),(0-99),(1-12),(1-31),(0-23),(0-59),(0-59) OK
Write Syntax	AT\$RTCTIME= <rtc_wkday>,<rtc_year>,<rtc_month>,<rtc_day>,<rtc_hour>,<rtc_min>,<rtc_sec>
Write Response	OK
Read Syntax	AT\$RTCTIME?
Read Response	\$RTCTIME: <rtc_wkday>, <rtc_year>, <rtc_month>, <rtc_day>,<rtc_hour>, <rtc_min>, <rtc_sec>” OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	Parameters are positional dependent, any parameter may be omitted with the use of the comma (’,’) as a place holder on command line. If a parameter is omitted then the current value in the hardware is used.
< rtc_wkday >	Current week day matching time day being set. The week day values range from 0-6, where; 0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday
< rtc_year >	The year to which the time is being set. The RTC supports years



AT\$RTCTIME	Real Time Clock Time
	2000-2099. The data is entered as a two digit value 0-99.
<rtc_month>	The month to which the time is being set. Values range from 1-12.
<rtc_day>	The day to which the time is being set. Values range from 1-31.
<rtc_hour>	The hour to which the time is being set. Values range from 0-24 for 24-Hour mode settings. NOTE: only 24-Hour mode currently supported.
<rtc_min>	The minute to which the time is being set. Values range from 0-59.
<rtc_sec>	The second to which the time is being set. Values range from 0-59.
Notes	This command is used to set the time for the RTC. Currently all time is based on 24-Hour time format.
Examples	at\$rtctime? \$RTCTIME: 01, 03, 10, 13, 14, 03, 2 OK  at\$rtctime=? \$RTCTIME: (0..6), (0..99), (1..12), (1..31), (0..23), (0..59), (0..59)  at\$rtctime=1,3,10,13,14,37,50 OK

# AT\$RTCWAKE Real Time Alarm Wake

AT\$RTCWAKE	Real Time Alarm Wake
Command Function	This command attempts to de-register from the network, at the end of a 5 second delay then powers down the modem so only the RTC is running. Upon the RTC alarm going off the modem will re-boot and initialize again. The command relies on the RTC Alarm feature being set prior, if system is to wake up at a preset time in the future.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$RTCWAKE
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
Notes	The AT\$RTCWAKE command powers down the modem, so only the RTC clock will be running. The modem will power up automatically only if the RTC Alarm feature has been, otherwise modem will remain powered off. See the AT\$RTCALRM command for setup of the RTC alarm.

# FOTA Commands

---

# AT\$FOTACFG Configure Firmware Upgrade

AT\$FOTACFG	Configure Firmware Upgrade
Command Function	This command allows the user to configure the modem Firmware Over The Air (FOTA) Server.
Syntax Query	AT\$FOTACFG=?
Syntax	\$FOTACFG: "host", (0-65535), "username", "password", (0,1), (0-20), (0,1), (0,1), (0,1) OK
Write Syntax	AT\$FOTACFG=<"ftpServerHostname">,<ftpPort>,<"ftpUsername">,<"ftpPassword">,<ftpMode>,<retries>,<reportMode>,<upgradeMode>,<upgradeStatus>
Write Response	OK
Read Syntax	AT\$FOTACFG?
Read Response	\$FOTACFG: <"ftpServerHostname">,<ftpPort>,<"ftpUsername">,<"ftpPassword">,<ftpMode>,<retries>,<reportMode>,<upgradeMode>,<upgradeStatus>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ftpServerHostname>	Either Fully Qualified Domain Name (FQDN) or a dotted-decimal IP address of the FTP server. The FQDN is resolved by DNS immediately prior to opening IP connection to FTP server. The FQDN may be up to a maximum of 63 characters. Default value: none.
<ftpPort>	The TCP port number on the FTP server. Port range: 0 - 65535 Default value: 21
<ftpUsername>	FTP login name on the FTP server. The username may be up to a maximum of 31 characters.

AT\$FOTACFG	Configure Firmware Upgrade
	Default value: none.
<ftpPassword>	<p>The FTP password for the FTP user. The password may be an empty string if the FTP user does not have a password. The password may be up to a maximum of 31 characters.</p> <p>Default value: none.</p>
<ftpMode>	<p>The FTP mode to use for data connections. FTP data connections may be active or passive. Passive mode is generally better to use with FTP servers located behind firewalls.</p> <p>Values:</p> <p>0 = active</p> <p>1 = passive</p> <p>Default value: 1 (passive mode).</p>
<retries>	<p>This is the maximum number of FTP file resumes to attempt before giving up on original AT\$FOTAGET command.</p> <p>The range of retries is 0 - 20.</p> <p>Default Value: 0.</p>
<reportMode>	<p>The reporting mode is used to control the level of status reporting during the FOTA process. Normal reporting mode sends only critical status updates, extended reporting mode sends informative and critical status updates.</p> <p>The command parameters are:</p> <p>0 = normal reporting</p> <p>1 = extended reporting</p> <p>Default value: 0 (normal reporting mode).</p>

AT\$FOTACFG	Configure Firmware Upgrade
<upgradeMode>	<p>The system behavior following successful FOTA file transfer to modem. The upgrade Mode may be either Automatic or Manual. Auto mode will immediately initiate FOTA upgrade when file transfer completes. Manual mode will wait after file transfer until an explicit upgrade command is issued (AT\$FOTAUPG).</p> <p>The command line parameters are:</p> <p>0 = auto</p> <p>1 = manual</p> <p>Default value: 0 (automatic mode).</p>
<upgradeStatus>	<p>Controls the generation of FOTA Finish status. FOTA Finish status is reported after the modem reboots from a FOTA upgrade or as a result of a fatal error. This status is only returned OTA.</p> <p>The command line parameters are:</p> <p>0 = disabled</p> <p>1 = enabled</p> <p>FOTA finish status is disabled by default.</p>
Notes	Refer to Novatel Wireless Application Note ENF000AN002 for more information on FOTA operations

# AT\$FOTAGET Get Firmware Upgrade OTA

AT\$FOTAGET	Get Firmware Upgrade OTA
Command Function	This command will initiate a connection to the FOTA Server to download the FOTA Delta File. The FOTA Server is defined with the \$FOTACFG command.
Syntax Query	AT\$FOTAGET=?
Syntax	\$FOTAGET: "remotefilename" OK
Write Syntax	AT\$FOTAGET=<"remotefilename">
Write Response	OK
Read Syntax	AT\$FOTAGET?
Read Response	\$FOTAGET: <fotaState>,<"remotefilename">,<retriesRemaining>,<appErrorCode>,<errorCode>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<fotaState>	Current FOTA mode indicating if FOTA is in the process of transferring a FOTA file.  0 = IDLE 1 = BUSY
<"remotefilename">	The upgrade package file name.

AT\$FOTAGET	Get Firmware Upgrade OTA
<retriesRemaining>	The number of FTP resumes remaining before terminating file transfer. This number starts with the retries specified in the AT\$FOTACFG command and is decremented for each FOTA restart which occurs.
<appErrorCode>	<p>Status of the last FOTA result.</p> <p>0 = success</p> <p>1 = failed to access package data</p> <p>2 = flash erase failed</p> <p>3 = flash write of package failed</p> <p>6 = package version or signature does not match</p> <p>7 = package version or signature does not match</p> <p>8 = could not install package in flash</p> <p>9 = previous version present and pkg Status pending</p> <p>10 = not enough space to write file</p> <p>11 = flash file open failed</p> <p>12 = flash file read failed or did not read requested size</p> <p>13 = flash file write failed</p> <p>14 = fota file larger than max allowed</p> <p>15 = flash file status failed</p> <p>16 = flash file seek failed</p> <p>17 = flash file close failed</p> <p>18 = failed to get flash sector address</p> <p>19 = flash file erase failed</p> <p>20 = checksum verification failed</p> <p>21 = failed to set update flag</p>
Notes	Refer to Novatel Wireless Application Note ENF000AN002 for more information on FOTA operations



# AT\$FOTAUPG Upgrade Current Firmware

AT\$FOTAUPG	Upgrade Current Firmware
Command Function	<p>This manual command is used to initiate a FOTA upgrade after FOTA file has been successfully transferred to the modem.</p> <p>This command should only be used if the manual mode is selected by AT\$FOTACFG or if the FOTA delta file is transferred to the modem manually using FTP or some other method.</p> <p>AT\$FOTAUPG should not be used if automatic mode is configured in AT\$FOTACFG because the upgrade process is started by the modem in automatic mode.</p> <p>AT\$FOTAUPG resets the modem and starts the FOTA upgrade agent process.</p>
Syntax Query	AT\$FOTAUPG=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$FOTAUPG
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	<p>Refer to Novatel Wireless Application Note ENF000AN002 for more information on FOTA operations.</p> <p>Refer to AT\$FOTACFG for automatic versus manual upgrade mode description.</p>
Example	N/A

# FTP Commands

---

# AT\$ATEXEC Execute AT Command File

<b>AT\$ATEXEC</b>	<b>Execute AT Command File</b>
<b>Command Function</b>	This command is used to execute an AT command file that is stored in FFS.
<b>Syntax Query</b>	AT\$ATEXEC =?
<b>Response</b>	\$ATEXEC: "filename", (0-2) OK
<b>Write Syntax</b>	AT\$ATEXEC =<"FILENAME ">,<OPTION>
<b>Response</b>	OK
<b>Read Syntax</b>	AT\$ATEXEC?
<b>Response</b>	\$ATEXEC: <"FILENAME">,<STATUS> OK
<b>Execute Syntax</b>	N/A
<b>Response</b>	N/A
<b>Unsolicited Response</b>	<p>The modem returns two responses after the command to run the script has been issued. The first response is an immediate acknowledgement of the command, either OK (if the command was entered correctly) or ERROR (if the command was not entered correctly). After the script has completed, the modem returns an Unsolicited Result Code (URC) indicating either success or the reason for the error.</p> <p>The following URCs may be returned:</p> <p>\$ATEXEC OK</p> <p>\$ATEXEC FAIL</p> <p>\$ATEXEC FILE OPEN ERROR</p> <p>\$ATEXEC FILE READ ERROR</p> <p>\$ATEXEC AT CMD ERROR, Line n (Where n is the line number of the first error in the script file)</p> <p>\$ATEXEC BUSY ERROR</p>
<b>Parameter Values</b>	
<b>&lt;"FILENAME"&gt;</b>	<p>Name of file to execute</p> <p>Maximum length is 15 characters.</p>

AT\$ATEXEC	Execute AT Command File
<OPTION>	AT command execution option 0 = Run To Completion 1 = Stop on Error 2 = Stop on Error and Reset Modem
<STATUS>	Status from Executing File 0 = Ok 1 = Error 2 = Open File Error 3 = File Read Error 4 = AT Command Error 5 = Busy 6 = Command Length Exceeded 7 = Retries Exceeded
Notes	AT\$FTPGET or AT\$FFS can be used to store files into the FFS.

# AT\$FFS Generic Flash File System Access

AT\$FFS	Generic Flash File System Access
Command Function	This command provides generic file read and write functionality to the embedded FFS in the Novatel Wireless modem.
Syntax Query	AT\$FFS=?
Response	\$FFS: (0-6,86),"filename",(0-524287),(0-100),("ASCII-HEX")
Write Syntax	AT\$FFS=<op>,<filename>,<seek>,<length>, <data>
Response	<p>The device response varies by &lt;op&gt;</p> <p><b>Response for Write (&lt;op&gt; = 0):</b> OK</p> <p><b>Response for Read (&lt;op&gt; = 1):</b> \$FFS: &lt;data&gt; OK</p> <p><b>Response for Delete (&lt;op&gt; = 2):</b> OK</p> <p><b>Response for Directory (&lt;op&gt; = 3):</b> \$FFS [&lt;filename&gt;] [&lt;filename&gt;] [...] OK</p> <p><b>Response for Directory and size (&lt;op&gt; = 4):</b> \$FFS: [&lt;file size&gt; &lt;filename&gt;] [&lt;file size&gt; &lt;filename&gt;]</p>

AT\$FFS	Generic Flash File System Access
	<p>[...]</p> <p><b>Response for Free and used space (&lt;op&gt; = 5):</b>  \$FFS: &lt;free_bytes&gt; Free Bytes, &lt;used_bytes&gt; Used Bytes</p> <p><b>Response for Delete All (&lt;op&gt; = 86):</b>  OK</p>
Read Syntax	N/A
Response	N/A
Execute Syntax	N/A
Response	N/A
Parameter Values	
<op>	FFS Operation 0 = write 1 = read 2 = delete 3 = directory 4 = directory with file size 5 = FFS free space and used space 6 = Output flash device name 86 = delete all files
<filename>	The filename in double quotes (i.e. "test.txt") Maximum length is 15 characters. Filename parameter is required for the write (0), read (1), and delete (2) operations.
<seek>	The position in the file to read or write given as a byte offset. Seek parameter is required for the write (0) and read (1), operations. Range: 0 - 524287.
<length>	The number of bytes to read or write.

AT\$FFS	Generic Flash File System Access
	Length parameter is only required for read and write operations. Range: 0 - 100
<data>	Only applicable to write operations as an input parameter. Also defines the output data format for the read operation. Limited to 100 bytes (200 characters). Data is encoded in ASCII hex (i.e. 0x12, 0x34, 0xab would be entered as "1234ab")
Notes	
Examples	<p>AT\$FFS=0,"test",0,8,"0102030405060708"</p> <p>OK</p> <p>AT\$FFS=1,"test",0,8</p> <p>\$FFS:</p> <p>0102030405060708</p> <p>OK</p> <p>AT\$FFS=3</p> <p>\$FFS:</p> <p>test</p> <p>OK</p> <p>AT\$FFS=4</p> <p>\$FFS:</p> <p>8 test</p> <p>OK</p> <p>AT\$FFS=5</p> <p>\$FFS: 675308 Free Bytes, 76324 Used Bytes</p> <p>OK</p> <p>AT\$FFS=2,"test"</p> <p>OK</p>

AT\$FFS	Generic Flash File System Access
	AT\$FFS=86 OK



# AT\$FTPABORT Aborts Current Data Operation With FTP Server

AT\$FTPABORT	Aborts Current Data Operation with FTP Server
Command Function	Interrupts the current data operation with the FTP server.
Syntax Query	AT\$FTPABORT=?
Response	OK
Write Syntax	N/A
Response	N/A
Read Syntax	AT\$FTPABORT?
Response	ERROR
Execute Syntax	AT\$FTPABORT
Response	OK
Parameter Values	N/A
Notes	If a data operation is not in operation, no action is performed. If a connection to an FTP server is not active, FTP INVALID CLIENT is returned.

# AT\$FTPBLK Change Current FTP BLock Size

AT\$FTPBLK	Change Current FTP Block Size
Command Function	Changes the current FTP block size
Syntax Query	AT\$FTPBLK=?
Response	\$FTPBLK: (128-1500) OK
Write Syntax	AT\$FTPBLK=<blocksize>
Response	OK
Read Syntax	AT\$FTPBLK?
Response	1500
Execute Syntax	N/A
Response	N/A
Parameter Values	
<blocksize>	FTP block size (128-1500)
Notes	The default blocksize is 1500. AT\$FTPBLK should be set prior to the AT\$FTPOPEN for the blocksize to have effect.

# AT\$FTPCFG Configure FTP Parameters

AT\$FTPCFG	Configure FTP parameters
Command Function	Configures FTP parameters required to open connection to FTP Server
Syntax Query	AT\$FTPCFG=?
Syntax	AT\$FTPCFG: "host", (0-65535), "username","password" OK
Write Syntax	AT\$FTPCFG=<host>,<portnum>,<username>,<password>
Write Response	OK
Read Syntax	AT\$FTPCFG?
Read Response	AT\$FTPCFG: <host>, <port>,<username>,<password>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<host>	Fully qualified domain name or dotted-decimal notation IP address of FTP server, character string up to 63 characters
<portnum>	FTP server port number, 0 – 65535
<username>	FTP server user name, character string up to 31 characters
<password>	FTP server password, character string up to 31 characters
Notes	AT\$FTPCFG must be entered initially before opening connection to FTP server. Configuration information will be saved to flash when flash write command is executed and will be re-used on subsequent FTP connections.

# AT\$FTPCHDIR Change Current Working Directory

<b>AT\$FTPCHDIR</b>	<b>Change current working directory</b>
<b>Command Function</b>	Changes the current working directory on the FTP server.
<b>Syntax Query</b>	AT\$FTPCHDIR=?
<b>Syntax</b>	\$FTPDIR: "directory" OK
<b>Write Syntax</b>	AT\$FTPCHDIR=<directory>
<b>Write Response</b>	OK
<b>Read Syntax</b>	AT\$FTPCHDIR?
<b>Read Response</b>	ERROR
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;directory&gt;</b>	Directory name on remote FTP server
<b>Notes</b>	AT\$FTPCHDIR can only be entered after the connection to the FTP server has been established. To navigate up a directory, use the ".." directory notation.

# AT\$FTPCLOSE Closes The Connection To The FTP Server

AT\$FTPCLOSE	Closes the Connection to FTP Server
Command Function	Closes the current connection with the remote FTP server.
Syntax Query	AT\$FTPCLOSE=?
Response	OK
Write Syntax	N/A
Response	N/A
Read Syntax	AT\$FTPCLOSE?
Response	ERROR
Execute Syntax	AT\$FTPCLOSE
Response	OK
Parameter Values	N/A
Notes	This command is only available when a connection to the FTP server is active. If a connection to an FTP server is not active, FTP INVALID CLIENT is returned.

# AT\$FTPDIR Directory Listing

AT\$FTPDIR	Directory Listing
Command Function	Prints a directory listing of the current working directory on the FTP server
Syntax Query	AT\$FTPDIR=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$FTPDIR?
Read Response	OK \$FTPDIR: <current directory information>
Execute Syntax	AT\$FTPDIR
Execute Response	OK <directory listing information>
Unsolicited Response	N/A
Parameter Values	N/A
Notes	AT\$FTPDIR can only be entered after the connection to the FTP server has been established. Directory listing information may be delayed due to the latency of the data connection to the FTP server.

# AT\$FTPGET Retrieve A Remote File

AT\$FTPGET	Retrieve a Remote File
Command Function	Retrieves a remote file from the FTP server for either output on the serial port or storage to the local flash file system.
Syntax Query	AT\$FTPGET=?
Syntax	\$FTPGET: "remotefile","localfile" OK
Write Syntax	AT\$FTPGET=<remotefile>,<localfile>
Write Response	OK
Read Syntax	AT\$FTPGET?
Read Response	\$FTPGET: <remotefile>,(SERIAL, <localfile>),<output>/<available>,<eof>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<remotefile>	File name on the remote FTP server.
<localfile>	File name to use for storage of the retrieved file in the local flash file system. Note: the local filename length is limited to 15 characters.
<output>	Number of bytes which have been transferred out the serial port
<available>	Number of bytes of file which are available for output to the serial port
<eof>	0 = file currently being transferred 1 = complete file has been transferred to FTP client
Notes	If local file is not specified, the remote file will be maintained in memory buffers until a subsequent FTP read request is performed to output a block of bytes onto the serial port. As the file is transmitted out the serial port, the FTP client will refill the memory buffers with additional file data from the FTP server.

# AT\$FTPOPEN Opens FTP Connection

<b>AT\$FTPOPEN</b>	<b>Opens FTP Connection</b>
<b>Command Function</b>	Opens FTP control connection to FTP server
<b>Syntax Query</b>	AT\$FTPOPEN=?
<b>Syntax</b>	\$FTPOPEN: ("ACTIVE,"PASSIVE") OK
<b>Write Syntax</b>	N/A
<b>Write Response</b>	N/A
<b>Read Syntax</b>	AT\$FTPOPEN?
<b>Read Response</b>	OK \$FTPOPEN: <status>
<b>Execute Syntax</b>	AT\$FTPOPEN="<mode>"
<b>Execute Response</b>	OK <status>
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;mode&gt;</b>	ACTIVE or PASSIVE Mode is an optional parameter. If not specified, the FTP connection will be in the PASSIVE mode.
<b>&lt;status&gt;</b>	FTP CONNECTION OPEN FTP CONNECTION CLOSED FTP STACK ERROR Status will display the state of the control connection with the remote FTP server.
<b>Notes</b>	AT\$FTPOPEN must be entered after FTP configuration information has been entered. If any FTP configuration information has been saved to flash, it will be used to open the connection to the FTP server. After the FTP open completes successfully, FTP CONNECTION OPEN will be printed, otherwise an error string will be printed.



# AT\$FTPPUT Store A Local File On The FTP Server

AT\$FTPPUT	Store a Local File On the FTP Server
Command Function	Stores a local file from the Flash File System onto the remote FTP server.
Syntax Query	AT\$FTPPUT=?
Response	\$FTPPUT:"remotefile","localfile" OK
Write Syntax	AT\$FTPPUT=<remotefile>,<localfile>
Response	OK
Read Syntax	AT\$FTPPUT?
Response	\$FTPPUT:<remotefile>,<localfile>,<totalWrote>/<totalSize>,<eof>
Execute Syntax	N/A
Response	N/A
Parameter Values	
<remotefile>	File name on the remote FTP server.
<localfile>	File name on the local Flash File System.
<totalWrote>	Number of bytes of file which have been written to FTP Server
<totalSize>	Size of file to be transferred to FTP Server (in bytes).
<eof>	0 = file is currently being transferred 1 = complete file has been transferred to FTP Server
Notes	

# AT\$FTPRST Restart The Last FTP GET Operation

AT\$FTPRST	Restart the Last FTP GET Operation
Command Function	Manual restart of an FTP GET file transfer from the last completed point in the file.
Syntax Query	N/A
Response	N/A
Write Syntax	AT\$FTPRST=<offset>
Response	OK
Read Syntax	N/A
Response	N/A
Execute Syntax	N/A
Response	N/A
Parameter Values	
<offset>	Number of bytes into the file to resume file transfer. (Optional)
Notes	<p>AT\$FTPRST will attempt to resume the last incomplete file transfer from the last point in the file successfully read. Offset is not required unless the automatically computed offset is to be overridden. The AT\$FTPRST command requires an RFC3659 compliant FTP server which implements the FTP REST command. AT\$FTPRST is not designed to work over a system restart.</p> <p>The user's current path must contain the file prior to issuing AT\$FTPRST or else the AT\$FTPRST command will fail.</p> <p>The only time it is necessary to send the \$FTPRST command is if a "FTP TIMEOUT" URC is received during a FTP transfer. The "FTP TIMEOUT" is typically caused by the loss of the network data connection for more than the FTP idle timeout, 60 seconds.</p>

# IP Router Commands

---

# AT\$FLFILT Filter ICMP Echo Request

AT\$FLFILT	Filter ICMP Echo Request
Command Function	This command allows the user to filter the ICMP echo request based against the Friends list
Syntax Query	AT\$FLFILT=?
Syntax	\$FLFILT: (0-1) OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$FLFILT?
Read Response	\$FLFILT: <enable> OK
Execute Syntax	AT\$FLFILT=<enable>
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	
<enable>	0 = do not filter ICMP echo requests 1 = filter ICMP echo requests against Friends List ** Default value is 1
Notes	Friends list entries are used without regard to the setting of the server bit.  This function will also filter ICMP Echo requests when in Dial-Up Networking Mode (\$HOSTIF=0)

# AT\$NETIP Display Network Assigned IP & DNS

<b>AT\$NETIP</b>	<b>Display Network Assigned IP &amp; DNS</b>
<b>Command Function</b>	This command allows the user to query the modem's network assigned IP.
<b>Syntax Query</b>	N/A
<b>Syntax</b>	N/A
<b>Write Syntax</b>	N/A
<b>Write Response</b>	N/A
<b>Read Syntax</b>	AT\$NETIP?
<b>Read Response</b>	<"IP">,<"DNS1">,<"DNS2">
<b>Execute Syntax</b>	N/A
<b>Execute Response</b>	N/A
<b>Unsolicited Response</b>	N/A
<b>Parameter Values</b>	
<b>&lt;IP&gt;</b>	network assigned IP
<b>&lt;DNS1&gt;</b>	network assigned DNS1
<b>&lt;DNS2&gt;</b>	network assigned DNS2
<b>Notes</b>	N/A

# AT\$PING Send ICMP Echo Request

AT\$PING	Send ICMP Echo Request
Command Function	This command allows the user to send an ICMP echo request and display the echo reply (ping).
Syntax Query	AT\$PING=?
Syntax	\$PING: "host",(1-5),(1-10) OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$PING=<"host">,<count>,<timeout>
Execute Response	Immediate response: "Pinging <IP address>"  After ICMP response: "Reply from <IP address>"
Unsolicited Response	N/A
Parameter Values	
<"host">	FQDN or IP address
<count>	Number of ICMP frames to send Range: 1-5 Default: 1
<timeout>	Maximum amount of time, in seconds, to wait for a response. Range: 1-10 Default: 5
Notes	The modem must have an IP address assigned to it by the network to send or receive ICMP packets.

# GPS Commands

---

# AT\$ALTOSI Set And Query GPS Overspeed Interval

AT\$ALTOSI	Set and Query GPS Overspeed Interval
Command Function	This command allows the user to define the criteria for a GPS overspeed event. A GPS overspeed event occurs when the minimum speed that is defined by the <speed> parameter is maintained for a specific duration of time.
Syntax Query	AT\$ALTOSI=?
Syntax	\$ALTOSI: (0 - 255),(0-65535),(0-1) OK
Write Syntax	AT\$ALTOSI=<speed>,<interval>,<sticky>
Write Response	OK
Read Syntax	AT\$ALTOSI?
Read Response	\$ALTOSI: <speed>,<interval>,<status>,<max_speed>,<duration> OK
Execute Syntax	AT\$ALTOSI
Execute Response	ERROR
Unsolicited Response	N/A
Parameter Values	
<speed>	Speed, in knots, must be met and/or exceeded to trigger the GPS overspeed event.
<interval>	Number of consecutive seconds for which <speed> must be maintained to trigger the GPS overspeed event.
<sticky>	Optional parameter to make overspeed parameters persistent. If 1, then speed/interval are written to flash immediately (no AT&W required). Default = 0.
<status>	If 1, then <max_speed> and <duration> represent a GPS overspeed interval that is currently active. If 0, they represent the previous GPS overspeed interval.



<b>AT\$ALTOSI</b>	<b>Set and Query GPS Overspeed Interval</b>
<b>&lt;max_speed&gt;</b>	The highest speed (in knots) that was attained in the current or previous GPS overspeed interval.
<b>&lt;duration&gt;</b>	Number of consecutive seconds that the speed was at or above <speed>.
<b>Notes</b>	If <speed> is set to zero, the GPS overspeed event is disabled.

# AT\$EXCACC Excessive Acceleration

AT\$EXCACC	Excessive Acceleration
Command Function	This command is used to configure the excessive acceleration event. Input event 198
Syntax Query	AT\$EXCACC=?
Syntax	\$EXCACC: (0-200),(1-20),(1-20) OK
Write Syntax	AT\$EXCACC=<THRESH>,<SET TIME >,<CLR TIME >
Write Response	OK
Read Syntax	AT\$EXCACC?
Read Response	\$EXCACC: <THRESH>,<SET TIME >,<CLR TIME> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<THRESH>	0 = Excessive Acceleration Event Disabled 1-200 = Acceleration Threshold (in Knots * 10)
<SET TIME>	1-20 = Number of seconds above threshold to set event
<CLR TIME>	1-20 = Number of seconds below threshold to clear event
Notes	A value of 100 for threshold equals 10 knots per second. This function uses GPS to determine acceleration.

# AT\$EXCDEC Excessive Deceleration

AT\$EXCDEC	Excessive Deceleration
Command Function	This command is used to configure the excessive deceleration event. Input event 199.
Syntax Query	AT\$EXCDEC=?
Syntax	\$EXCDEC: (0-200),(1-20),(1-20) OK
Write Syntax	AT\$EXCDEC=<THRESH>,<SET TIME >,<CLR TIME >
Write Response	OK
Read Syntax	AT\$EXCDEC?
Read Response	\$EXCDEC: <THRESH>,<SET TIME >,<CLR TIME> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<THRESH>	0 – Excessive Deceleration Event Disabled 1-200 – Deceleration Threshold (in Knots * 10)
<SET TIME>	1-20 – Number of seconds above threshold to set event
<CLR TIME>	1-20 – Number of seconds below threshold to clear event
Notes	A value of 100 for threshold equals 10 knots per second. This function uses GPS to determine deceleration.

# AT\$GEOFNC Geo Fencing A Circular Area

AT\$GEOFNC	Geo Fencing a Circular Area
Command Function	This command allows a user to send a GPS message when the device moves in or out of a geographical area.
Syntax Query	AT\$GEOFNC=?
Syntax	\$GEOFNC: (1-25),(0-1000000),(-90.0 - +90.0),(-180.0 - +180.0) OK
Write Syntax	AT\$GEOFNC=<fenceNum> <radius>,<latitude>,<longitude>
Write Response	OK
Read Syntax	AT\$GEOFNC?
Read Response	\$GEOFNC: <fenceNum>,<radius>,<latitude>,<longitude> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<fenceNum>	Defines the fence number
<radius>	Defines radius of the circle from given Latitude and Longitude coordinates (in meters)
<latitude>	Defines the latitude for the center point of a circle
<longitude>	Defines the longitude for the center point of a circle
Notes	<p>An AT\$EVENT command has to be set to send a GPS message to the remote host when entering or exiting the fenced area.</p> <p>Although this command accepts latitude/longitude parameters with up to 15 characters, internally the value is stored as a C float type which has less precision (but requires half the storage size). The float type is capable of storing accuracy commensurate with the GPS receiver's capability, but the queried latitude/longitude values of the AT\$GEOFNC command may differ from the input parameters due to this precision limitation.</p>

# AT\$GFDBNC Set Geofence Debounce Count

AT\$GFDBNC	Set Geofence Debounce Count
Command Function	This command allows the user to set the number of consecutive geofence positions required to trigger an 'inside geofence' or 'outside geofence' event. It prevents a false reading when the device is on the threshold of the geofence.
Syntax Query	AT\$GFDBNC=?
Syntax	\$GFDBNC: (0-250),(0-250) OK
Write Syntax	AT\$GFDBNC=<out_cnt>,<in_cnt>
Write Response	OK
Read Syntax	AT\$GFDBNC?
Read Response	\$GFDBNC: <out_cnt>, <in_cnt> OK
Execute Syntax	N/A
Execute Response	
Unsolicited Response	N/A
Parameter Values	
<out_cnt>	Consecutive GPS position reports outside a geofence required to trigger 'O' condition for geofence input event (see \$EVENT)
<in_cnt>	Consecutive GPS position reports inside a geofence required to trigger 'I' condition for geofence input event (see \$EVENT)
Notes	The GPS reporting interval varies depending on the product. for the MT-Gx and MTxxxx products, the updates are sent every one second.

# AT\$GFDEL Delete A Range Of Geo-Fences

AT\$GFDEL	Delete a Range of Geo-Fences
Command Function	This command deletes a range of geo-fences.
Syntax Query	AT\$GFDEL=?
Syntax	\$GFDEL: (1-25),(1-25) OK
Write Syntax	AT\$GFDEL=<start>,<stop>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<start>	First geo-fence index in range to be deleted
<stop>	Last geo-fence index in range to be deleted.
Notes	N/A

# AT\$GFIDX# Geo Fencing Index

AT\$GFIDX#	Geo Fencing Index
Command Function	This command allows the user to query the modem for a single geofence from storage. The modem returns the index, radius, longitude and latitude. Longitude and latitude data is formatted for use in the GPS. Replace the # with the index number noted below.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$GFIDX<index>?
Read Response	
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	NA
Parameter Values	
<index>	(1-25) - Index to the geofence stored in the modem
Notes	N/A

# AT\$GPSCMD GPS Command

AT\$GPSCMD	GPS Command
Command Function	This command allows the user to configure the state of the GPS module.
Syntax Query	AT\$GPSCMD=?
Syntax	\$GPSCMD: (0-3) OK
Write Syntax	AT\$GPSCMD=<cmd>
Write Response	OK
Read Syntax	AT\$GPSCMD?
Read Response	\$GPSCMD: <cmd> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<cmd>	0 - Disables the GPS 1 - Cold start 2 - Hot start 3 - Stop the current GPS fix
Notes	<p>0 - GPS is in an idle state</p> <p>1 - Commands GPS to begin acquisition from a cold start.</p> <p>2 - Hot start can only be accomplished if it has valid time, ephemeris, almanac and position data.</p> <p>3 - Some Novatel Wireless products do not tracking GPS, and is defined as GPS fix stop (for power consumption)</p> <p>After a power up or reset, the modem will attempt an assisted cold start. This is defined as the modem supplying the GPS receiver any valid time, ephemeris, almanac, and position data that the modem might have stored.</p>



# AT\$GPSFD Restore GPS Filter Defaults

AT\$GPSFD	Restore GPS Filter Defaults
Command Function	<p>This command restores GPS filter defaults for the \$GPSQUAL, \$GPSFLT, and \$GPSRST commands without impacting the rest of the system configuration.</p> <p>This command is intended to be used after a software upgrade to take advantage of new filter settings without having to perform a full AT&amp;F.</p> <p>This command accepts one parameter (a 'sticky' flag). If the sticky flag is set, the GPS filter defaults will be persistent. If the sticky flag is not set, the GPS filter defaults will not be maintained past the next modem reset.</p>
Syntax Query	AT\$GPSFD=?
Syntax	\$GPSFD: (0-1) OK
Write Syntax	AT\$GPSFD=<sticky>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<sticky>	0 = The GPS modem defaults will not be maintained past the next modem reset. 1 = The GPS modem defaults will be persistent.
Notes	

# AT\$GPSFLT GPS Speed Filter

AT\$GPSFLT	GPS Speed Filter
Command Function	<p>At times the GPS chipset used by this device will report non-zero speeds when the device is not moving. Normally, the positions are accurate but the false speeds cause errors in the generation of the GPS idle, distance and odometer features. This filter is usually able to detect these false motions and clamp the reported speed to 0.0 knots while still reporting valid position data.</p> <p>Under normal circumstances, the default values should provide acceptable GPS performance for most applications. Please contact Novatel Wireless customer support if you feel the filter settings need to be changed for your application.</p>
Syntax Query	AT\$GPSFLT=?
Syntax	<p>If the \$GPSFLT debug flag is not set for expanded AT command response, the response will be as follows:</p> <p>\$GPSFLT: (0-2),(0-255),(0-180),(0-255),(0-255),(0-180),(0-180),(0-255),(0-255)</p> <p>OK</p> <p>If the \$GPSFLT debug flag is set for expanded AT command response, the response will be as follows:</p> <p>\$GPSFLT: md,tr,hdgD,tiAc,n4aAc,tiHdg,n4aHdg,pDop,dbg,unc,mot,ignB,tAdj</p> <p>OK</p>
Write Syntax	AT\$GPSFLT=<md>,<tr>,<hdgD>,<tiAc>,<n4aAc>,<tiHdg>,<n4aHdg>,<pDop>,<dbg>,<unc>,<mot>,<ignB>,<tAdj>
Write Response	OK
Read Syntax	AT\$GPSFLT?
Read Response	<p>AT\$GPSFLT: &lt;md&gt;,&lt;tr&gt;,&lt;hdgD&gt;,&lt;tiAc&gt;,&lt;n4aAc&gt;,&lt;tiHdg&gt;,&lt;n4aHdg&gt;,&lt;pDop&gt;,&lt;dbg&gt;,&lt;unc&gt;,&lt;mot&gt;,&lt;ignB&gt;,&lt;tAdj&gt;</p> <p>OK</p>
Execute Syntax	N/A

AT\$GPSFLT	GPS Speed Filter
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<md>	<p>0 = filter is disabled</p> <p>1 = filter is enabled</p> <p>2 = automatic (default). Filter determines whether device is in parked mode or driving mode by determining the number of seconds the device has been idle. If in parked mode, filter is enabled. If in driving mode, filter is disabled. This improves the performance of the GPS idle, distance, and odometer features when coming off stop signs and red lights, while still stopping most of the false speed-related events when the device is parked.</p>
<tr>	<p>0-255 = Trust threshold.</p> <p>With each GPS report, this filter uses the other filter settings to assess a confidence level to determine if indicated motion seems valid. This confidence level is either incremented or reset with each report. The confidence level is converted into a trust factor which is compared against this threshold value. If the trust factor is less than this threshold and the device is in the parked motion state, the speed is clamped to 0.0 knots. If the trust factor is above or equal to this threshold, the speed from the \$GPRMC is allowed through and the device transitions into the driving state.</p> <p>Default is 16.</p>
<hdgD>	<p>0-180 = Heading delta threshold.</p> <p>This filter calculates the heading from the two most recent position reports and compares it to the heading provided in the \$GPRMC sentence. If the delta between the two headings is greater than this threshold (in degrees), the confidence level is reset to zero.</p> <p>Default is 30 degrees.</p>
<tiAc>	<p>0-255 = TI acceleration threshold.</p> <p>This filter calculates the velocity change from the reported \$GPRMC velocity in the two most recent position reports. If the reported</p>

AT\$GPSFLT	GPS Speed Filter
	<p>velocity change is greater than this threshold (in knots/sec), the confidence level is reset to zero.</p> <p>Default is 10 knots/second.</p>
<n4aAc>	<p>0-255 = Novatel Wireless acceleration threshold.</p> <p>This filter calculates the velocity change from the distance between the two most recent position locations and based on the time delta. If the calculated velocity change is greater than this threshold (in knots/sec), the confidence level is reset to zero.</p> <p>Default is 10 knots/second.</p>
<tiHdg>	<p>0-180 = TI heading change threshold.</p> <p>This filter calculates the heading change from the reported \$GPRMC heading in the two most recent position reports. If the heading change is greater than this threshold (in degrees/second), the confidence level is reset to zero.</p> <p>Default is 30 degrees/second.</p>
<n4aHdg>	<p>0-180 = Novatel Wireless heading change threshold.</p> <p>This filter calculates the heading using the two most recent position locations and based on the time delta. If the difference between the two most-recent heading calculations is greater than this threshold (in degrees/second), the confidence level is reset to zero.</p> <p>Default is 30 degrees/second.</p>
<pDop>	<p>Obsolete. This value is no longer used. The field is maintained for backwards compatibility.</p>
<dbg>	<p>Bit mask made up of the following bit definitions:</p> <p>1 = Display all filter actions to serial port. If \$GPSQUAL or \$GPSFLT filters take any action the change or invalidate the GPS report, setting this bit will allow the code to display the action that was taken.</p>

AT\$GPSFLT	GPS Speed Filter
	<p>2 = Display runtime filter info to serial port. Allows code to display calculations/variables used by the filter.</p> <p>4 = Display geofence timing debug info.</p> <p>8 = System test flag.</p> <p>16 = GPS log flag. Enables code to log raw GPS data to a file in flash for post-mortem analysis for certain filtering events.</p> <p>32 = GPS message flag. Enabled code to send a message over the air and to the serial port, indicating that an abnormal GPS event has been detected and logged.</p> <p>64 = Raw NMEA flag. Display raw NMEA input on the serial port.</p> <p>128 = Expand \$GPSFLT=? and \$GPSQUAL=? Responses to provide a verbose explanation of each parameter.</p>
<unc>	<p>0-255 = Position uncertainty threshold.</p> <p>If internal position uncertainty value is greater than this value, the confidence level is reset to zero.</p> <p>Default is 75.</p>
<mot>	<p>0-999 = Motion transition threshold (in seconds).</p> <p>If device has been idle (no velocity) for this number of seconds, transition from driving mode to parked mode.</p> <p>Default is 120 seconds.</p>
<ignB>	<p>0-255 = Ignition bias.</p>

AT\$GPSFLT	GPS Speed Filter
	<p>For devices that are connected to the ignition via the white wire, the code can detect when the ignition is turned on/off and adjust the trust threshold accordingly. (Vehicles with the ignition off would only be moving if being towed. Vehicles with the ignition on will probably be driving if not moving already.)</p> <p>For devices that are not connected to the ignition, this value should be set to 0.</p> <p>Default is 6.</p>
<tAdj>	<p>0 = Do not adjust RTC time if RTC time disagrees with GPS time.</p> <p>1 = Adjust RTC time if RTC time and GPS time differ by more than 5 seconds.</p> <p>Default is 1.</p>
Notes	<p>There have also been cases where slow Time To First Fix (TTFF) reports have been attributed to discrepancies between the RTC time and the GPS time. When GPS filtering is enabled, the code ensures the RTC time is always with +/-5 seconds of the GPS time. This time sync feature is disabled by setting \$GPSFLT=0.</p> <p>If GPS filtering is modified or disabled, performance degradation may result.</p> <p>With the implementation of this filter, the \$ODOCFG command is now obsolete. For backwards compatibility, the \$ODOCFG command will still be accepted, but the code will take no action on its parameters.</p>

# AT\$GPSLCL Configure Sending Of GPS Message To The Serial Port

AT\$GPSLCL	Configure Sending of GPS Message to the Serial Port
Command Function	This command allows the user to configure sending of GPS data on the serial port.
Syntax Query	AT\$GPSLCL=?
Syntax	\$GPSLCL: (0-1),(0-127) OK
Write Syntax	AT\$GPSLCL=<option>,<nmeaMsgs>
Write Response	OK
Read Syntax	AT\$GPSLCL?
Read Response	\$GPSLCL: <option>,<nmeaMsgs> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	

AT\$GPSLCL	Configure Sending of GPS Message to the Serial Port												
<option>	<p>0 - Disable sending of GPS data to the local port when the device is in AT command mode (Default)</p> <p>1 - Enable sending of GPS NMEA ASCII data to the local port when the device is in AT command mode</p>												
<nmeaMsgs>	<p>This field is the bit-wise OR of the type of messages desired. The user has following message options to select from.</p> <p>Decimal Format</p> <table> <tr> <th>User Selectable</th><th>Type of NMEA Message</th></tr> <tr> <td>1</td><td>GGA</td></tr> <tr> <td>4</td><td>GSA</td></tr> <tr> <td>8</td><td>GSV</td></tr> <tr> <td>16</td><td>RMC</td></tr> <tr> <td>64</td><td>PENFG</td></tr> </table>	User Selectable	Type of NMEA Message	1	GGA	4	GSA	8	GSV	16	RMC	64	PENFG
User Selectable	Type of NMEA Message												
1	GGA												
4	GSA												
8	GSV												
16	RMC												
64	PENFG												
Notes	is used for debug purpose as directed by Novatel Wireless Technical Support Personnel. It is not a standard NMEA message.												



# AT\$GPSLNA Enable/Disable Internal LNA

AT\$GPSLNA	Enable/Disable Internal LNA
Command Function	This command is used to enable and disable the internal Low Noise Amplifier (LNA)
Syntax Query	\$GPSLNA=?
Syntax	\$GPSLNA: (0-2) OK
Write Syntax	\$GPSLNA=<cfg>
Write Response	OK
Read Syntax	\$GPSLNA?
Read Response	\$GPSLNA: =<cfg>,<actual> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<cfg>	0 = Internal LNA enabled 1 = Internal LNA disabled 2 = Automatic (default)
<actual>	0 = Internal LNA enabled 1 = internal LNA disabled
Notes	

# AT\$GPSOSI Set And Query GPS Overspeed Interval

AT\$GPSOSI	Set and Query GPS Overspeed Interval
Command Function	This command allows the user to define the criteria for a GPS overspeed event. A GPS overspeed event occurs when the minimum speed that is defined by the <speed> parameter is maintained for a specific duration of time.
Syntax Query	AT\$GPSOSI=?
Syntax	\$GPSOSI: (0-255),(0-65536),(0-1) OK
Write Syntax	AT\$GPSOSI=<speed>,<interval>,<sticky>
Write Response	OK
Read Syntax	AT\$GPSOSI?
Read Response	\$GPSOSI: <speed>, <interval>, <status>, <max_speed>, <duration> OK
Execute Syntax	AT\$GPSOSI
Execute Response	ERROR
Unsolicited Response	N/A
Parameter Values	
<speed>	Speed, in knots, must be met and/or exceeded to trigger the GPS overspeed event.
<interval>	Number of consecutive seconds for which <speed> must be maintained to trigger the GPS overspeed event.
<sticky>	Optional parameter to make overspeed parameters persistent. If 1, then speed/interval are written to flash immediately (no AT&W required). Default = 0.
<status>	If 1, then <max_speed> and <duration> represent a GPS overspeed interval that is currently active. If 0, they represent the previous GPS overspeed interval.

<b>AT\$GPSOSI</b>	<b>Set and Query GPS Overspeed Interval</b>
<b>&lt;max_speed&gt;</b>	The highest speed (in knots) that was attained in the current or previous GPS overspeed interval.
<b>&lt;duration&gt;</b>	Number of consecutive seconds that the speed was at or above <speed>.
<b>Notes</b>	If <speed> is set to zero, the GPS overspeed event is disabled.

# AT\$GPSQUAL GPS Quality Filters

AT\$GPSQUAL	GPS Quality Filters
Command Function	This command allows the user to define extra criteria for the event engine to use before it reports a position fix as valid.
Syntax Query	AT\$GPSQUAL=?
Syntax	If the \$GPSFLT debug flag is not set for expanded AT command response, the response will be as follows:  \$GPSQUAL: (0-255),(0-255),(0-30)  OK
Write Syntax	AT\$GPSQUAL=<flg>,<hdop>,<sats>
Write Response	OK
Read Syntax	AT\$GPSQUAL?
Read Response	\$GPSQUAL:<flg>,<hdop>,<sats>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<flg>	<p>Bit mask made up of the following bit definitions:</p> <p>1 = Mark GPS report invalid if \$GPGSA fix type is 2 (2D fix). This bit is provided for backwards compatibility with legacy products. It has not been shown to improve performance with this product.</p> <p>2 = Mark GPS report invalid if the report is based on a propagated fix.</p> <p>4 = Request POSITION EXTENDED report (in place of the basic POSITION report) from GPS chipset. This report includes additional info that the basic POSITION report does not.</p> <p>8 = Request MEASUREMENT EXTENDED report. This report includes information about satellite speeds which can be used in filtering.</p> <p>Default is 12 (request both POSITION EXTENDED and MEASUREMENT EXTENDED reports).</p>
<hdop>	<p>0 = Ignore HDOP when filtering. (default)</p> <p>1-255 = Mark GPS report invalid if HDOP value from \$GPGSA</p>

AT\$GPSQUAL	GPS Quality Filters
	<p>sentence is less than or equal to this indicated HDOP limit.</p> <p>This field is provided for backwards compatibility with legacy products. It has not been shown to improve performance with this product.</p>
<sats>	<p>0 (default) = Disregard # of satellites during filtering.</p> <p>1-30 = Mark GPS report invalid if the # of satellites in reported solution (from \$GPGLGA sentence) is below this value.</p>
Notes	<p>The units of the uncertainty threshold should not be taken literally. For example, a position uncertainty threshold of 100 meters does not guarantee that the actual location will be within 100 meters of the actual position.</p> <p>Please consult with Novatel Wireless technical support personnel before changing uncertainty thresholds.</p>
Example:	

# AT\$GPSRD Read Current GPS ASCII Data

AT\$GPSRD	Read Current GPS ASCII data												
Command Function	This command allows a user to read current NMEA format GPS data.												
Syntax Query	AT\$GPSRD=?												
Syntax	\$GPSRD: [(0-7F),(0-127)],(0-1) OK												
Write Syntax	N/A												
Write Response	N/A												
Read Syntax	AT\$GPSRD=<nmeaMsgs>,<decimal>												
Read Response	"\$GPG....." OK												
Execute Syntax	N/A												
Execute Response	N/A												
Unsolicited Response	NA												
Parameter Values	The output NMEA sentence depends on whether the <nmeaMsgs> parameter is entered in Hex or Decimal format. By default, the <decimal> parameter is not required and <nmeaMsgs> parameter has to be entered as HEX value without the preceding "0x" characters as outlined in Hex Format table below.												
<nmeaMsgs>	<p>This field is the sum of the type of NMEA messages desired. A user has the following message options to select from. Maximum value for &lt;nmeaMsgs&gt; in this case would be 7F in Hex format or 127 in decimal format.</p> <p>Hex Format</p> <table> <thead> <tr> <th>User Selectable</th><th>Type of NMEA Message</th></tr> </thead> <tbody> <tr> <td>0x01</td><td>GGA</td></tr> <tr> <td>0x04</td><td>GSA</td></tr> <tr> <td>0x08</td><td>GSV</td></tr> <tr> <td>0x10</td><td>RMC</td></tr> <tr> <td>0x40</td><td>PENFG</td></tr> </tbody> </table>	User Selectable	Type of NMEA Message	0x01	GGA	0x04	GSA	0x08	GSV	0x10	RMC	0x40	PENFG
User Selectable	Type of NMEA Message												
0x01	GGA												
0x04	GSA												
0x08	GSV												
0x10	RMC												
0x40	PENFG												

AT\$GPSRD	Read Current GPS ASCII data												
	<p>Decimal Format</p> <table> <tr> <th>User Selectable</th><th>Type of NMEA Message</th></tr> <tr> <td>1</td><td>GGA</td></tr> <tr> <td>4</td><td>GSA</td></tr> <tr> <td>8</td><td>GSV</td></tr> <tr> <td>16</td><td>RMC</td></tr> <tr> <td>64</td><td>PENFG</td></tr> </table>	User Selectable	Type of NMEA Message	1	GGA	4	GSA	8	GSV	16	RMC	64	PENFG
User Selectable	Type of NMEA Message												
1	GGA												
4	GSA												
8	GSV												
16	RMC												
64	PENFG												
<decimal>	<p>1 = &lt;nmeaMsg&gt; value has to be sum of User Selectable values from decimal table format</p> <p>0 = select values out of hex table format</p>												
Notes	The \$PENFG (Proprietary ENFora Gps) is used for debug purpose as directed by Novatel Wireless Technical Support Personnel. It is not a standard NMEA message.												

# AT\$GPSRST Set And Query GPS Reset Interval

AT\$GPSRST	Set and Query GPS Reset Interval
Command Function	<p>This command allows the user to specify the duration of a continuous GPS no lock condition after which the GPS chip will be reset.</p> <p>The GPS will often take several minutes to get a lock when power has been cycled to the device. However, if valid RTC time is available, then the lock should be acquired more quickly. The second parameter allows the user to cut down the time without a lock before resetting the GPS chip when outside of the cold start window.</p>
Syntax Query	AT\$GPSRST=?
Syntax	\$GPSRST: (0-255),(0-255) OK
Write Syntax	AT\$GPSRST=<coldStartInterval>,<warmStartInterval>
Write Response	OK
Read Syntax	AT\$GPSRST?
Read Response	\$GPSRST: <coldStartInterval>,<warmStartInterval> <filtCnt> OK
Execute Syntax	AT\$GPSRST
Execute Response	ERROR
Unsolicited Response	N/A
Parameter Values	
<coldStartInterval>	When RTC time is not valid (year < 2010), this is the time (in tens of minutes) of GPS outage after which to reset the GPS chip and resume positioning.



AT\$GPSRST	Set and Query GPS Reset Interval
<warmStartInterval>	When RTC time is valid (year >= 2010) and this value is zero, then coldStartInterval is used as defined above. When RTC time is valid (year >= 2010) and this value is greater than zero, this is the time (in minutes) of GPS outage after which to reset the GPS chip and resume positioning.
Notes	If <coldStartInterval> is set to 0, the GPS chip will not be reset after a GPS outage of any duration.
Example	AT\$GPSRST=1,2 Code will allow 10 minutes to acquire a GPS lock before resetting the GPS chip following a power cycle (RTC time not valid). Once a lock has been acquired (RTC time automatically set when a lock has been acquired), the code will allow two minutes of 'no GPS lock' time before resetting the GPS chip. Since RTC time is maintained through an ignition reset, the two minute limit would be in effect following an ignition reset.

# AT\$GPSVAL GPS Speed Validation

AT\$GPSVAL	GPS Speed Validation
Command Function	This command is used to turn GPS speed filtering on and off.
Syntax Query	AT\$GPSVAL=?
Syntax	\$ GPSVAL: (0-1) OK
Write Syntax	AT\$ GPSVAL =<MODE>
Write Response	OK
Read Syntax	AT\$ GPSVAL?
Read Response	\$ GPSVAL: <MODE> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<MODE>	0 - Use default for GPS Speed filter 1 - Disable GPS Speed filter
Notes	The GPS speed filter here is the same as the GPS speed filter in \$GPSFLT.
Examples	

# AT\$ODOMETER GPS Trip Odometer

AT\$ODOMETER	GPS Trip Odometer
Command Function	The \$ODOMETER command records how far the vehicle has traveled total, or in one trip. The user can reset the odometer at the beginning of a new trip. This is identical to the \$TODOM feature, but allows the user to reset this odometer without resetting the other odometer.
Syntax Query	AT\$ODOMETER=?
Syntax	\$ODOMETER: (0-4000000000)
Write Syntax	AT\$ODOMETER=1234 (where 1234 is distance in meters)
Write Response	OK
Read Syntax	AT\$ODOMETER?
Read Response	\$ODOMETER xxxx (xxxx=distance traveled in meters)
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	N/A
Notes	<p>The user is able to set a seed value for the Trip Odometer starting at 0 but not higher than the maximum value of 4000000000.</p> <p>The AT&amp;F command will not reset the odometer value to 0.</p> <p>The Virtual Odometer reading would be a 4-byte value starting from 0 to 4000000000 (maximum of approximately 2500000 miles before it rolls over to 0)</p> <p>The unit for Virtual Odometer shall be in METERS.</p> <p>The Virtual Odometer history shall be updated every second</p> <p>The Virtual Odometer history shall be saved once a minute in modem's memory. This value shall be retained through an internal or external reset and can be read upon the next power up or during run time mode. The delta distance traveled between the minute marks could be lost due to an unexpected external or non-modem originated reset. However, the total distance traveled till the prior minute would still be preserved.</p>

# AT\$PLYCLR Clear Polygonal Geofence

AT\$PLYCLR	Clear Polygonal Geofence
Command Function	This command allows a user to delete all the points for the indicated polygonal geofence (see \$PLYFN#).
Syntax Query	AT\$PLYCLR=?
Syntax	\$PLYCLR: (0-24) OK
Write Syntax	AT\$PLYCLR=<id>
Write Response	OK
Read Syntax	N/A
Read Response	N/A
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<id>	0-24: identifier of geographical polygon (see \$PLYFN#)
Notes	

# AT\$PLYFN# Geo Fencing A Polygonal Area

AT\$PLYFN#	Geo Fencing a Polygonal Area
Command Function	<p>This command allows a user to define up to 25 separate polygonal geographical areas (# can be any number from 0 to 24). The event engine (see AT\$EVENT) can be set up to send a message when the device moves in or out of the geographical area defined by the polygon.</p> <p>Each command defines one of 26 possible vertices or corners. The line segments of the polygon are generated by connecting the non-zero vertices in sequence (for example, point 0 is connected to point 1, which is connected to point 2, etc.). The polygon can be generated in either a clockwise or counterclockwise sequence (see Notes section below for additional details). You can specify your polygon with less than 26 vertices, but the first and last points of the polygon still must be identical</p> <p>The query returns all non-zero locations defined for the polygon as well as all (0,0) vertex locations that serve as a polygon terminator.</p>
Syntax Query	AT\$PLYFN#=?
Syntax	\$PLYFN#: (0-25),(-90.0 - +90.0),(-180.0 - +180.0) OK
Write Syntax	AT\$PLYFN#=<idx>,<latitude>,<longitude>
Write Response	OK
Read Syntax	AT\$PLYFN#?
Read Response	\$PLYFN#: PtIdx Latitude Longitude <idx> <latitude> <longitude> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<idx>	0-25: index for this vertex of the polygon

AT\$PLYFN#	Geo Fencing a Polygonal Area
<latitude>	Defines the latitude for this vertex of the polygon in decimal degrees (for example, 32 degrees 30 minutes 0 seconds would be represented as 32.5 degrees since 30 minutes is exactly 1/2 of a degree). The latitude must be specified in 15 characters (including minus sign and decimal point) or less. This is the same representation for latitude as used in the AT\$GEOFNC command.
<longitude>	Defines the longitude for this vertex of the polygon in decimal degrees (for example, -96 degrees 45 minutes 0 seconds would be represented as -96.75 degrees since 45 minutes is exactly 3/4 of a degree). The longitude must be specified in 15 characters (including minus sign and decimal point) or less. This is the same representation for longitude as used in the AT\$GEOFNC command.
<b>Notes</b>	<p>In general, it is always a good idea to verify the integrity of your polygon description with a mapping tool such as Google Earth. The most common mistake with polygons is to mistype one of the fractional digits in one of the coordinates. When this happens, the polygon you've created with your mapping tool is not the polygon you've entered into this command and you will not get your desired results.</p> <p>The code implements the pnpoly algorithm. The pnpoly algorithm is copyright © 1970-2003, Wm. Randolph Franklin.</p> <p>Regardless of how many points you specify for your polygon, all 26 points are fed to the algorithm. Due to the mathematical nature of this algorithm, the polygon will not work if vertices of the same polygon lie on opposite sides of the 180 degree longitude divide.</p> <p>Although this command accepts latitude/longitude parameters with up to 15 characters, internally the value is stored as a C float type which has less precision (but requires half the storage size). The float type is capable of storing accuracy commensurate with the GPS receiver's capability, but the queried latitude/longitude values of the AT\$PLYFN# command may differ from the input parameters due to this precision limitation.</p>
<b>Example</b>	<p>The following commands create a crude diamond-shaped polygon #9 just west of the Novatel Wireless offices in Richardson, Texas:</p> <pre> AT\$PLYFN9=1,32.995498,-96.722064 AT\$PLYFN9=2,33.000392,-96.715028 AT\$PLYFN9=3,33.008747,-96.722466 AT\$PLYFN9=4,33.001330,-96.732982 AT\$PLYFN9=5,32.995498,-96.722064 </pre>

AT\$PLYFN#	Geo Fencing a Polygonal Area																					
	<p>Here's the query response for that polygon definition:</p> <p>AT\$PLYFN9?</p> <p>\$PLYFN9: PtIdx   Latitude   Longitude</p> <table><tr><td>0</td><td>0.0000000</td><td>0.0000000</td></tr><tr><td>1</td><td>32.9954987</td><td>-96.7220612</td></tr><tr><td>2</td><td>33.0003929</td><td>-96.7150269</td></tr><tr><td>3</td><td>33.0087471</td><td>-96.7224655</td></tr><tr><td>4</td><td>33.0013313</td><td>-96.7329788</td></tr><tr><td>5</td><td>32.9954987</td><td>-96.7220612</td></tr><tr><td>6</td><td>0.0000000</td><td>0.0000000</td></tr></table> <p>OK</p>	0	0.0000000	0.0000000	1	32.9954987	-96.7220612	2	33.0003929	-96.7150269	3	33.0087471	-96.7224655	4	33.0013313	-96.7329788	5	32.9954987	-96.7220612	6	0.0000000	0.0000000
0	0.0000000	0.0000000																				
1	32.9954987	-96.7220612																				
2	33.0003929	-96.7150269																				
3	33.0087471	-96.7224655																				
4	33.0013313	-96.7329788																				
5	32.9954987	-96.7220612																				
6	0.0000000	0.0000000																				

# AT\$PWRSAPV Enable Power Save Mode

AT\$PWRSAPV	Enable Power Save Mode
Command Function	This command allows a user to put the device in low power mode at the <timeout> interval after the Ignition line drops. The ignition line has to be connected per the user manual for this feature to work properly. A user has the capability of getting a notification when the device entering low power mode or returns to normal operating mode.
Syntax Query	AT\$PWRSAPV=?
Syntax	\$PWRSAPV: (0-1),(0-65535),(0-1) OK
Write Syntax	AT\$PWRSAPV=<ign>,<timeout>,<reg>
Write Response	OK
Read Syntax	AT\$PWRSAPV?
Read Response	\$PWRSAPV: 0,0,0 OK
Execute Syntax Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ign>	0 - disable the Ignition feature 1 - enter low power mode after Ignition signal went low and timeout has expired
<timeout>	0 - 65535 seconds. Timeout value after which the unit will enter low power mode. Unit will work in normal mode until the timeout has expired.
<reg>	0 - remain registered with network during low power mode 1 - reset modem when entering normal power mode
Notes	An AT\$EVENT command has to be set to send a GPS message to the remote host when entering or exiting power save mode.  The modem Ignition (switched power) Input must be connected to the vehicles ignition line for this function to work.



# AT\$TODOM Virtual Trip Odometer

AT\$TODOM	Virtual Trip Odometer
Command Function	The \$TODOM command records how far the vehicle has traveled. This is identical to the \$ODOMETER feature, but allows the user to reset this trip odometer without resetting the other odometer.
Syntax Query	AT\$TODOM=?
Syntax	\$TODOM:(0-40000000000) OK
Write Syntax	AT\$TODOM=1234 (where 1234 is distance in meters)
Write Response	OK
Read Syntax	AT\$TODOM?
Read Response	\$TODOM: xxxx (xxxx=distance traveled in meters)
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A

<b>AT\$TODOM</b>	<b>Virtual Trip Odometer</b>
<b>Parameter Values</b>	N/A
<b>Notes</b>	<p>The user is able to set a seed value for the Trip Odometer starting at 0 but not higher than the maximum value of 4000000000.</p> <p>The AT&amp;F command will not reset the odometer value to 0. (see example below)</p> <p>The Trip Odometer reading would be a 4-byte value starting from 0 to 4000000000 (maximum of approximately 2500000 miles before it rolls over to 0)</p> <p>The unit for Trip Odometer shall be in meters.</p> <p>The Trip Odometer history shall be updated every second.</p> <p>The Trip Odometer history shall be saved once a minute in modem's memory. This value shall be retained through an internal or external reset and can be read upon the next power up or during run time mode. The delta distance traveled between the minute marks could be lost due to an unexpected reset. However, the total distance traveled till the prior minute would still be preserved.</p> <p>The \$ODOMETER value can trigger the event engine via input event 70. The \$TODOM value can trigger the event engine via input event 91.</p>
<b>Example</b>	<p>Reset Trip Odometer to 0:</p> <p>\$TODOM=0</p>

# Battery Commands

---

The backup battery commands only apply to MT 3000 Series devices that contain a backup battery. Do not attempt to use these commands on devices without backup batteries.

## AT\$OBDBAT Enable/Disable Backup Battery

AT\$OBDBAT	Enable/Disable Backup Battery
Command Function	This command is used to enable/disable the internal backup battery. If this is disabled the unit will power off as soon as main power is removed.
Syntax Query	AT\$OBDBAT=?
Syntax	\$OBDBAT: (0-1) OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$OBDBAT?
Read Response	\$OBDBAT: <status>,<voltage> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<status>	0 = Internal Battery is Disabled 1 = Internal Battery is Enabled (default value)

<b>AT\$OBDBAT</b>	<b>Enable/Disable Backup Battery</b>
<b>&lt;voltage&gt;</b>	Internal battery voltage, displayed in millivolts.
<b>Notes</b>	<p>Products that do not have a battery installed will return the response of ERROR when AT\$OBDBAT is issued.</p> <p>The displayed voltage reading is only meaningful when running on battery power. When external power is available, the displayed voltage will reflect the charging voltage.</p> <p>The device will shut down if the internal backup battery voltage drops to 3.00 V</p>
<b>Example</b>	<p>AT\$OBDBAT?</p> <p>\$OBDBAT: 1,4753</p> <p>OK</p> <p>This example shows the read operation and response for a device where the internal backup battery is enabled and the voltage reading is 4.753 V.</p>

# AT\$OBDBBT Backup Battery Timer

AT\$OBDBBT	Backup Battery Timer
Command Function	This command is used to configure the length of time that the unit will run on internal battery power after main power is removed.
Syntax Query	AT\$OBDBBT=?
Syntax	\$OBDBBT: (30-6000)
Write Syntax	AT\$OBDBBT=<time>
Write Response	OK
Read Syntax	AT\$OBDBBT?
Read Response	\$OBDBBT: <time>,<time_remaining>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<time>	Time (in seconds) to remain running after main power has been removed Range: 30-6000 Default: 300
<time_remaining>	Time (in seconds) remaining before shutting down. This value is calculated by subtracting the elapsed time since the loss of main power from the run time specified by this command.
Notes	
Example	

# AT\$OBDBQS Backup Battery Quick Shutdown

AT\$OBDBQS	Backup Battery Quick Shutdown
Command Function	This command is designed to be used before shipping or storing the unit to preserve the power in the internal battery.
Syntax Query	AT\$OBDBQS=?
Syntax	\$OBDBQS: (0-1) OK
Write Syntax	AT\$OBDBQS=<status>
Write Response	OK
Read Syntax	AT\$OBDBQS?
Read Response	\$OBDBQS: <status> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<status>	0 = Backup battery quick shutdown is disabled (default) 1 = Backup battery quick shutdown is enabled
Notes	
Example	

# OBD Commands

---



# AT\$OBDACL Acceleration Thresholds

AT\$OBDACL	OBD 2 Acceleration Thresholds
Command Function	This command is used to set the three acceleration thresholds.
Syntax Query	AT\$OBDACL=?
Syntax	\$OBDACL: (1-3),(0-4000),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDACL=<ALERT>,<THRHL D>,<SET TIME>,<CLR TIME>
Write Response	OK
Read Syntax	AT\$OBDACL?
Read Response	\$OBDACL: <THR LD>,<SET TIME>,<CLR TIME><THR LD>,<SET TIME>,<CLR TIME><THR LD>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Acceleration alert number
<THRHL D>	Acceleration in milli Gs. 0 is the default value and is used to disable that threshold.
<SET TIME>	Time above threshold to set alert in seconds
<CLR TIME>	Time below threshold to clear alert in seconds
Notes	
Examples	

# AT\$OBDADN Set Down Accelerometer Threshold

AT\$OBDADN	OBD2 Set Down Accelerometer threshold
Command Function	This command is used to set the 'down' accelerometer threshold.
Syntax Query	AT\$OBDADN=?
Syntax	\$OBDADN: (1),(0-4000),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDADN=<ALERT>,<THRHL D>,<SET TIME>,<CLR TIME>
Write Response	OK or ERROR
Read Syntax	AT\$OBDADN?
Read Response	\$OBDADN: <THRHL D>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Acceleration alert number (must be 1)
<THRHL D>	Acceleration in milli Gs 0 is the default value and is used to disable that threshold.
<SET TIME>	Time above threshold to set alert in seconds (default 1)
<CLR TIME>	Time below threshold to clear alert in seconds (default 2)
Notes	N/A

# AT\$OBDADT Auto Discovery Timer

AT\$OBDADT	OBD2 Auto Discovery Timer
Command Function	This command is used to set the OBD2 auto discovery timer.
Syntax Query	AT\$OBDADT=?
Syntax	\$ OBDADT: (0-65535) OK
Write Syntax	AT\$ OBDADT =<DISCOVERY TIMER>
Write Response	OK
Read Syntax	AT\$ OBDADT?
Read Response	\$ OBDADT: < DISCOVERY TIMER >,< DISCOVERY TIME REMAINING > OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<DISCOVERY TIMER>	OBD2 Auto Discovery Timer 0 - Auto Discovery Timer Disabled 1-65535 Time in seconds
<DISCOVERY TIME REMAINING>	Time remaining on OBD2 Auto Discovery Timer 0-65535 Time in seconds
Notes	N/A

# AT\$OBDALT Set Left Turn Accelerometer Threshold

AT\$OBDALT	OBD2 Set Left Turn Accelerometer Threshold
Command Function	This command is used to set the two left turn accelerometer thresholds.
Syntax Query	AT\$OBDALT=?
Syntax	\$OBDALT: (1-2),(0-4000),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDALT=<ALERT>,<THRHL>,<SET TIME>,<CLR TIME>
Write Response	OK or ERROR
Read Syntax	AT\$OBDALT?
Read Response	\$OBDALT: <THRHL>,<SET TIME>,<CLR TIME>,<THRHL>,<SET TIME>,<CLRTIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Acceleration alert number
<THRHL>	Acceleration in milli Gs 0 is the default value and is used to disable that threshold.
<SET TIME>	Time above threshold to set alert in seconds (default 1)
<CLR TIME>	Time below threshold to clear alert in seconds (default 2)
Notes	N/A

# AT\$OBDAM Set Any Motion Parameters

AT\$OBDAM	Set Any Motion Parameters
Command Function	This command allows the user to set the 'any motion' parameters of the accelerometer. The 'any motion' setting allows users to trigger an event when the accelerometer senses motion in any axis.
Syntax Query	AT\$OBDAM=?
Syntax	\$OBDAM: (0-1),(0-255),(0-3),(0-64800) OK
Write Syntax	AT\$OBDAM= <enable>,<threshold>,<count>,<holdtime>
Write Response	OK
Read Syntax	AT\$OBDAM?
Read Response	\$OBDAM: <enable>,<threshold>,<count>,<holdtime>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<enable>	0 = Disable the 'any motion' feature 1 = Enable the 'any motion' feature (default)
<threshold>	Accelerometer 'any motion' threshold. This value represents the level of movement required to be considered 'any motion.' Lower values are more sensitive to motion. The <threshold> values are raw hardware units ( <u>not</u> mili-G's).  Range: 0 - 255 (Default value = 10)

AT\$OBDAM	Set Any Motion Parameters
<count>	Accelerometer 'any motion' count Range: 0 - 3 (Default value = 3)
<holdtime>	Number of seconds of no motion before the 'any motion' event is cleared. Range: 0 - 64800 (Default value = 180)
Notes	None yet.

# AT\$OBDART Set Right Turn Threshold

AT\$OBDART	OBD2 Set Right Turn Threshold
Command Function	This command is used to set the two right turn accelerometer thresholds.
Syntax Query	AT\$OBDART=?
Syntax	\$OBDART: (1-2),(0-4000),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDART=<ALERT>,<THRHL D>,<SET TIME>,<CLR TIME>
Write Response	OK or ERROR
Read Syntax	AT\$OBDART?
Read Response	\$OBDART: <THR LD>,<SET TIME>,<CLR TIME>,<THR LD>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Acceleration alert number
<THRHL D>	Acceleration in milli Gs 0 is the default value and is used to disable that threshold.
<SET TIME>	Time above threshold to set alert in seconds (default 1)
<CLR TIME>	Time below threshold to clear alert in seconds (default 2)
Notes	N/A

# AT\$OBDAUP OBD2 Set Up Accelerometer Threshold

AT\$OBDAUP	OBD2 Set Up Accelerometer Threshold
Command Function	This command is used to set the 'up' accelerometer threshold.
Syntax Query	AT\$OBDAUP=?
Syntax	\$OBDAUP: (1),(0-4000),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDAUP=<ALERT>,<THRHL D>,<SET TIME>,<CLR TIME>
Write Response	OK or ERROR
Read Syntax	AT\$OBDAUP?
Read Response	\$OBDAUP: <THRHL D>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Acceleration alert number (must be 1)
<THRHL D>	Acceleration in milli Gs 0 is the default value and is used to disable that threshold.
<SET TIME>	Time above threshold to set alert in seconds (default 1)
<CLR TIME>	Time below threshold to clear alert in seconds (default 2)
Notes	N/A



# AT\$OBDCUT Clear The OBD2 User Trip Odometer

AT\$OBDCUT	Clear the OBD2 User Trip Odometer
Command Function	This command is used to set the User trip odometer to zero. This is used in conjunction with Input Event 192 - OBD User Trip Odometer.
Syntax Query	AT\$OBDCUT=?
Response	OK
Write Syntax	N/A
Response	N/A
Read Syntax	N/A
Response	N/A
Execute Syntax	AT\$OBDCUT
Response	OK
Parameter Values	N/A
Notes	N/A
Example	AT\$STOATEV=1,AT\$OBDCUT // send stored AT event to reset trip distance AT\$EVENT=20,0,192,3000,2147483647 //Trigger trip distance at 3000 meters AT\$EVENT=20,3,41,2,540016710 // send acknowledged UDP message AT\$EVENT=20,3,44,1,0 // send stored at command tin index 1

# AT\$OBDDCL Deceleration Thresholds

AT\$OBDDCL	OBD2 Deceleration Thresholds
Command Function	This command is used to set the three deceleration thresholds.
Syntax Query	AT\$OBDDCL=?
Syntax	\$OBDDCL: (1-3),(0-4000),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDDCL=<ALERT>,<THRHL D>,<SET TIME>,<CLR TIME>
Write Response	OK
Read Syntax	AT\$OBDDCL?
Read Response	\$OBDDCL: <THRLD>,<SET TIME>,<CLR TIME><THRLD>,<SET TIME>,<CLR TIME><THRLD>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Deceleration alert number
<THRHL D>	Deceleration in milli Gs 0 is the default value and is used to disable that threshold.
<SET TIME>	Time above threshold to set alert in seconds
<CLR TIME>	Time below threshold to clear alert in seconds
Notes	
Examples	

# AT\$OBDDSC Discovery Configuration

AT\$OBDDSC	OBD 2 Discovery Configuration
Command Function	This command is used to set the OBD2 protocol discovery method
Syntax Query	AT\$OBDDSC=?
Syntax	\$OBDDSC: (0-7, 255) OK
Write Syntax	AT\$OBDDSC=<PROTOCOL>
Write Response	OK
Read Syntax	AT\$ OBDDSC?
Read Response	\$OBDDSC:< PROTOCOL> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<PROTOCOL>	OBD2 Protocol Discovery 0 – ISO 15765 250 kHz 11 bit 1 – ISO 15765 500 kHz 11 bit 2 – ISO 15765 250 kHz 29 bit 3 – ISO 15765 500 kHz 29 bit 4 – J1850 PWM 5 – J1850 VPW 6 – ISO 9141 2 7 – ISO 14230 255 – Auto Discovery
Notes	
Examples	

# AT\$OBDECT Engine Coolant Temperature Threshold

AT\$OBDECT	OBD2 Engine Coolant Temperature Threshold
Command Function	This command is used to get or set the Engine Coolant Temperature Threshold.
Syntax Query	AT\$OBDECT=?
Syntax	\$OBDECT: (0-255),(30-1000),(30-1000) OK
Write Syntax	AT\$ OBDECT =<value>, <set time>, <clear time>
Write Response	OK
Read Syntax	AT\$ OBDECT?
Read Response	\$ OBDECT: <value>, <set time>, <clear time> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<value>	0 = threshold is disabled 1 = -39 degrees Celcius 255 = 215 degrees Celcius
<set time>	Time above threshold in seconds before event is set
<clear time>	Time below threshold in seconds before event is cleared
Notes	N/A

# AT\$OBDEES Excessive Engine Speed Thresholds

AT\$OBDEES	OBD2 Excessive Engine Speed Thresholds
Command Function	This command is used to set the three excessive engine speed thresholds.
Syntax Query	AT\$OBDEES=?
Syntax	\$OBDEES: (1-3),(0-65535),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDEES=<ALERT>,<THRHLD>,<SET TIME>,<CLR TIME>
Write Response	OK
Read Syntax	AT\$OBDEES?
Read Response	\$OBDEES: <THRLD>,<SET TIME>,<CLR TIME><THRLD>,<SET TIME>,<CLR TIME><THRLD>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Excessive engine speed alert number
<THRHLD>	Engine speed in ¼ RPM 0 is the default setting and is used to disable that threshold
<SET TIME>	Time above threshold to set alert in seconds
<CLR TIME>	Time below threshold to clear alert in seconds
Notes	
Examples	

# AT\$OBDFAC Set Thresholds To Factory Defaults

AT\$OBDFAC	OBD2 Set Thresholds to Factory Defaults
Command Function	This command is used to set all of the OBD2 thresholds to factory default values.
Syntax Query	AT\$OBDFAC=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$OBDFAC
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	AT\$OBDFAC resets all aux processor based settings including the accelerometer configuration
Examples	

# AT\$OBDGSP Use OBD Speed In Place Of GPS Speed

AT\$OBDGSP	Use OBD Speed in place of GPS Speed
Command Function	This command is used to select GPS or OBD generated speed in binary output messages – Bit Field Table 0 bit 12
Syntax Query	AT\$OBDGSP=?
Syntax	\$OBDGSP: (0-1) OK
Write Syntax	AT\$OBDGSP =<USE OBD SPEED>
Write Response	OK
Read Syntax	AT\$OBDGSP?
Read Response	\$OBDGSP: <USE OBD SPEED> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<USE OBD SPEED>	Use OBD Speed 0 – Use GPS speed 1 – Use OBD speed
Notes	

# AT\$OBDIDL Idle Speed Threshold

AT\$OBDIDL	OBD2 Idle Speed Threshold
Command Function	This command is used to set the idle speed threshold.
Syntax Query	AT\$OBDIDL=?
Syntax	\$OBDIDL: (0-255),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDIDL=<THRHL>,<SET TIME>,<CLR TIME>
Write Response	OK
Read Syntax	AT\$OBDIDL?
Read Response	\$OBDIDL: <THRHL>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<THRHL>	Idle speed in KPH
<SET TIME>	Time above threshold to set alert in seconds
<CLR TIME>	Time below threshold to clear alert in seconds
Notes	When the threshold is set to zero the event is disabled.
Examples	



# AT\$OBDIGN OBD2 Ignition Off Configuration

AT\$OBDIGN	OBD2 Ignition Off Configuration
Command Function	This command is used to configure ignition off detection.
Syntax Query	AT\$OBDIGN=?
Syntax	\$OBDIGN: (0-600),(0-1) OK
Write Syntax	AT\$ OBDIGN =<OBD OFF>,< IGN OFF >
Write Response	OK
Read Syntax	AT\$ OBDIGN?
Read Response	\$ OBDIGN: < OBD OFF >, < IGN OFF > OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
< OBD OFF >	OBD2 Off Time 0-600 Time in seconds after vehicle and engine speed are zero before dropping OBD communications with the ECU.
< IGN OFF >	0 – Ignition off when vehicle and engine speed are zero. 1 – Ignition off after OBD2 off time expires
Notes	The recommended setting for hybrids and stop/start vehicles is 180,1. A longer time can be used safely if the ECU stops communicating after the key is removed. There is a power consumption trade off for ECUs that continue transmitting after the key is removed. The default setting and recommended setting for all other vehicles is 60,0.
Examples	

# AT\$OBDLBL Low Battery Level Threshold

AT\$OBDLBL	OBD2 Low Battery Level Threshold
Command Function	This command is used to set the low battery level threshold.
Syntax Query	AT\$OBDLBL=?
Syntax	\$OBDLBL: (0-25500),(150-1000),(150-1000) OK
Write Syntax	AT\$OBDLBL=<THRHL>,<SET TIME>,<CLR TIME>
Write Response	OK
Read Syntax	AT\$OBDLBL?
Read Response	\$OBDLBL: <THRHL>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<THRHL>	Low battery level threshold in millivolts
<SET TIME>	Time below threshold to set alert in seconds
<CLR TIME>	Time above threshold to clear alert in seconds
Notes	When the threshold is set to zero the event is disabled. The typical component tolerance is +/- 4.5%
Examples	

# AT\$OBDLED OBD2 LED Timer

AT\$OBDLED	OBD2 LED Timer
Command Function	This command is used to set the length of time that the LEDs will blink after a power on reset.
Syntax Query	AT\$OBDLED=?
Syntax	\$ OBDLED: (0-240) OK
Write Syntax	AT\$ OBDLED =<LED TIMER>
Write Response	OK
Read Syntax	AT\$ OBDLED?
Read Response	\$ OBDLED: < LED TIMER >< LED TIME REMAINING > OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<LED TIMER>	OBD2 LED Timer 0 - LED Timer Disabled (LEDs will always blink) 1-240 Time in minutes that LEDs will blink
<LED TIME REMAINING>	Time remaining on OBD2 LED Timer 0-240 Time in minutes
Notes	

# AT\$OBDLFL Low Fuel Level Threshold

AT\$OBDLFL	OBD2 Low Fuel Level Threshold
Command Function	This command is used to set the low fuel level threshold.
Syntax Query	AT\$OBDLFL=?
Syntax	\$OBDLFL: (0-100),(1-1000),(1-1000) OK
Write Syntax	AT\$OBDLFL=<THRHL>,<SET TIME>,<CLR TIME>
Write Response	OK
Read Syntax	AT\$OBDLFL?
Read Response	\$OBDLFL: <THRHL>,<SET TIME>,<CLR TIME>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<THRHL>	Low fuel level threshold in percent
<SET TIME>	Time below threshold to set alert in seconds
<CLR TIME>	Time above threshold to clear alert in seconds
Notes	When the threshold is set to zero the event is disabled.
Examples	

# AT\$OBDSAV Save OBDII Thresholds To Memory

AT\$OBDSAV	Save OBD2 Thresholds to memory
Command Function	This command is used to save all of the OBD2 thresholds to flash memory.
Syntax Query	AT\$OBDSAV=?
Syntax	OK
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$OBDSAV
Execute Response	OK
Unsolicited Response	N/A
Parameter Values	N/A
Notes	If AT\$OBDSAV is not sent, any changes made to OBD or accelerometer commands are maintained through a reset (AT\$RESET), but are lost if the power is cycled.

# AT\$OBDSPD Vehicle Speed Thresholds

AT\$OBDSPD	OBD2 Vehicle Speed Thresholds
Command Function	This command is used to set the three excessive vehicle speed thresholds.
Syntax Query	AT\$OBDSPD=?
Syntax	\$OBDSPD: (1-3),(0-255),(0-1000),(0-255) OK
Write Syntax	AT\$OBDSPD=<ALERT>,<THRHLD>,<SET TIME>,<CLR THRLD>
Write Response	OK
Read Syntax	AT\$OBDSPD?
Read Response	\$OBDSPD: <THRLD>,<SET TIME>,<CLR THRHLD><THRLD>,<SET TIME>,<CLR THRHLD><THRLD>,<SET TIME>,<CLR THRHLD>
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<ALERT>	Excessive vehicle speed alert number
<THRHLD>	Vehicle speed in KPH
<SET TIME>	Time above threshold to set alert in seconds
<CLR THRLD>	Vehicle speed in KPH
Notes	When the set and clear thresholds are set to zero the event is disabled.
Examples	

# AT\$OBDTYP Protocol Type

AT\$OBDTYP	OBD2 Protocol Type
Command Function	This command is used to query the OBD2 protocol type.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$OBDTYP?
Read Response	\$OBDTYP: <OBD TYPE>,<OBD COMM > OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<OBD Type>	OBD2 Protocol Type 0 = ISO 15765 250 kHz 11 bit 1 = ISO 15765 500 kHz 11 bit 2 = ISO 15765 250 kHz 29 bit 3 = ISO 15765 500 kHz 29 bit 4 = J1850 PWM 5 = J1850 VPW 6 = ISO 9141 2 7 = ISO 14230 255 = Unknown
<OBD COMM>	OBD2 Communication Status 0 - Not Active 1 - Active
Notes	
Examples	





# AT\$OBDVER Auxiliary Processor

AT\$OBDVER	OBD2 Auxiliary Processor
Command Function	This command is used to query the software version of the auxiliary processor.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	AT\$OBDVER?
Read Response	\$OBDVER: <SW VER> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<SW Ver>	Auxiliary processor software version
Notes	
Examples	

# Miscellaneous Commands

---

# AT\$IGNDBNC Ignition Debounce

AT\$IGNDBNC	Ignition Debounce
Command Function	This command allows the user to set ignition debounce time used for the event engine. The ignition line has to be valid for the specified amount of time before the event: GPIO-8 in the event engine will be triggered.
Syntax Query	AT\$IGNDBNC=?
Syntax	\$IGNDBNC: (1-4) OK
Write Syntax	AT\$IGNDBNC=<debounceTimeout>
Write Response	OK
Read Syntax	AT\$IGNDBNC?
Read Response	\$IGNDBNC: 0
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<delay>	1 - 4 seconds. This field specifies the debounce timeout value.
Notes:	The "reset upon ignition" reset interval will follow the setting of AT\$IGNDBNC. \$IODBNC is affected by this command. Changes made affect GPIO8 in \$IGNDBNC? writes made by \$IGNDBNC will be seen by \$IODBNC?

# AT\$HBRST Automatic Modem Reset

AT\$HBRST	Automatic Modem Reset
Command Function	This command allows the user to program the reset interval and enable/disable ignition resets on supported devices.
Syntax Query	AT\$HBRST=?
Syntax	\$HBRST:(0-168),(0-1*),(0-1),(0-1*) OK
Write Syntax	AT\$HBRST=<hours>,<ign rst inhibit>,<motion>,<ignition>
Write Response	OK
Read Syntax	AT\$HBRST?
Read Response	\$HBRST:<hours>,<ign rst inhibit>,<motion>,<ignition> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A
Parameter Values	
<hours>	0 = Automatic reset turned off 1-168 = Number of hours until the modem resets
<ign rst inhibit>	0 = inhibit off (default). Modem will be reset when ignition on event is detected. 1 = Inhibit reset upon ignition on  Note: If the device is being reset by ignition input (switched power), the frequency of the heartbeat reset can be set as high as 168 hours. If ignition resets are inhibited, the maximum heartbeat reset allowed is 24 hours. Therefore, if <ign rst inhibit> = 1, <hours> must = (1-24)  * Note: On devices that do not support ignition detection, this parameter value can only be set to 1.

AT\$HBRST	Automatic Modem Reset
<motion>	<p>0 = Postpone automatic reset when &lt;hours&gt; threshold is reached if device is in 'driving' motion state until mode transitions to 'parked' motion state.</p> <p>1= Ignore motion state when implementing automatic reset logic.</p>
<ignition>	<p>0 = Postpone automatic reset when &lt;hours&gt; threshold is reached if device is in 'ignition on' state until mode transitions to 'ignition off' state.</p> <p>1 = Ignore ignition state when implementing automatic reset logic.</p> <p>* Note: On devices that do not support ignition detection, this parameter value can only be set to 1.</p>
Notes	<p>* On devices that do not support ignition detection (such as the MT 1000), the only acceptable value for the &lt;ign rst inhibit&gt; and &lt;ignition&gt; parameters is 1.</p> <p>The time until the modem resets is an approximate value.</p> <p>If the device is being reset by ignition input (switched power), the frequency of the heartbeat reset can be set as high as 168 hours. If ignition resets are inhibited, the maximum heartbeat reset allowed is 24 hours.</p>

# AT\$OFF Power Off Command

AT\$OFF	Power Off Command
Command Function	This command allows the user to perform a software-controlled shutdown. The modem gracefully deregisters from the network before powering down so it may take a few seconds before current consumption decreases.
Syntax Query	N/A
Syntax	N/A
Write Syntax	N/A
Write Response	N/A
Read Syntax	N/A
Read Response	N/A
Execute Syntax	AT\$OFF
Execute Response	None, modem powers down
Unsolicited Response	N/A
Parameter Values	None
Notes	

# AT%SLEEP Select Level Of Sleep Mode

AT%SLEEP	Select level of sleep mode
Command Function	This command allows the user to select the level of sleep the modem will enter during periods of inactivity.
Syntax Query	AT%SLEEP=?
Syntax	%SLEEP: (0-4) OK
Write Syntax	AT%SLEEP=<mode>
Write Response	OK
Read Syntax	AT%SLEEP?
Read Response	%SLEEP: <mode> OK
Execute Syntax	N/A
Execute Response	N/A
Unsolicited Response	N/A

AT%SLEEP	Select level of sleep mode
Parameter Values	
< mode >	<p>0 = no sleep</p> <p>1= Small</p> <p>2 = Big</p> <p>3 = Big + Deep</p> <p>4 = Small+ Big +Deep</p> <p>No Sleep: all sections of the modem will remained powered on and ready for use.</p> <p>Small: All functions are active and perform normally. Some peripherals are in sleepstate.</p> <p>Big: The radio and peripherals are asleep. The UART is awake and able to receivedata on the serial port. The radio will wakeup periodically check for pages form thewireless Network.</p> <p>Deep: only the 32 MHZ clock and a small portion of the DSP are running, all othersections of the module are in a sleep state. The UART is asleep, but will wakeup with the first character received (this character will be lost). The modem willwakeup periodically to check for incoming pages form the wireless carrier.</p>
Notes	<p>If %SLEEP &lt;mode&gt; of 3 or 4 is selected, and the modem has entered Deep sleep, the UART will miss the first character that is sent over the serial port. This first character will wake up the UART and subsequent characters will be accepted by the UART. Default setting is 2</p>



# Event Tables

---

# Event Engine

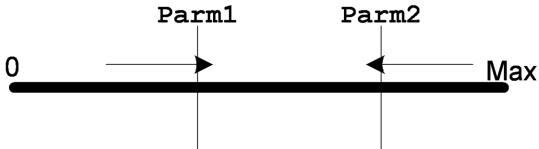
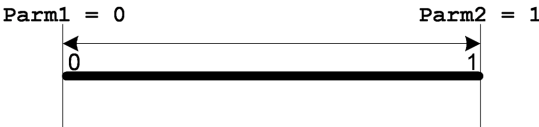
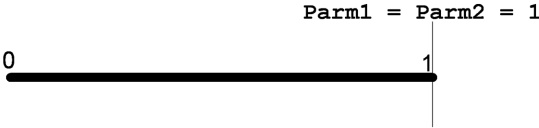
The Novatel Wireless Event Engine is a proprietary state machine that can be initiated by one or many triggering input events (which can be logically “ANDed” and/or “ORed”) to generate one or many pre-defined outputs. The Event Engine allows simple, fast development and deployment into end solutions without the need of proprietary development platforms or embedded applications.


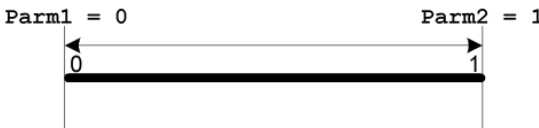
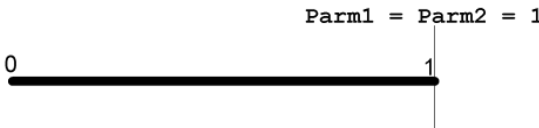
## Event Category

The <event category> parameter defines the actual Input or Output Event number and their valid range for <parm1> and <parm2>.

## Event Type

The <event type> parameter defines the type of event: Input or Output. An Input event can be defined as: Transition, Occurrence, or Input. The Output event is executed when input event conditions are met.

Value	Type of event	Description
0	Transition Trigger	<p>A transition Trigger is defined as an input condition, defined by &lt;event category&gt;, whose value was previously &lt;parm1&gt; or less is now greater than &lt;parm1&gt; and less than &lt;parm2&gt; or was greater or equal to &lt;parm2&gt; is now less than &lt;parm2&gt; but greater than &lt;parm1&gt;. The output event would be executed when an input &lt;event category&gt; requirements are satisfied or transition to the value set by &lt;parm1&gt; and &lt;parm2&gt; when they are equal. &lt;parm1&gt; should be the min value and &lt;parm2&gt; should be the max value.</p> <p><b>Example 1:</b></p>  <p>An output event will be executed when the value of an input event exceeds &lt;Parm1&gt; (previously it was &lt;Parm1&gt; or less) or decreases to a value less than &lt;Parm2&gt; (previously it was &lt;Parm2&gt; or greater).</p> <p><b>Example 2:</b></p>  <p>An output event will be executed when the value of an input event is 0 (previously it was anything else but 0) and &lt;Parm1&gt; along with &lt;Parm2&gt; is set to 0.</p>
		<p><b>Example 3:</b></p>  <p>An output event will be executed when the value of an input event is 1 (previously it was anything else but 1) and &lt;Parm1&gt; along with &lt;Parm2&gt; is set to 1.</p>
1	Occurrence Trigger	<p>An Occurrence Trigger is defined as an input condition, defined by &lt;event category&gt;, whose current value is greater than or equal to &lt;parm1&gt; and less than or equal to &lt;parm2&gt;.</p>

Value	Type of event	Description
		<p>The output event would be executed when an input &lt;event category&gt; requirements are satisfied or transition to the value set by &lt;parm1&gt; and &lt;parm2&gt; when they are equal. &lt;parm1&gt; should be the min value and &lt;parm2&gt; should be the max value.</p> <p><b>Example 1:</b></p>  <p>An output event will be executed when the current value of an input event is between &lt;Parm1&gt; and &lt;Parm2&gt; including boundary conditions.</p> <p><b>Example 2:</b></p>  <p>An output event will be executed when the value of the input event changes from 0 to 1 or vice-versa.</p>
		<p><b>Example 3:</b></p>  <p>An output event will be executed when the value of the input event is 1 and &lt;Parm1&gt; along with &lt;Parm2&gt; is set to 1.</p>
2	Input Trigger	<p>An Input Trigger is defined as an input condition, defined by &lt;event category&gt;, that should be used as a logical AND condition to another input condition defined as Transition Trigger or an Occurrence Trigger.</p> <p>An Output event is not triggered when Input Trigger condition is valid. The input event, defined as Input Trigger, is valid when within the event range defined by &lt;parm1&gt; and &lt;parm2&gt; or when &lt;parm1&gt; and &lt;parm2&gt; are equal.</p>

Value	Type of event	Description
3	Output	<p>An Output event is executed when all input event conditions (defined as Transition Trigger, Occurrence Trigger, or Input Trigger) for that particular &lt;event group&gt; are met.</p> <p><i>Output Event Note: ASCII messages sent by the Event Engine have a maximum length of 250 characters (not including the optional header). Any ASCII messages exceeding this length are truncated.</i></p>

## Input Event Table

The following table defines the values for <event category>, <parm1> and <parm2> parameter for input events defined as a Transition Trigger, Occurrence Trigger, or Input Trigger.

Event Category	Parm1	Parm2	Description
0 - 2	N/A	N/A	Reserved GPIO4 - General purpose Input/Output #4 - GPS LED
3	0 or 1	0 or 1	0 = Low 1 = High
4 - 6	N/A	N/A	Reserved GPIO8 - General Purpose Input/Output #8 - GSM LED
7	0 or 1	0 or 1	0 = Low 1 = High
8	1	1	Modem power up indication
9	0 to 5	0 to 5	Modem registration (see AT+CREG command description for registration status information)
10	0 to 8	0 to 8	Modem Data Network registration (see AT%CGREG command description for Data Network registration status information) Receipt of IP address.
11	0 or 1	0 or 1	0 = No IP address 1 = Valid IP address obtained
12	1	1	Timer 1 (set by AT\$EVTIM1)
13	1	1	Timer 2 (set by AT\$EVTIM2)
14	1	1	Timer 3 (set by AT\$EVTIM3)

Event Category	Parm1	Parm2	Description
15	1	1	Timer 4 (set by AT\$EVTIM4) GPS Distance (unit of measurement is: meters)
16	0 to 1000000	1000000	This input event has been replaced by the AT\$ODOMETER (input category 70) or AT\$TODOM (input Category 91) input events for more accurate data on newer devices and scripts. The event remains valid for legacy applications.
17	0 to 250	250	Current Velocity (unit of measurement is: Knots)
18 - 20	N/A	N/A	Reserved Geo Fence #1. See AT\$GEOFNC command for details on setting a circular geo-fence
21	0 or 1	0 or 1	0 = Leaving Geofence area 1 = Entering Geofence area
22	0 or 1	0 or 1	Geo Fence #2
23	0 or 1	0 or 1	Geo Fence #3
24	0 or 1	0 or 1	Geo Fence #4
25	0 or 1	0 or 1	Geo Fence #5 MT Power Save Event
26	0 or 1	0 or 1	0 = Exit Power Save Mode 1 = Enter Power Save Mode GPS Status
27	0 or 1	0 or 1	0 = Invalid GPS data 1 = Valid GPS data
28	1	1	**RTC Alarm Input
29	0 to 1000000	1000000	Invalid GPS data for a period of time (unit of measurement is: seconds)
30	0 to 1000000	1000000	Unit staying Idle in one place (unit of measurement is: seconds) Geo Fence #6. See AT\$GEOFNC command for details on setting a circular geo-fence
31	0 or 1	0 or 1	0 = Leaving Geofence area 1 = Entering Geofence area
32	0 or 1	0 or 1	Geo Fence #7
33	0 or 1	0 or 1	Geo Fence #8
34	0 or 1	0 or 1	Geo Fence #9
35	0 or 1	0 or 1	Geo Fence #10

Event Category	Parm1	Parm2	Description
36	0 or 1	0 or 1	Geo Fence #11
37	0 or 1	0 or 1	Geo Fence #12
38	0 or 1	0 or 1	Geo Fence #13
39	0 or 1	0 or 1	Geo Fence #14
40	0 or 1	0 or 1	Geo Fence #15
41	0 or 1	0 or 1	Geo Fence #16
42	0 or 1	0 or 1	Geo Fence #17
43	0 or 1	0 or 1	Geo Fence #18
44	0 or 1	0 or 1	Geo Fence #19
45	0 or 1	0 or 1	Geo Fence #20
46	0 or 1	0 or 1	Geo Fence #21
47	0 or 1	0 or 1	Geo Fence #22
48	0 or 1	0 or 1	Geo Fence #23
49	0 or 1	0 or 1	Geo Fence #24
50	0 or 1	0 or 1	Geo Fence #25
51	0	0	<p>**Input Event Counter. This event will occur when a counter reaches the maximum number of a selected Input event count.</p> <p>New SMS indication.</p>
52	0 or 1	0 or 1	<p>0 = SMS message read</p> <p>1 = New SMS message received</p>
53	0 to -1	0 to -1	<p>Current Input Event Counter count that can be used as an AND condition with other input events</p> <p>Has the user programmed any geo-fence? Normally this can be found by sending AT\$GEOFNC? command and verifying it manually based on the response sent by the device</p>
54	0 or 1	0 or 1	<p>0 = geo-fence does not exist</p> <p>1 = at least one geo fence was created</p>
55 - 56	N/A	N/A	Reserved
57	0-2147483647	0-2147483647	Messages to be sent Over-The Air (OTA) Exist
58 - 59	N/A	N/A	Reserved
60	0 - 9999	0 - 9999	Number of Unsent Messages (\$msglogrd count)
61	0 - 100	0 - 100	Memory full percentage (\$msglogrd)
62	0 or 1	0 or 1	<p>Motion Status</p> <p>0 = Stopped</p>

Event Category	Parm1	Parm2	Description
			1 = Moving
			Power Source
63	0 or 1	0 or 1	0 = main power not applied 1 = main power applied
64-65	N/A	N/A	Reserved
66	1	1	Timer 5 (set by AT\$EVTIM5)
67	1	1	Timer 6 (set by AT\$EVTIM6)
68	1	1	Timer 7 (set by AT\$EVTIM7)
69	1	1	Timer 8 (set by AT\$EVTIM8)
70	0- 214748364- 7	0- 214748364- 7	Current \$ODOMETER value
71	N/A	N/A	Reserved
			GPS Overspeed
72	0-1	0-1	0 = A GPS overspeed interval has ended 1 = A GPS overspeed interval has begun
73 - 90	N/A	N/A	Reserved
91	0	214748364- 7	Trip odometer (distance in meters)
92 - 99	N/A	N/A	Reserved
100	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 0
101	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 1
102	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 2
103	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 3



Event Category	Parm1	Parm2	Description
104	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 4
105	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 5
106	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 6
107	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 7
108	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 8
109	- 214748364- 8 to 214748364- 7	- 214748364- 8 to 214748364- 7	User variable 9
110 - 118	N/A	N/A	Reserved
119	0 or 1	0 or 1	0= An alternate GPS overspeed interval has ended 1= An alternate GPS overspeed interval has begun
120	0 or 1	0 or 1	Polygon Geofence #0. See \$ATPLYFN# for details on setting a polygonal geofence. 0 = Leaving Geofence area 1 = Entering Geofence area
121	0 or 1	0 or 1	Polygon Geofence #1
122	0 or 1	0 or 1	Polygon Geofence #2
123	0 or 1	0 or 1	Polygon Geofence #3
124	0 or 1	0 or 1	Polygon Geofence #4
125	0 or 1	0 or 1	Polygon Geofence #5

Event Category	Parm1	Parm2	Description
126	0 or 1	0 or 1	Polygon Geofence #6
127	0 or 1	0 or 1	Polygon Geofence #7
128	0 or 1	0 or 1	Polygon Geofence #8
129	0 or 1	0 or 1	Polygon Geofence #9
130	0 - 214748364 7	0 - 214748364 7	Bit-mapped value indicating if/why GPS data is being invalidated via \$GPSQUAL filtering:
			0 = data is good
			1 = delay interval in progress due to earlier detection of bad uncertainty, heading change, speed, distance and/or acceleration
			2 = velocity fix flag invalid
			4 = uncertainty value > threshold
			8 = \$GPGSA fix type not 3D
			16 = \$GPGSA HDOP < threshold
			32 = GPS date invalid
			64 = GPS time interval suspect
			128 = speed > threshold
131	N/A	N/A	256 = acceleration over one sec > threshold
			512 = heading change over one sec > threshold
			1024 = # of satellites being tracked < threshold
			2048 = distance over one sec > threshold
			Reserved
132	0 or 1	0 or 1	Polygon Geofence #10
133	0 or 1	0 or 1	Polygon Geofence #11
134	0 or 1	0 or 1	Polygon Geofence #12
135	0 or 1	0 or 1	Polygon Geofence #13
136	0 or 1	0 or 1	Polygon Geofence #14
137	0 or 1	0 or 1	Polygon Geofence #15
138	0 or 1	0 or 1	Polygon Geofence #16
139	0 or 1	0 or 1	Polygon Geofence #17
140	0 or 1	0 or 1	Polygon Geofence #18
141	0 or 1	0 or 1	Polygon Geofence #19
142	0 or 1	0 or 1	Polygon Geofence #20
143	0 or 1	0 or 1	Polygon Geofence #21
144	0 or 1	0 or 1	Polygon Geofence #22

Event Category	Parm1	Parm2	Description
145	0 or 1	0 or 1	Polygon Geofence #23
146	0 or 1	0 or 1	Polygon Geofence #24
147	1	1	This event is triggered at the completion of a FOTA upgrade after the modem has rebooted with the new firmware (old firmware if the upgrade failed).
148 - 159	N/A	N/A	Reserved
160			Accelerometer direction calibration complete
			The following Bit mask were added to Message 2 to enable sending OBD2 information:
			0x00000100 - Accelerometer X, Y, Z values
			ASCII - 3 values of 5 digits each, values are milliGs for X, Y, Z
			Binary - 3 values of 2 bytes each (signed), values are milliGs for X, Y, Z
			OBDII Acceleration Threshold 1
161	0 or 1	0 or 1	0 = Cleared 1 = Set
162	0 or 1	0 or 1	OBDII Acceleration Threshold 2
163	0 or 1	0 or 1	OBDII Acceleration Threshold 3
164	0 or 1	0 or 1	OBDII Deceleration Threshold 1
165	0 or 1	0 or 1	OBDII Deceleration Threshold 2
166	0 or 1	0 or 1	OBDII Deceleration Threshold 3
167	0 or 1	0 or 1	OBDII Engine Speed Threshold 1
168	0 or 1	0 or 1	OBDII Engine Speed Threshold 2
169	0 or 1	0 or 1	OBDII Engine Speed Threshold 3
170	0 or 1	0 or 1	OBDII Low Fuel Indication
171	0 or 1	0 or 1	OBDII Engine Idle
172	0 or 1	0 or 1	OBDII Speed Threshold 1
173	0 or 1	0 or 1	OBDII Speed Threshold 2
174	0 or 1	0 or 1	OBDII Speed Threshold 3
175	0 or 1	0 or 1	OBDII Low Battery Indication
176	0 or 1	0 or 1	OBDII Malfunction Indicator Light
177	0 or 1	0 or 1	OBDII Low Power Mode (Modem Off)
178	0 or 1	0 or 1	OBDII Ignition
183	1	1	Timer 9 (set by AT\$EVTIM9)
184	1	1	Timer 10 (set by AT\$EVTIM10)
185	0 or 1	0 or 1	OBD Protocol Discovered

Event Category	Parm1	Parm2	Description
186	0 or 1	0 or 1	OBD Auto-discovery time out
187-191	N/A	N/A	Reserved
	1 to 2147483647	1 to 2147483647	OBD Trip Odometer (distance in meters)
192			The event will fire when the trip odometer is in between the settings of param 1 and param 2.
193-197	NA	NA	Reserved
198	0 or 1	0 or 1	Excessive Acceleration
199	0 or 1	0 or 1	Excessive Deceleration
200	0 or 1	0 to 43200	Triggered when RTC reaches the target check in time (set by AT\$CHKIN)
201	0 or 1	0 or 1	Engine Coolant Temperature above Threshold

## Output Event Table

The below table defines the values for <event category>, <parm1> and <parm2> parameter for output events defined as Output.

Event Category	Parm1	Parm2	Description
0 - 10	N/A	N/A	Reserved
11	0	0	Set GPIO #4 configured as Output to Low (0)
12 - 14	N/A	N/A	Reserved
15	0	0	Set GPIO #8 configured as Output to Low (0)
16 - 18	N/A	N/A	Reserved
19	0	0	Set GPIO #4 configured as Output to High (1)
20 - 22	0	0	Reserved
23	0	0	Set GPIO #8 configured as Output to High (1)
24 - 26	N/A	N/A	Reserved
27	0	0	Toggle GPIO #4 configured as Output
28 - 30	N/A	N/A	Reserved
31	0	0	Toggle GPIO #8 configured as Output
32 - 34	N/A	N/A	Reserved
35	<a href="#">See GPIO Flash Table</a>		Flash GPIO #4 configured as Output (GPS LED)
36 - 38	N/A	N/A	Reserved

Event Category	Parm1	Parm2	Description
39	<a href="#">See GPIO Flash Table</a>		Flash GPIO #8 configured as Output (GSM LED)
40			Generate and transmit one UDP Message to first IP address listed in \$FRIEND command and port number listed in \$UDPAPI command based on Parm1 and Parm2 values
41	0 to 214783647	<a href="#">See Bit-Field Table</a>	Generate and transmit a UDP message with Acknowledge. This message is controlled by \$ACKTM command for number of retries sent. This message has to be acknowledged to avoid sending of retries.
42			Generate and transmit one UDP Message to all IP address listed in \$FRIEND command and port number listed in \$UDPAPI command based on Parm1 and Parm2 values
			Resets the timer (Timer #1 - Timer #8) specified by Parm1 to the time (in seconds) specified by Parm2. Parm2, when set to 0, resets the timer to the time last set by \$EVTIMx command or previous output event 43 execution.
43	1 - 8	0	A value other than 0 would set the timer to expire at the new specified interval (e.g. xx,3,43,1,180 would set timer 1 to expire in 180 seconds). When used with a value other than 0, this is equivalent to invoking \$EVTIMx directly from the event engine and subsequent AT&F commands will save the new value to flash.
44	1 - 25	0	Execute AT command stored at index number of the \$STOATEV command. Parm1 identifies the index number.
45	0 to 2147483647	<a href="#">See Bit-Field Table</a>	Sends data over SMS to All SMS destination addresses configured via \$SMSDA command. (For select \$SMSDA entries, see event categories 54-58)
46	N/A	N/A	Reserved
47	0	0 to -1	Input Event Counter
48	0	0 to -1	Input Event Counter reset to value stated by parm2
49	1 - 25	0 - 1000000	Set geo-fence specified by parm1 to current latitude & longitude with radius specified by parm2
50	0 - 57	0 to -1	Emulate AT\$EVTEST command via event engine. Parm1 is the input event number while Parm2 is the value to emulate for the input event
51	N/A	N/A	Reserved

Event Cat-egory	Parm1	Parm2	Description
52	0 to -1	<a href="#">See Bit-Field Table</a>	Generate and transmit one TCP/IP Message to IP address & port number listed by \$FRIEND command based on Parm1 and Parm2 values  Sets periodic RTC alarm in minutes, hours, days, or months.  Parm1 indicates the frequency with which to generate the message.  Parm2 indicates the unit with which to generate the message.
53	0 - 99	0 - 8	For example:  Parm1 Parm2 Result  11 RTC Alarm occurs every minute 12 RTC Alarm occurs every hour 14 RTC Alarm occurs every day 18 RTC Alarm occurs every month
54	0 to 214748364-7		Sends data over SMS to the first indexed SMS destination address configured via \$SMSDA command.
55	0 to 214748364-7		Sends data over SMS to the second indexed SMS destination address configured via \$SMSDA command.
56	0 to 214748364-7	<a href="#">See Bit-Field Table</a>	Sends data over SMS to the third indexed SMS destination address configured via \$SMSDA command.
57	0 to 214748364-7		Sends data over SMS to the fourth indexed SMS destination address configured via \$SMSDA command.
58	0 to 214748364-7		Sends data over SMS to the fifth indexed SMS destination address configured via \$SMSDA command.
59	N/A	N/A	Reserved
60	0 to -1	<a href="#">See Bit-Field Table</a>	Generate and transmit message to main serial port based on Parm1 and Parm2 values in ASCII format only.
61 - 124	N/A	N/A	Reserved
125	0 - 9	- 214748364-8 to	Sets user variable indicated by parm1 to value of parm2 (for example, if parm1 is 7 and parm2 is 50, this output event would set user variable 7 to 50)

Event Cat- egory	Parm1	Parm2	Description
		214748364- 7	
126	0 - 9	- 214748364- 8 to 214748364- 7	Increments user variable indicated by parm1 by value of parm2 (for example, if parm1 is 4, parm2 is 100, and user variable 4 was 200 prior to this event, user variable 4 would be incremented to 300 by this event)
127	0 - 9	- 214748364- 8 to 214748364- 7	Decrements user variable indicated by parm1 by value of parm2 (for example, if parm1 is 9, parm2 is 10, and user variable 9 was 50 prior to this event, user variable 9 would be decremented to 40 by this event)
128	0 - 9	- 214748364- 8 to 214748364- 7	Copies value of a system variable into user variable indicated by parm1. Parm2 is used as an index to determine the system variable that will be copied (see User Variable Index Table).
129 - 146	N/A	N/A	Reserved
147	0	1	Save circular GPS AI2 buffer to /cst/ai2log in FFS (used to debug unexpected GPS behavior).
148	0 to 214783647	<a href="#">See Bit Field Table</a>	Generate and Transmit TCP message to IP address & port listed in \$FRIEND
149	0 to 214783647	<a href="#">See Bit Field Table</a>	Generate and Transmit UDP message to IP first address listed in \$FRIEND, port listed in \$UDPAPI
150	0 to 214783647	<a href="#">See Bit Field Table</a>	Generate and Transmit UDP message to IP address listed in \$FRIEND requiring ACK, port listed in \$UDPAPI
151	0 to 214783647	<a href="#">See Bit Field Table</a>	Generate and Transmit UDP message to IP all addresses listed in \$FRIEND requiring ACK, port listed in \$UDPAPI
152	0	0	Send UDP message to the check-in servers specified in the \$FRIEND list. This event sends a wakeup style message to every server in the friends list with usage=4.
153-200	N/A	N/A	Reserved
201	0 to 1	0 to 1	Main Power Removed 0 = main power applied 1 = main power not applied

# Bit Field Tables

## Bit-Field Table Selection

Use the table below to determine which of the four bit-field tables (0-3) to use for the Parm2 value.

Bit 31	Bit 30	Description
0	0	Table selector 0. Format message based on Parm2 values using Message Format Table 0 (legacy format)
0	1	Reserved
1	0	Table selector 2. Format message based on Parm2 values using Message Format Table 2 This table is for User Variables.
1	1	Reserved

## Bit-Field Table 0 – Legacy (0,0)

The Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

Bit	Description
Bit 0:	1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format
Bit 1:	1 = add parm1 data to UDP message (4 - bytes in Binary format, 11 - bytes of data in ASCII format) 0 = do not add parm1 data to outbound UDP message
Bit 2:	1 = add \$MDMID value (22 - bytes of ASCII data - irrespective of Bit- 0 setting) 0 = do not add \$MDMID value
Bit 3:	1 = add GPIO data and direction (2 bytes binary or 6 bytes ASCII) * 0 = do not add GPIO data *GPIO 8 data and direction will not be reported for this product. OBD ignition state will be reported in its place.
Bit 4:	Peak Engine Speed (LSB=1/4 RPM) The reported peak engine speed is the peak from the last time that the engine



speed exceeded the Excessive Engine Speed 1 threshold.

- Bit 5: 1 = add ADC-2 value (2 bytes binary or 5 bytes ASCII)  
0 = do not add ADC-2 value
- 1 = Message is stored in non-volatile memory until it can be sent, regardless of network status
- Bit 6: 0 = Code checks network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and message queue has few or no messages pending), the message is stored in non-volatile message queue until it can be sent. Otherwise, the message is deleted.
- Bit 7: 1 = add input <event category> number (1 - byte in binary format, 3 - bytes in ASCII format)  
0 = do not add input <event category> number
- Bit 8: 1 = add GPS data (3 - bytes of Date information in Binary format or up to 80 - bytes of \$GPGGA NMEA message if Bit-0 is set to 0)  
0 = do not add this particular field of GPS data
- 1 = add 1-byte of GPS STATUS information in Binary  
0 = do not add this particular field of GPS data
- Bit 9: NOTE: The values for the GPS status are:  
0 - no GPS fix  
1 - valid GPS fix  
9 - data is last valid GPS fix (only used if bit 19 is 1)
- Bit 10: 1 = add GPS data (3 - bytes of Latitude information in Binary format or up to 80 - bytes of \$GPGSA NMEA message if Bit-0 is set to 0)  
0 = do not add this particular field of GPS data
- Bit 11: 1 = add GPS data (4 - bytes of Longitude information in Binary format or up to 80 - bytes of \$GPGSA NMEA message if Bit-0 is set to 0)  
0 = do not add this particular field of GPS data
- Bit 12: 1 = add GPS data (2 - bytes of Velocity information in Binary format or up to 80 - bytes of \$GPRMC NMEA message if Bit-0 is set to 0)  
0 = do not add this particular field of GPS data
- Bit 13: 1 = add 2-bytes of GPS HEADING information in Binary  
0 = do not add this particular field of GPS data
- Bit 14: 1 = add GPS data (3 - bytes of Time information in Binary format or 0 bytes if Bit-0 is set to 0)  
0 = do not add this particular field of GPS data
- Bit 15: 1 = add GPS data (3 - bytes of Altitude information in Binary format or 0 bytes

- if Bit-0 is set to 0) 0 = do not add this particular field of GPS data  
1 = add GPS data (1 - byte of Number Of Satellites In View information in Binary format or 0 bytes if Bit-0 is set to 0)
- Bit 16: 0 = do not add this particular field of GPS data
- BIT 17: 1 = Disable sending message when in Low Power (\$PWRSAV) Mode  
0 = Enable sending message when in Low Power (\$PWRSAV) Mode
- Bit 18: 1 = send this OTA message via SMS when Data Network services is not available  
0 = send this OTA message via Data Network only  
1 = send Last Valid GPS data if current data is invalid  
0 = send current GPS data - valid or invalid
- Bits 19: NOTE: If the msg format is ASCII, this bit is 1 and the GPS fix is not valid, the GPS status field in the RMC message will be '9', indicating that the data is from the last valid GPS fix. If the msg format is binary, see bit 9.
- Bit 20: 1 = add GPS Odometer reading (4 - bytes of Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0)  
0 = do not add this particular field of GPS data
- Bit 21: 1 = add RTC time (6 - bytes of RTC time in Binary format or 13 - bytes if Bit-0 is set to 0)  
0 = do not add RTC time with GPS data
- Bit 22: 1 = Replace/append modem ID field with 10-byte modem ID (including one leading and one ending space character) if bit-0 is set to 0. Replace/append it with 8-bytes long modem ID value if bit-0 is set to 1 (no leading or ending space characters in binary mode). (NOTE: bit-22 setting overrides bit-2 setting)  
0 = send the modem ID as defined by Bit-2
- Bit 23: Reserved  
1 = add GPS overspeed data (6 - bytes of Overspeed information in Binary format or 6 to 18 - bytes if Bit-0 is set to 0).
- Bit 24: Binary format: xxyyzz: xx is speed specified by AT\$GPSOSI (unit: knots);  
yy is the maximum speed incurred during the interval (unit: knots, 1/10 knot accuracy);  
zz is the interval duration (unit: seconds);
- ASCII format: " x y z": space delineated, length of each field varies with its value

0 = do not add this particular field of GPS data

1= Add cell information as follows: (See GSM0000TN012 - Engineering Mode Manual for details of the %EM AT command)

Bit 25: If Binary format (Bit0=1) is selected, please refer to the "Bit 25 Binary Format" table  
If ASCII format (Bit0=0) is selected, please refer to the "Bit 25 Binary Format" table

0 = Do not add PCELL data

1 = add Alternate GPS overspeed data (6 - bytes of Overspeed information in Binary format or 6 to 18 - bytes if Bit-0 is set to 0).

Bit 26: Binary format: xxyyzz: xx is speed specified by AT\$GPSOSI (unit: knots);  
yy is the maximum speed incurred during the interval (unit: knots, 1/10 knot accuracy);  
zz is the interval duration (unit: seconds);

ASCII format: " x y z": space delineated, length of each field varies with its value

0 = do not add this particular field of GPS data

OBD2 Info (VIN, OBD protocol, Novatel Wireless Firmware version, RSSI)

1 = Add OBDII Data (VIN, OBD Protocol, PKG, RSSI)

0 = Do not add OBDII Data

Bit 27: Data Format:  
VIN  
Binary - 20 bytes (nul char padding at end of VIN)  
ASCII - a space delimiter followed by 20 chars for the VIN.  
The VIN is padded with leading spaces if less than 20 chars.

OBD Protocol

Binary - 1 byte

ASCII - a space delimiter followed by 3 digits for the protocol.

If the protocol is less than 3 digits, it will be padded with leading spaces.

FW Version

Binary - 4 bytes

ASCII - a space delimiter followed by 6 chars for the FW version

RSSI (received signal strength indicator)

Binary - 1 byte

ASCII - a space delimiter followed by 2 digits for the RSSI

1 = Add OBDII Malfunction Indicator Light (MIL) Data

0 = Do not add OBDII MIL Data

Bit 28:

Data Format:

Binary - 1 byte containing the number of MIL codes to follow. If 1 or more codes, each code is 2 additional bytes.

ASCII - a space delimiter followed by 2 digits containing the number of MIL codes. If 1 or more codes, the count is followed by a space and 4 hexadecimal chars for each code. Subsequent codes are separated by a comma.

1 = Add OBDII Trip Odometer Data

0 = Do not add OBDII Trip Odometer Data

BIT 29:

Data Format:

Binary - 4 bytes

ASCII - a space delimiter followed by 7 digits for the trip distance.

Bits 30 -  
31:

Reserved

## Bit-Field Table 2 – (1,0)

This table is for User Variables.

The Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

Bit	Description
Bit 0:	1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format
Bit 1:	1 = Add parm1 data to message (4 bytes in binary format, 11 bytes of data in ASCII format) 0 = do not add parm1 data to outbound message
Bit 2:	1 = add \$MDMID value (22 bytes of ASCII data irrespective of Bit 0 setting) 0 = do not add \$MDMID value
Bit 3 is least significant in the following description:	
000 = Do not add user variables to message.	
001 = Add user variables 0-9 to message, starting with user variable 0 (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format). Spaces are inserted between the different user variables.	
010 = Add only user variable 0 to message (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format).	
011 = Add only user variables 0-1 to message(4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format). Spaces are inserted between the different user variables.	
Bits 3-5:	100 = Add only user variables 0-2 to message (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format). Spaces are inserted between the different user variables.
	101 = Add only user variables 0-3 to message (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format). Spaces are inserted between the different user variables.
	110 = Add only user variables 0-5 to message (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format). Spaces are inserted between the different user variables.
	111 = Add only user variables 0-7 to message (4 bytes per user variable in binary format, 8 hex bytes per user variable in ASCII format). Spaces are inserted between the different user variables.
Bit 6:	1 =Message is stored in non-volatile memory until it can be sent, regardless of network status. 0 = Check network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and

message queue has few or no messages pending), the message is stored in the non-volatile message queue until it can be sent. Otherwise, the message is deleted.

1 = add input <event category> number (1 byte in binary format, 3 bytes in ASCII format)

Bit 7:

0 = do not add input <event category> number

Bits 8-14 Reserved

Bit 15 is the least significant in the following description:

000 = Do not add GPS data to message.

001 – Include GPS latitude, longitude, velocity and heading in message.

If Bit0 = 1 (binary), GPS data is added to the message in the following sequence:

Latitude = 3 bytes. For example, if \$GPRMC latitude is 3259.815430,N, then the three bytes would be 31BDA7 (3259815).

Longitude = 4 bytes. For example, if \$GPRMC longitude is 09642.854492,W, then the four bytes would be FF6CDC9A (-9642854).

Velocity = 2 bytes in tenths of knots. For example, if \$GPRMC velocity is 21.1 knots, then the two bytes would be 00D3 (211).

Heading = 2 bytes in tenths of degrees. For example, if \$GPRMC heading is 357.9 degrees, then the two bytes would be 0DF8 (3579).

Bits 15-17

If Bit0 = 0 (ASCII), GPS data is formatted as described above, then converted to ASCII HEX (effectively doubling the number of bytes required to contain the information) before being inserted into the message. There are no spaces between the various GPS data fields.

010 – If Bit0 = 0 (ASCII), include \$GPRMC sentence in message. If Bit0 = 1 (binary), include GPS date, latitude, longitude, velocity, heading and time in message in the following sequence:

Date = 3 bytes. For example, if \$GPRMC date is 290611, then the three bytes would be 046F33 (290611).

Latitude = 3 bytes. For example, if \$GPRMC latitude is 3259.815430,N, then the three bytes would be 31BDA7 (3259815).

Longitude = 4 bytes. For example, if \$GPRMC longitude is 09642.854492,W, then the four bytes would be FF6CDC9A (-9642854).

Velocity = 2 bytes in tenths of knots. For example, if \$GPRMC velocity is 21.1 knots, then the two bytes would be 00D3 (211).

Heading = 2 bytes in tenths of degrees. For example, if \$GPRMC heading is 357.9 degrees, then the two bytes would be 0DF8 (3579).

Time = 3 bytes. For example, if \$GPRMC time is 162916.00, then the three bytes would be 027c64 (162916).

011 – Include GPS date, latitude, longitude, velocity and time in message.

If Bit0 = 1 (binary), GPS data is added to the message in the following sequence:

Date = 3 bytes. For example, if \$GPRMC date is 290611, then the three bytes would be 046F33 (290611).

Latitude = 3 bytes. For example, if \$GPRMC latitude is 3259.815430,N, then the three bytes would be 31BDA7 (3259815).

Longitude = 4 bytes. For example, if \$GPRMC longitude is 09642.854492,W, then the four bytes would be FF6CDC9A (-9642854).

Velocity = 2 bytes in tenths of knots. For example, if \$GPRMC velocity is 21.1 knots, then the two bytes would be 00D3 (211).

Time = 3 bytes. For example, if \$GPRMC time is 162916.00, then the three bytes would be 027c64 (162916).

If Bit0 = 0 (ASCII), GPS data is formatted as described above, then converted to ASCII HEX (effectively doubling the number of bytes required to contain the information) before being inserted into the message. There are no spaces between the various GPS data fields.

100 – Include GPS date, latitude, longitude, velocity, heading, date, time, altitude and number of satellites used in solution in message.

If Bit0 = 1 (binary), GPS data is added to the message in the following sequence:

Date = 3 bytes. For example, if \$GPRMC date is 290611, then the three bytes would be 046F33 (290611).

Latitude = 3 bytes. For example, if \$GPRMC latitude is 3259.815430,N, then the three bytes would be 31BDA7 (3259815).

Longitude = 4 bytes. For example, if \$GPRMC longitude is 09642.854492,W, then the four bytes would be FF6CDC9A (-9642854).

Velocity = 2 bytes in tenths of knots. For example, if \$GPRMC velocity is 21.1 knots, then the two bytes would be 00D3 (211).

Heading = 2 bytes in tenths of degrees. For example, if \$GPRMC heading is 357.9 degrees, then the two bytes would be 0DF8 (3579).

Time = 3 bytes. For example, if \$GPRMC time is 162916.00, then the three bytes would be 027c64 (162916).

Altitude = 3 bytes. For example, if \$GPGLL altitude is 177.7 meters, then the three bytes would be 0000B1 (177).

Number of satellites = 1 byte. For example, if \$GPGLL number of satellites being tracked is 10, then this byte would be 0A (10).

If Bit0 = 0 (ASCII), GPS data is formatted as described above, then converted to ASCII HEX (effectively doubling the number of bytes required to contain the information) before being inserted into the message. There are no spaces between the various GPS data fields.

101 – Reserved for future expansion (no assignments currently performed).

110 – Reserved for future expansion (no assignments currently performed).

111 – Reserved for future expansion (no assignments currently performed).

1 = send this OTA message via SMS when Data Network services is not available

Bit 18:  
0 = send this OTA message via Data Network only

Bits 19-  
20      Reserved

1 = add RTC time (6 bytes of RTC time in binary format or 13 bytes in ASCII format)

Bit 21:  
0 = do not add RTC time

1 = Replace/append modem ID field with 10-byte modem ID (including one leading and one ending space character) in ASCII format. Replace/append modem ID with 8 bytes long modem ID value in binary format (no leading or ending space characters in binary format).

Bit 22:  
(NOTE: Bit 22 setting overrides Bit 2 setting) 0 = send the modem ID as defined by Bit 2

0 = send the modem ID as defined by Bit 2

Excessive Acceleration Data

ASCII format: 14 bytes HHMMSS,LLL,MMM

HHMMSS - time at start of event

LLL - length of event in seconds

Bit 23:  
MMM - max acceleration/deceleration

Binary format: 5 bytes -

3 bytes (HMS) time at start of event

1 byte length of event in seconds



1 byte max acceleration/deceleration

Excessive Deceleration Data

ASCII format: 14 bytes HHMMSS,LLL,MMM

HHMMSS - time at start of event

LLL - length of event in seconds

Bit 24: MMM - max acceleration/deceleration

Binary format: 5 bytes -

3 bytes (HMS) time at start of event

1 byte length of event in seconds

1 byte max acceleration/deceleration

## User Variable Index Table

This table is used only with Output Event 128

Parm2      System Variable Copied to User Variable

(For example, AT\$EVENT=99,3,128,3,9 would copy value of Input Event 9 (GSM registration status) into User Variable 3).

NOTE: All the following system variables are not supported by all devices. Ensure your device supports the system variable before attempting to use it with user variables.

For Serving Cell and Neighbor Cell values, see GSM0000GN012 – Engineering Mode Manual for details of the %EM command.

-500      Hardware ID

-499 to -474 Reserved

-473      Copies Neighbor Cell 5 signal strength. Equivalent to AT%EM=2,3

-472      Copies Neighbor Cell 5 absolute radio frequency channel number (ARFCN). Equivalent to AT%EM=2,3

-471      Copies Neighbor Cell 5 cell ID. Equivalent to AT%EM=2,3

-470      Copies Neighbor Cell 5 location area code. Equivalent to AT%EM=2,3

-469 to -464 Reserved

- 463 Copies Neighbor Cell 4 signal strength. Equivalent to AT%EM=2,3
- 462 Copies Neighbor Cell 4 absolute radio frequency channel number (ARFCN).  
Equivalent to AT%EM=2,3
- 461 Copies Neighbor Cell 4 cell ID. Equivalent to AT%EM=2,3
- 460 Copies Neighbor Cell 4 location area code. Equivalent to AT%EM=2,3
- 459 to -454 Reserved
- 453 Copies Neighbor Cell 3 signal strength. Equivalent to AT%EM=2,3
- 452 Copies Neighbor Cell 3 absolute radio frequency channel number (ARFCN).  
Equivalent to AT%EM=2,3
- 451 Copies Neighbor Cell 3 cell ID. Equivalent to AT%EM=2,3
- 450 Copies Neighbor Cell 3 location area code. Equivalent to AT%EM=2,3
- 449 to -444 Reserved
- 443 Copies Neighbor Cell 2 signal strength. Equivalent to AT%EM=2,3
- 442 Copies Neighbor Cell 2 absolute radio frequency channel number (ARFCN).  
Equivalent to AT%EM=2,3
- 441 Copies Neighbor Cell 2 cell ID. Equivalent to AT%EM=2,3
- 440 Copies Neighbor Cell 2 location area code. Equivalent to AT%EM=2,3
- 439 to -434 Reserved
- 433 Copies Neighbor Cell 1 signal strength. Equivalent to AT%EM=2,3
- 432 Copies Neighbor Cell 1 absolute radio frequency channel number (ARFCN).  
Equivalent to AT%EM=2,3
- 431 Copies Neighbor Cell 1 cell ID. Equivalent to AT%EM=2,3
- 430 Copies Neighbor Cell 1 location area code. Equivalent to AT%EM=2,3
- 429 to -424 Reserved
- 423 Copies Neighbor Cell 0 signal strength. Equivalent to AT%EM=2,3
- 422 Copies Neighbor Cell 0 absolute radio frequency channel number (ARFCN).  
Equivalent to AT%EM=2,3
- 421 Copies Neighbor Cell 0 cell ID. Equivalent to AT%EM=2,3
- 420 Copies Neighbor Cell 0 location area code. Equivalent to AT%EM=2,3
- 419 to -407 Reserved
- 406 Copies Serving Cell timing advance. Equivalent to AT%EM=2,1
- 405 Copies Serving Cell signal strength. Equivalent to AT%EM=2,1

-404	Copies Serving Cell absolute radio frequency channel number (ARFCN). Equivalent to AT%EM=2,1
-403	Copies Serving Cell cell ID. Equivalent to AT%EM=2,1
-402	Copies Serving Cell location area code. Equivalent to AT%EM=2,1
-401	Copies Serving Cell MNC (0x00MMNNCC, where MM, NN, and CC are the hex values of the ASCII representations of the MNC). Equivalent to AT%EM=2,4
-400	Copies Serving Cell MCC (0x00MMCCcc, where MM, CC, and cc are the hex values of the ASCII representations of the MCC) . Equivalent to AT%EM=2,4
-399 to -303	Reserved
-302	Copies Software version (for example, if version is 1.1.1.8, value would be 0x00001118)
-301	Copies product ID
-300	Copies \$usrval value
-299 to -51	Reserved
-50	Engine Coolant Temperature
-49 to -42	Reserved
	Copies GPIO value status where LSB represents GPIO0. For example: 0x00165432 2=binary 0010 (so GPIO1=1; GPIO0, GPIO2, GPIO3=0) 3=binary 0011 (so GPIO4,GPIO5=1; GPIO6,GPIO7=0) 4=binary 0100 (so GPIO10=1; GPIO8,GPIO9,GPIO11=0) 5=binary 0101 (so GPIO12,GPIO14=1; GPIO13,GPIO15=0) 6=binary 0110 (so GPIO17,GPIO18=1; GPIO16,GPIO19=0) 1=binary 0001 (so GPIO20=1) Note: GPIO numbers >8 are not supported on the MT-Gμ
	Copies GPIO direction status where LSB represents GPIO0 (1=input,0-output). For example: 0x00165432 2=binary 0010 (so GPIO1=input; GPIO0, GPIO2, GPIO3= outputs) 3=binary 0011 (so GPIO4,GPIO5=inputs; GPIO6,GPIO7=outputs) 4=binary 0100 (so GPIO10=input; GPIO8,GPIO9,GPIO11=outputs) 5=binary 0101 (so GPIO12,GPIO14=inputs; GPIO13,GPIO15=outputs) 6=binary 0110 (so GPIO17,GPIO18=inputs; GPIO16,GPIO19=outputs)
-40	

1=binary 0001 (so GPIO20=input)

Note: GPIO numbers >8 are not supported on the MT-G $\mu$

-39 to -22	Reserved
-21	Copies current RTC time: 0x00HHMMSS where HH = hour (0-23), MM = minute (0-59), SS = second (0-59)
-20	Copies current RTC date: 0x00YYMMDD where YY = last two digits of year (00-99), MM = month (1-12), DD = day of month (1-31)
-19 to -9	Reserved
-8	Copies current count of event timer 8 in seconds (equivalent to \$EVTIMQRY=8)
-7	Copies current count of event timer 7 in seconds (equivalent to \$EVTIMQRY=7)
-6	Copies current count of event timer 6 in seconds (equivalent to \$EVTIMQRY=6)
-5	Copies current count of event timer 5 in seconds (equivalent to \$EVTIMQRY=5)
-4	Copies current count of event timer 4 in seconds (equivalent to \$EVTIMQRY=4)
-3	Copies current count of event timer 3 in seconds (equivalent to \$EVTIMQRY=3)
-2	Copies current count of event timer 2 in seconds (equivalent to \$EVTIMQRY=2)
-1	Copies current count of event timer 1 in seconds (equivalent to \$EVTIMQRY=1) Copies value of Input Event 0 (GPIO1).
0	0 = Low 1 = High Copies value of Input Event 1 (GPIO2).
1	0 = Low 1 = High Copies value of Input Event 2 (GPIO3).
2	0 = Low 1 = High Copies value of Input Event 3 (GPIO4).
3	0 = Low 1 = High Copies value of Input Event 4 (GPIO5).
4	0 = Low 1 = High
5	Copies value of Input Event 5 (GPIO6).

	0 = Low
	1 = High
	Copies value of Input Event 6 (GPIO7).
6	0 = Low
	1 = High
	Copies value of Input Event 7 (GPIO8).
7	0 = Low
	1 = High
	Copies value of Input Event 8 (modem power up indication).
8	Always 1.
	Copies value of Input Event 9 (modem registration).
9	See AT+CREG command description for registration status information.
	Copies value of Input Event 10 (modem Data Network registration).
10	See AT%CGREG command for Data Network registration status information.
	Copies value of Input Event 11 (Receipt of IP address).
11	0 = No IP address
	1 = Valid IP address obtained
	Copies value of Input Event 12 (Timer 1 status).
12	0 = Timer not expired
	1 = Timer expired
	Copies value of Input Event 13 (Timer 2 status).
13	0 = Timer not expired
	1 = Timer expired
	Copies value of Input Event 14 (Timer 3 status).
14	0 = Timer not expired
	1 = Timer expired
	Copies value of Input Event 15 (Timer 4 status).
15	0 = Timer not expired
	1 = Timer expired
16	Copies value of Input Event 16 (GPS distance in meters)
17	Copies value of Input Event 17 (Maximum velocity in knots)

18	Copies value of Input Event 18 (ADC1 status)
19 to 20	Reserved
	Copies value of Input Event 21 (Geofence #1)
21	0 = Leaving geofence area 1 = Entering geofence area
	Copies value of Input Event 22 (Geofence #2)
22	0 = Leaving geofence area 1 = Entering geofence area
	Copies value of Input Event 23 (Geofence #3)
23	0 = Leaving geofence area 1 = Entering geofence area
	Copies value of Input Event 24 (Geofence #4)
24	0 = Leaving geofence area 1 = Entering geofence area
	Copies value of Input Event 25 (Geofence #5)
25	0 = Leaving geofence area 1 = Entering geofence area
	Copies value of Input Event 26 (MT Power Save Event)
26	0 = Exit Power Save Mode 1 = Enter Power Save Mode
	Copies value of Input Event 27 (GPS status)
27	0 = Invalid GPS data 1 = Valid GPS data
28	Copies value of Input Event 28 (RTC Alarm Input)
29	Copies value of Input Event 29 (Invalid GPS data for a period of seconds)
30	Copies value of Input Event 30 (Unit staying Idle in one place for a period of seconds)
	Copies value of Input Event 31 (Geofence #6)
31	0 = Leaving geofence area 1 = Entering geofence area
	Copies value of Input Event 32 (Geofence #7)
32	0 = Leaving geofence area

1 = Entering geofence area  
 Copies value of Input Event 33 (Geofence #8)

33 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 34 (Geofence #9)

34 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 35 (Geofence #10)

35 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 36 (Geofence #11)

36 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 37 (Geofence #12)

37 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 38 (Geofence #13)

38 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 39 (Geofence #14)

39 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 40 (Geofence #15)

40 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 41 (Geofence #16)

41 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 42 (Geofence #17)

42 0 = Leaving geofence area  
 1 = Entering geofence area  
 Copies value of Input Event 43 (Geofence #18)

43

	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 44 (Geofence #19)
44	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 45 (Geofence #20)
45	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 46 (Geofence #21)
46	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 47 (Geofence #22)
47	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 48 (Geofence #23)
48	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 49 (Geofence #24)
49	0 = Leaving geofence area
	1 = Entering geofence area
	Copies value of Input Event 50 (Geofence #25)
50	0 = Leaving geofence area
	1 = Entering geofence area
51	Copies value of Input Event 51 (Input Event Counter)
	Copies value of Input Event 52 (New SMS indication)
52	0 = SMS message read
	1 = New SMS message received
53	Copies value of Input Event 53 (Current Input Event Counter count that can be used as an AND condition with other input events)
	Copies value of Input Event 54 (Does any geofence exist?)
54	0 = geofence does not exist
	1 = at least one geofence was created



55 to 64	Reserved
65	Copies value of Input Event 64 (Receipt of incoming call with Call Identifier matching one of the numbers configured via the \$EVCID command)
	Copies value of Input Event 66 (Timer 5 status).
66	0 = Timer not expired 1 = Timer expired
	Copies value of Input Event 67 (Timer 6 status).
67	0 = Timer not expired 1 = Timer expired
	Copies value of Input Event 68 (Timer 7 status).
68	0 = Timer not expired 1 = Timer expired
	Copies value of Input Event 69 (Timer 8 status).
69	0 = Timer not expired 1 = Timer expired
70	Copies value of Input Event 70 (Odometer in meters)
71	Reserved
	Copies value of Input Event 72 (GPS overspeed)
72	0 = interval has ended 1 = interval has begin
73 to 85	Reserved
86 to 99	Reserved
100	Copies value of Input Event 100 (User variable 0)
101	Copies value of Input Event 101 (User variable 1)
102	Copies value of Input Event 102 (User variable 2)
103	Copies value of Input Event 103 (User variable 3)
104	Copies value of Input Event 104 (User variable 4)
105	Copies value of Input Event 105 (User variable 5)
106	Copies value of Input Event 106 (User variable 6)
107	Copies value of Input Event 107 (User variable 7)
108	Copies value of Input Event 108 (User variable 8)
109	Copies value of Input Event 109 (User variable 9)

110-191	Reserved
192	Copies value of Input Event 192 (OBD Trip odometer in meters)

## GPIO Flash Table

Parm1

Bits 16 – 31 determine the low signal state while bits 0 – 15 determine the high signal state. A value of 0 for bits 16 – 31 indicates the GPIO will remain in low signal state for the same amount of time as the high signal state (50% duty cycle). The high or low states are measured in multiples of ¼ seconds. The toggle count is set by Parm2

Parm2

The flashing GPIO event will cause the GPIO output state to toggle at time 0 to the opposite state prior to starting the GPIO output flash event processing. This counts as toggle #1. An even number of toggle count will force a final state which is the same as the initial state. An odd number of toggle count will force the final state to be opposite of the initial GPIO output condition. 0 = toggle forever.



Warning:  
Flashing GPIO 3 on the MT 3000 will damage the device.

## OBD Bit Field Tables

### OBD BIT-FIELD TABLE SELECTION

Use the table below to determine which of the four bit-field tables (0-3) to use for the Parm2 value.

Bit-Field Table Selection

Bit 31 Bit 30 Description

0	0	Table selector 0. Format message based on Parm2 values using Message Format Table 0 (OBD format)
0	1	Reserved
1	0	Reserved
1	1	Reserved

### BIT-FIELD TABLE 0 – OBD (0,0)

The Parm2 value is obtained as a result of selecting individual bit-fields from the table below.

#### Bit-Field Table 0 – OBD (0,0)

Parm2	Description
Bit 0:	1 = send all data generated as a result of this table in Binary format 0 = send all data generated as a result of this table in ASCII format
Bit 1:	1 = add parm1 data to UDP message (4 – bytes in Binary format, 11 – bytes of data in ASCII format) 0 = do not add parm1 data to outbound UDP message
Bit 2:	1 = add \$MDMID value (22 – bytes of ASCII data – irrespective of Bit- 0 setting) 0 = do not add \$MDMID value
Bit 3:	Reserved Peak Engine Speed (LSB=1/4 RPM)
Bit 4:	The reported peak engine speed is the peak from the last time that the engine speed exceeded the Excessive Engine Speed 1 threshold.
Bit 5:	Reserved 1 =Message is stored in non-volatile memory until it can be sent, regardless of network status. 0 = Code checks network status before storing message in non-volatile memory. If it appears that the message can be sent out immediately (network status is clear and message queue has few or no messages pending), the message is stored in the non-volatile message queue until it can be sent. Otherwise, the message is deleted.
Bit 6:	1 = add input <event category> number (1 – byte in binary format, 3 – bytes in ASCII format) 0 = do not add input <event category> number
Bit 7:	1 = add GPS data (3 – bytes of Date information in Binary format or up to 80 – bytes of \$GPGGA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 8:	1 = add 1-byte of STATUS information in Binary 0 = do not add this particular field of GPS data
Bit 9:	1 = add GPS data (3 – bytes of Latitude information in Binary format or up to 80 – bytes of \$GPGSA NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 10:	1 = add GPS data (4 – bytes of Longitude information in Binary format or up to two 80 – bytes of \$GPGSV NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 11:	1 = add GPS data (4 – bytes of Longitude information in Binary format or up to two 80 – bytes of \$GPGSV NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data

Bit 12:	1 = add GPS data (2 - bytes of Velocity information in Binary format or up to 80 - bytes of \$GPRMC NMEA message if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 13:	1 = add 2-bytes of HEADING information in Binary 0 = do not add this particular field of GPS data
Bit 14:	1 = add GPS data (3 - bytes of Time information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 15:	1 = add GPS data (3 - bytes of Altitude information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 16:	1 = add GPS data (1 - byte of Number Of Satellites In View information in Binary format or 0 bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 17:	Reserved
Bit 18:	1 = send this OTA message via SMS when GPRS services is not available 0 = send this OTA message via GPRS only
Bit 19:	1 = send Last Valid GPS data if current data is invalid 0 = send current GPS data - valid or invalid
Bit 20:	1 = add Odometer reading (4 - bytes of Odometer information in Binary format or 11 - bytes if Bit-0 is set to 0) 0 = do not add this particular field of GPS data
Bit 21:	1 = add RTC time (6 - bytes of RTC time in Binary format or 13 - bytes if Bit-0 is set to 0) 0 = do not add RTC time with GPS data
Bit 22:	1 = Replace/append modem ID field with 10-byte modem ID (including one leading and one ending space character) if bit-0 is set to 0. Replace/append it with 8-bytes long modem ID value if bit-0 is set to 1 (no leading or ending space characters in binary mode.) (NOTE: bit-22 setting overrides bit-2 setting) 0 = Sent the modem ID as defined by Bit-2
Bit 23:	Reserved
Bit 24:	Reserved
Bit 25:	1 = add fuel level Reserved
Bit 26:	Reserved
Bit 27:	OB2 Info (VIN, OBD protocol, Novatel Wireless Firmware version, RSSI) 1 = Add OBDII Data (VIN, OBD Protocol, PKG, RSSI) 0 = Do not add OBDII Data Data Format:

VIN:

Binary - 20 bytes (null char padding at end of VIN)

ASCII - a space delimiter followed by 20 chars for the VIN. The VIN is padded with leading spaces if less than 20 chars.

OBD Protocol:

Binary - 1 byte

ASCII - a space delimiter followed by 3 digits for the protocol. If the protocol is less than 3 digits, it will be padded with leading spaces

FW Version:

Binary - 4 bytes

ASCII - a space delimiter followed by 6 chars for the FW version

RSSI (received signal strength indicator):

Binary - 1 byte

ASCII - a space delimiter followed by 2 digits for the RSSI

1 = Add OBDII Malfunction Indicator Light (MIL) Data

0 = Do not add OBDII MIL Data

Data Format:

Bit 28:

Binary - 1 byte containing the number of MIL codes to follow. If 1 or more codes, each code is 2 additional bytes.

ASCII - a space delimiter followed by 2 digits containing the number of MIL codes. If 1 or more codes, the count is followed by a space and 4 hexadecimal chars for each code. Subsequent codes are separated by a comma.

1 = Add OBDII Trip Odometer Data

0 = Do not add OBDII Trip Odometer Data

BIT 29:

Data Format: Binary - 4 bytes

ASCII - a space delimiter followed by 7 digits for the trip distance.

Bits 30 – 31: Reserved

# Appendix - Result Codes

---

# Appendix A – Result Codes

## Result Codes

Modem Verbose Response	Modem Terse Response	Definition
OK	0	command successful completed; ready
CONNECT	1	entering data transfer state
RING	2	Ring indication detected
NO CARRIER	3	connection terminated
ERROR	4	Command abnormally completed, ready
NO DIALTONE	6	Dial tone not found
BUSY	7	Busy signal detected
NO ANSWER	8	connection completion timeout

## UNSOLICITED RESULT Codes

Result Code	Definition	AT Command
+CCCM: <ccm>	Current call meter value	AT+CACM=1
+CCWA: <number>,<type> ,<class> [,<alpha>]	Call Waiting Status	AT+CCWA=1
+CLAV: <code>	ME Language Change	AT+CLAE=1
+CLIP: <number>,<type>[,<subaddr> , <satype> [,<alpha>]]	Calling Line Identification Presentation	AT+CLIP=1
+CME ERROR: <err>	ME Error Result Code	AT+CMEE=x
+COLP: <number>,<type>[,<subaddr> , <satype>[,<alpha>]]	Connected Line Identification Presentation	AT+COLP=1
+CR: <type>	Service Reporting Control	AT+CR=1
\$CREG: <stat>[,<lac>,<ci>]	Registration status indication	AT+CREG=1
+CRING: <type>	Incoming Call Indication	AT+CRC=1
+CSSI: <code1>[,<index>]	Supplementary Services Result Code	AT+CSSN=1,1
+CSSU: <code2>[,<index>[,<number> , <type>[,<subaddr>,<satype>]]]	Supplementary Services Result Code	AT+CSSN=1,1

Result Code	Definition	AT Command
+CUSD: <m>[,<str>,<dcs>]	Indication of Incoming USSD String	AT+CUSD=1
\$CGREG: <stat>[,<lac>,<ci>]	Registration Status	AT+CGREG=1

## SMS Unsolicited Result Codes

Result Code	Definition	AT Command
+CMTI: <mem>,<index>	Indication of new short message	AT+CNMI=1,1
+CMT: <length><CR><LF><pdu>	Short Message output Directly to TE (PDU mode)	AT+CNMI=1,2
+CBM: <sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data>	Incoming Cell Broadcast Message routed directly to TE	AT+CNMI=1,0,2
+CDS: <length><CR><LF><pdu>	SMS status report routed directly to the TE	AT+CNMI=1,0,0,1, AT+CSMP=49,



# Appendix - Error Codes

---

# Appendix B – Error Codes

## General Error Codes

Modem Numeric Response	Modem Verbose Response
0	phone failure
1	no connection to phone
2	phone-adaptor link reserved
3	operation not allowed
4	operation not supported
5	
6	
7	
10	
11	
12	
13	
14	
15	
16	incorrect password
17	
18	
20	memory full
21	invalid index
22	not found
23	memory failure
24	text string too long
25	invalid characters in text string
26	dial string too long
27	invalid characters in dial string
30	no network service

Modem Numeric Response	Modem Verbose Response
31	network timeout
32	network not allowed - emergency calls only
40	network personalization PIN required
41	network personalization PUK required
42	network subset personalization PIN required
43	network subset personalization PUK required
44	service provider personalization PIN required
44	service provider personalization PIN required
45	service provider personalization PUK required
46	corporate personalization PIN required
47	corporate personalization PUK required
48	
49	
100	unknown

## SMS Error Codes (+CMS)

Modem Numeric Response	Modem Verbose Response
1	unassigned (unallocated) number
8	operator determined barring
10	call barred
21	short message transfer rejected
27	destination out of service
28	unidentified subscriber
29	facility rejected
30	unknown subscriber
38	network out of order
41	temporary failure
42	congestion

Modem Numeric Response	Modem Verbose Response
47	resources unavailable, unspecified
50	requested facility not subscribed
69	requested facility not implemented
81	invalid short message transfer ref. value
95	invalid message, unspecified
96	invalid mandatory information
97	message type non-existent or not implemented
98	message not compatible with SM protocol state
99	information element non-existent or not impl.
111	protocol error, unspecified
127	interworking, unspecified
128	telematic interworking not supported
129	short message type 0 not supported
130	cannot replace short message
143	unspecified TP-PID error
144	data coding scheme (alphabet) not supported
145	message class not supported
159	unspecified TP-DCS error
160	command cannot be actioned
161	command unsupported
175	unspecified TP-Command error
176	TPDU not supported
192	SC busy
193	no SC subscription
194	SC system failure
195	invalid SME address
196	destination SME barred
197	SM rejected-duplicate SM
208	SMS storage full

Modem Numeric Response	Modem Verbose Response
209	no SMS storage capability
210	error in MS
211	memory capacity exceeded
255	unspecified error cause
300	ME failure
301	SMS service of ME reserved
302	operation not allowed
303	operation not supported
304	invalid PDU mode parameter
305	invalid text mode parameter
310	
311	
312	
313	
314	
315	
316	
317	
318	
320	memory failure
321	invalid memory index
322	memory full
330	SMSC address unknown
331	no network service
332	network timeout
340	no +CNMA acknowledgement expected
500	unknown error
512	failed to abort
197	SM rejected-duplicate SM

# Release Causes For Extended Error Reporting (+CEER)

Error	Description
1	unassigned number
-1,255	no error
3	no route to destination
6	channel unacceptable
8	operator determined barring
16	normal call clearing
17	user busy
18	no user responding
19	user alerting
21	call rejected
22	number changed
26	non selected user clearing
27	destination out of order
28	invalid number format
29	facility rejected
30	response to status enquiry"
31	normal
34	no channel available
38	network out of order
41	temporary failure
42	switching equipment congestion
43	access information discarded
44	requested channel unavailable
47	resources unavailable
49	quality of service unavailable
50	requested facility unsubscribed

Error	Description
55	incoming calls barred within CUG
57	bearer capability not authorized
58	bearer capability not available
63	service not available
65	bearer service not implemented
68	ACM reached ACM maximum
69	facility not implemented
70	only restricted bearer cap. avail.
79	service not implemented
81	invalid TI
87	no member of CUG
88	incompatible destination
91	invalid transit network selection
95	incorrect message
96	invalid mandatory information
97	message type not implemented
98	message type incompatible
99	info element not implemented
100	conditional info element error
101	message incompatible
102	recovery on time expiry
101	unsuccessful GPRS attach
102	unsuccessful PDP context activation
103	GPRS detach
104	GPRS PDP context deactivation
128	No Service
202	timer 303 expiry
203	establishment failure
210	no error

Error	Description
211	operation failed
212	timeout
213	bearer service not compatible

## Extended Error Codes

Error Code	Description
0	"parameter not allowed"
1	"data corrupted",
2	"internal error",
3	"call table full",
4	"service table full"
5	"call not found",
6	"no data-call supported"
7	"one call on hold",
8	"hold call not supported for this type"
9	"number not allowed by FDN"
10	"number not allowed by BDN"
11	"parallel USSD not supported"
12	"fax minimum speed condition"
13	"conflict with command details"
14	"not allowed by ALS-Lock"
15	"IMEI illegal"
16	



# Index

\$

\$ACKTM	93
\$ALTOSI	196
\$APIOPT	105
\$APIPWD	109
\$AREG	23
\$ATEXEC	175
\$ATPASSWD	35
\$CHKIN	110, 114, 280
\$CMDPORT	54, 130
\$CREG	24
\$DEVTyp	26
\$ETSAV	140
\$EVCID	141
\$EVDEL	137, 143, 156
\$EVDELA	144
\$EVDELr	145
\$EVENT	132, 137, 142, 146, 151, 156, 200, 225, 228, 293
\$EVGQRY	148
\$EVNTRY	149
\$EVTEST	150, 281
\$EVTIM	151
\$EVTIMQRY	152
\$EVTOFF	124, 153
\$EVTQRY	154
\$EXCACC	198
\$EXCDEC	199
\$FFS	177
\$FID	112
\$FLFILT	192
\$FOTACFG	168, 172-173
\$FOTAGET	169, 171
\$FOTAUPG	170, 173
\$FRIEND	109, 113, 130, 134, 137
\$FTPABORT	181
\$FTPBLK	182
\$FTP CFG	183
\$FTPCHDIR	184
\$FTPCLOSE	185
\$FTPDIR	186
\$FTPGET	176, 187
\$FTPOPEN	182, 188
\$FTPPUT	189
\$FTP RST	190
\$GEOFNC	200, 226, 274

\$GFDBNC	201
\$GFDEL	202
\$GFIDX	203
\$GPSCMD	204
\$GPSFD	205
\$GPSFLT	206
\$GPSLCL	211
\$GPSLNA	213
\$GPSOSI	214, 286
\$GPSQUAL	216
\$GPSRD	218
\$GPSRST	220
\$GPSVAL	222
\$HBRST	264
\$IDENT	116
\$IGNDBNC	263
\$MDMID	117, 128
\$MDN	6
\$MDSTAT	26
\$MLQSIZ	118
\$MSGLOGCL	122
\$MSGLOGDMP	123
\$MSGLOGEN	125
\$MSGLOGRD	125-126
\$MSGSND	129
\$MSID	7
\$NETIP	28, 30, 193
\$NETMON	30
\$NWACTIVATION	32
\$NWGMR	8
\$OBDACL	237
\$OBDADN	238
\$OBDADT	239
\$OBDALT	240
\$OBDAM	241
\$OBDART	243
\$OBDAUP	244
\$OBDBAT	232
\$OBDBBT	234
\$OBDBQS	235
\$OBDCUT	245
\$OBDDCL	246
\$OBDDSC	247
\$OBDECT	248
\$OBDEES	249
\$OBDFAC	250
\$OBDGSP	251
\$OBDIDL	252
\$OBDIGN	253
\$OBDLBL	254
\$OBLED	255
\$OBDLFL	256
\$OBDSAV	56, 257

\$OBDSPD	258	\$WAKEUP	113, 117, 136
\$OBDTYP	259		
\$OBDVER	261	<u>%</u>	
\$ODOMETER	223, 274	%CGREG	273, 297
\$OFF	266	%EM	293
\$OTASP	33	%SLEEP	267
\$PAD DST	130	%SNCNT	138
\$PING	194		
\$PKG	9, 26	<u>&amp;</u>	
\$PLYCLR	224	&F	54, 115, 122, 128, 144, 205, 223, 230, 281
\$PWRS AV	228	&V	55
\$QCDNSP	37	&W	56-61, 140, 147, 150, 153, 196, 214
\$QCDNSS	38		
\$QCMGD	66	<u>+</u>	
\$QCMGF	67, 69	+CACM	307
\$QCMGL	68	+CCWA	307
\$QCMGR	70	+CEER	51
\$QCMGS	71	+CGMI	13
\$QCMGW	72	+CGMM	14
\$QCMIP	39	+CGMR	15
\$QCMIPEP	40	+CGREG	308
\$QCMIPGETP	41	+CGSN	16
\$QCMIPMASS	42	+CLAE	307
\$QCMIPMHSS	43	+CLIP	307
\$QCMIPNAI	44	+CMEE	307
\$QCMSS	65	+CMGD	80
\$QCNMI	73	+CMGF	79, 82, 84
\$QCPMS	75	+CMGL	81
\$QCSMP	77	+CMGR	83
\$RESET	52, 257	+CMGS	3, 84
\$RTCALRM	159, 166	+CMGW	85
\$RTCCLRA	160-161	+CMSS	86
\$RTCRSET	162	+CNMI	87, 308
\$RTCTIME	164	+CNUM	46
\$RTCWAKE	166	+COLP	307
\$SMSDA	130-131, 133	+CPMS	90
\$SMSDAEN	132-133	+CR	307
\$SMSURC	78	+CRC	3, 307
\$SPC	45	+CREG	273, 297, 307
\$SPN	11	+CSMP	308
\$SRN	12	+CSQ	49
\$STOATEV	130, 155, 245	+CSS	27, 47
\$TCPAPI	95, 101, 130	+CSSN	307
\$TCPIDLETO	97	+CUSD	308
\$TCPRESTR T	98, 101	+GMI	17
\$TCPRETRYTO	99	+GMM	18
\$TCP SRC	100	+GMR	19
\$TCPSTATS	101	+GSN	20
\$TCPURC	103		
\$TODOM	229		
\$UDPMSG	134		
\$USRVAL	157		
\$V	53		

## A

ATE	57
ATI	3, 21
ATQ	58
ATS3	59
ATS4	60
ATS5	61
ATV	62
ATX	63