AirPrime MC/SL-Series (UMTS/LTE)

Extended AT Command Reference

SIERRA WIRELESS

2130616
Rev. 8

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. The Sierra Wireless modem can transmit signals that could interfere with this equipment.

Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

Limitation of Liability

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Patents

This product may contain technology developed by or for Sierra Wireless Inc.

This product includes technology licensed from QUALCOMM®.

This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group.

Copyright

Trademarks

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

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

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

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

Other trademarks are the property of their respective owners.

Contact Information

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

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

www.sierrawireless.com

Revision History

Revision number	Release date	Changes
2.8	Jul 2008	• Deprecated !GPSENABLE (for MC8780 and newer), !CUSTOM="SKUID", !GPSIPADDR, !GPSPORTID • Removed !CUSTOM="NOATTACH" • Updated !CUSTOM="GPSENABLE"), !GBAND, !DTMEN • Added !GPSMTLRSETTINGS, !GPSSUPLURL, !NVPLMNCLR, !CUSTOM="CSVOICEREJECT" • Updated 'Supporting Mini Cards' for all commands • Updated minimum firmware revision requirements
2.9	Oct 2008	• Updated modules list (added MC8791V, MC8792V) • Updated minimum firmware revision requirements • Added !GPSAUTOSTART • Added SIM Application Toolkit chapter with commands: !STKC, !STKGC, !STKCR, !STKPD, !STKMS, !STKPLI, and details of !STKN notifications • Updated !AVSETPROFILE • Updated !LEDCTRL
2.10	Apr 2009	• Removed !CUSTOM="SWOCENABLE" customization; !NVSWOC, !DTEMP, !NVPLMNCLR • Updated !CUSTOM="AUTONETWORKMODE" and "MEPLOCK" customizations; !REL, !DASCHAN, !DASBAND, !BSMCCHECK, !LEDCTRL, !GPSLOC, !CMEN, !PCTEMPLIMITS (cross-reference), !NVPORTMAP, !NVMUXMODE (removed 80/81 cross-reference to !NVPORTMAP), !GPSAUTOSTART • Added !MAPUART, !DIO, !DIOCFG, !AIN, !MAPMTPDP, !WGETWK, !WWKUP, • Added general and AT-command indexes
2.11	Jun 2009	• Added !BZBUZZ, !DAWSTXPWR, !UDUSBCOMP • Updated !CUSTOM="CSVOICEREJECT" • Added escape sequence guard time information
2.12	Nov 2009	• Standardized command detail format • Replaced device-specific references with chipset-specific references • Added PAD Commands chapter • Added !CWSETUP, !CWSTATS, !PING • Updated !CUSTOM: added "CPASCWRINGIND", "SIMLPM", "USBDMDISABLE", "WAPPUSHDATA" customizations; clarified "PCSCDISABLE" values. • Updated !AVTONEPLAY, !AVDTMFVOLDB: removed MIDI from <method> • Updated audio profile default settings (Table 7-2 on page 129, Table 7-2 on page 111) • Clarified !NVPLMN usage
3.0	Mar 2010	• New corporate branding • Updated descriptions of !AV* commands for 'instant update' • Added !AVINBANDRANGE, !AVRXVOLDB, !AVTONESETTINGS, !GPSSUPLPID • Updated !GPSPORTID description • Updated !GPSSUPLURL parameters • Updated !CUSTOM: added "TRUFLOWDISABLE" and "DISSTACK" customizations.

Revision number	Release date	Changes
4	November 2010	• Added SL808x devices • Removed MSM7200A/MSM7201A (MC8785V) • Updated !GPSPORTID, !GPSSUPLURL descriptions • Updated !DAWINFO, !UDUSBCOMP outputs • Updated !CUSTOM: added “GPSLPM”, “GPSREFLOC”, “GPSSEL”, “PPPPRO-FAUTH”, “RFINDENABLE”, “USBSERIALENABLE” customizations • Added !BAND, !GPSNMEACONFIG, !GPSLBSSETTINGS • Updated !AVTONEPLAY • Added !DAAGCTON, !NVSPCODE
5	March 2011	• Removed !AVEXTPCMSEL, !GPSENABLE, !NVENGPS • Removed MSM6280, MSM7200, MSM7201 • Added MDM6200, MDM8200A, MDM9200, MDM9600 • Added !ALLUP, !AVRXPCMIIRFLTR, !AVTXPCMIIRFLTR, !BZBUZZ, !BZBUZZPLAY, !CHAN, !DALGAVGAGC, !DALGRXAGC, !DALGTXAGC, !DIVERSITY, !IMSTESTMODE, !KEYOFF, !KEYON, !PACKAGE, $QCAGC, !RX2, !RX2AGC, !RXAGC, !SKU, !TX, !TXAGC, !UDPID, +WHCNF • Updated !DASBAND, !DIOCFG, !REL, +WWKUP • Added !CUSTOM customizations: GPSSUPLSETID, HWCFGLOCK

6	July 2011	• New chapter—OMA-DM Commands (!IDSAUTOFOTA, !IDSAUTOSDM, !IDSCONFIGACC, !IDSCREATEACC, !IDSDFLTACC, !IDSFUMOROOT, !IDSPID, !IDSROAM, !IDSSUPPORT)
		• New chapter - SAR Backoff and Thermal Control Commands (!MAXPWR, !SARBACKOFF, !SARSTATE, !SARSTATEDFLT, !THERMCONFIG, !THERM-DELTATX, !THERMDELTATXTEMP, !THERMENABLE, !THERMINFO, !THERMTHRESHOLD, !THERMTIMERS)
		• New commands—!ANTSEL, !LTENAS, !PADENDLISTEN, !PADSWITCH, !SELACQ, !STKAUTOCR, !STKEVENT, !STKEVENTLIST, !STKVER,
		• Updated commands:
		  • Removed MDM6200 support—!AVCODECRXG, !AVCODECTXG, !AVRXPCMFLTR, !AVTXPCMFLTR
		  • Updated gain range and gain calculation for QSC6270—!AVCODECSTG, !AVCODECTXG
		  • Added MDM6200/MSM6290 support—!AVRXPCMIIRFLTR, !AVTXPCMIIRFLTR
		  • !BAND—Corrected <index> type
		  • !BZBUZZ—Added query and query list syntaxes
		  • !CUSTOM—Added customizations (“LTEFIRST”, “LTESMS”, “SINGLEAPNSWITCH”)
		  • !MAPUART—Added usage note for SL809x
		  • PAD Commands—Added multipad support
		  • !PCTEMPLIMITS—Added MDM6200 limits
		  • !PCVOLTLIMITS—Fixed parameters, added MDM6200 and MDM8200A limits
		  • !SCPROFSWOPT—Added new <swOption> values
		  • !SKU—Updated chipsets, and <SKU> parameter values
		  • Added Setup Event List command—!STKC, !STKCR, !STKGC
		  • STK command (<cmdId>) parameters—Updated <DCS*> and <default> parameters where !STKVER? returns ‘1’; added Table 10-3, Setup Event List parameters, on page 223

Revision number	Release date	Changes
7	September 2011	• New section (I2S Audio Commands)—!AVCODECBRG, !AVCODECCFG, !AVCODECRED, !AVCODECRST, !AVCUSTI2CCFG, !AVMODESET, !AVREGVALWID, !AVSETSAMP, !AVUSEMCU • !CUSTOM—Removed customizations (“LTEFIRST”, “LTESMS”, “SINGLEAPNSWITCH”)

8	April 2012	• New commands—!BEP, !DALSRXBW, !DALSTXBW, !GPSLBSAPN, !GPSNMEASENTENCE, !GPSSUPLVER, !GPSXTRAAPN, !SIMRFSC, !SIMRSTC, !STKDTMF
		• New MDM8200A-specific commands—!AVEC, !AVRXDECGAIN, !AVRXPCMI-IRFLTR, !AVRXSPKRGAIN, !AVRXVOLDB, !AVSETDEV, !AVSETPROFILE, !AVSETVOL, !AVTXENCGAIN, !AVTXMICGAIN, !AVTXPCMIIRFLTR, !AVTXVOL
		• Updated commands:
		  • !ANTSEL—Added fourth GPIO
		  • !AVCODECRXG—Removed 6 and 7 from list of valid <profile> values
		  • !AVCODECSTG—Removed 6 and 7 from list of valid <profile> values. Updated gain formula. Corrected unity gain value.
		  • !AVCODECTXG—Removed 6 and 7 from list of valid <profile> values
		  • !AVMICGAIN—Removed 6 and 7 from list of valid <profile> values. Added chipset-specific examples.
		  • !BAND—Added C850, C1900, and W1700 bands
		  • !CUSTOM customizations—Added CFUNPERSISTEN, WAKEHOSTEN. Updated GPSENABLE, GPSSEL, GPSSUPLSETID, MUXMODE, PPPPRO-FAUTH, RADIORESET, STKUIEN.
		  • !DALGAVGAGC—Updated usage note
		  • !GPSCLRASSIST—Updated parameters for MDM9200 exceptions.
		  • !GPSTRANSSEC—Updated <security> parameter for MDM9200.
		  • !MAXPWR—Added MDM6200 support; removed 3 (GSM) from valid <tech> values
		  • !PADCONF—Updated description, <idle> parameter description, and <interchar> parameter units
		  • !PADSETUP—Added <addrtype> parameter, and updated <ipaddr> parameter examples
		  • !PADSWITCH—Updated <ipaddr> parameter examples
		  • !SARBACKOFF, !SARSTATE, !SARSTATEDFLT—Added MDM6200 support
		  • !UDUSBCOMP—Updated Query List response for DIP to show QMI interface
		  • +WHCNF—Added buzzer to list of supported <type> values.
		• Deprecated !GPSPROTOSEL (for QSC6270)
		• Updated Audio profiles on page 128 (updated supported profiles by module type, added new tables for default settings, updated default gain values)

Contents

1: About This Guide

Introduction

This document describes proprietary, password-protected AT commands available for Sierra Wireless AirPrime™ MC/SL-series intelligent embedded modules. These commands are intended for use by OEMs, and are supplemental to the standard AT commands for GSM devices defined by the 3GPP (3rd Generation Partnership Project) in *TS 27.007 AT command set for User Equipment (UE)* and *TS 27.005 Use of Data Terminal Equipment—Data Circuit terminating Equipment (DTE-DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (BSE)*.

The Sierra Wireless document *UMTS Modems - Supported AT Command Reference (document number 2130617)* identifies supported standard AT commands and provides details where commands vary from the standards. It also describes proprietary commands that are *not* password-protected.

Note: If a command is only partially protected (for example, the execution form is restricted, but the query form is not), the non-protected form of the command also appears in the 'Supported' reference.

The AT commands in this document are specific to the Sierra Wireless embedded modems listed in Table 1-1.

Table 1-1: Supported AirPrime modules[a]

Chipset	Device	Firmware revision (minimum)[b]
MDM6200	SL809X	P0.0
MDM6270	SL8081[c] SL8083[c] SL8085[c]	S2.0.0.9
MDM8200	MC8700	M2_0_4_0ap
MDM8200A	MC8704 MC8705	(MC8704) T2_0_1_4 (MC8705) T1_0_1_1
MDM8220	MC8801	N_0_0_0ap
MDM9200	MC7700 MC7710	9200X_01.00.00.00
MDM9600	MC7750	9600M_01.00.00.00

Table 1-1: Supported AirPrime modules[a] (Continued)

Chipset	Device	Firmware revision (minimum)[b]
MSM6290	MC8790/ MC8790V MC8791V MC8792V MC8795V	K1_0_2_8ap
QSC6270	SL8080 SL8081[c] SL8082 SL8083[c] SL8084 SL8985[c]	S1.0 S2.0 (Voice/GPS devices)

a. Changes made to this document after a device reaches end-of-life may not apply to that. Refer to the Revision History on page 5 for a summary of changes made in recent releases of this document.
b. Any exceptions are indicated in the command detail sections.
c. SL8081, SL8083, and SL8085 may use either MDM6270 or QSC6270 chipsets.

Note: When designing applications that use these AT commands, use Watcher™ (and other Sierra Wireless applications) as functionality templates to ensure proper use of command groups. For questions or concerns relating to command implementation, please contact your Sierra Wireless account representative.

Command access

Most of the commands in this reference are password-protected. To use these commands, you must enter the correct password using AT!ENTERCND on page 28. Once the password is entered, all commands are available and remain available until the modem is reset or powered off and on.

The password assigned to **AT!ENTERCND** is unique to each carrier and is configured onto the modem during manufacture. If you do not know your password, contact your Sierra Wireless Account Manager.

Command timing

Interval timing

Some commands require time to process before additional commands are entered. For example, the modem returns OK when it receives **AT!DAFTMACT**. If **AT!DASBAND** is received too soon after this, the modem returns an error.

When building automated test scripts, ensure that sufficient delays are embedded, where necessary, to avoid these errors.

Escape sequence guard time

The AT escape sequence "**+++**" requires a guard time of 1.0 seconds before and after it is used.

Result codes

Result codes are not shown in the command tables unless special conditions apply. Generally the result code OK is returned when the command has been executed. ERROR may be returned if parameters are out of range, and is returned if the command is not recognized or is not permitted in the current state or condition of the modem.

References

This guide covers the command sets used by OEMs, designers and testers of Sierra Wireless AirPrime MC/SL-series intelligent embedded modules. It does not deal with operational use commands. For normal operations of the modem, consult the *UMTS Modems Supported AT Command Reference (document number 2130617)*.

You may also want to consult the other documents available on our website at www.sierrawireless.com.

Terminology and acronyms

This document makes wide use of acronyms that are in common use in data communications and cellular technology.

Current firmware versions

Version

To determine your firmware revision:

- Enter the identification command **AT+GMR**
  The modem responds with version information for software, firmware, and hardware.

  The details following the revision number include Sierra Wireless information on the specific build followed by the date and time of the build.

  +GMR: … F/W VER: R1_0_0_…

Upgrading

If your modem firmware is an earlier version, you can acquire updated firmware by contacting your account manager.

Document structure

This document describes the proprietary commands listed in the tables below—each table corresponds to a specific chapter.

AT Password Commands—Commands used to enable access to password-protected AT commands and to set the AT command password.

Table 1-2: AT password commands

Command	Description	Page
!ENTERCND	Enable access to password-protected commands	28
!SETCND	Set AT command password	28

Modem Status, Customization, and Reset Commands—Commands used to determine modem status, adjust customization settings, and reset the modem.

Table 1-3: Modem status commands

Command	Description	Page
!AIN	Read analog value from AUXV1	32
!ANTSEL	Set/query external antenna select configuration	33
!BAND	Select/return frequency band set	34
!BPLMNTIMER	Set/return data inactivity timer value	36
!BSHWID	Generate frequency on buzzer pin	38
!BSMCCHECK	Return modem hardware version	37
!BZBUZZ	Generate frequency on buzzer pin	38
!BZBUZZPLAY	Generate melody on buzzer pin	39
!CMEN	Enable/disable HSPA Compressed Mode stack functionality	39
!CUSTOM	Set/return customization settings	40
!CWSETUP	Set Connection Watchdog parameters	46
!CWSTATS	Query/clear Connection Watchdog statistics	47
!DARPEN	Enable/disable DARP for SAIC	47
!DIO	Read/write from/to Digital I/O (DIO) channel	48
!DIOCFG	Configure DIO channels	49
!DTMEN	Enable/disable Dual Transfer Mode stack functionality	51
!EDAEN	Configure protocol stack for EDA	52
!EQEN	Enable/disable WCDMA L1 equalizer	52
!GCFEN	Enable/disable GCF test mode	53
!GCFUIMTYPE	Set/return current SIM type	53

Table 1-3: Modem status commands (Continued)

Command	Description	Page
!HSDCAT	Set/return HSDPA category	54
!HSUCAT	Set/report HSUPA category	54
!LEDCTRL	Set/report LED control pattern	55
!LTENAS	Configure LTE NAS settings	56
!MAPMTPDP	Configure MT PDP port mapping	57
!MAPUART	Map UART interface to a service	58
!MXPORTMAP	Set/report MUX mode port mappings	58
!NASREL	Set/report supported NAS release compliance version	59
!NVENCRYPTIMEI	Write unencrypted IMEI to modem	60
!NVMUXMODE	Set/report USB descriptor MUX mode	61
!NVNSCODE	Return Network Subset codes	62
!NVOEM	Set/report values of non-volatile (NV) memory items	63
!NVPLMN	Provision PLMN list for Network Personalization locking	64
!NVPORTMAP	/Change modem port mappings in non-MUX mode	65
!NVSPCODE	Provision Network Service Provider code list	65
!PACKAGE	Return package version string	66
!PCINFO	Return power control status information	67
!PCOFFEN	Set/return Power Off Enable state	68
!PCTEMPLIMITS	Set/report temperature state limit values	69
!PCVOLTLIMITS	Set/report power supply voltage state limit values	70
!PING	Ping an IP address	71
!PRIID	Set/report module PRI part number and revision	71
!REL	Set/report active protocol/revision	72
!SCANTHRESH	Set/report WCDMA Scan Threshold	73
!SCPROFSWOPT	Set/report profile’s software options	74
!SCROPROF	Set/report profile’s read-only flag	74
!SELACQ	Select RAT acquisition order	75
!SIMRFSC	Set/report SIM refresh reset notification state	76
!SIMRSTC	Set/report SIM refresh reset notification state	77
!SKU	Read modem’s SKU	77
!SLEEP	Set/report Sleep Enable state	78

Table 1-3: Modem status commands (Continued)

Command	Description	Page
!UDPID	Set/report product ID in USB descriptor	78
!UDUSBCOMP	Set/report USB interface configuration	79
!UOOS	Set/report UMTS 'Out of Service' parameters	80
+WGETWK	Return wake-up event type	80
+WHCNF	Activate/deactivate modem hardware features	81
+WWKUP	Enable/disable wake-up signals	83

Diagnostic Commands—Commands used to select frequency bands and diagnose problems.

Table 1-4: Diagnostic commands

Command	Description	Page
!CMUX	Implement multiplexing mode	86
!CMUXLPBK	Configure ports into loopback mode/query loopback state	86
!DIVERSITY	Enable/disable CDMA receive diversity	87
!ERR	Display diagnostic information	87
!GBAND	Set/return the current operating band	88
$QCPDPP	Set/report PDP-IP connection authentication parameters	89
$QCTER	Set/report TE-DCE baud rate	89
!RXDEN	Enable/disable WCDMA/LTE receive diversity	90
!UMTSCHAN	Set/report Priority UMTS Channel Selection state	91

Test Commands—Commands required to place the modem in particular modes of operation, test host connectivity, and to configure the transmitters and receivers for test measurements.

Table 1-5: Test commands

Command	Description	Page
!ALLUP	Turn transmitter on/off and simulate 'All UPs'	96
!BEP	Return MT RSSI and RSQ	97
!CHAN	Tune synthesizer to channel/band	98
!DAAGCTON	Return C/N (carrier to noise ratio) while in factory test mode	98
!DAFTMACT	Put modem into Factory Test Mode	99
!DAFTMDEACT	Put modem into online mode from Factory Test Mode	99
!DAGGAVGRSSI	Return averaged RSSI value in dBm (GSM only)	99

Table 1-5: Test commands (Continued)

Command	Description	Page
!DAGGRSSI	Return the RSSI value in dBm (GSM only)	100
!DAGGRSSIRAW	Return raw RSSI value	100
!DAGINFO	Return GSM mode RF information (GSM only)	101
!DAGSLOCK	Return synthesizer lock state	101
!DAGSRXBURST	Set GSM receiver to burst mode	102
!DAGSRXCONT	Set GSM receiver continuously on	102
!DAGSTXBURST	Set GSM transmitter to burst mode	103
!DAGSTXFRAME	Set GSM Tx frame structure	104
!DALGAVGAGC	Return averaged Rx AGC value (LTE only)	105
!DALGRXAGC	Return Rx AGC value (LTE only)	106
!DALGTXAGC	Return Tx AGC value and transmitter parameters (LTE only)	107
!DALSRXBW	Set LTE Rx bandwidth (LTE only)	108
!DALSTXBW	Set LTE Tx bandwidth (LTE only)	109
!DAOFFLINE	Place modem offline	109
!DASBAND	Set frequency band	110
!DASCHAN	Set modem channel (frequency)	111
!DASLNAGAIN	Set LNA gain state	112
!DASPDM	Set PDM value	113
!DASTXOFF	Turn Tx PA off	113
!DASTXON	Turn Tx PA on	114
!DAWGAVGAGC	Return averaged Rx AGC value (WCDMA only)	114
!DAWGRXAGC	Return Rx AGC value (WCDMA only)	115
!DAWINFO	Return WCDMA mode RF information (WCDMA only)	116
!DAWSCONFIGRX	Set WCDMA receiver to factory calibration settings	117
!DAWSPARANGE	Set PA range state machine	118
!DAWSSCHAIN	Enable secondary receive chain (WCDMA only)	118
!DAWSCHAINTCM	Place receive chain in test call mode (WCDMA only)	118
!DAWSTXCW	Set waveform used by the transmitter	119
!DAWSTXPWR	Set desired Tx power level (WCDMA mode only)	119
!IMSTESTMODE	Enable/disable IMS test mode	120
!KEYOFF	Key off the transmitter	120

Table 1-5: Test commands (Continued)

Command	Description	Page
!KEYON	Key on the transmitter	121
!OSDSM	Display memory usage for DSM buffer pools	121
$QCAGC	Read Rx AGC (CDMA and WCDMA modes)	122
!RX2	Turn second receiver on/off	122
!RX2AGC	Read second receiver Rx AGC	123
!RXAGC	Read first receiver Rx AGC	123
!TX	Turn transmitter on/off	123
!TXAGC	Set desired Tx AGC	124

Memory Management Commands—Commands that control the data stored in non-volatile memory of the modem.

Table 1-6: Memory management commands

Command	Description	Page
!NVDEF	Reset non-volatile memory	126
!NVRESTORE	Restore backup data	126

Voice Commands—Voice commands (Supported on voice-enabled modems only. For example, MC8795V and SL8080)

Table 1-7: Voice commands

Command	Description	Page
!AVAUDIOLPBK	Enable/disable an audio loopback	132
!AVCODECRXG	Set/report CODEC Rx gain	132
!AVCODECSTG	Set/report CODEC sidetone gain	134
!AVCODECTXG	Set/report CODEC Tx gain	135
!AVDEF	Set audio settings to default values	136
!AVDTMFTXG	Set/report the DTMF Tx gain	137
!AVDTMFVOLDB	Set/report volume for each DTMF volume level in Rx direction	138
!AVEC	Set/report the echo cancellation setting	139
!AVEXTPCMCFG	Configure external PCM interface	140
!AVEXTPCMSTOPCLKOFF	Prevent/allow external PCM interface clock from turning off	141
!AVINBANDRANGE	Specify Progress Descriptor value range for in-band signaling	142
!AVMICGAIN	Set/report microphone gain	143
!AVNS	Enable/disable noise suppression	144

Table 1-7: Voice commands (Continued)

Command	Description	Page
!AVRXAGC	Set/report Rx AVC/AGC configuration	145
!AVRXPCMFLTR	Set/report the Rx PCM filter tap	146
!AVRXPCMIIRFLTR	Set/report the Rx PCM IIR filter parameters	148
!AVRXVOLDB	Set/report volume for each voice volume level in Rx direction	150
!AVSETPROFILE	Configure and activate profile	151
!AVSN	Set/report audio revision number	152
!AVTONEPLAY	Play DTMF tone	152
!AVTONESETTINGS	Enable/disable playing of locally-generated DTMF tones	153
!AVTXAGC	Set Tx AGC	154
!AVTXPCMFLTR	Set/report Tx PCM filter tap	155
!AVTXPCMIIRFLTR	Set/report the Tx PCM IIR filter parameters	156
!AVTXVOL	Set Tx volume	158

I2S Audio Commands—I2S audio codec commands (Supported on I2S audio-enabled modems only. For example, MC8704.)

Table 1-8: I2S audio commands

Command	Description	Page
!AVCODECBRG	Configure codec registers to make call	165
!AVCODECCFG	Configure codec register	165
!AVCODECRED	Read a codec register	166
!AVCODECRST	Configure codec registers for reset	166
!AVCUSTI2CCFG	Configure external codec I2C details	167
!AVEC	Set/report the echo cancellation setting	168
!AVMODESET	Select codec mode	168
!AVREGVALWID	Set codec register bit width	169
!AVRXDECGAIN	Set/report voice decoder gain	169
!AVRPCMIIRFLTR	Set/report the Rx PCM IIR filter parameters	170
!AVRXSPKRGAIN	Set/report audio profile speaker gain	172
!AVRXVOLDB	Set/report Rx voice volume	172
!AVSETDEV	Set audio profile Rx and Tx mute states	173
!AVSETPROFILE	Configure and activate profile	174
!AVSETSAMP	Set I2S sample rate	175

Table 1-8: I2S audio commands (Continued)

Command	Description	Page
!AVSETVOL	Set audio profile default volume level	175
!AVTXENCGAIN	Set/report audio profile encoder gain	176
!AVTXMICGAIN	Set/report audio profile microphone gain	176
!AVTXPCMIIRFLTR	Set/report the Tx PCM IIR filter parameters	177
!AVTXVOL	Set Tx volume	178
!AVUSEMCU	Select codec controller	178

GPS Commands—Supported on GPS-enabled modems only.

Table 1-9: GPS commands

Command	Description	Page
!GPS3RDPARTYXFER	Initiate Location Service (LCS) third party transfer location request	181
!GPSAUTOSTART	Configure GPS auto-start features	182
!GPSCLRASSIST	Clear specific GPS assistance data	183
!GPSCOLDSTART	Clear all GPS assistance data	184
!GPSEND	End an active session	184
!GPSFIX	Initiate GPS position fix	185
!GPSIPADDR	Set/report IP address to use over TCP/IP	186
!GPSKEEPWARM	Configure Keep Warm functionality	186
!GPSLBSAPN	Set GPS LBS APNs	187
!GPSLBSSETTINGS	Set default GPS location fix options	188
!GPSLOC	Return last known location of the modem	189
!GPSMTLRSETTINGS	Set/report MT location request settings	190
!GPSNIQOSTIME	Set/report GPS QoS timeout period for network-initialized fixes	191
!GPSNMEACONFIG	Enable and set NMEA data output rate	191
!GPSNMEASENTENCE	Set/report NMEA sentence type	192
!GPSPORTID	Set/report port ID to use over TCP/IP	193
!GPSPOSMODE	Configure support for GPS positioning modes	194
!GPSPROTOSEL	Control GPS protocol selection	195
!GPSSATINFO	Request satellite information	196
!GPSSTATUS	Request current status of a position fix session	197
!GPSSUPLPID	Set/report supplementary channel connection profile ID	198

Table 1-9: GPS commands (Continued)

Command	Description	Page
!GPSSUPLURL	Set/report SUPL server URL	198
!GPSSUPLVER	Set/report SUPL server version	199
!GPSTRACK	Initiate local tracking (multiple fix) session	200
!GPSTRANSSEC	Control GPS transport security	201
!GPSXTRAAPN	Set GPS XTRA APNs	202
!GPSXTRADATAENABLE	Set/report GPS XTRA settings	203
!GPSXTRADATAURL	Set/report GPS XTRA data server URLs	204
!GPSXTRAINITDNLD	Initiate gpsOneXTRA data download and inject operation	204
!GPSXTRASTATUS	Return current status of gpsOneXTRA	205
!GPSXTRATIME	Inject GPS or UTC time into gpsOneXTRA system	206
!GPSXTRATIMEENABLE	Set/report GPS XTRA time settings	207
!GPSXTRATIMEURL	Set/report GPS XTRA SNTP server URLs	208

STK Commands—SIM Application Toolkit commands (Supported on MDM6270/MDM8200/MSM6290/QSC6270-based modems.)

Table 1-10: SIM Application Toolkit commands

Command	Description	Page
!STKAUTOCR	Configure host responses to SIM commands	213
!STKC	Receive unsolicited SIM command	214
!STKCR	Respond to unsolicited SIM command	215
!STKDTMF	Send DTMF string on active call	216
!STKEVENT	Notify SIM when monitored STK event occurs	216
!STKEVENTLIST	Return list of host-monitored STK events	217
!STKGC	Retrieve data for unsolicited SIM command	218
!STKMS	Request menu item selection or help from SIM	219
!STKPD	Update STK supported features profile	220
!STKPLI	Record local provisioning information	222
!STKVER	Display STK version	222

PAD Commands—Commands for configuring, initiating, and disconnecting PAD (Packet Assembler/Disassembler) connections. (Supported on the MC8790V/91V/92V/95V and SL8080/82/84 modems.)

Table 1-11: PAD commands

Command	Description	Page
!PADCONF	Configure profile options	244
!PADCONN	Initiate PAD client connection	245
!PADDISCONN	Disconnect PAD connection	246
!PADENDLISTEN	Disable PAD server	246
!PADFILTER	Set IP address filters for TCP PAD server	247
!PADLISTEN	Initiate PAD server connection	248
!PADSETUP	Set/query PAD (Packet Assembler/Disassembler) profile connection parameters	249
!PADSWITCH	Switch active PAD session	250

OMA-DM Commands—Commands used to configure DM (Device Management) accounts, sessions, and host–device–server interactions.

Table 1-12: OMA-DM commands

Command	Description	Page
!IDSAUTOFOTA	Configure automatic settings for FOTA updates	252
!IDSAUTOSDM	Configure Subscriber Device Management response to server request	253
!IDSCONFIGACC	Configure DM account authentication mode and XML format	254
!IDSCREATEACC	Enter DM account credentials	255
!IDSDFLTACC	Set DM account to use for device-initiated sessions	256
!IDSFUMOROOT	Set DM Tree root path for FUMO node	256
!IDSPID	Set profile ID for DM data connection types	257
!IDSROAM	Configure DM client roaming support	257
!IDSSUPPORT	Configure DM sessions	258

SAR Backoff and Thermal Control Commands—Commands used to configure SAR backoff options, and thermal mitigation algorithm parameters and limits.

Table 1-13: SAR backoff and thermal control commands

Command	Description	Page
!MAXPWR	Set/report maximum Tx power	260
!SARBACKOFF	Set/report maximum Tx power limit	261

Table 1-13: SAR backoff and thermal control commands (Continued)

Command	Description	Page
!SARSTATE	Set/report SAR backoff state	262
!SARSTATEDFLT	Set/report default SAR backoff state	262
!THERMCONFIG	Set/report thermal mitigation configuration options	263
!THERMDELTATX	Set/report amount to reduce maximum Tx power	264
!THERMDELTATXTEMP	Set/report amount power backoff temperature threshold	265
!THERMENABLE	Enable/disable thermal mitigation	265
!THERMINFO	Display thermal mitigation information	266
!THERMTHRESHOLD	Set/report thermal threshold, mitigation threshold, and hysteresis	267
!THERMTIMERS	Set/report thermal mitigation algorithm timer details	268

Conventions

The following format conventions are used in this reference:

Character codes or keystrokes that are described with words or standard abbreviations are shown within angle brackets using a different font, such as <CR> for Carriage Return and <space> for a blank space character.

Numeric values are decimal unless prefixed as noted below.

Hexadecimal values are shown with a prefix of 0x, i.e. in the form 0x3D.

Binary values are shown with a prefix of 0b, i.e. in the form 0b00111101.

Command and register syntax is noted using an alternate font: **!CHAN=<c>[,b]**. The leading "**AT**" characters are not shown but must be included before all commands except as noted in the reference tables.

Characters that are required are shown in uppercase; parameters are noted in lowercase. Required parameters are enclosed in angle brackets (**<n>**) while optional parameters are enclosed within square brackets (**[x]**). The brackets are not to be included in the command string.

Commands are presented in table format. Each chapter covers the commands related to that subject and presents a summary table to help you locate a needed command. Commands are in ASCII alphabetical order in the body of each chapter.

Any default settings are noted in the command tables. Note that these are the factory default settings and *not* the default parameter value assumed if no parameter is specified.

Result Code This is a numeric or text code that is returned after all commands (except resets)—text codes are returned if verbose responses are enabled. Only one result code is returned for a command line regardless of the number of individual commands contained on the line.

Response This term indicates a response from the modem that is issued prior to a result code. Reading registers or issuing commands that report information will provide a response followed by a result code unless the command generates an error.

Responses and result codes from the modem, or host system software prompts, are shown in this font:

CONNECT 14400

2: AT Password Commands

Introduction

AT commands described in this document are password-protected. This chapter describes how to enter and change the password.

Command summary

Table 2-1 on page 27 lists the commands described in this chapter.

Table 2-1: AT password commands

Command	Description	Page
!ENTERCND	Enable access to password-protected commands	28
!SETCND	Set AT command password	28

Command reference

Table 2-2: AT command password details

Command	Description
!ENTERCND Supporting chipsets • All *Note: The execution operation ("=") is not password-protected.*	**Enable access to password-protected commands** Before you can use any password-protected AT commands, you must enter the password correctly using this command. The initial password is configured onto the modem during manufacture. You can change the password using !SETCND. If you do not know the password, contact your Sierra Wireless Account Manager. Once the password has been entered correctly, the password-protected AT commands are available until the modem is reset or powered off and on. **Usage:** • Execution: **AT!ENTERCND=<"key">** Response: OK Purpose: Unlock password-protected commands. • Query: **AT!ENTERCND?** Response: <key> *(if unlocked)* Purpose: This command is password-protected. After entering the password correctly using the execution operation ("="), you can use this command to display the password as a reminder. **Parameters:** <"key"> (Password stored in NV memory) • Password must be entered with quotation marks. (For example, AT!ENTERCND="ExamplePW".) • Password length: 4–10 characters (0–9, A–Z, upper or lower case) • Characters may be entered in ASCII format, or in Hex format. (For example: "myPass3" or "ABCDEF01234".)
!SETCND Supporting chipsets • All	**Set AT command password** Change the password used for the !ENTERCND command. (Before you can change the password using **!SETCND**, you must enable access to this command using **!ENTERCND**.) **Usage:** • Execution: **AT!SETCND=<"key">** Response: OK Purpose: Sets <"Key"> as the new password for accessing protected commands. **Parameters:** <"key"> (New password) • Password must be entered with quotation marks (for example, AT!SETCND="NewPW"). • Password length: 4–10 characters (0–9, A–Z, upper or lower case) • Characters may be entered in ASCII format, or in Hex format. (For example: "myPass3" or "ABCDEF01234".) **Warning:** *Do NOT enter a null password (that is, the <"Key"> cannot be "") — you will NOT be able to use password-protected commands, and will have to contact Sierra Wireless for help to reset the password.*

3: Modem Status, Customization, and Reset Commands

Introduction

This chapter describes commands used to reset the modem, adjust customization settings, retrieve the hardware version, and monitor the temperature, voltage, and modem status.

Command summary

Table 3-1 lists the commands described in this chapter.

Table 3-1: Modem status commands

Command	Description	Page
!AIN	Read analog value from AUXV1	32
!ANTSEL	Set/query external antenna select configuration	33
!BAND	Select/return frequency band set	34
!BPLMNTIMER	Set/return data inactivity timer value	36
!BSHWID	Generate frequency on buzzer pin	38
!BSMCCHECK	Return modem hardware version	37
!BZBUZZ	Generate frequency on buzzer pin	38
!BZBUZZPLAY	Generate melody on buzzer pin	39
!CMEN	Enable/disable HSPA Compressed Mode stack functionality	39
!CUSTOM	Set/return customization settings	40
!CWSETUP	Set Connection Watchdog parameters	46
!CWSTATS	Query/clear Connection Watchdog statistics	47
!DARPEN	Enable/disable DARP for SAIC	47
!DIO	Read/write from/to Digital I/O (DIO) channel	48
!DIOCFG	Configure DIO channels	49
!DTMEN	Enable/disable Dual Transfer Mode stack functionality	51
!EDAEN	Configure protocol stack for EDA	52
!EQEN	Enable/disable WCDMA L1 equalizer	52
!GCFEN	Enable/disable GCF test mode	53
!GCFUIMTYPE	Set/return current SIM type	53
!HSDCAT	Set/return HSDPA category	54

Table 3-1: Modem status commands (Continued)

Command	Description	Page
!HSUCAT	Set/report HSUPA category	54
!LEDCTRL	Set/report LED control pattern	55
!LTENAS	Configure LTE NAS settings	56
!MAPMTPDP	Configure MT PDP port mapping	57
!MAPUART	Map UART interface to a service	58
!MXPORTMAP	Set/report MUX mode port mappings	58
!NASREL	Set/report supported NAS release compliance version	59
!NVENCRYPTIMEI	Write unencrypted IMEI to modem	60
!NVMUXMODE	Set/report USB descriptor MUX mode	61
!NVNSCODE	Return Network Subset codes	62
!NVOEM	Set/report values of non-volatile (NV) memory items	63
!NVPLMN	Provision PLMN list for Network Personalization locking	64
!NVPORTMAP	/Change modem port mappings in non-MUX mode	65
!NVSPCODE	Provision Network Service Provider code list	65
!PACKAGE	Return package version string	66
!PCINFO	Return power control status information	67
!PCOFFEN	Set/return Power Off Enable state	68
!PCTEMPLIMITS	Set/report temperature state limit values	69
!PCVOLTLIMITS	Set/report power supply voltage state limit values	70
!PING	Ping an IP address	71
!PRIID	Set/report module PRI part number and revision	71
!REL	Set/report active protocol/revision	72
!SCANTHRESH	Set/report WCDMA Scan Threshold	73
!SCPROFSWOPT	Set/report profile's software options	74
!SCROPROF	Set/report profile's read-only flag	74
!SELACQ	Select RAT acquisition order	75
!SIMRFSC	Set/report SIM refresh reset notification state	76
!SIMRSTC	Set/report SIM refresh reset notification state	77
!SKU	Read modem's SKU	77
!SLEEP	Set/report Sleep Enable state	78
!UDPID	Set/report product ID in USB descriptor	78

Table 3-1: Modem status commands (Continued)

Command	Description	Page
!UDUSBCOMP	Set/report USB interface configuration	79
!UOOS	Set/report UMTS 'Out of Service' parameters	80
+WGETWK	Return wake-up event type	80
+WHCNF	Activate/deactivate modem hardware features	81
!WWKUP	Enable/disable wake-up signals	83

Command reference

Table 3-2: Modem status, customization, and reset commands

Command	Description
!AIN Supporting chipsets: • MSM6290	**Read analog value from AUXV1** Return the raw ADC value from AUXV1. **Usage:** • Query: **AT!AIN?** Response: !AIN: <value> OK Purpose: Return the raw ADC value from AUXV1. **Parameters:** <value> (Analog value from AUXV1) • Valid range: 0–255

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!ANTSEL Supporting chipsets: • MDM9200 (min f/w rev: 9200X 1.0 Release 2) • MDM9600	**Set/query external antenna select configuration** (This command is supported only on Mini Card devices.) Configure the device to drive (high or low) up to four GPIOs for specific bands. (If a GPIO is not needed for a specific band, it is identified as not required.) When the device switches to a configured band, the GPIOs are driven as specified, and the host uses those GPIOs to tune the external antenna appropriately. Note that this feature is independent of the radio technology being used. For example, Band 5 corresponds to any 850-band technology (CDMA, WCDMA, LTE, GSM). *Note: Any change to GPIO configurations take effect after the modem is reset.* *Note: System level testing should be performed to ensure that the antenna switching feature does not introduce any handover issues. The tunable antenna should be designed to ensure that it can retune in < 5 µs (recommended) and < 10 µs (maximum).* **Usage:** • Execution: **AT!ANTSEL=<band>, <gpio1>, <gpio2>, <gpio3>[, <gpio4>]** Response: OK Purpose: Configure the GPIOs for the specified <band>. • Query: **AT!ANTSEL?** Response: BAND <band *a*>: <gpio1>, <gpio2>, <gpio3>[, <gpio4>] BAND <band *b*>: <gpio1>, <gpio2>, <gpio3>[, <gpio4>] ... OK Purpose: Display the current external antenna select configuration. • Query List: **AT!ANTSEL=?** Purpose: Display valid execution format and parameter values. **Parameters:** <band> (RF band) • 3GPP band number. For a full listing of 3GPP band numbers, see Table 4-2 on page 280. • Valid range: 0–60. Band support is product specific—see the device's Product Specification or Product Technical Specification document for details. <gpio1>, <gpio2>, <gpio3>, <gpio4> (GPIO configurations. Note: <gpio4> availability is device-specific—see the appropriate Product Technical Specification for details.) • 0=Logic low • 1=Logic high • 2=Not used for antenna selection (Default value for <gpio4>.)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!BAND Supporting chipsets: • MDM6200 • MDM6270 • MDM8220 • MDM9200 • MDM9600 • QSC6270 *Note: The 'Basic' command and response versions are used if you haven't entered the required password. (See Command access on page 14.)* *Note: The 'Basic' commands and responses are also described in the AirCard/ AirPrime Supported AT Command Reference.*	**Select/return frequency band set** Configure the modem to operate on a set of frequency bands, look up available sets, create new sets, or return the current selection. **Usage:** • Execution (Basic): **AT!BAND=<Index>** Response: OK Purpose: Select an existing set of bands. • Execution (Extended): **AT!BAND=<Index>,"<Name>",<GWmask>[,<Lmask>]** Response: OK Purpose: Create a new set of bands. • Query: **AT!BAND?** Response: Index, Name[, GW Band Mask [, L Band Mask]] <Index>, <Name>[, <GWmask> [, <Lmask>]] OK *or* *(If the current band mask doesn't match a band set)* Unknown band mask. Use AT!BAND to set band. <Index> OK Purpose: Report the current band selection. (<GWmask> and <Lmask> may only appear in Extended responses.) • Query List: **AT!BAND=?** Response: Index, Name[, GW Band Mask [L Band Mask]] <Index1>, <Name1>[, <GWmask1> [, <Lmask1>]] ... <IndexN>, <NameN>[, <GWmaskN> [, <LmaskN>]] OK Purpose: Display allowed <Index> values and descriptions of the associated band sets. (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!BAND (continued)	**Select/return frequency band set (continued)** Parameters: <Index> (Index of a band set. Use the Query List command to display all supported sets) • Valid range: 0–13 (Hexadecimal. There are 20 possible values.) <Name> (Name of the band set) • ASCII string—Up to 30 characters <GWmask> (GSM/WCDMA bands included in the set) • Format: 32-bit bitmask • Valid values: • 0000000000000003—C850 • 0000000000000004—C1900 • 0000000000000080—G1800 • 0000000000000300—G900 (EGSM/GSM) • 0000000000080000—G850 • 0000000000200000—G1900 • 0000000000400000—W2100 • 0000000000800000—W1900 • 0000000002000000—W1700 • 0000000004000000—W850 • 0000000008000000—W800 • 0002000000000000—W900 <Lmask> (LTE bands included in the set) • Format: 32-bit bitmask • Valid values: • 0000000000000001—Band 1 0000000000000002—Band 2 ... 0000004000000000—Band 39 0000008000000000—Band 40

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!BPLMNTIMER Supporting chipsets: • All	**Set/return data inactivity timer value** Set or return the GPRS data inactivity timer 'expiry' value. The timer represents the time elapsed since the last time there was PS data traffic – the timer resets to zero each time PS data traffic is received. After the timer expires, a background PLMN scan to find a home or more-preferred PLMN network occurs (note that while the scan is in progress, data traffic is blocked). **Usage:** • Execution: **AT!BPLMNTIMER=<timer>** Result: OK Purpose: Set the inactivity timer. • Query: **AT!BPLMNTIMER?** Response: !BPLMNTIMER: Recent Activity Timer (s): <timer> OK *or* !BPLMNTIMER: Recent Activity Timer is not set, default is used OK Purpose: Display the current <timer> setting. **Parameters:** <timer> (Data inactivity timer, in seconds) • Valid range: 0–65535 • Default: 45
!BSHWID Supporting chipsets: • All, with these exceptions: • MDM8200 (min f/w rev: M2.0 Release 1) • MSM6290 (min f/w rev: K1_1_1_10ap or K2_0_3_1ap) *Note: This command is not password-protected.*	**Return modem hardware ID** Return the modem's hardware ID number. **Usage:** • Query: **AT!BSHWID?** Response: <HW ID value> OK Purpose: Return the modem's <HW ID value>. **Parameters:** <HW ID value> (Modem's hardware ID) • Valid range: 0–63

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!BSMCCHECK Supporting chipsets: • All *Note: This command is not password-protected.*	**Return modem hardware version** Return the modem's hardware (board) version number based on the hardware resistor. A similar command, **^HVER**, returns the version number based on the FSN (see the *UMTS Modems Supported AT Command Reference, Document 2130617)*. For MC879xV, use **^HVER.** **Usage:** • Query: **AT!BSMCCHECK?** Response: <hwRev> OK Purpose: Return the modem's <HW ID value>. **Parameters:** <version> (Modem's hardware version) • Valid values: • "ENG1_1" – Modem uses an Eng 1.1 board • "ENG2_0" – Modem uses an Eng 2.0 board • "SPR1_D" – Modem uses an SPR 1D board • "SPR1_V" – Modem uses an SPR 1V board • "ENGXXX" – Modem uses a newer board that cannot be identified by the current firmware version • "PCB_ID_x"—MDM8200 devices only. ('x' is the hardware revision level)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!BZBUZZ Supporting chipsets: • MDM6200	**Generate frequency on buzzer pin** Generate a pre-defined frequency on the module's BUZZER_EN pin using the modem's internal PWM (pulse wave modulation) generator. *Note: If BUZZER_EN is configured as a PWM output, use this command to generate a specific frequency, or use !BZBUZZPLAY to generate a predefined melody.* **Usage:** • With BUZZER_EN configured as PWM output, use this command or !BZBUZZPLAY to control the signal: Execution: **AT!BZBUZZ=<status>, <freq>** Response: OK Purpose: Play (or stop playing) the frequency on the PWM output. • With BUZZER_EN configured as GPO (general purpose output), use this command to control the signal: Execution: **AT!BZBUZZ=<status>** Response: OK Purpose: Set the BUZZER_EN pin to high or low. • Query: **AT!BZBUZZ?** Response: !BZBUZZ: <status>[, <freq>] OK Purpose: Display current buzzer output state. • Query list: **AT!BZBUZZ=?** Purpose: Display valid values for <status> and <freq> parameters. **Parameters:** <status> (BUZZER_EN pin output state) • 0=Off (low voltage) • 1=On (high voltage) <freq> (Predefined frequency in Hz) • Valid values: • 0 • 32 • 64 • 85 • 128 • 171 • 195 • 256 • 293 • 391 • 512 • 586 • 781 • 1172 • 1563 • 2344 • 3125 • 4688

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!BZBUZZPLAY Supporting chipsets: • MDM6200	**Generate melody on buzzer pin** Generate a pre-defined melody on the module's BUZZER_EN pin using the modem's internal PWM (pulse wave modulation) generator. *Note: BUZZER_EN must be configured as a PWM output. To generate a frequency instead of a melody, use !BZBUZZPLAY.* **Usage:** • Query: **AT!BZBUZZPLAY=<melody>** Response: OK Purpose: Play the selected melody using the module's buzzer. **Parameters:** <melody> (Predefined melody) • Valid range: 1–7
!CMEN Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Enable/disable HSPA Compressed Mode stack functionality** Enable or disable HSDPA and HSUPA (for supporting modules) compressed mode functionality in the protocol stack. Typically, this command is used during testing of production networks. **Usage:** • Execution: **AT!CMEN=<enableDPA>[, <enableUPA>]** Response: OK Purpose: Enable or disable HSDPA and HSUPA compressed modes. • Query: **AT!CMEN?** Response: !CMEN: HSDPA Compressed Mode: <enableDPA> HSUPA Compressed Mode: <enableUPA> OK Purpose: Display current state of HSDPA and HSUPA compressed modes. If the parameters have not previously been set, the default values are returned. • Query list: **AT!CMEN=?** Purpose: Display valid values for <enableDPA> and <enableUPA> parameters. **Parameters:** <enableDPA> (Enable/disable HSDPA compressed mode) • 0 = Disable compressed mode • 1 = Enable compressed mode (Default) — This value is used for normal operations. <enableUPA> (Enable/disable HSUPA compressed mode) • 0 = Disable compressed mode • 1 = Enable compressed mode (Default) — This value is used for normal operations. • 2 = Enable compressed mode but don't broadcast

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CUSTOM Supporting chipsets: • All *Note: Some customizations may not be available for certain chipsets, firmware revisions, or devices.*	**Set/return customization settings** Set or return several customization values. **Usage:** • Execution: **AT!CUSTOM=<customization>, <value>** Response: OK Purpose: Assign <value> to a specific <customization> setting. • Query: **AT!CUSTOM?** Response: (list of enabled <customization>s) OK Purpose: Display customizations that are currently enabled. • Query list: **AT!CUSTOM=?** Purpose: Return a list of valid <customization> values. **Parameters:** <value> (Value being assigned to a specific <customization> setting) • Descriptions are included in each of the customizations described below. • Numeric value. Valid range depends on the <customization> type. <customization> (String identifying customization setting. The default value for all customizations is 0.) *Note: Use quotation marks around the customization string. For example, AT!CUSTOM="CSDOFF",0.* • "AUTONETWORKMODE"—Indicate if UE should revert to Automatic Network mode after 60 seconds of Manual Network mode. <value>: • 0 = Remain in Manual. • 1 = Revert to Automatic. • 2 = Remain in Manual if UE is attached to the network, otherwise switch to Automatic. • "CFUNPERSISTEN"—Enable/disable persistence (across power cycles) of AT+CFUN setting. <value>: • 0 = Disable (+CFUN setting does not persist across power cycle) • 1 = Enable (+CFUN setting persists across power cycle) • "CPASCWRINGIND"—Set incoming (RINGING) vs. in-progress call priority for +CPAS command. <value>: • 0 = Incoming call does not take priority over a call already in progress (Default) • 1 = Incoming call takes priority over a call already in progress • "CSDOFF"—Enable/disable ability of UE to initiate CS calls. <value> • 0 = Enable • 1 = Disable (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CUSTOM (continued)	**Set—query customization settings (continued)** • "CSVOICEREJECT"—Enable/disable ability to ignore incoming voice call pages on 3G channels. <value>: • 0 = Process pages as per 3GPP TS 24.008 specification (default) • 1 = Ignore paging (type 1 and 2) messages • 2 = Reject call setup (voice and circuit-switched VT), returning cause code 88 (Incompatible destination) • "DISFDNPDPCHK"—Enable/disable FDN check for PS data calls. <value>: • 0 = Allow FDN checking • 1 = Disable FDN when checking PDP activation number • 2 = Disable FDN when checking SMSC address • 3 = Disable FDN when checking PDP activation number and when checking SMSC address • "DISSTACK"—Enable/disable Adaptive Multi-Rate (AMR) codec <value> (Bitmask): • 00000000 = (Default) Enable AMR, including AMR-WB • 00001000 = Disable AMR, including AMR-WB • 00010000 = Disable AMR-WB only • "GPSENABLE"—Enable the Mobile Originated (MO) and/or Mobile-Terminated (MT) GPS feature (and present the NMEA port to the host). <value>: • 0 = Disable • 1 = MT & MO enabled • 2 = MO enabled only • 3 = MT enabled only • (MDM9200 3.0 release 2 or higher) 4 = NMEA port enabled; MT & MO enabled (unless GPS_DISABLE pin is asserted) • (MDM9200 3.0 release 2 or higher) 5 = NMEA port enabled; MO enabled (unless GPS_DISABLE pin is asserted) • (MDM9200 3.0 release 2 or higher) 6 = NMEA port enabled; MT enabled (unless GPS_DISABLE pin is asserted) • "GPSLPM"—Enable/disable GPS in Low Power Mode. <value>: • 0 = Enable (Default) • 1 = Disable • "GPSREFLOC"—Enable/disable reference GPS location reporting. <value>: • 0 = Enable (Default) • 1 = Disable (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CUSTOM (continued)	**Set/query customization settings (continued)** • "GPSSEL"—Select GPS antenna (useful only for devices with both a GPS and a shared GPS/Rx diversity antenna). <value>: • 0 = Use dedicated GPS antenna (Default) • 1 = Use shared GPS/Rx diversity antenna • 2 = Use dedicated GPS antenna, with bias voltage disabled • "GPSSUPLSETID"—Value used in the SUPL POS INIT message's SET ID field (Note: Not supported by MDM9200) <value>: • 0 = IMSI (Default) • 1 = MSISDN • "HPPLMNSCDIS"—Set HPLMN scan constraints. <value>: • 0 = Include foreign MCC • 1 = Reject foreign MCC • "HSDPATEST"—Enable/disable HSDPA test channel interface. <value>: • 0 = Disable • 1 = Enable • "HSICON"—Control HSPA icon behavior. <value>: • 0 = Maintain HSPA icon if the bearer is revoked but remains on the same cell • 1 = Revert to the UMTS icon if the bearer is revoked • "HWCFGLOCK"—Prevent UART reconfiguration. <value>: • 0 = Disable (UART cannot be reconfigured) • 1 = Enable (UART can be reconfigured) • "ISVOICEN"—Enable/disable voice functionality. <value>: • 0 = Disable voice-related CnS objects • 1 = Enable voice-related CnS objects • 2 = Disable voice on both CnS and AT interfaces • (Note: Voice functionality is available on the AT interface when <value> = 0 or 1.) • "MEPCODE"—Enable/disable prompt for MEP code when incorrect SIM is inserted. <value>: • 0 = Disable • 1 = Enable (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CUSTOM (continued)	**Set/query customization settings (continued)** • "MEPLOCK"—Set MEP locking status. <value>: • 0, 2, 4, 6, 8, 10, 12, 14, where <value> is represented by Bits 0–7: • Bit 0: Reserved, always 0 • Bit 1: Network personalization: • 0 = Not permanent • 1 = Permanent • Bit 2: Network subset personalization: • 0 = Not permanent • 1 = Permanent • Bit 3: Service provider personalization: • 0 = Not permanent • 1 = Permanent • Bits 4–7: Reserved, always 0 • Example: If <value> = 0, locking can be removed by the host using +CPIN or +CLCK with the correct unlock code. If <value> = 2 (permanent network personalization), locking cannot be removed. • "MUXMODE"—Override the MUX mode setting in the USB descriptor. (Note: Not supported by MDM9200) <value>: • 0 = Do not override • 1 = Enable MUX mode • 2 = Disable MUX mode • "NOGPRS"—Enable/disable indicator display (GPRS, EDGE, WCDMA, etc.). <value>: • 0 = Enable • 1 = Disable • "NOROAM"—Enable/disable roaming indicator display. <value>: • 0 = Enable • 1 = Disable • "PCSCDISABLE"—Determine functionality of PCSC, GSM Algorithm and Authenticate commands, and +CIMI command. <value>: • 0–7 (Default value = 0—all functions enabled) • Bit 0: PCSC (0=Enable, 1=Disable) • Bit 1: GSM Algorithm and Authenticate commands (0=Enable, 1=Disable) • Bit 2: AT+CIMI outputs IMSI (0=Enable, 1=Disable) • "PPPPROFAUTH"—Indicate source of authorization information (username, password) for PPP session. (Note: Not supported by MDM9200) <value>: • 0 = (Default) Use profile 1, and obtain username/password from host • 1 = Obtain username/password from default profile • 2 = Obtain username/password from profile #1 (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CUSTOM (continued)	**Set/query customization settings (continued)** • "PRLREGION"—Region-specific scanning algorithm. <value>: • 0 = Default (internal) • 1 = Europe/rest of the world • 2 = North America • 3 = Australia • 4 = Japan • 5 or greater = Reserved • "PUKPRMPT"—Indicate if host will prompt for PUK code if maximum number of PIN unlock retries is exceeded. <value>: • 0 = Don't prompt • 1 = Prompt • "RADIORESET"—Indicate if modem should reset when coming out of Low Power Mode (that is, if **+CFUN=1**). (Note: Not supported by MDM9200) <value>: • 0 = No reset (default) • 1 = No reset • 2 = Reset • "RFINDENABLE" (MC8795V with GPS-enabled only)—Enable/disable RF indicator for DIO-3 and DIO-4, showing GPS ON/OFF status and operating band. <value>: • 0 = Disable (Default) • 1 = Enable • When enabled: DIO-3 (GPS status): 0 = GPS disabled; 1 = GPS enabled DIO-4 (Operating band): 0 = 850/1900; 1 = 900/1800/2100 • "SCANPROF"—Enable/disable profile scan (try all profiles configured on card until successful connection is found; the activated context becomes the new default). <value>: • 0 = Disable • 1 = Enable • "SIMLPM"—Indicate default SIM power state during Low Power Mode. <value>: • 0 = Default (device-dependent) • 1 = Do not power down SIM in LPM • 2 = Power down SIM in LPM (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CUSTOM (continued)	**Set/query customization settings (continued)** • "SKUID"—*(Deprecated: Use !PRIID instead of this customization)* Assign a unique SKU ID to the modem. <value>: • Valid range: 0–255 • "STARTLPM"—Enable/disable LPM (Low Power Mode) as the startup mode for the user equipment. <value>: • 0 = Disabled • 1 = Enabled (start UE in LPM) • "STKUIEN"—Enable/disable SIM toolkit UI. • MDM9200 (while in QMI mode) (f/w rev: SWI9200X_3.0-Release2, SWI9200M_3.5-Beta3) <value>: • 0, 1 = Enable for QMI interface • 2 = Enable for AT interface • All other chipsets (including MDM9200 not in QMI mode): <value>: • 0 = Disable • 1 = Enable for CnS interface (AT disabled) • 2 = Enable for AT interface (CnS disabled) • "TRUFLOWDISABLE"—Enable/disable TRU-Flow <value>: • 0 = TRU-Flow is enabled (default) • 1 = TRU-Flow is disabled • "USBDMDISABLE"—Enable/disable the USB DM port. <value>: • 0 = Enable (default) • 1 = Disable • "USBSERIALENABLE"—Use IMEI as USB serial number. <value>: • 0 = Do not use IMEI as USB serial number (default) • 1 = Use IMEI as USB serial number • "WAKEHOSTEN"—Enable/disable host wake-up via SMS or incoming data packet. <value>: • 0 = Disable (Host will not wake when SMS or incoming data packet is received) • 1 = Wake host when simple SMS is received. • 2 = Wake host when incoming data packet is received. • "WAPPUSHDATA"—Enable/disable processing of WAP Push data. <value>: • 0 = Disable • 1 = Enable

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!CWSETUP Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MSM6290 (min f/w rev: K2.0 Release 2) • QSC6270 (min f/w rev: S2.0) *Note: This command is not password protected.*	**Set Connection Watchdog parameters** Set the parameters used by the connection watchdog (CW) process. **Usage:** • Execution: **AT!CWSETUP=<mode>, <pingIP>, <dnsName>, <rxInt>, <valInt>, <reset>** Response: OK *or* ERROR Purpose: Set new CW parameters as specified. • Query: **AT!CWSETUP?** Response: AT!CWSETUP:<mode>, <pingIP>, <dnsName>, <rxInt>, <valInt>, <reset> OK Purpose: Return current CW parameters. **Parameters:** <mode> (CW operation mode) • 0 = Disabled • 1 = Uses ping method • 2 = Uses DNS lookup method <pingIP> (Ping server IP address) • Standard IP address format. For example, 192.168.0.255 <dnsName> (Domain name for DNS lookup) • Length: Up to 128 characters • Example: “www.sierrawireless.com” • If <mode=1>, the name can be a null string (“”) <rxInt> (Interval between checks for new received data) • The number of minutes the modem waits between checks for new received data. • Valid range: 1–1440 <valInt> (Interval between CW method validation attempts) • The number of minutes the modem waits between attempts to validate the CW method, until successful. • Valid range: 1–110 <reset> (Reset required on bad connection detection) • 0 = Reestablish data connection when CW detects a bad connection (Reset not required). • 1 = Reset modem when CW detects a bad connection.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!CWSTATS
Supporting chipsets:
• MDM6200
• MDM6270 (min f/w rev: S2.0)
• MSM6290 (min f/w rev: K2.0 Release 2)
• QSC6270 (min f/w rev: S2.0)
Note: This command is not password protected.
	Query/clear Connection Watchdog statistics
	Returns connection watchdog (CW) statistics collected since the last time the device powered up, or since the statistics were cleared.
	This command also used to clear the statistics.
	Usage:
	• Query: **AT!CWSTATS?**
	Response: State: <state>
	Check Counter: <checkCounter>
	Reset/Disconnect Count: <resetCounter>
	Purpose: Return current CW statistics.
	• Execution: **AT!CWSTATS=0**
	Response: OK
	or ERROR
	Purpose: Reset <counter> and <resetCounter> to 0.
	Parameters:
	<state> (Current CW state)
	• Valid values:
	  • Off
	  • Validating
	  • Active
	<checkCounter> (Number of times CW checked connection (via ping/DNS lookup) since last power-up)
	• Range: 0–65535
	<resetCounter> (Number of times CW caused a reset or disconnect)
	• Range: 0–65535

!DARPEN Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Enable/disable DARP for SAIC** Enable or disable Downlink Advanced Receiver Performance (DARP) for Single-Antenna Interference Cancellation (SAIC). **Usage:** • Execution: **AT!DARPEN=<enableFlag>** Response: OK Purpose: Enable or disable SAIC-DARP. • Query: **AT!DARPEN?** Response: !DARPEN: <enableFlag> OK Purpose: Display the current <enableFlag> setting—this shows whether SAIC-DARP is enabled or disabled. If the command returns ERROR, SAIC-DARP is assumed to be enabled. • Query list: **AT!DARPEN=?** Purpose: Display a list of valid <enableFlag> values. **Parameters:** <enableFlag> (Enable/disable SAIC-DARP mode) • 0 = Disable SAIC-DARP • 1 = Enable SAIC-DARP (Default) — This value is used for normal operations.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!DIO Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MSM6290 (min f/w rev: K1_1_1_3ap) • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Read/write from/to Digital I/O (DIO) channel** Read from or write to a DIO channel that has been configured and enabled using !DIOCFG. **Usage:** • Execution: **AT!DIO=<channel>, <output value>]** Response: OK Purpose: Output (write) a logic low or logic high to the specified DIO channel. • Query: **AT!DIO?<channel>** Response: !DIO: <channel>,<input value> OK Purpose: Read the logic level at the specified DIO channel (the last value written to the DIO channel). • Query list: **AT!DIO=?** Purpose: Display the 'write' command format and allowed parameter values. **Parameters:** <channel> (Digital I/O channel) • Valid range: 1–[number of channels available] (See !DIOCFG for channel details.) <input value> (Logic level on specified <channel>) • 0 = Logic low • 1 = Logic high <output value> (Logic level output to specified <channel>) • 0 = Output logic low • 1 = Output logic high

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!DIOCFG Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MSM6290 (min f/w rev: K1_1_1_3ap) • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Configure DIO channels** Configure specific GPIOs for DIO (digital I/O) operation. • If a DIO channel is disabled, the GPIO maintains its default functionality. • After a channel is configured and enabled, it can be used for input or output (as appropriate) using !DIO. See the tables below for channel configuration details. • Input DIO channels—Choice to register level change notifications associated with the corresponding DIO channels is specified as part of the configuration. • Output DIO channels—Initial value to write to the DIO channel is specified as part of the configuration. **MSM6290-based devices** Input channels can be configured for pull-up (P-Up) or pull-down (P-Dn) operation—some restrictions apply to GPIOs connected to external resistors.

DIO channel	Host connector pin	GPIO	Input			Output
			Pull-up	Pull-down	Rext[a]	
1	MIO_06	32	✔[b]		100k	✔[b]
2	MIO_16	28	✔[b]	✔[b]	*[c]	✔[b]
3	MIO_28	43	✔[b]	✔[b]		✔[b]
4	MIO_30	27	✔[b]		2k2	✔[b]

a. External pull-up resistor
b. Supports programmable option
c. Some development kits have this pin pulled up via 10K for SD support

MDM6200-based devices

DIO channel	Host connector pin	GPIO
1[a]	GPIO_0	87
2	GPIO_1	77
3	GPIO_2	76
4	GPIO_3	75

a. Cannot program as input or output if Tx burst indication has been enabled by +WTBI.

MDM6270/QSC6270-based devices

DIO channel	Host connector pin	GPIO
1	GPIO_0	26
2	GPIO_1	25
3	GPIO_2	24
4	BUZZER_EN	29

(Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!DIOCFG (continued)	**Configure DIO channels (continued)**
	Usage:
	• Execution: First use: **AT!DIOCFG=<channel>, <enable>, <type>, <initval/notify>**
	Second use: **AT!DIOCFG=<channel>, <enable>**
	Response: OK
	or ERROR
	Purpose: Configure the specified <channel> (all parameters required), or enable/disable the already configured <channel> (only <channel> and <enable> are required).
	An ERROR is returned if the specified DIO channel is not configured.
	• Query: **AT!DIOCFG?<channel>**
	Response: !DIOCFG: <channel>,<enable>,<type>,<initval/notify>
	or ERROR *(Unconfigured channel or channel out of range)*
	Purpose: Display the current configuration for the specified <channel>.
	• Query list: **AT!DIOCFG=?**
	Purpose: Display the execution command format and allowed parameter values.
	Parameters:
	<channel> (DIO channel)
	• Valid range: 1–[number of channels available]
	<enable> (Enable/disable DIO channel flag)
	• 0 = Disable
	• 1 = Enable
	<type> (DIO channel Input/output type)
	• 0 = Output
	• 1 = Input with pull-up resistor
	• 2 = Input with pull-down resistor
	<initval/notify> (Initial output value, or input change notification option)
	• Inputs
	  • 0 = Disable input level change notifications
	  • 1 = Enable input level change notifications
	• Outputs
	  • 0 = Set output to logic low at power-up
	  • 1 = Set output to logic high at power-up

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!DTMEN Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Enable/disable Dual Transfer Mode stack functionality** Enable or disable Dual Transfer Mode (DTM) and Enhanced DTM (EDTM) functionality in the stack. *Note: The command is only available on devices that support DTM. The second parameter (<enableEDTM>) is only available if EDTM is also supported.* **Usage:** • Execution: **AT!DTMEN=<enableDTM>** *(if EDTM is not supported)* *or* **AT!DTMEN=<enableDTM, <enableEDTM>** *(if EDTM is not supported)* Response: OK Purpose: Enables or disables DTM, and EDTM (if supported). • Query: **AT!DTMEN?** Response: !DTMEN: *(if EDTM is not supported)* <enableDTM> OK *or* !DTMEN: *(if EDTM is supported)* DTM: 01 EDTM: 01 OK Purpose: Indicates the current state (disabled/enabled) of DTM and, if supported, EDTM support. If the command returns ERROR, DTM and EDTM are assumed to be enabled. • Query list: **AT!DTMEN=?** Purpose: Returns a list of valid <enableDTM> and, if supported, valid <enableEDTM> values. **Parameters:** <enableDTM> (Enable/disable Dual Transfer Mode) • 0 = Disable DTM • 1 = Enable DTM (Default) — Value used for normal operations. <enableEDTM> (Enable/disable Enhanced Dual Transfer Mode) • 0 = Disable EDTM • 1 = Enable EDTM (Default) — Value used for normal operations.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!EDAEN Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Configure protocol stack for EDA** Enable or disable EDA (Extended Dynamic Allocation) functionality in the stack. *Note: This command is only supported on devices that support EDA.* **Usage:** • Execution: **AT!EDAEN=<enableFlag>** Response: OK Purpose: Enable or disable EDA. • Query: **AT!EDAEN?** Response: !EDAEN: <enableFlag> OK Purpose: Display the current <enableFlag> setting—this shows whether EDA is enabled or disabled. If ERROR is returned, assume that EDA is enabled. • Query list: **AT!EDAEN=?** Purpose: Return a list of valid <enableFlag> values. **Parameters:** <enableFlag> (Enable/disable EDA) • 0 = Disable • 1 = Enable (Default) — This value is used for normal operations.
!EQEN Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Enable/disable WCDMA L1 equalizer** Enable or disable the modem's WCDMA L1 equalizer. **Usage:** • Execution: **AT!EQEN=<enableFlag>** Response: OK Purpose: Enable/disable the L1 equalizer. • Query: **AT!EQEN?** Response: !EQEN: <enableFlag> OK Purpose: Return the current <enableFlag> value. • Query List: **AT!EQEN=?** Purpose: Return a list of supported <enableFlag> values. **Parameters:** <enableFlag> (Enable/disable L1 equalizer) • 0 = Disable • 1 = Enable (Default)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!GCFEN Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Enable/disable GCF test mode** Place the modem in GCF testing mode or normal operating mode. **Usage:** • Execution: **AT!GCFEN=<enableFlag>** Response: OK Purpose: Place the modem in GCF testing mode or normal operating mode. • Query: **AT!GCFEN?** Response: !GCFEN: <enableFlag> OK Purpose: Display the modem's current mode. • Query List: **AT!GCFEN=?** Purpose: Return a list of supported <enableFlag> values. **Parameters:** <enableFlag> (Enable/disable GCF testing) • 0 = Disable GCF test mode (Default) — This value is used for normal operations. • 1 = Enable GCF test mode.
!GCFUIMTYPE Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Set/return current SIM type** Indicate (for GCF testing) the type of SIM that is installed in the module. **Usage:** • Query: **AT!GCFUIMTYPE?** Response: !GCFUIMTYPE: <simType> Purpose: Return the type of SIM that is installed in the module (the current <simType> value). • Query list: **AT!GCFUIMTYPE=?** Purpose: Return a list of supported SIM types. • Execution: **AT!GCFUIMTYPE=<simType>** Response: OK Purpose: Indicate the type of SIM that is installed—the SIM type (2G SIM or 3G USIM) determines how the module behaves for GCF testing and normal operation. During GCF testing, using the 2G SIM type enables the module to pass some tests that cannot be passed using the 3G USIM type. **Parameters:** <simType> (Installed SIM type) • 0 = 2G SIM — This value is required to pass GCF testing. • 1 = 3G USIM (Default) — This value should be used for normal operations.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!HSDCAT Supporting chipsets: • All *Note: This command is not password-protected.*	**Set/return HSDPA category** Indicate the UE's current HSDPA category. **Usage:** • Execution: **AT!HSDCAT=<category>** Response: OK Purpose: Set the HSDPA category. • Query: **AT!HSDCAT?** Response: !HSDCAT: <category> OK Purpose: Return the current HSDPA <category> • Query list: **AT!HSDCAT=?** Purpose: Return a list of supported <category> values. **Parameters:** <category> (HSDPA category) • Valid values: 6, 8, 12 (For descriptions of HSDPA categories, see Table C-1 on page 277.) *Note: Older model UEs do not support the higher-speed HSDPA categories.*
!HSUCAT Supporting chipsets: • MDM6200 • MDM8200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 *Note: This command is not password-protected.*	**Set/report HSUPA category** Indicate the UE's current HSUPA category. **Usage:** • Execution: **AT!HSUCAT=<category>** Response: OK Purpose: Set the HSUPA category. • Query: **AT!HSUCAT?** Response: !HSUCAT: <category> OK Purpose: Return the current HSUPA <category> • Query list: **AT!HSUCAT=?** Purpose: Return a list of supported <category> values. **Parameters:** <category> (HSUPA category) • Valid values: 3, 5 (For descriptions of HSUPA categories, see Table C-2 on page 277.) *Note: Older model UEs do not support HSUPA.*

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!LEDCTRL Supporting chipsets: • All, with these exceptions: • MSM6290 (min f/w rev: K1_1_0_5ap) *Note: MDM6200, MDM6270, MSM6290, and QSC6270 include <invert> parameter.*	**Set/report LED control pattern** Control the pattern-flashing behavior of the LED. See the device's Product Specification or Product Technical Specification document for default behavior. **Usage:** • Execution: **AT!LEDCTRL=<index>, <period>,<ontime>,<invert>** Response: OK Purpose: Set the LED behavior for the indicated pattern and indicate if the on/off pattern should be inverted. Note: <invert> may not be available on some chipsets. • Query: **AT!LEDCTRL?** Response: <index1>:<period>,<ontime>,<invert> ... <indexn>:<period>,<ontime>,<invert> Purpose: Return a list of supported patterns. Note: <invert> may not be available on some chipsets. **Parameters:** <index> (Pattern number) • 0 = OFF pattern • 1 = Searching for service • 2 = Attached to network • 3 = Connected (has an active context) • 4 = Low power (airplane) mode • 5 = Connected and transferring data (has an active context, and packet switched data is being passed). <period> (Length of repeating pattern) • <period> ≥ <ontime> • Unit of measurement: 100 ms <ontime> (LED stays on for this amount of time per <period>) • <ontime> ≤ <period> • Unit of measurement: 100 ms <invert> (Keep or reverse the on/off LED pattern) • 0 = Display normal LED pattern • 1 = Invert the LED pattern (<period> - <ontime>). For example, if the LED is normally on for 50 ms and off for 25 ms, it is now off for 50 ms and on for 25 ms. **Examples:** • AT!LEDCTRL=1,52,50 While searching for a network, the LED stays on for 5 seconds out of every 5.2 seconds. • AT!LEDCTRL=3,5,4 While in a call, the LED stays on for 0.4 seconds out of every 0.5 seconds. • AT!LEDCTRL=3,5,4,0 Same behavior. LED stays on for 0.4 seconds out of every 0.5 seconds. • AT!LEDCTRL=3,5,4,1 Invert the LED behavior. The LED stays off for 0.4 out of every 0.5 seconds.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!LTENAS Supporting chipsets: • MDM8220 • MDM9200 • MDM9600 *Note: This command is not password-protected.*	**Configure LTE NAS settings** Configure LTE NAS-related settings. **Usage:** • Execution: **AT!LTENAS=<bitmask><FPLMNtime>** Response: <bitmask> <FPLMNtime> Purpose: Set LTE NAS-related settings. • Query: **AT!LTENAS?** Response: LSTI: <lsti> GERAN Cap: <geran> Disable GUTI Security check: <guti> Temp Forbidden PLMN: <FPLMNtime> OK Purpose: Report the current settings. • Query List: **AT!LTENAS=?** Purpose: Return the command format and the supported parameter values. **Parameters:** <bitmask> (8-bit mask that identifies LTE NAS-related settings) • Displayed/entered as hexadecimal value • Bit 0: <lsti> • Bit 1: <geran> • Bit 2: <guti> <FPLMNtime> (Temporary Forbidden PLMN backoff time) • Displayed/entered as hexadecimal value • 0x0=Disable feature (Default) • 0x1–0xFFFFFFFE=Backoff time (in ms) • 0xFFFFFFFF=Use the user equipment's Timer T3402 time as backoff time <lsti> (LTE/SAE Trial Initiative) • 0=Disabled • 1=Enabled <geran> (GERAN Cap) • 0=Disabled • 1=Enabled <guti> (GUTI and NAS security check) • 0=Disabled • 1=Enabled

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!MAPMTPDP
Supporting chipsets:
• All, with these exceptions:
  • MDM8200 (min f/w rev: M2.0 Release 1)
  • MSM6290 (min f/w rev: K1.1 Release 2)
 	
Note: This command is not password-protected.
	Configure MT PDP port mapping
	Configure the SIO port mapping for the mobile terminated (MT) PDP context.
	MT PDP can be mapped over:
	• AT command port—The modem alerts the host by sending a RING. The number of rings is set using the **!S0** command. The host must respond with ATA within 5 seconds, otherwise the modem will reject the MT PDP call.
	• NDIS—The modem alerts the NDIS driver of the MT PDP call.
	
	Usage:
	• Execution: **AT!MAPMTPDP=<service>**
	Response: OK
	Purpose: Map the MT PDP context to the specified service. This takes effect after the modem is reset.
	• Query: **AT!MAPMTPDP?**
	Response: !MAPMTPDP: <service>
	OK
	Purpose: Reports the current service mapping.
	• Query List: **AT!MAPMTPDP=?**
	Purpose: Return the command format and the supported <service> values.
	
	Parameters:
	<service> (Supported service)
	• 0 = MT PDP disabled
	• 1 = AT command
	• 2 = Reserved
	• 3 = NDIS (default)
	• 4 = Reserved
	• 5 = PDP1 service (not supported)
	• 6 = PDP2 service (not supported)
	• 7 = PDP3 service (not supported)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!MAPUART Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 (min f/w rev: K1.1 Release 1) • QSC6270 (min f/w rev: S2.0)	**Map UART interface to a service** Map the UART interface to a service (in MUX or non-MUX mode). Any change to the service mapping takes effect after the modem is reset. Once the UART interface is mapped, an application can access the functions offered by the service over the UART interface. **Usage:** • Execution: **AT!MAPUART=<service>** Response: OK *or* ERROR *(unsupported service)* Purpose: Map a supported service to the UART interface. • Query: **AT!MAPUART?** Response: !MAPUART: <service> OK Purpose: Report which service is currently mapped to the UART interface. • Query List: **AT!MAPUART=?** Purpose: Return the command format and the supported <service> values. **Parameters:** <service> (Supported services) • 0 = No service mapped; UART disabled • 1 = AT command processor/data service (MUX/non-MUX) • 2 = Diagnostic Message (DM) service (MUX/non-MUX) • 3 = Reserved • 4 = NMEA (GPS) service (non-MUX) • 5 = PDP1 service (non-MUX) • 6 = PDP2 service (non-MUX) • 7 = PDP3 service (non-MUX) (Default)
!MXPORTMAP Supporting chipsets: • MDM6200 • MDM6270 • MSM6290 • QSC6270	**Set/report MUX mode port mappings** Set the modem's MUX-mode DLCI port mappings for AT and PDP2. **Usage:** • Query: **AT!MXPORTMAP?** Response: <mode> OK Purpose: Report the current port mapping. • Execution: **AT!MUXMODE=<mode>** Response: OK Purpose: Select the new port mapping. **Parameters:** <mode> (Port map used in MUX-mode) • 00 = Default AT uses DLCI port 2 PDP2 uses DLCI port 6 • 01 = Carrier-specific AT uses DLCI port 6 PDP2 uses DLCI port 2

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NASREL Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.*	**Set/report supported NAS release compliance version** Configure the modem to support a specific NAS (Non-Access Stratum) release compliance version. **Usage:** • Execution: **AT!NASREL=<nasrel>** Response: OK Purpose: Set the desired version (<nasrel>). • Query: **AT!NASREL?** Response: !NASREL: NAS Release: Release 5 *(or Release 99)* OK Purpose: Report the NAS release compliance version currently being used. • Query List: **AT!NASREL=?** Purpose: Return the command format (for !NASREL =) and the supported parameter values. **Parameters:** <nasrel> (NAS release compliance version) • 00 = Release 99 (Default) • 01 = Release 5 *Note: If you use **!REL**, use the default value (1) for that command's <sgsnr> and <mscr> parameters. You must choose the appropriate compliance version using **!NASREL**.*

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NVENCRYPTIMEI Supporting chipsets: • All	**Write unencrypted IMEI to modem** Write an unencrypted IMEI to a modem *if* the modem does not already have an IMEI—the command can only be used once per modem. The IMEI is a fifteen digit string formed by concatenating the following elements: TAC code (8 digits) SN (Serial number) (6 digits) CheckDigit (1 digit calculated from TAC code and SN) The CheckDigit is calculated as follows: **1.** Label the fourteen digits in the TAC and SN as: TAC: D14..D7 SN: D6..D1 For example: TAC = 12345678 (‘1’ is D14, ‘8’ is D7) SN = 901234 (‘9’ is D6, ‘4’ is D1) **2.** Double the value of each odd-labeled digit (D13, D11, ..., D1). **3.** Add the values of each individual digit from the result of Step 2. **4.** Add the even-labeled digits (D14, D12, ..., D2) to the result of Step 3. **5.** Check the last digit of the result of Step 4. If it is ‘0’, the CheckDigit is 0; if it is not ‘0’, subtract it from 10 to get the CheckDigit. For example: TAC (12345678) SN (901234) Step 1: Label the digits of the TAC and SN. D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 1 2 3 4 5 6 7 8 9 0 1 2 3 4 Step 2: Double the odd-labeled values: D14 **D13** D12 **D11** D10 **D9** D8 **D7** D6 **D5** D4 **D3** D2 **D1** 1 **4** 3 **8** 5 **12** 7 **16** 9 **0** 1 **4** 3 **8** Step 3: Add *each* digit of the odd-labeled values: 4 + 8 + (1 + 2) + (1 + 6) + 0 + 4 + 8 = 34 Step 4: Add each digit of the even-labeled values to the Step 3 total: 1 + 3 + 5 + 7 + 9 + 1 + 3 +34 = 63 Step 5: Check last digit of Step 4 total. CheckDigit = 10 - 3 = 7 Result: IMEI = TAC:SN:CheckDigit = 123456789012347 **Usage:** • Execution: **AT!NVENCRYPTIMEI=<P1>, <P2>, <P3>, <P4>, <P5>, <P6>, <P7>, <P8>** Response: OK Purpose: Write the unencrypted IMEI to the modem. (Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NVENCRYPTIMEI (continued)	**Write unencrypted IMEI to modem (continued)** **Parameters:** <P1> to <P8> (IMEI segments) • <P1> = IMEI[0..1]; <P2> = IMEI[2..3]; ...; <P8> = IMEI[14..15] • <P1> to <P4> represent the TAC • <P5> to <P7> represent the SNR • <P8> represents the CheckDigit plus a padding digit (‘0’) **Example:** Using the example IMEI shown above: AT!NVENCRYPTIMEI=12,34,56,78,90,12,34,70
!NVMUXMODE Supporting chipsets: • MDM6270 • QSC6270	**Set/report USB descriptor MUX mode** Change the mode (MUX/non-MUX) in which the Windows driver operates. In non-MUX mode, supported ports are: • Data (endpoint 2), HIP carrying CnS and DM (endpoint 4), and AT (endpoint 5). These default port assignments can be changed using the **!NVPORTSET** *(UMTS Modems Supported AT Command Reference, Document 2130617)* command. In MUX mode, services are multiplexed over endpoint 2. Supported ports are AT (DLCI-1), DM (DLCI-2), HIP carrying CnS (DLCI-3), and Data (DLCI-5). NMEA is also supported on DLCI-4 for GPS-enabled modules. *Note: This command requires the following minimum Windows driver revisions:* *- USB: 2.2.1.0* *- NDIS: 2.2.0.0* **Usage:** • Query List: **AT!NVMUXMODE=?** Purpose: Return a list of valid <mode> values. • Query: **AT!NVMUXMODE?** Response: <mode> OK Purpose: Return the current <mode> setting. • Execution: **AT!NVMUXMODE=<mode>** Response: OK Purpose: Set MUX mode on or off, or indicate that the current setting should be used. **Parameters:** <mode> (MUX mode state) • 0 = Default. Use the current MUX mode setting in the USB descriptor • 1 = Windows driver operates in MUX mode • 2 = Windows driver operates in non-MUX mode

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NVNSCODE Supporting chipsets: • All, with these exceptions: • MSM6290 (min f/w rev: K1_0_2_11ap)	**Return Network Subset codes** Return the provisioned list of Network Subset codes used for Network Subset Personalization locking. **Usage:** • Query: **AT!NVNSCODE?** Response: (list of <NSCode> items) OK Purpose: Return the list of Network Subset codes used for Network Subset Personalization locking. • Execution: **AT!NVNSCODE=<NSCode>** Response: OK Purpose: Add a network subset code to the list used for Network Subset Personalization locking. **Parameters:** <NSCode> (Network Subset Code) • Format: 99—Two digit numeric code

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NVOEM Supporting chipsets: • All *Note: The read version (!NVOEM?<item>) of this command is not password-protected.*	**Set/report values of non-volatile (NV) memory items** Read or write the values of specific modem operation values held in NV memory. **Usage:** • Query List: **AT!NVOEM=?** Purpose: Return a list of NV items that can be read or written. • Query: **AT!NVOEM?<item>** Response: <value1> ... <valuen> OK Purpose: Return the current configuration of the specified <item> in 1-byte <value>s (displayed as hexadecimal values) • Execution: **AT!NVOEM=<item>, <value1>, ..., <valuen>** Response: OK Purpose: Write the specified configuration <values> (1-byte each) to the NV <item> **Parameters:** <item> (A supported NV item) • Valid values: • "GMSCLASS"—GPRS MS class • "EMSCLASS" — EDGE MS class • "FTM_MODE" — FTM enable • "GERANFP1" — GERAN FP 1 enable • "GSMA5ALG" — A5 encryption support • "GEAALG" — GEA encryption support *Note: Do not type the quotation marks when you enter the command.* <value> (Single byte of NV configuration item) • Represented as hexadecimal ASCII • Valid range: '00'–'FF' **Example 1 – Read:** Input: AT!NVOEM?GMSCLASS Output: 0C OK **Example 2 – Assign:** Input: AT!NVOEM=GMSCLASS, 0C Output: OK

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NVPLMN Supporting chipsets: • All	**Provision PLMN list for Network Personalization locking** Provision the list of PLMN (MCC/MNC pairs) used for Network Personalization locking. **Usage:** • Query: **AT!NVPLMN?** Response: <MCC> <MNC> ... OK Purpose: Return a list of NV items that can be read or written. • Execution: **AT!NVPLMN=<MCC1>, <MNC1>, ..., <MCCn>, <MNCn>** Response: OK Purpose: Add up to six MCC/MNC pairs to the PLMN list Note: Execution can be performed one time only (all MCC/MNC pairs must be set at the same time). **Parameters:** <MCC> (Mobile Country Code) • 3 digits <MNC> (Mobile Network Code) • 2 digits *Note: This command has no effect on operations when using Sierra Wireless' 27.010 MUX mode drivers.*

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!NVPORTMAP Supporting chipsets: • MSM6290 *Note: The modem must be in online mode to use this command.*	**/Change modem port mappings in non-MUX mode** Change the modem port mappings in non-MUX mode—the modem must be in online mode (not FTM mode) for this to work. **Usage:** • Execution: **AT!NVPORTMAP=<normMode>[, <diagMode>]** Response: OK Purpose: Set normal mode and diagnostic mode port mappings. • Query: **AT!NVPORTMAP?** Response: <normMode>,<diagMode> OK Purpose: Return current port mappings for normal mode and diagnostic mode. **Parameters:** <normMode> (Port map in normal mode) • 00 = AT→UART • 04 = AT→EP2, DM→UART • 05 = AT→EP5, Data→EP2 (Recommended default) • 06 = AT→EP2, Data→EP5 <diagMode> (Port map in diagnostic mode (optional)) • 00 = AT→UART • 04 = AT→EP2, DM→UART • 05 = AT→EP5, Data→EP2 • 06 = AT→EP2, Data→EP5 *Note: This command has no effect on operations when using Sierra Wireless' 27.010 MUX mode drivers.*
!NVSPCODE Supporting chipsets: • All, with these exceptions: • MSM6290 (min f/w rev: K2_0_7_24ap)	**Provision Network Service Provider code list** Provisions the list of Network Service Provider codes that are used for Network Service Provider Personalization locking. **Usage:** • Query: **AT!NVSPCODE?** Response: SP Code: (list of <SP Code> values) OK Purpose: Provision the list of Network Service Provider codes. • Execution: **AT!NVSPCODE=<SP Code>** Response: OK Purpose: Add a Network Service Provider code to the list. **Parameters:** <SP Code> (Service Provider Code) • Format: 1–2 digits

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!PACKAGE Supporting chipsets: • MDM9600 *Note: This command is not password-protected.*	**Return package version string** This command returns the package version loaded in the modem. **Usage:** • Query: **AT!PACKAGE?** Response: !PACKAGE:<versionString> OK Purpose: Return the package version string. **Parameters:** <versionString> • Character string • Example: MC7750_01.00.02.03_00_VZW_011.006_000

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!PCINFO
Supporting chipsets:
• All
 	
Note: This command is not password-protected.
	Return power control status information
	Return the modem's power control status information.
	Usage:
	• Query: **AT!PCINFO?**
	Response: State: <state>
	LPM force flags: W_DISABLE: <0|1>, User:<0|1>, Temp:<0|1>, Volt:<0|1> *(0=Did not cause, 1=Caused)*
	W_DISABLE: <w_disable>
	Poweroff mode: <pwroff>
	User initiated LPM: <userlpm>
	OK
	Purpose: Return power control information.
	
	Note: In firmware revisions D0_0_4_1ap and earlier, "Poweroff enabled: <pwroff>" replaces "Poweroff mode: <pwroff>"
	
	Parameters:
	<state> (The modem's power mode)
	• 0 = Low Power Mode (LPM)
	• 1 = Online
	• 2 = Offline
	• 3 = Power off (internal)
	• 4 = Initialization (internal)
	
	<LPM force flags> (Conditions that caused modem to enter LPM. 0=did not cause, 1 = caused)
	• W_DISABLE: W_DISABLE is asserted
	• USER: CnS/AT command was issued
	• TEMP: Temperature is outside operational limits
	• VOLT: Voltage is outside operational limits
	
	<w_disable> (Current state of W_DISABLE)
	• 0 = De-asserted
	• 1 = Asserted
	
	<pwroff>
	• Firmware revisions D0_0_4_1ap and earlier: State of Power Off Enable feature—preset by device manufacturer:
	  • 0 = Disabled
	  • 1 = Enabled
	• Firmware revisions D0_0_4_2ap and later: Current power off mode—preset by device manufacturer:
	  • 0 = Enter LPM when W_DISABLE is asserted
	  • 1 = Power down modem when W_DISABLE is asserted
	  • 2 = Ignore changes on W_DISABLE
	
	<userlpm> (State of user-initiated low power mode)
	• 0 = Disabled (normal power mode)
	• 1 = Enabled (low power mode)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!PCOFFEN
Supporting chipsets:
• All
	Set/return Power Off Enable state
	The modem can be configured to enter low power mode or power off when W_DISABLE is asserted. (This is called the Power Off Enable feature.)
	Use this command to indicate or set the Power Off Enable feature state.
	Usage:
	• Execution: **AT!PCOFFEN=<state>**
	Response: OK
	Purpose: Set the current state.
	• Query: **AT!PCOFFEN?**
	Response: <state>
	OK
	Purpose: Report the current <state>.
	Parameters:
	<state> (Current state of Power Off Enable)
	• 0 = Modem will enter LPM (low power mode) when W_DISABLE is asserted.
	• 1 = (Mini Card devices only) Modem will power off when W_DISABLE is asserted.
	• 2 = Ignore changes on W_DISABLE.
	• Default value:
	  • MDM6200, MDM6720, QSC6720: 0 (Enter LPM when W_DISABLE is asserted)
	  • All other chipsets: 1 (Power off when W_DISABLE is asserted))

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!PCTEMPLIMITS Supporting chipsets: • All	**Set/report temperature state limit values** Certain modem functionality is affected by the modem's temperature state. The possible temperature states are high critical, high warning, high normal, low normal, and low critical. Use this command to report or set the limits that correspond to these temperature states. To display the current temperature and temperature state, see **AT!PCTEMP** in *UMTS Modems Supported AT Command Reference, Document 2130617*. **Usage:** • Execution: **AT!PCTEMPLIMITS=<hc>,<hw>,<hn>,<ln>,<lc>** Response: OK Purpose: Set the temperature limits for each state (all five values must be specified). • Query: **AT!PCTEMPLIMITS?** Response: HI CRIT: <hc> HI WARN: <hw> HI NORM: <hn> LO NORM: <ln> LO CRIT: <lc> Purpose: Return the temperature limits for each state. **Parameters:** <hc> (High Critical) • Temperature limit varies by device (see device Product Specification Document or Product Technical Specification) <hw> (High Warning) • Temperature limit varies by device (see device Product Specification Document or Product Technical Specification) <hn>(High Normal) • Temperature limit varies by device (see device Product Specification Document or Product Technical Specification) <ln> (Low Normal) • Temperature limit varies by device (see device Product Specification Document or Product Technical Specification) <lc> (Low Critical) • Temperature limit varies by device (see device Product Specification Document or Product Technical Specification)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!PCVOLTLIMITS Supporting chipsets: • All	**Set/report power supply voltage state limit values** Certain modem functionality is affected by the modem's power supply voltage state. The possible voltage states are high critical, high normal, low normal, low warning, and low critical. Use this command to report or set the limits that correspond to these voltage states. **Usage:** • Execution: **AT!PCVOLTLIMITS=<hc>,<hn>,<ln>,<lw>,<lc>** Response: OK Purpose: Set the voltage limits for each state (all five values must be specified). • Query: **AT!PCVOLTLIMITS?** Response: HI CRIT: <hc> HI NORM: <hn> LO NORM: <ln> LO WARN: <lw> LO CRIT: <lc> Purpose: Return the voltage limits for each state. **Parameters:** <hc> (High Critical) • Voltage limit varies by device (see device Product Specification Document or Product Technical Specification) <hw> (High Normal) • Voltage limit varies by device (see device Product Specification Document or Product Technical Specification) <ln> (Low Normal) • Voltage limit varies by device (see device Product Specification Document or Product Technical Specification) <lw> (Low Warning) • Voltage limit varies by device (see device Product Specification Document or Product Technical Specification) <lc> (Low Critical) • Voltage limit varies by device (see device Product Specification Document or Product Technical Specification)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!PING
Supporting chipsets:
• All, with these exceptions:
  • MDM8200 (min f/w rev: M2.0 Release 2)
  • MSM6290 (min f/w rev: K1.1 Release 2)
	Ping an IP address
	Ping an IP address, waiting for a specified (or default) timeout period for a response.
	Usage:
	• Execution: **AT!PING=<ipAddr>[,<timeout>]**
	Response: !PING: Reply from <ipAddr> time=<1-10000> ms
	OK
	or !PING: Timeout from <ipAddr>
	OK
	or ERROR
	Purpose: Send a ping request to the specified <ipAddr> and wait for a response. An ERROR is returned if there is no data connection, or if the modem was unable to open a socket, or if the ping response did not match the request.
	Parameters:
	<ipAddr> (IP address being pinged)
	• Standard IP address format. For example, 192.168.0.255
	<timeout> (Time to wait for a ping response)
	• 0 = Default (4000 ms)
	• 1–10000 = Timeout period in milliseconds

!PRIID Supporting chipsets: • All *Note: The read version (!PRIID?) of this command is not password-protected.*	**Set/report module PRI part number and revision** Report or set the module's PRI part number and PRI revision. **Usage:** • Query: **AT!PRIID?** Response: PRI Part Number: <priPn> Revision: <priRevDisplay> OK Purpose: Return the module's PRI part number (<priPn>) and revision (<priRev-Display>). • Execution: **AT!PRIID=<priPN>,<priRev>** Response: OK Purpose: Set the module's PRI part number (<priPn>) and revision (<priRev>). **Parameters:** <priPn> (PRI part number) • 7-digit ASCII number • Example: 9991234 <priRev> (PRI revision number being written to the module) • 4-digit ASCII: XXYY (implied '.' between XX and YY) • Example: 0100 <priRevDisplay> (PRI revision number being read from the module) • 4-digit ASCII: XX.YY • Example: 01.00

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!REL Supporting chipsets: • All *Note: Only the execution operation ("=") is password-protected.* *Note: The actual parameter values or ranges used in the query (=?) and execution (=) forms of this command may vary slightly from this description depending on the device and firmware revision used.* *Note: Devices should always use the default value (1) for <sgsnr> and <mscr>, and use **AT!NASREL** to choose the NAS Release Compliance version (Release 5 or Release 99).*	**Set/report active protocol/revision** Configure the modem to use specific protocol, SGSN, and MSC revisions, or indicate the current settings. **Usage:** • Execution: **AT!REL=<wcdmarrc> [,<sgsnr>, <mscr>]** Response: OK Purpose: Set the desired protocol (<wcdmarrc>), SGSN revision (<sgsnr>), and MSC revision (<mscr>). • Query: **AT!REL?** Response: !REL: Protocol: Release 5 *(from <wcdmarrc>)* SGSN Revision: Dynamic *(from <sgsnr>)* MSC Revision: Dynamic *(from <mscr>)* OK Purpose: Report the current operating protocol, SGSN revision, and MSC revision. • Query List: **AT!REL=?** Purpose: Return the command format (for !REL =) and the supported values for each parameter (the supported ranges depend on modem models—see the parameter descriptions for details). **Parameters:** <wcdmarrc> (Protocol) • Default value is the highest release supported by the device. • Two-digit number corresponding to 3GPP release (!REL=? shows valid values) • Example: 00 = Release 99 <sgsnr> (SGSN revision) • Two-digit number corresponding to SGSN revision (!REL=? shows valid values) • Example: 00 = Release 97 • nn = Dynamic—Uses the revision broadcast by the network <mscr> (MSC revision) • Two-digit number corresponding to MSC revision (!REL=? shows valid values) • Example: 00 = Release 97 • nn = Dynamic—Uses the revision broadcast by the network

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!SCANTHRESH Supporting chipsets: • MDM6270 • MSM6290 • QSC6270	**Set/report WCDMA Scan Threshold** Set the WCDMA Scan Threshold. The scan threshold is set to -103 dBm by default, but can be adjusted if necessary to improve scan performance by reducing 'false positives'. During WCDMA acquisition, L1 (stack layer 1) uses a raw scan every ten channels to discover potential candidates, followed by a finer scan based on those candidates. For embedded modules in host devices, the default threshold can be very close to the noise floor of the host. As a result, there are a number of 'false positives', which significantly slows down the scan time for WCDMA. **Caution:** *If you adjust the scan threshold to reduce or eliminate false positives, you also reduce the ability of the UE to acquire systems with weak signal strength.* **Usage:** • Execution: **AT!SCANTHRESH=<threshold>** Response: OK Purpose: Set the threshold value • Query: **AT!SCANTHRESH?** Response: !SCANTHRESH: Scan Threshold (dBm): <threshold> OK *or* !SCANTHRESH: Scan Threshold is not set, default is used. OK Purpose: Indicate the current <threshold> setting. **Parameters:** <threshold> (WCDMA Scan Threshold, in dBm) • Valid range: -103 to -98 • Default value: -103

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!SCPROFSWOPT Supporting chipsets: • All	**Set/report profile's software options** Set or display the software option byte of the specified profile. **Usage:** • Execution: **AT!SCPROFSWOPT=<pid>,<swOption>** Response: OK Purpose: Set the profile's software options. • Query: **AT!SCPROFSWOPT?<pid>** Response: !SCPROFSWOPT: <pid>, <swOption> OK Purpose: Return current software options for the specified profile (<pid>). **Parameters:** <pid> (PDP context definition) • Valid range: 1–16 — A valid profile ID <swOption> (8-bit mask that identifies enabled/disabled software) • Bit 0: Enabled profile (0–NDIS, 1–DUN) • Bit 1: TurboTCP (0–Enable, 1–Disable) • Bit 2: WINS (0–Disable, 1–Enable) • Bit 3: DNS negotiation (0–Enable, 1–Disable) • Bit 4: User defined default (0—Disable, 1—Enable) • Bit 5: Prompt for username (0—Disable, 1—Enable) • Bit 6: Profile visibility (0—Visible, 1—Hidden) • Bit 7: Reserved
!SCROPROF Supporting chipsets: • All	**Set/report profile's read-only flag** Set or display the read-only flag of a profile. **Usage:** • Query: **AT!SCROPROF?<pid>** Response: !SCROPROF: <pid>,<ro_flag> Purpose: Return the flag value for the specified profile (<pid>). • Execution: **AT!SCROPROF=<ro_flag>,<pid>** Response: OK Purpose: Set the read-only flag for the specified profile. **Parameters:** <pid> (PDP context definition) • Valid range: 1–16 — A valid profile ID that will be used as the default <ro_flag> (Read Only flag) • 0 = Not write-protected • 1 = Write-protected

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!SELACQ Supporting chipsets: • MDM8220 (min f/w rev: N2.0 Release 5) • MDM9200 • MDM9600	**Select RAT acquisition order** Select the acquisition order for RATs (Radio Access Technologies). **Usage:** • Execution: **AT!SELACQ=<mode1>[,<mode2>[,<mode3>[,<mode4>[,<mode5>]]]]** Response: OK Purpose: Indicate the acquisition order for up to five RATs. See <mode> parameter description for details. • Query: **AT!SELACQ?** Response: <mode1> <mode2> <mode3> <mode4> <mode5> Purpose: Show the current acquisition order for the supported RATs. • Query list: **AT!SELACQ=?** Purpose: Display valid execution format and parameter values. **Parameters:** <mode*n*> (RAT types) • Valid values (shown in default order): • "CDMA" • "LTE" • "WCDMA" • "HDR" • "GSM" • If the execution format is issued with fewer than five RATs, the missing entries are appended based on the default order shown above. • Example: If the command is issued as **AT!SELACQ=HDR,CDMA,GSM** Then **AT!SELACQ?** will show: HDR CDMA GSM LTE WCDMA *Note: Even if the device does not support a specific RAT (for example, CDMA), the RAT will still appear in the Query response.*

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!SIMRFSC Supporting chipsets: • MSM6290 (min f/w rev: K2_0_7_52)	**Set/report SIM refresh reset notification state** Set or report the state of the unsolicited SIM refresh reset notification (!SIMRFSN). *Note: This command should be used instead of !SIMRSTC, which has been deprecated.* **Usage:** • Execution: **AT!SIMRFSC=<n>** Response: OK Purpose: Enable/disable the SIM refresh reset notification. • Query: **AT!SIMRFSC?** Response: !SIMRFSC: <n> Purpose: Show the current state of the SIM refresh reset notification. • Query list: **AT!SIMRFSC=?** Purpose: Display valid execution format and parameter values. • Unsolicited notification: **!SIMRFSN: <event>** Purpose: Alert host when the SIM is refreshed via a proactive REFRESH command. **Parameters:** <n> (Notification state) • 0=Disable • 1=Enable <event> (Notification event type) • 0=SIM is usable and full service can start. For example, this event would occur after a proactive REFRESH command. • 1=Warm reset has been performed as a result of a REFRESH with RESET command, which implicitly asks for the PIN verification procedure again. *Note: "!SIMFRSN: 1" is equivalent to the "!SIMRSTN" notification associated with !SIMRSTC.* *Note: For an explanation of the REFRESH command, see ETSI TS 101 223, section 6.4.7.*

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!SIMRSTC Supporting chipsets: • MDM8220 (min f/w rev: N2_0_8_3) • MDM9200 (min f/w rev: SWI9200X_00.07.04.01) • MDM9600 (min f/w rev: SWI9600M_01.00.07.01) • MSM6290 (min f/w rev: K2_0_7_42)	**Set/report SIM refresh reset notification state** Set or report the state of the unsolicited SIM refresh reset notification (!SIMRSTN). *Note: MSM6290—Deprecated. Use !SIMRFSC instead.* **Usage:** • Execution: **AT!SIMRSTC=<n>** Response: OK Purpose: Enable/disable the SIM refresh reset notification. • Query: **AT!SIMRSTC?** Response: !SIMRSTC: <n> Purpose: Show the current state of the SIM refresh reset notification. • Query list: **AT!SIMRSTC=?** Purpose: Display valid execution format and parameter values. **Parameters:** <n> (Notification state) • 0=Disable • 1=Enable
!SKU Supporting chipsets: • MDM9200 (min f/w rev: SWI9200X_00.07.03.01) • MDM9600 (min f/w rev: SWI9600m_01.00.06.03)	**Read modem's SKU** This command returns the modem's Sierra Wireless SKU identification. **Usage:** • Query: **AT!SKU?** Response: SKU: <sku> OK Purpose: Read the SKU from the modem. **Parameters:** <sku> (SKU number) • Valid values: • Numeric string (e.g. "1584083") • "Unset" if no SKU has been set

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!SLEEP Supporting chipsets: • All	**Set/report Sleep Enable state** The modem supports a low-power sleep state that occurs when the feature is enabled *and* specific operating conditions are met (for example, there is no data traffic over USB, no OTA traffic, etc.). Use this command to indicate the current state of the Sleep Enable feature, and to set the state of the feature. **Usage:** • Query: **AT!SLEEP?** Response: <state> OK Purpose: Report the current <state>. • Execution: **AT!SLEEP=<state>** Response: OK Purpose: Set the current state. **Parameters:** <state> (Sleep state enabled/disabled) • 0 = Disabled (Default) – The modem cannot enter the sleep state under any conditions. • 1 = Enabled – The modem can enter the sleep state when all conditions are met.
!UDPID Supporting chipsets: • All	**Set/report product ID in USB descriptor** Use this command to set the device's product ID in the USB descriptor. (Some devices support more than one product ID.) **Usage:** • Execution: **AT!UDPID=<product_id>** Response: OK Purpose: Set the product ID in the USB descriptor. • Query: **AT!UDUSBCOMP?** Response: <product_id> OK Purpose: Report the product ID that is stored in the USB descriptor. • Query List: **AT!UDPID=?** Purpose: Display a list of available product IDs for the device. **Parameters:** <product_id> • Hexadecimal ASCII value. • Valid range: 0000–FFFF

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

!UDUSBCOMP
 	
Supporting chipsets:
• All, with these exceptions:
  • MDM8200 (min f/w rev: M2.0 Release 1)
	Set/report USB interface configuration
	
	Use this command with modems that have been configured with multiple USB compositions.
	
	By default, devices are typically configured to use a USB composition that presents a minimal set of interfaces. If the device also supports other compositions, this command is used to choose from any of the supported compositions.
	
	Note: MDM6270/MSM6290/QSC6270-based devices can use either their default VID/PID (traditional interface, Direct IP not supported), or the Direct IP VID/PID (Direct IP interface).
	
	Usage:
	• Execution: **AT!UDUSBCOMP=<device_comp>**
	Response: OK
	Purpose: Set the current composition. For the change to take effect, you must reset the modem.
	• Query: **AT!UDUSBCOMP?**
	Response: !UDUSBCOMP: <device_comp>
	OK
	Purpose: Report the current interface composition.
	• Query List: **AT!UDUSBCOMP=?**
	Response: Example responses, details may differ:
	Traditional interface:
	`0 - HIP DM NMEA AT MDM1 MDM2 MDM3 MS SUPPORTED`
	`1 - HIP DM NMEA AT MDM1 MS NOT SUPPORTED`
	`2 - HIP DM NMEA AT NIC1 MS NOT SUPPORTED`
	`3 - HIP DM NMEA AT MDM1 NIC1 MS NOT SUPPORTED`
	`4 - HIP DM NMEA AT NIC1 NIC2 NIC3 MS NOT SUPPORTED`
	`5 - HIP DM NMEA AT ECM1 MS NOT SUPPORTED`
	
	`OK`
	
	Direct IP interface:
	`0 - HIP DM NMEA AT MDM1 MDM2 MDM3 MS NOT SUPPORTED`
	`1 - HIP DM NMEA AT MDM1 MS NOT SUPPORTED`
	`2 - HIP DM NMEA AT NIC1 MS SUPPORTED`
	`3 - HIP DM NMEA AT MDM1 NIC1 MS SUPPORTED`
	`4 - HIP DM NMEA AT NIC1 NIC2 NIC3 MS SUPPORTED`
	`5 - HIP DM NMEA AT ECM1 MS SUPPORTED`
	`6 - DM NMEA AT QMI NOT SUPPORTED`
	
	`OK`
	Purpose: Report the available interface compositions (<device_comp>)—the device can use any of the compositions that are listed as “SUPPORTED”.
	
	Parameters:
	<device_comp> (USB composition)
	• Integer value, 0 or greater
	• Use **AT!UDUSBCOMP=?** to view the configurations available for the device. Available configurations are identified as “SUPPORTED”.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
!UOOS Supporting chipsets: • MDM8200 • MDM8200A • MSM6290	**Set/report UMTS 'Out of Service' parameters** Set required and optional 'Out of Service' (OOS) parameters. **Usage:** • Execution: **AT!UOOS=<sleep>[, <totalscan>, <umtsscan>, <gsmscan>]** Response: OK Purpose: Set the required parameter (<sleep>) and the scan times for UMTS and GSM bands • Query: **AT!UOOS?** Response: !UOOS: Deep Sleep (s): <sleep> Total Scan Time (ms): <totalscan> UMTS Scan Time (ms): <umtsscan> GSM Scan Time (ms): <gsmscan> OK Purpose: Report the current OOS settings. **Parameters:** <sleep> (UMTS OOS deep sleep time) • 0–0xFFFF = Number of seconds <totalscan> (Sum of <umtsscan> and <gsmscan>) • Manually verify that this value is entered as the sum of <umtsscan> and <gsmscan> to ensure that the OOS parameters are updated correctly. • 0–0xFFFF = Number of milliseconds <umtsscan> (OOS UMTS bands scan time) • 0–0xFFFF = Number of milliseconds <gsmscan> (OOS GSM bands scan time) • 0–0xFFFF = Number of milliseconds
+WGETWK Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MSM6290 (min f/w rev: K1_1_1_9ap) • QSC6270 (min f/w rev: S2.0)	**Return wake-up event type** When a wake-up signal is received from the modem, use this command to identify the event type that caused it to be sent. (If any other wake-up events occurred while the wake-up signal was being sent, they are ignored.) When this command is executed, the last wake-up event is identified and then cleared from memory. **Usage:** • Execution: **AT+WGETWK** Response: +WGETWK: <event> Purpose: Report the wake-up event type that caused the modem to send a wake-up signal to the host device. **Parameters:** <event> (Wake-up event type) • Valid range: 0–3 • 0 = No event occur ed • 1 = Ring received • 2 = Radio coverage restored • 3 = SMS received

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

+WHCNF
Supporting chipsets:
• MDM6200
• MDM6270 (min f/w rev: S0_0_2_1)
• QSC6270 (min f/w rev: S0_0_2_1)
	Activate/deactivate modem hardware features
	Activate, deactivate, or return the current operating state of hardware features (LED, SIM, and UART).
	Usage:
	• Execution: **AT+WHCNF=<type>, <mode>[, <uartgroup>[, <uartlinedcd>[, <uartlinedtr>[, <uartlinedsr>[, <uartsleep>]]]]]**
	Response: OK
	Purpose: Activate/deactivate the specified hardware <type>.
	• Query: **AT+WHCNF?**
	Response: +WHCNF: <type>,<status>[,<resetFlag>]
	or +WHCNF: <type>, <status>,<resetFlag>,<uartgroup>,<uartlinedcd>,<uartlinedtr>, <uartlinedsr>,<uartsleep>]
	...
	OK
	Purpose: Display the current activation state and settings for each hardware feature.
	• Query List: **AT+WHCNF=?**
	Purpose: Display valid execution formats and parameter values.
	Parameters:
	<type> (Hardware functionality)
	• 1 = LED
	• 4 = InSIM switch (MDM6270/QSC6270 only)
	• 6 = UART
	• 7 = Buzzer (MDM6200 only)
	<mode> (Requested operation)
	• For <type = 1> (LED)
	  • 0 = Deactivate
	  • 1 = Activate
	  • 2 = Interrogate status (Note: The execution response returns the current operating state of the <type>—deactivated or activated. The Query format can be used to return the same information for all of the hardware types at one time.)
	• For <type = 4> (InSIM switch)
	  • 0 = External SIM privileged (Default)
	  • 1 = External SIM only
	  • 2 = Internal SIM only
	  • 3 = Get inSIM switch selected mode
	• For <type = 6> (UART)
	  • 0 = Deactivate
	  • 1 = Activate
	  • 2 = Interrogate status (Note: The execution response returns the current operating state of the <type>—deactivated or activated. The Query format can be used to return the same information for all of the hardware types at one time.)
	(Continued on next page)

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description

+WHCNF (continued)	**Activate/deactivate modem hardware features (continued)**
	• For <type = 7> (Buzzer)
	  • 0 = BUZZER_EN outputs a PWM signal (Default)
	  • 1 = BUZZER_EN outputs as a GPO
	<uartgroup> (Pin type duplexed for extended UART signals—DCD/DSR/DTR)
	• 0 = Extended UART (8-wire) disabled. (4-wire enabled)
	• 1 = GPIO
	• 2 = PCM
	• 3 = SPI
	<uartlinedcd> (DCD line)
	• 0 = Disabled
	• 1 = Enabled
	<uartlinedtr> (DTR line)
	• 0 = Disabled
	• 1 = Enabled
	<uartlinedsr> (DSR line)
	• 0 = Disabled
	• 1 = Enabled
	<uartsleep> (UART sleep flag)
	• Sleep flag is referenced only if DTR line is disabled.
	• 0 = Sleep enabled
	• 1 = Sleep disabled
	• Behavior summary:
	  • <uartgroup = 1|2|3> && <uartlinedtr = 1>: UART goes to sleep when DTR is deasserted and wakes when DTR is asserted.
	  • <uartgroup = 0> && <uartsleep = 0>: UART goes to sleep 15 seconds after bootup (if no activity), and after 4 seconds of normal inactivity. UART wakes when activity is detected on Rx line; the port may be used after short delay (~1 second).
	  • <uartgroup = 0> && <uartsleep = 1>: UART never sleeps.

Table 3-2: Modem status, customization, and reset commands (Continued)

Command	Description
+WWKUP Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MSM6290 (min f/w rev: K_1_1_9ap) • QSC6270 (min f/w rev: S2.0)	**Enable/disable wake-up signals** Enable (or disable) the output of wake-up signals from the modem to the host device in response to specific wake-up event types. These settings are persistent (remain unchanged over a modem power cycle). • (MSM6290) The modem can send wake-up signals (using the Ring Indicator (RI) on the Mini Card host interface) to the host when the following events occur: ring received, radio coverage resolved, and SMS received. • (MDM6200/MDM6270/QSC6270) WAKE_N pin is used as RI to output the wake-up signal to the host. Functionality is available even when the UART interface is disabled. You must enable the UART interface using !MAPUART=1 for the RI signal to be output. Signal timing: • Ring received event—Assert RI for 5.9 seconds, then deassert for 0.1 seconds. Repeat for number of ring cycles (network dependent). • Radio coverage restored/SMS received events—Assert RI for 5.9 seconds, then deassert. Any wake-up events that occur while RI is being asserted (a wake-up signal is already being output) are ignored until RI is deasserted. To identify the event that generated the wake-up signal, see +WGETWK on page 80. **Usage:** • Execution: **AT+WWKUP=<bitmask>** Response: OK *or* ERROR Purpose: Enable or disable reporting of wake-up events. • Query: **AT+WWKUP?** Response: +WWKUP: <bitmask> Purpose: Display the current state of wake-up event reporting. **Parameters:** <bitmask> (Wake-up event reporting states) • 8-bit field (range 00-07) • Bit values (1=Enable, 0=Disable) • Bit 0: Ring received (Default = 1). The RING indication on the AT command port is not affected by this setting—it is still issued even if RI is deasserted. • Bit 1: Radio coverage restored (Default = 0) • Bit 2: SMS received (Default = 1) • Bit 3–7: Reserved (All must be 0)

4: Diagnostic Commands

Introduction

This chapter describes commands used to diagnose modem problems.

Command summary

The table below lists the commands described in this chapter.

Table 4-1: Diagnostic commands

Command	Description	Page
!CMUX	Implement multiplexing mode	86
!CMUXLPBK	Configure ports into loopback mode/query loopback state	86
!DIVERSITY	Enable/disable CDMA receive diversity	87
!ERR	Display diagnostic information	87
!GBAND	Set/return the current operating band	88
$QCPDPP	Set/report PDP-IP connection authentication parameters	89
$QCTER	Set/report TE-DCE baud rate	89
!RXDEN	Enable/disable WCDMA/LTE receive diversity	90
!UMTSCHAN	Set/report Priority UMTS Channel Selection state	91

Command reference

Table 4-2: Diagnostic command details

Command	Description
!CMUX Supporting chipsets: • MDM6200 • MDM6270 • MSM6290 • QSC6270	**Implement multiplexing mode** This command implements the functionality of +CMUX (27.007 standard command). It is provided to ensure command consistency with Sierra Wireless CDMA modules. See *3GPP TS 27.007* for command syntax and arguments.
!CMUXLPBK Supporting chipsets: • MDM6200 • MSM6290	**Configure ports into loopback mode/query loopback state** Use this command to select a multiplexed port and to loop data traffic between the host and the modem over that port. The modem is effectively in loopback mode only after the modem is put into MUX mode (using **+CMUX**). You can, however, issue **!CMUXLPBK** before or after **+CMUX.** Loopback mode is NOT supported on the MUX control channel, since flow control and other modem signaling is passed on this port. **Usage:** • Execution: **AT!CMUXLPBK=<bitmask>** Response: OK Purpose: Select a specified multiplexed port. • Query: **AT!CMUXLPBK?** Response: !CMUXLPBK: <bitmask> OK Purpose: Return the current setting. You can have two ports in loopback mode. When the modem is in loopback mode, the loopback occurs just as data is passed to the application layer of the modem. Therefore, the processing time of this command includes the processing overhead of packet disassembly and reassembly as the packets are received and echoed back to the host. To exit loopback mode when the AT data port is in loopback, you must reset the modem. To exit loopback mode on other ports, enter the command with the appropriate bit in the bitmask set to 0. To disable loopback mode on all ports, enter AT!CMUXLPBK=0x00. **Parameters:** <bitmask> (16-bit hexadecimal bitmask that identifies the port) • Bit 0: AT data port • Bit 1: Diag port • Bit 2: HIP (CnS) port • Bit 3: Reserved • Bit 4: PDP1 port • Bit 5: Reserved • Bit 6: Reserved • Bit 7–15: Reserved **Examples:** AT!CMUXLPBK=0x01—Sets the AT data port into loopback mode AT!CMUXLPBK=0x10—Sets PDP1 into loopback mode

Table 4-2: Diagnostic command details (Continued)

Command	Description
!DIVERSITY Supporting chipsets: • MDM9600 *Note: This command is not password-protected.*	**Enable/disable CDMA receive diversity** Enable or disable CDMA receive diversity (1X and 1xEV-DO), or establish receive diversity as the primary path. **Usage:** • Execution: **AT!DIVERSITY=<cdma_diversity>,<hdr_diversity>** Response: OK Purpose: Set the current receive diversity state. • Query: **AT!DIVERSITY?** Response: CDMA Diversity: <cdma_diversity> HDR Diversity: <hdr_diversity> OK Purpose: Return the current receive 1X (CDMA) and 1xEV-DO (HDR) diversity <state>. **Parameters:** <cdma_diversity> (Current/requested 1X receive diversity state) • 0 = Rx diversity disabled • 1 = Rx diversity enabled <hdr_diversity> (Current/requested 1xEV-DO receive diversity state) • 0 = Rx diversity disabled • 1 = Rx diversity enabled
!ERR Supporting chipsets: • All	**Display diagnostic information** This command is used to display diagnostic information (logged error conditions) that Sierra Wireless uses to assist in resolving technical issues. **Usage:** • Execution: **AT!ERR=0** Response: OK Purpose: Clear the logged error conditions. Use this command before running tests to make sure that details displayed using AT!ERR are relevant to the tests being performed. • Query: **AT!ERR** Response: 00 [F] <count> <file> <line> ... nn [F] <count> <file> <line> OK Purpose: Return all logged error conditions that are stored in NVRAM. **Parameters:** <count> (Number of occurrences) • Valid range: 0x00–0xFF <file> (Log file name) • Name of log file using ASCII characters <line> (Line number in log file) • Valid range: 1–99999

Table 4-2: Diagnostic command details (Continued)

Command	Description
!GBAND Supporting chipsets: • All	**Set/return the current operating band** Read or set the current operating bands (1 or more) of the modem. **Usage:** • Execution: **AT!GBAND=<bandmask>** Response: OK Purpose: Set the operating bands. • Query: **AT!GBAND?** Response: !GBAND: <band>, <bandmask> OK Purpose: Return the current operating band and the mask of preferred bands. **Parameters:** <band> (Current WCDMA or GSM operating band) • ASCII string (For example: “WCDMA1900”, “GSM850”, etc.) <bandmask> (Supported bands—use a bitwise OR to choose multiple bands) • GSM850: 0000000000080000 • GSM900: 0000000000000300 • GSM1800: 0000000000000080 • GSM1900: 0000000000200000 • WCDMA2100: 0000000000400000 • WCDMA1900: 0000000000800000 • WCDMA850: 0000000004000000 • WCDMA800: 0000000008000000 • WCDMA900: 0002000000000000 • Any band: 000000003FFFFFFF (autoband—the modem acquires any available band) **Example:** You can set the modem to use multiple bands using a bitwise OR. For example, to set the modem to use the GSM850 and GSM900 bands, you would issue the command AT!GBAND=00080300: 0000000000080000 (GSM850) 0000000000000300 (GSM900) --- 0000000000080300 *Note: It is not possible to perform handoffs to bands not in the currently selected band mask.* *Note: Due to stack implementation, the query form of the command (!GBAND?) reports WCDMA800 for both WCDMA800 and WCDMA850.*

Table 4-2: Diagnostic command details (Continued)

Command	Description
$QCPDPP Supporting chipsets: • All	**Set/report PDP-IP connection authentication parameters** Set authentication requirements (username, password) for PDP-IP packet data calls for each supported profile. **Usage:** • Execution: **AT$QCPDPP=<cid>, <auth_type>, <password>, <username>** Response: OK *or* ERROR Purpose: Set, for each <cid> (profile), the type of authorization required when establishing a connection. • Query: **AT$QCPDPP?** Response: $QCPDPP: <cid>, <auth_type>[,<username>] (repeat for each <cid>) Purpose: Return, for each profile, the authorization type required and the username used for PAP authentication. Note: Passwords are not displayed. **Parameters:** <cid> • Supported profiles • Valid range: 1–16 <auth_type> • 0 = None—Username and password not required • 1 = PAP—Username and password accepted • 2 = CHAP—Username and password (secret) accepted <password> • Supplied by network provider • Required for <auth_type> = 1 and 2 <username> • Supplied by network provider • Required for <auth_type> = 1 and 2 • For <auth_type> = 2, the username can be any dummy value
$QCTER Supporting chipsets: • All	**Set/report TE-DCE baud rate** Set the baud rate at which DCE accepts commands over the UART interface (the command does nothing when the USB interface is used). **Usage:** • Execution: **AT$QCTER=<rate>** Response: OK Purpose: Set the default baud rate and store it in NV RAM. • Query: **AT$QCTER?** Response: <rate> Purpose: Return the current default TE-DCE baud rate. • Query List: **AT$QCTER=?** Purpose: Return a list of supported baud rates. **Parameters:** <rate> • Use the $QCTER=? command to list all supported baud rates.

Table 4-2: Diagnostic command details (Continued)

Command	Description
!RXDEN Supporting chipsets: • MDM6200 • MDM8200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 *Note: Only the execution operation ("=") is password-protected.*	**Enable/disable WCDMA/LTE receive diversity** Enable or disable WCDMA/LTE receive diversity, or establish receive diversity as the primary path. The new state takes effect the next time the modem is reset. (To enable/disable CDMA diversity, see !DIVERSITY on page 87.) *Note: To change from <state=0> to <state=2> (or from <state=2> to <state=0>, you must issue AT!RXDEN=1, reset the modem, and then make the final state change.* **Usage:** • Execution: **AT!RXDEN=<state>** Response: OK Purpose: Set the current receive diversity state. • Query: **AT!RXDEN?** Response: !RXDEN: <state> OK Purpose: Return the current receive diversity <state>. • Query List: **AT!RXDEN=?** Purpose: Return a list of available <state> values to use in this command. **Parameters:** <state> (Current/ requested receive diversity state) • 0 = Rx diversity disabled • 1 = Rx diversity enabled • 2 = Rx diversity is primary path

Table 4-2: Diagnostic command details (Continued)

Command	Description
!UMTSCHAN Supporting chipsets: • All *Note: This command is not password-protected.*	**Set/report Priority UMTS Channel Selection state** Set the modem to scan a 'priority' UMTS channel *when the modem powers up,* before scanning other frequencies. This allows you to use the modem on a test cell network (that uses the priority UMTS channel) in an area where a UMTS network already exists—the modem attempts to access the test network even if a cell on the normal UMTS network would provide a better connection. *Note: If the signal is lost on the priority UMTS channel, the modem scans for another channel. To force the modem to rescan for the priority UMTS channel, you must power cycle the modem.* **Usage:** • Execution: **AT!UMTSCHAN=<flag>[,<DL_UARFC>]** Response: OK Purpose: Enable/disable the feature and power cycle the modem, forcing cell reselection. • Query: **AT!UMTSCHAN?** Response: Enable: <flag> Channel: <DL_UARFCN> OK Purpose: Return the current <flag> setting, and DL_UARFCN (if <flag> = 1). **Parameters:** <flag> • 0 = Disable priority UMTS channel selection. When the modem power cycles, cell reselection occurs using normal scanning procedure. • 1 = Enable priority UMTS channel selection. When the modem power cycles, cell reselection occurs—the modem scans DL_UARFC first, then uses normal scanning procedure if it is unavailable. <DL_UARFCN> (Valid downlink UARFCN based on the operating band) • Only used when <flag> = 1 (enable) • Examples: 10700, 9800

5: Test Commands

Introduction

To obtain regulatory approval and carrier approvals for your product, you may be required to perform tests on the radio component of the embedded modem. This chapter describes AT commands used to perform those tests.

In most cases the modem must be in a particular mode before you can issue the AT commands to perform particular tests. Therefore, the order in which you issue certain commands is important. Three AT commands are important in setting the mode:

- !DAFTMACT—puts the modem in factory test mode (a non-signaling mode). You must issue **AT!DAFTMACT** before issuing any other command that starts with "!DA".
- !DASBAND—selects the frequency band.

  You must execute **AT!DASBAND** to select an LTE band to run these commands that test the LTE transceiver:
  - !DALGAVGAGC
  - !DALGRXAGC
  - !DALGTXAGC

  You must execute **AT!DASBAND** to select a WCDMA band to run these commands that test the WCDMA transceiver:
  - !DAWGRXAGC
  - !DAWGAVGAGC
  - !DAWSTXCW
  - !DAWSPARANGE
  - !DASTXOFF
  - !DASTXON
  - !DAWSCONFIGRX

  You must execute **AT!DASBAND** to select a GSM band to run these commands that test the GSM transceiver:
  - !DAGSRXBURST
  - !DAGSRXCONT
  - !DAGGRSSI
  - !DAGGAVGRSSI
  - !DAGGRSSIRAW
  - !DAGSTXFRAME
- !DASCHAN—selects the channel. This command must be run after you have selected the band with **!DASBAND**. (If you don't select a channel, the modem uses a default.)

Command summary

The table below lists the commands described in this chapter.

Table 5-1: Test commands

Command	Description	Page
!ALLUP	Turn transmitter on/off and simulate 'All UPs'	96
!BEP	Return MT RSSI and RSQ	97
!CHAN	Tune synthesizer to channel/band	98
!DAAGCTON	Return C/N (carrier to noise ratio) while in factory test mode	98
!DAFTMACT	Put modem into Factory Test Mode	99
!DAFTMDEACT	Put modem into online mode from Factory Test Mode	99
!DAGGAVGRSSI	Return averaged RSSI value in dBm (GSM only)	99
!DAGGRSSI	Return the RSSI value in dBm (GSM only)	100
!DAGGRSSIRAW	Return raw RSSI value	100
!DAGINFO	Return GSM mode RF information (GSM only)	101
!DAGSLOCK	Return synthesizer lock state	101
!DAGSRXBURST	Set GSM receiver to burst mode	102
!DAGSRXCONT	Set GSM receiver continuously on	102
!DAGSTXBURST	Set GSM transmitter to burst mode	103
!DAGSTXFRAME	Set GSM Tx frame structure	104
!DALGAVGAGC	Return averaged Rx AGC value (LTE only)	105
!DALGRXAGC	Return Rx AGC value (LTE only)	106
!DALSRXBW	Set LTE Rx bandwidth (LTE only)	108
!DALSTXBW	Set LTE Tx bandwidth (LTE only)	109
!DALGTXAGC	Return Tx AGC value and transmitter parameters (LTE only)	107
!DAOFFLINE	Place modem offline	109
!DASBAND	Set frequency band	110
!DASCHAN	Set modem channel (frequency)	111
!DASLNAGAIN	Set LNA gain state	112
!DASPDM	Set PDM value	113
!DASTXOFF	Turn Tx PA off	113
!DASTXON	Turn Tx PA on	114
!DAWGAVGAGC	Return averaged Rx AGC value (WCDMA only)	114

Table 5-1: Test commands (Continued)

Command	Description	Page
!DAWGRXAGC	Return Rx AGC value (WCDMA only)	115
!DAWINFO	Return WCDMA mode RF information (WCDMA only)	116
!DAWSCONFIGRX	Set WCDMA receiver to factory calibration settings	117
!DAWSPARANGE	Set PA range state machine	118
!DAWSSCHAIN	Enable secondary receive chain (WCDMA only)	118
!DAWSCHAINTCM	Place receive chain in test call mode (WCDMA only)	118
!DAWSTXCW	Set waveform used by the transmitter	119
!DAWSTXPWR	Set desired Tx power level (WCDMA mode only)	119
!IMSTESTMODE	Enable/disable IMS test mode	120
!KEYOFF	Key off the transmitter	120
!KEYON	Key on the transmitter	121
!OSDSM	Display memory usage for DSM buffer pools	121
$QCAGC	Read Rx AGC (CDMA and WCDMA modes)	122
!RX2	Turn second receiver on/off	122
!RX2AGC	Read second receiver Rx AGC	123
!RXAGC	Read first receiver Rx AGC	123
!TX	Turn transmitter on/off	123
!TXAGC	Set desired Tx AGC	124

Command reference

Table 5-2: Test command details

Command	Description
!ALLUP Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Turn transmitter on/off and simulate 'All UPs'** This command turns the transmitter on/off and simulates an 'All UPs' Tx condition. **Usage:** • Execution: **AT!ALLUP=<value>** Response: OK Purpose: Turn transmitter on/off and simulate All UPs Tx condition. **Parameters:** <value> (State of All UPs simulation) • 0—All UPs off • 1—All UPs on

Table 5-2: Test command details (Continued)

Command	Description

!BEP
Supporting chipsets:
• MDM6270
• MSM6290 (min f/w rev: K2_0_8_1)
• QSC6270
	Return MT RSSI and RSQ
	This command returns the RSSI (Received Signal Strength Indicator) and RSQ (Received Signal Quality) from the MT.
	This command extends the +CSQ command by reporting the mean BEP (bit error probability) value during EGPRS TBF mode (rather than the '99' reported by +CSQ).
	Note: When the device is in Dual Transfer Mode, EGPRS BEP reporting is suspended, and the command reports RSQ for the CS connection.
	Usage:
	• Execution: **AT!BEP**
	Response: !BEP: <rssi>,<rsq value>,<rsq type>
	OK
	or
	+CME ERROR: <err>
	OK
	Purpose: Return the MT RSSI and RSQ.
	Parameters:
	<rssi> (MT RSSI converted to dBm, in 2 dBm steps)
	• 0 = -113 or less
	• 1 = -111
	• 2 = -109
	• ...
	• 30 = -53
	• 31 = -51 or greater
	• 99 = Not known, or not detectable
	<rsq value> (MT RSQ value)
	• Value depends on <rsq type>
	• If <rsq type> = 0 (RXQUAL)
	  • 0..7 (As defined in TS 45.008 subclause 8.2.4)
	• If <rsq type> = 1 or 2 (Mean BEP for GMSK/8PSK)
	  • 0..31 (As defined in TS 45.008 subclause 8.2.5)
	• If <rsq type> = 99 (Not known or not detectable)
	  • 99 (Not known or not detectable)
	<rsq type> (RSQ measurement type)
	• 0=RXQ (Received signal quality)
	• 1=Mean BEP for GMSK
	• 2=Mean BEP for 8PSK
	• 99=Not known or not detectable

Table 5-2: Test command details (Continued)

Command	Description
!CHAN Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Tune synthesizer to channel/band** This command tunes the synthesizer to a specified channel and band, or reports the current tuning (including changes made using !KEYON on page 121). If the band is omitted, the modem uses the current band setting, changing only the channel. The channel setting on entry to diagnostic mode is determined by the prior activity of the modem. **Usage:** • Execution: **AT!CHAN=<channel>[,<band>]** Response: OK Purpose: Tune the synthesizer to the specified channel and band. • Query: **AT!CHAN?** Response: Channel = <channel> OK Purpose: Return the last channel that the synthesizer attempted to tune to. **Parameters:** <channel> • 0–799, 991–1023 (Cellular) • 0–1200 (PCS) <band> • 0=US Cellular • 1=US PCS • 2=GPS
!DAAGCTON Supporting chipsets: • MDM8200 • MSM6290	**Return C/N (carrier to noise ratio) while in factory test mode** Return the carrier-to-noise ratio when the modem is in FTM (Factory Test Mode). **Usage:** • Query: **AT!DAAGCTON** Response: <c_n> <errorcode> OK Purpose: Returns carrier-to-noise ratio. **Parameters:** <c_n> (Carrier-to-noise ratio in dB/Hz) • Format: x.x (fixed point decimal) <errorcode> • 0=OK

Table 5-2: Test command details (Continued)

Command	Description
!DAFTMACT Supporting chipsets: • All	**Put modem into Factory Test Mode** Place the modem in FTM (Factory Test Mode). FTM is a non-signaling mode that allows the radio component to be manually configured to conduct certain types of tests. AT commands that start with "!DA" are only available when the modem is in FTM mode. *Note: When this command executes successfully, the modem responds with the value 290300. Any other response indicates an error.* **Usage:** • Query: **AT!DAFTMACT** Response: 290300 *(Success. Any other response indicates an error.)* OK Purpose: Place modem in FTM mode.
!DAFTMDEACT Supporting chipsets: • All	**Put modem into online mode from Factory Test Mode** This command takes the modem out of FTM and puts the modem back into online mode. (The command !DAFTMACT puts the modem into FTM.) *Note: When this command executes successfully, the modem responds with the value 290400. Any other response indicates an error.* **Usage:** • Query: **AT!DAFTMDEACT** Response: 290400 *(Success. Any other response indicates an error.)* OK Purpose: Place modem in online mode (from FTM mode).
!DAGGAVGRSSI Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Return averaged RSSI value in dBm (GSM only)** Return an averaged RSSI (Received Signal Strength Indicator) value in dBm. **Usage:** • Execution: **AT!DAGGAVGRSSI=<channel>, <LNA Index>** Response: OK Purpose: Return the averaged RSSI for the specified channel and LNA offset index. **Parameters:** <channel> (Channel number for the band specified using **!DASBAND**) • Valid values depend on the selected band <LNA Index> (LNA offset index) • 0=R0 (highest gain) • 1=R1 • 2=R2 • 3=R3 (lowest gain) *Note: **!DASBAND** must be issued before you can use **!DAGGAVGRSSI**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAGGRSSI Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Return the RSSI value in dBm (GSM only)** Return the RSSI (Received Signal Strength Indicator) value in dBm using the calibration offsets (valid in GSM burst mode only—the command !DAGSRXBURST puts the modem into burst mode). **Usage:** • Execution: **AT!DAGGRSSI** Response: Channel:<chan> LNA:<lna> RXPWR:<pwr> dBm OK Purpose: Return the averaged RSSI for the specified channel and LNA offset index. **Parameters:** <chan> (Channel number for the band specified using **!DASBAND**) • Valid values depend on the selected band <lna> (Current LNA state) • Valid range: 0–3 <pwr> (RSSI converted to dBm) • Signed value based on Rx power *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGGRSSI**.*
!DAGGRSSIRAW Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Return raw RSSI value** Return a 32-bit raw RSSI value (valid in GSM mode only). The value is an average over multiple bursts. **Usage:** • Execution: **AT!DAGGRSSIRAW** Response: <rssi> OK Purpose: Return the raw RSSI value. **Parameters** <rssi> (RSSI, averaged over multiple bursts) • 32-bit value *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGGRSSIRAW**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAGINFO Supporting chipsets: • All *Note: The modem must be in online mode to use this command.*	**Return GSM mode RF information (GSM only)** Return RF information for GSM mode. *Note: **!DAWINFO** provides RF information for WCDMA mode.* **Usage:** • Execution: **AT!DAGINFO** Response: Channel: <channel> RSSI:<rssi> LNA:<lna> RXPWR:<rxPwr> dBm SNR:<snr> DC Offset_I:<iOffset> DC Offset Q:<qOffset> Freq. Offset:<freqOffset> Timing Offset:<timingOffset> OK Purpose: Return the RF information. **Parameters** <channel> (GSM channel) • Valid range: 0–65535 <rssi> (Receive power) • Valid range: 0x00000000–0xFFFFFFFF <lna> (LNA gain state) • Valid range: 0–65535 <rxPwr> (Rx power in dBm (0.25 units)) • Valid range: 0–65535 <snr> (Signal-to-noise ratio) • Valid range: 0–65535 <iOffset> (I offset) • Valid range: 0–65535 <qOffset> (Q offset) • Valid range: 0–65535 <freqOffset> (Frequency offset) • Valid range: 0–65535 <timingOffset> (Timing offset) • Valid range: 0–65535
!DAGSLOCK Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Return synthesizer lock state** Return a value indicating the lock state of the RF synthesizers. **Usage:** • Execution: **AT!DAGSLOCK** Response: <sLockState> OK Purpose: Return the synthesizer lock state. **Parameters** <sLockState> (Synthesizer lock state) • 0=One or more synthesizers are out of lock • 1=All synthesizers locked *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGSLOCK**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAGSRXBURST Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set GSM receiver to burst mode** Set the receiver to start or stop sending bursts. The receiver must be in burst mode to read the RSSI. (The command !DAGGRSSI returns the RSSI value.) **Usage:** • Execution: **AT!DAGSRXBURST=<function>** Response: <function> OK Purpose: Set the receiver to burst mode **Parameters:** <function> • 0=Get RSSI (Burst mode) • 2=Stop continuous Rx *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGSRXBURST**.*
!DAGSRXCONT Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set GSM receiver continuously on** Set the GSM receiver so that it is continuously on and not bursting. **Usage:** • Execution: **AT!DAGSRXCONT=<function>** Response: <function> OK Purpose: Set the receiver to continuously on. **Parameters:** <function> • 3=Receiver continuously on • 4=Receiver off *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGSRXCONT**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAGSTXBURST Supporting chipsets: • All	**Set GSM transmitter to burst mode** Set the transmitter to start or stop sending bursts. **Usage:** • Execution: **AT!DAGSTXBURST=<source>, <TSCindex>, <burstdur>** Response: <source> <TSCindex> <burstdur> OK Purpose: Set the receiver to start/stop sending bursts. **Parameters:** <source> • 0=Random data • 1=Tone • 2=Buffer data <TSCindex> (Training sequence index) • Valid range: 0–9 <burstdur> Burst duration: • 1=Continuous *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGSTXBURST**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAGSTXFRAME Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set GSM Tx frame structure** This command configures the Tx slots for GSM operation. It must be issued eight times to set all eight slots. **Usage:** • Execution: **AT!DAGSTXFRAME=<slotnum>, <onoff>, <pwr>, <mcs>** Response: <slotnum> <onoff> <pwr> <mcs> OK Purpose: Set the Tx frame structure. **Parameters:** <slotnum> (Slot number) • Valid range: 0–7 (eight available Tx slots) <onoff> (Enable/disable the specified slot) • 0=Off (disable) • 1=On (enable) <pwr> (Slot power level) • Measured in dB*100 • Maximum values: • GMSK Mode 850/900 bands: 3200 (32 dBm) 1800/1900 bands: 2900 (29 dBm) • 8PSK (EDGE) Mode 850/900 bands: 2700 (27 dBm) 1800/1900 bands: 2600 (26 dBm) <mcs> (Modulation code scheme) • Valid range: 0–8 (MCS1 to MCS9) *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAGSTXFRAME**.*

Table 5-2: Test command details (Continued)

Command	Description
!DALGAVGAGC Supporting chipsets: • MDM9200 • MDM9600 *Note: The modem must be in LTE mode to use this command.*	**Return averaged Rx AGC value (LTE only)** Return the averaged AGC (Automatic Gain Control) readings for a specific uplink channel on the main and diversity paths. **Usage:** • Execution: **AT!DALGAVGAGC=<channel>, <LNA Index>** Response: Paths: <paths> Rx<n>: AGC: <agc> dBm LNA: <lna> Chain: <chain> Rx<n>: AGC: <agc> dBm LNA: <lna> Chain: <chain> OK Purpose: Return the averaged AGC for <channel> on the main and diversity paths. **Parameters:** <channel> (Uplink channel number (UARFCN) for the band specified using **!DASBAND**) • Valid values depend on the selected band <LNA Index> (LNA offset index) • 0=R0 (Highest gain) • 1=R1 • 2=R2 • 3=R3 (Lowest gain) <paths> (Number of receive paths) • 2 <agc> (AGC value in dBm) • Valid values: Dynamic Rx range <chain> (Receive paths) • 0=Rx Main • 1=Rx Diversity *Note: **!DASBAND** and **!DALSRXBW** must be issued before you can use **!DALGAVGAGC**.*

Table 5-2: Test command details (Continued)

Command	Description

!DALGRXAGC
Supporting chipsets:
• MDM9200
• MDM9600
	Return Rx AGC value (LTE only)
	Return the Rx AGC (Automatic Gain Control) value and LNA gain states for each RF path.
	The AGC value can be converted to RSSI (Received Signal Strength Indicator) in dBm:
	if (<AGC_value> < 511)
	<RX_dBm> = -106 + ((<AGC_value> + 512) / 12)
	else
	<RX_dBm> = -106 + (((<AGC_value>-1024) + 512) / 12)
	Usage:
	• Execution: **AT!DALGRXAGC** or **AT!DALGRXAGC?**
	Response: <AGC value>
	OK
	Purpose: Return the <AGC value> for either the main or diversity path. If no <path> is specified, the main path is assumed.
	Parameters:
	<path> (For modules supporting diversity)
	• 0=Main path
	• 1=Diversity path
	<AGC value> (Rx AGC value for specified path)
	• Valid range: -512 to +511
	*Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DALGRXAGC**.*

Table 5-2: Test command details (Continued)

Command	Description
!DALGTXAGC Supporting chipsets: • MDM9200 • MDM9600	**Return Tx AGC value and transmitter parameters (LTE only)** Return the Tx AGC (Automatic Gain Control) value and other transmitter parameters. *Note: This command works only in an active call (for example, when connected to a call box or live network).* **Usage:** • Execution: **AT!DALGTXAGC** or **AT!DALGTXAGC?** Response: Paths: <paths> Tx<n>:AGC: <agc> dBm RBi: <rbi> RB: <rbn> PA: <pa> TxGainIdx: <txgi> MTPL: <mtpl> dBm IQgain: <iq> MPR: <mpr> AMPR: <ampr> NS: <ns> SARmpr: <sarmpr> PDet Mode: <mode> PDetAGC: <pagc> PDet: <pdbm> Traw: <traw> Tscaled: <tscaled> Tidx: <tidx> Trem: <trem> OK Purpose: Return transmitter parameters and the transmit <AGC value>. **Parameters:** <paths> (Number of transmit paths) • 1 (Tx) <agc> (Tx AGC value in dBm) • Valid range: -70 to +23 <rbi> • Start resource block index <rbn> (Number of resource blocks) • Valid range: 0–50 <pa> (PA gain state) • Valid range: 0–3 <txgi> • Tx gain index <mtpl> (Max Tx power limit) • Max value: +23 <iq> • Digital IQ gain scaling <mpr> (Maximum power reduction) • See 3GPP 36.101 for details <ampr> (Additional Max power reduction) • See 3GPP 36.101 for details <ns> (Network Signaled (NS) value) • See 3GPP 36.101 for details (Continued on next page)

Table 5-2: Test command details (Continued)

Command	Description

!DALGTXAGC (continued)	**Return Tx AGC value and transmitter parameters (LTE only) (continued)**
	<mode> (HDET (power detector) mode)
	• Valid values:
	  • L (Lower power)
	  • H (Higher power)
	<padc>
	• HDET ADC
	<pdbm>
	• HDET dBm
	<traw> (Raw thermistor ADC value)
	• Valid range: 0–4095
	<tscaled> (Scaled thermistor value)
	• Valid range: 0–255
	• Value is scaled from <traw> based on calibrated min/max <traw> values for the supported temperature range.
	<tidx> (Temperature compensation bin)
	• Valid range: 0–7
	<trem>
	• Temperature compensation remainder bin
	Note: !DASBAND and !DASCHAN must be issued before you can use !DALGTXAGC.

!DALSRXBW Supporting chipsets: • MDM9200 (min f/w rev: SWI9200M_1.0 Release 5, or SWI9200X_3.0 Release 2) • MDM9600 (min f/w rev: SWI9600M_01.00.09.04)	**Set LTE Rx bandwidth (LTE only)** Set the LTE Rx bandwidth. Requirements: • **!DASBAND** must be issued before you can use this command. • This command must be issued before you can use **!DALGAVGAGC**. **Usage:** • Execution: **AT!DALSRXBW=<bw>** Response: OK Purpose: Set the LTE Rx bandwidth. **Parameters:** <bw> (LTE bandwidth) • 0=1.4 MHz • 1=3 MHz • 2=5 MHz • 3=10 MHz • 4=15 MHz • 5=20 MHz

Table 5-2: Test command details (Continued)

Command	Description
!DALSTXBW Supporting chipsets: • MDM9200 (min f/w rev: SWI9200M_1.0 Release 5, or SWI9200X_3.0 Release 2) • MDM9600 (min f/w rev: SWI9600M_01.00.09.04)	**Set LTE Tx bandwidth (LTE only)** Set the LTE Tx bandwidth. Requirements: • **!DASBAND** must be issued before you can use this command. **Usage:** • Execution: **AT!DALSTXBW=<bw>** Response: OK Purpose: Set the LTE Tx bandwidth. **Parameters:** <bw> (LTE bandwidth) • 0=1.4 MHz • 1=3 MHz • 2=5 MHz • 3=10 MHz • 4=15 MHz • 5=20 MHz
!DAOFFLINE Supporting chipsets: • All	**Place modem offline** Put the modem offline. **Usage:** • Execution: **AT!DAOFFLINE** Response: OK Purpose: Put the modem offline. **Parameters:** None

Table 5-2: Test command details (Continued)

Command	Description

!DASBAND
Supporting chipsets:
• All, with these exceptions:
  • MSM6290 (min f/w rev: K1.0 Release 12 (for W900 band))
  • MDM9200 (min f/w rev: 3.00.05.04 Release 1)
 	
*Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*
	Set frequency band
	Set the modem to use a particular frequency band. You must use this command to select an appropriate band before running LTE, WCDMA, or GSM commands. See page 93.
	Usage:
	• Execution: **AT!DASBAND=<rfband>**
	Response: <rfband>
	OK
	Purpose: Set frequency band.
	Parameters:
	<rfband> (Unique value corresponding to an RF band and technology.)
	• This is a unique value that maps to an RF band and technology. It is not an actual 3GPP band number. For example, '18' is GSM 850, which corresponds to 3GPP band 5 (on a GSM network).
	• Band support is product specific—see the device's Product Specification or Product Technical Specification document for details.
	• Examples (for a full listing, see Table D-1 on page 279):
	  • GSM
	  • 10=GSM 900
	  • 11=GSM 1800
	  • 12=GSM 1900
	  • 18=GSM 850
	  • WCDMA
	  • 9=WCDMA 2100
	  • 16=WCDMA 1900B
	  • 22=WCDMA 850
	  • 29=WCDMA 900 (BC8)
	  • LTE
	  • 34=LTE B1
	  • 35=LTE B7
	  • 36=LTE B13
	  • 37=LTE B17
	  • 42=LTE B4
	  • 44=LTE B3
	  • 47=LTE B8
	  • 56=LTE B20

Table 5-2: Test command details (Continued)

Command	Description
!DASCHAN Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set modem channel (frequency)** Set the modem to operate on a particular frequency channel. Before using this command, use the command !DASBAND (described on page 110) to set the band. Once a channel is set, the modem continues to use that channel until the modem is reset or powered off and on. **Usage:** • Execution: **AT!DASCHAN=<rfchannel>** Response: <rfchannel> OK Purpose: Set modem channel (frequency). **Parameters:** <rfchannel> (Uplink channel number (ARFCN)—depends on frequency band being used) • 128–251: GSM 850 MHz • 1–24: GSM 900 MHz • 975–1023: GSM 900 MHz • 512–885: GSM 1800 MHz • 512–810: GSM 1900 MHz • 9612–9888: WCDMA 2100 • 9262–9538: WCDMA 1900 • 4132–4233: WCDMA 850 • 2712–2863: WCDMA 900 • 18000–18599: LTE B1 • 19200–19949: LTE B3 • 19950–20399 LTE B4 • 20750–21449: LTE B7 • 21450–21799: LTE B8 • 23180–23279: LTE B13 • 23730–23849: LTE B17 • 24150–24449: LTE B20 *Note: **!DASBAND** must be issued before you can use **!DASCHAN**.*

Table 5-2: Test command details (Continued)

Command	Description
!DASLNAGAIN Supporting chipsets: • All	**Set LNA gain state** Set the LNA (Low Noise Amplifier) range for the main or diversity path (if applicable), in either WCDMA or GSM mode. *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DASLNAGAIN**.* **Usage:** • Execution: **AT!LNAGAIN=<gain index>[, <path>]** Response: <gain index> OK Purpose: Set the LNA gain state for either the main or diversity paths. **Parameters:** <gain index> • 0=R0 (highest gain) Approximate switch from low to high gain: WCDMA (< -72 dBm); GSM (< -73 dBm) • 1=R1 Approximate switch from low to high gain: WCDMA (< -72 up to -46 dBm); GSM (< -73 up to -58 dBm) • 2=R2 Approximate switch from low to high gain: WCDMA (< -46 up to -36 dBm); GSM (< -58 up to -41 dBm) • 3=R3 (lowest gain) Approximate switch from low to high gain: WCDMA (> -36 dBm); GSM (< -41 dBm) *Note: The LNA gain state is set based on the expected receive power level. The gain state values listed above are provided as a guideline. The values are approximations and subject to change over time. The values are different than those from high to low gain.* <path> (For modules supporting diversity) • 0=Main path • 1=Secondary (diversity) path

Table 5-2: Test command details (Continued)

Command	Description
!DASPDM Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set PDM value** Adjust the PDM (Pulse Duration Modulation), allowing you to apply frequency offset to the LO (Local Oscillator) or Tx AGC. When you adjust the Tx AGC (<PDM ID> = 2), the modem does not use a calibrated result but uses the raw AGC value. The resulting change in Tx power will vary from modem to modem, so it is usually necessary to tune this value by executing the command repeatedly with different settings for the <PDMvalue> until you obtain the desired Tx power. When adjusting the tracking LO, you also need to execute the command repeatedly with different settings for the <PDMvalue> until you obtain the desired frequency offset. *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DASPDM**.* **Usage:** • Execution: **AT!DASPDM=<PDM ID>, <PDMvalue>** Response: <PDM ID> <PDMvalue> OK Purpose: Set the tracking LO and Tx AGC PDM. **Parameters:** <PDM ID> (LO (Local Oscillator) or Tx AGC (Automatic Gain Control) to adjust) • 0=Tracking LO adjust (GSM only) • 2=Tx AGC adjust (WCDMA only) • 4=Tracking LO adjust (WCDMA only) <PDMvalue> (Frequency offset value) • If <PDM ID>=0: 0–511 • If <PDM ID>=2: 0–511 • If <PDM ID>=5: 0–65536
!DASTXOFF Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Turn Tx PA off** Turn the transceiver PA off, after it has been turned on with !DASTXON. *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DASTXOFF**.* **Usage:** • Execution: **AT!DASTXOFF** Response: OK Purpose: Turn the Tx PA off. **Parameters:** None

Table 5-2: Test command details (Continued)

Command	Description
!DASTXON Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Turn Tx PA on** Turn on the transceiver PA (either the WCDMA PA or the GSM PA, depending on the mode set with !DASBAND). The PA then remains on until you turn it off using the !DASTXOFF command, or until you reset or power the modem down and up. *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DASTXON**.* **Usage:** • Execution: **AT!DASTXON** Response: OK Purpose: Turn the Tx PA on. **Parameters:** None
!DAWGAVGAGC Supporting chipsets: • All	**Return averaged Rx AGC value (WCDMA only)** Return the averaged AGC (Automatic Gain Control) reading for a specific band for either the main path or diversity path (if applicable). **Usage:** • Execution: **AT!DAWGAVGAGC=<channel>, <LNA Index>[, <path>]** Response: <agc> OK Purpose: Return the averaged AGC for <channel> on the main path or diversity path. **Parameters:** <channel> (Uplink channel number (UARFCN) for the band specified using **!DASBAND**) • Valid values depend on the selected band <LNA Index> (LNA offset index) • 0=R0 (Highest gain) • 1=R1 • 2=R2 • 3=R3 (Lowest gain) <path> (For modules supporting diversity) • 0=Main path • 1=Diversity path <agc> (Averaged Rx AGC in dBm) • Example: -78.9 *Note: **!DASBAND** must be issued before you can use **!DAWGAVGAGC**.*

Table 5-2: Test command details (Continued)

Command	Description

!DAWGRXAGC
Supporting chipsets:
• All
	Return Rx AGC value (WCDMA only)
	Return the Rx AGC (Automatic Gain Control) value of the main path or diversity path (if applicable).
	This value can be converted to RSSI (Received Signal Strength Indicator) in dBm:
	if (<AGC_value> < 511)
	<RX_dBm> = -106 + ((<AGC_value> + 512) / 12)
	else
	<RX_dBm> = -106 + (((<AGC_value>-1024) + 512) / 12)
	Usage:
	• Execution: **AT!DAWGRXAGC?[<path>]**
	Response: <AGC value>
	OK
	Purpose: Return the <AGC value> for either the main or diversity paths. If no <path> is specified, the main path is assumed.
	Parameters:
	<path> (For modules supporting diversity)
	• 0=Main path
	• 1=Diversity path
	<AGC value> (Rx AGC value for specified path)
	• Valid range: -512 to +511
	*Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWGRXAGC**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAWINFO Supporting chipsets: • All *Note: The modem must be in online mode to use this command.*	**Return WCDMA mode RF information (WCDMA only)** Return RF information for WCDMA mode when the modem is in CELL_DCH (Designated Channel) state. *Note: !DAGINFO provides RF information for GSM mode.* **Usage:** • Execution: **AT!DAWINFO** Response: RXAGC:<rxAGC> TXAGC:<txAGC> TXADJ:<txAdj> TXLIM:<txLim> LNA:<lnaRange> PA ON:<paOn> TX ON:<txOn> PA Range:<paRange> RxD RXAGC:<RXDrxAGC> RxD LNA:<RXDlnaRange> HDET:<hdet> OK Purpose: Return the RF information. **Parameters:** <rxAGC> (Rx AGC value) • Valid range: 0–65535 <txAGC> (Tx AGC value) • Valid range: 0–65535 <txAdj> (Tx AGC value after linearization (adjustment)) • Valid range: 0–65535 <txLim> (Tx AGC limit) • Valid range: 0–65535 <lnaRange> (State of the LNA) • Valid range: 0–65535 <paOn> (State of PA_ON0) • Valid range: 0–65535 <txOn> (State of TX_ON) • Valid range: 0–65535 <paRange> (State of PA_R1: PA_R0) • Valid range: 0–65535 <RXDrxAGC> (RxD Rx AGC value) • Valid range: 0–65535 <RXDlnaRange> (State of the RxD LNA) • Valid range: 0–65535 <hdet> (Raw HDET (High Power Detector) data) • HDET information appears for devices using firmware revisions M3.0 Release 1 or higher • Valid range: 0–255

Table 5-2: Test command details (Continued)

Command	Description
!DAWSCONFIGRX Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set WCDMA receiver to factory calibration settings** Configure the WCDMA receiver according to factory calibration settings stored in the modem's NV (Non-Volatile memory). This allows for accurate measurement of Rx AGC levels. The command performs these steps: 1. Sets the channel. 2. Selects and sets LNA range (or LNA gain). 3. Sets the VGA gain offset based on the channel. 4. Sets the LNA range offset. **Usage:** • Execution: **AT!DAWSCONFIGRX=<channel>,<Rx_Level_dBm>** Response: <LNA Index>, <LNA Value> OK Purpose: Configure the receiver. **Parameters (Input):** <channel> (Uplink channel number (ARFCN)) • Value based on the selected band <Rx_Level_dBm> (Approximate signal level (in dBm) being applied to the modem receiver) • Valid range: -113 to 20 **Parameters (Output):** <LNA Index> (LNA offset index) • 0=R0 (highest gain) • 1=R1 • 2=R2 • 3=R3 (lowest gain) <LNA Value> • Internal use only *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWSCONFIGRX**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAWSPARANGE Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set PA range state machine** Set the PA range state machine in WCDMA operation. **Usage:** • Execution: **AT!DAWSPARANGE=<PA range>** Response: <PA range> OK Purpose: Set the PA range state machine. **Parameters:** <PA range> • 0=Low gain state of the PA — Limited to about 16 dBm output power (R0=0, R1=0) • 3=High gain state of the PA — Up to the maximum output power of the modem (R0=1, R1=1) *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWSPARANGE**.*
!DAWSSCHAIN Supporting chipsets: • All	**Enable secondary receive chain (WCDMA only)** Enable or disable the secondary receive chain. **Usage:** • Execution: **AT!DAWSSCHAIN=<state>** Response: OK Purpose: Enable or disable the secondary receive chain. **Parameters:** <state> (Requested state for secondary receive chain) • 0=Off (Disable) • 1=On (Enable) *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWSSCHAIN**.*
!DAWSCHAINTCM Supporting chipsets: • All	**Place receive chain in test call mode (WCDMA only)** Place one or both of the primary and secondary receive chains in test call mode. **Usage:** • Execution: **AT!DAWSCHAINTCM=<chain>** Response: OK Purpose: Place requested receive chain(s) in test call mode. **Parameters:** <chain> (Receive chain to place in test call mode) • 0=Main • 1=Secondary • 2=Both *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWSCHAINTCM**.*

Table 5-2: Test command details (Continued)

Command	Description
!DAWSTXCW Supporting chipsets: • All *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set waveform used by the transmitter** Set the waveform used by the transmitter—the modem can transmit either in carrier wave or WCDMA modulated. **Usage:** • Execution: **AT!DAWSTXCW=<waveform>** Response: OK Purpose: Set the transmitter waveform. **Parameters:** <waveform> (Waveform used by the transmitter) • 0=WCDMA • 1=Carrier wave (no modulating signal applied) *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWSTXCW**.*
!DAWSTXPWR Supporting chipsets: • All, with these exceptions: • MDM8200 (min f/w rev: M2.0 Release 1) *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set desired Tx power level (WCDMA mode only)** Set the desired Tx power level in dBm. When this occurs, the PA range and PDM are automatically updated as well. (When this command is used, you do not need to use **!DAWSPARANGE** and **!DASPDM**.) **Usage:** • Execution: **AT!DAWSTXPWR=<dBm>[,<use_temp_comp>]** Response: <pa_range>, <pdm> OK Purpose: Set the Tx power level to the requested <dBm> level, and automatically set the PA range and PDM. **Parameters:** <dBm> (Desired Tx power in dBm) • Valid range: -57 to 28 <use_temp_comp> (Apply temperature compensation to set desired Tx power.) • Frequency compensation is always applied. • 0=No (default) • 1=Yes <pa_range> (PA range set as a result of the command) • Valid range: 0–3 <pdm> (PDM set as a result of the command) • Valid range: 0–255 *Note: **!DASBAND** and **!DASCHAN** must be issued before you can use **!DAWSTXPWR**.*

Table 5-2: Test command details (Continued)

Command	Description
!IMSTESTMODE Supporting chipsets: • MDM9600	**Enable/disable IMS test mode** Enable/disable IMS (IP Multimedia Subsystem) test mode. If IMS test mode is enabled: • IMS registration attempts will not occur • SMS is not supported **Usage:** • Execution: **AT!IMSTESTMODE=<mode>** Response: OK Purpose: Enable/disable IMS test mode. • Query: **AT!IMSTESTMODE?** Response: IMS Test Mode Enabled *or* IMS Test Mode Disabled Purpose: Return the current state of IMS Test Mode. **Parameters:** <mode> (IMS Test Mode state) • 0=Disable • 1=Enable
!KEYOFF Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Key off the transmitter** Turn off the transmitter's power amplifier. The transmitter circuitry remains powered until !TX on page 123 (!Tx=0) is used. **Usage:** • Execution: **AT!KEYOFF** Response: OK Purpose: Turn off the transmitter's power amplifier. **Parameters:** None

Table 5-2: Test command details (Continued)

Command	Description
!KEYON Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Key on the transmitter** Tune the radio, key the transmitter at maximum, and select pseudo-random noise or a sine wave signal. The power amplifier is set to maximum gain and the output power limit is disabled. *Note: This command enables the transmitter circuitry—!TX does not need to be used before !KEYON.* To turn off the power amplifier, use !KEYOFF on page 120. To turn off the transmitter, use !TX on page 123. **Usage:** • Execution: **AT!KEYON=<channel>,<band>,<wave>** Response: OK Purpose: Key on the transmitter **Parameters:** <channel> • 0–799, 991–1023 (Cellular) • 0–1200 (PCS) <band> • 0=US Cellular • 1=PCS <wave> • 0=PN (pseudo-random noise) • 1=SINE (sine wave)
!OSDSM Supporting chipsets: • All	**Display memory usage for DSM buffer pools** Display memory usage of the DSM (Distributed Shared Memory) buffer pools. This provides a means of detecting memory leaks. **Usage:** • Query: **AT!OSDSM?** Response: Pool n Size <size> Free <free>/<max> Min <min free> … *(list displays one line for each buffer pool from 0 to n)* OK Purpose: Display memory usage. **Parameters:** All values are ASCII strings representing hexadecimal numbers from 0x0000 to 0xFFFF. <size> (Size (in bytes) of each item in the pool) • Valid range: 0x0000–0xFFFF <free> (Number of items currently available in the pool) • Valid range: 0x0000–0xFFFF <max> (Total number of items in the pool) • Valid range: 0x0000–0xFFFF <min free> (Lowest number of free items since power-on) • Valid range: 0x0000–0xFFFF

Table 5-2: Test command details (Continued)

Command	Description

$QCAGC
Supporting chipsets:
• MDM9600
 	
*Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*
	Read Rx AGC (CDMA and WCDMA modes)
	This command reads the Rx AGC in dBm for CDMA and WCDMA modes.
	Usage:
	• Execution: **AT!QCAGC=<band>,<rx_chan>,<path>**
	Response: RSSI: <rssi>
	OK
	Purpose: Read Rx AGC for specified band, channel, and path.
	• Query List: **AT$QCAGC=?**
	Purpose: Return a list of supported parameters.
	Parameters:
	<band>
	• Valid bands:
	  • "CDMA_800"
	  • "CDMA_1900"
	  • "WCDMA_IMT"
	  • "WCDMA_800"
	  • "WCDMA_900"
	  • "WCDMA_1900"
	<rx_chan> (Rx channel)
	• Valid channels by <band>:
	  • CDMA_800: 1–799, 991–1023
	  • CDMA_1900: 0–1199
	  • WCDMA_IMT: 10562–10842
	  • WCDMA_800: 4357–4467, 1007–1087
	  • WCDMA_900: 2937–3097
	  • WCDMA_1900: 9662–9942, 412–687
	<path> (Receiver path)
	• Valid paths:
	  • "MAIN"
	  • "AUX"
	<rssi> (RSSI AGC reading in dBm)
	• Example: -52

!RX2 Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Turn second receiver on/off** Turn the second receiver on or off. **Usage:** • Execution: **AT!RX2=<state>** Response: OK Purpose: Turn the second receiver on or off. **Parameters:** <state> • 0=Turn receiver off • 1=Turn receiver on

Table 5-2: Test command details (Continued)

Command	Description
!RX2AGC Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Read second receiver Rx AGC** Return the Rx AGC value of the second receiver. **Usage:** • Query: **AT!RX2AGC?** Response: RX2AGC = <agc> = [integer agc]dBm OK Purpose: Return second receiver's AGC. **Parameters:** <agc> (Rx AGC value) • Valid values: 0x000–0x3FF • 0x000 is most positive, 0x3FF is lowest • 0x200 represents 50% duty cycle
!RXAGC Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Read first receiver Rx AGC** Return the Rx AGC value of the first receiver. **Usage:** • Query: **AT!RXAGC?** Response: RXAGC = <agc> = [integer agc]dBm OK Purpose: Return first receiver's AGC. **Parameters:** <agc> (Rx AGC value) • Valid values: 0x000–0x3FF • 0x000 is most positive, 0x3FF is lowest • 0x200 represents 50% duty cycle
!TX Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Turn transmitter on/off** Turn the transmitter on or off. **Usage:** • Execution: **AT!TX=<state>** Response: OK Purpose: Turn the transmitter on or off. **Parameters:** <state> • 0=Turn transmitter off • 1=Turn transmitter on

Table 5-2: Test command details (Continued)

Command	Description
!TXAGC Supporting chipsets: • MDM9600 *Note: The modem must be in FTM mode to use this command—use **!DAFTMACT** to enter FTM mode.*	**Set desired Tx AGC** Set the desired Tx AGC. **Usage:** • Query: **AT!TXAGC=<agc>** Response: OK Purpose: Return first receiver's AGC. **Parameters:** <agc> (Tx AGC value) • Valid values: 0x000–0x1FF • 0x000 is lowest, 0x1FF is most positive • 0x100 represents 50% duty cycle

6: Memory Management Commands

Introduction

The modem has 2 MB of non-volatile memory that is used to store:

- Factory calibration data
- Settings made in a host application such as Watcher

The commands in this chapter allow you to back up and restore the data in non-volatile memory.

Command summary

The table below lists the commands described in this chapter:

Table 6-1: Memory management commands

Command	Description	Page
!NVDEF	Reset non-volatile memory	126
!NVRESTORE	Restore backup data	126

Command reference

Table 6-2: Memory management command details

Command	Description
!NVDEF Supporting chipsets: • All	**Reset non-volatile memory** **Warning:** *This command erases all calibration data, customizations, etc. First use **!NVBACKUP** to save these settings, and then, after using this command, use **!NVRESTORE** (p.126) to restore the settings.* *Note: Profiles (PDP contexts) are not restored using AT commands. The host application is responsible for implementing this task.* Reset non-volatile memory to default values and then restore the modem's FSN. All calibration data, customizations, etc., are removed. *Note: This command may take 20–30 seconds to complete.* **Usage:** • Execution: **AT!NVDEF** Response: !NVDEF: Wiping NV, restoring defaults for S1614350476E10F *(FSN shown is an example)* NV Items Defaulted: <defaulted> OK Purpose: Clear and reset NV items. **Parameters:** <defaulted> (Number of NV items defaulted) • Valid range: 0–65535 (This number varies depending on firmware version.)
!NVRESTORE Supporting chipsets: • All	**Restore backup data** Restore items to non-volatile memory that were backed up with the command **AT!NVBACKUP**, and return the number of NV items restored. **Usage:** • Execution: **AT!NVRESTORE=<category>** Response: !NVRESTORE: NV Items Restored: <restored> OK Purpose: Clear and reset NV items. **Parameters:** <category> (Item type to be restored) • 0=Factory items (RF calibration data) • 1=OEM items (Factory configuration, unless user has performed an AT!NVBACKUP=1) • 2=User items (Configuration as of most recent firmware download, including all user customizations) <restored> (Number of NV items restored) • Valid range: 0–255

7: Voice Commands

Introduction

Note: This chapter applies to modems supporting analog and PCM digital audio interfaces. For modems supporting I2S audio, see I2S Audio Commands on page 163.

Voice-enabled Sierra Wireless Mini Card modems (for example, MC8795V) have built-in audio support that allows the modems to be used as mobile phones.

Note: Voice-enabled SL808x modems support the general audio functionality described in this chapter. Specific details are to be determined.

The modems all support a single analog audio interface that can be used for handset-style applications or headset applications, and a PCM digital audio interface. The modems have a wide range of software-controlled audio filtering and amplification stages, which minimize the amount of external circuitry required on the host system.

At its most basic configuration, the host audio system could contain:

- A microphone/speaker combination (handset)

  —or—

- A headset jack

The audio pass band for the primary and secondary audio paths extends from 300 Hz to 3.4 kHz in both the receive and transmit directions.

These transmit audio features are supported (depending on module type):

- Adjustable gain up to +40 dB
- Several adjustable filtering stages (high-pass and slope filters)
- Noise cancellation
- Configurable echo cancellation for various acoustic environments. (For example, headset, handset, and speakerphone.)
- Adjustable sidetone from mute to unity gain

These receive audio features are supported:

- Adjustable gain up to +24 dB
- High-pass filter stage
- Output driver stages, which can drive speakers directly

The modems also provide 13-tap FIR (Finite Impulse Response) filtering for both the receive and transmit paths to equalize the acoustic response of the speaker and microphone.

The embedded module can serve as an integral component of a more complex audio system, such as a PDA with a separate codec interfacing with the main processor. In this case, the interface between the modem and PDA codec can be as simple as line-level audio with no transducer considerations. Phone-oriented functions such as echo cancellation and FIR filtering are typically left to the module, while path-switching and transducer interfaces are the responsibility of the PDA codec. Functions such as adjustable gain and volume settings, DTMF and ringer tone generation, and mixing can be accomplished in either codec, depending on the architecture of the particular product. The interface between the module and host audio systems is usually the primary audio interface set to line-level amplitudes, routed as differential pairs for noise immunity.

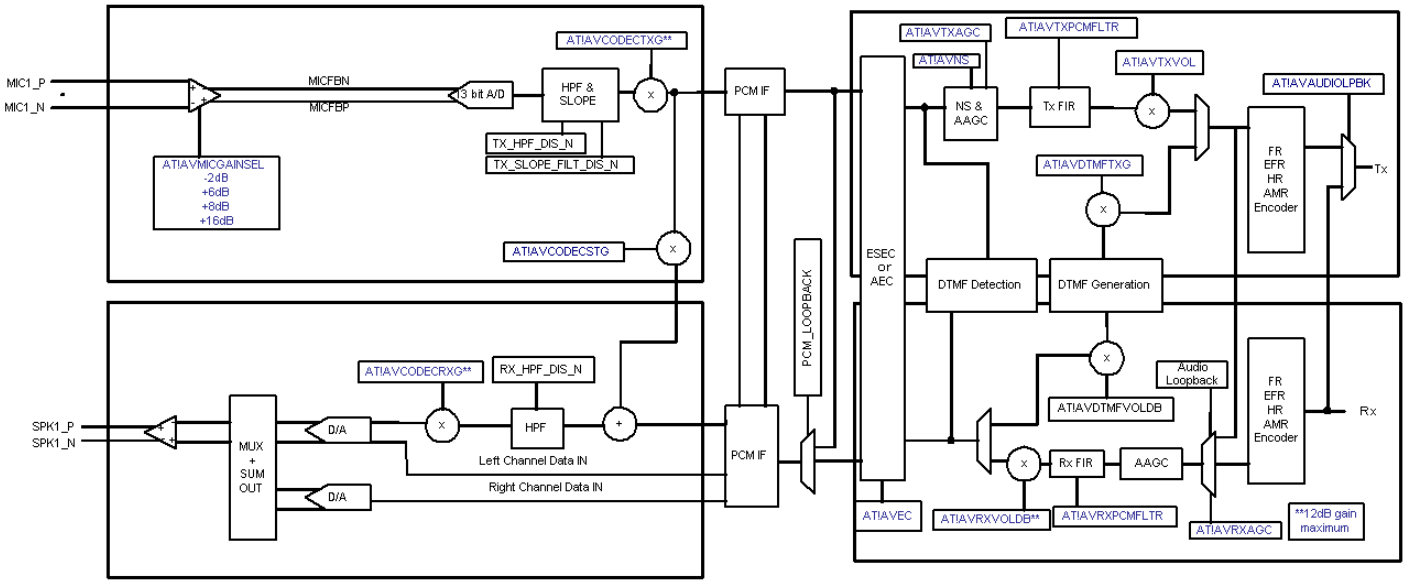

Figure 7-1: Mini Card audio block diagram

Audio profiles

The AT command set allows you to have different audio configurations for different purposes. As an example, assume you are embedding the module in a device that has a handset mode and a speakerphone mode. Assume also that you want to use different transmit gain, noise suppression, and echo cancellation settings in each mode. You can store the settings for each mode in separate audio profiles, then activate the appropriate profile as your application switches to handset or speakerphone mode.

The AirPrime UMTS/LTE embedded module supports some combination of the following audio profiles, depending on the module type:

- 0—Handset (MSM6290, QSC6270)
- 1—Headset (MSM6290, QSC6270)

- 2—Car kit (MSM6290, QSC6270)
- 3—Speaker phone (MSM6290, QSC6270)
- 4—Auxiliary (MSM6290, QSC6270)
- 5—TTY (TeleTYpe—a device that allows speech and hearing-impaired people to use a phone) (MSM6290, QSC6270)
- 6—Auxiliary external PCM (128 kHz clock) (MSM6290, QSC6270)
- 7—Primary external PCM (2 MHz clock) (MDM6200, MSM6290, QSC6270)

AT commands that are used to change the audio configuration have a profile parameter; any changes you make are applied to the specified profile. The default audio configurations for each profile are shown in Table 7-1, Table 7-2, and Table 7-3.

Table 7-1: MDM6200 audio profile default settings

Setting	Profile ID
	7
Automatic Gain Control (Tx) **AT! AVTXAGC**	Off
Noise Suppression (Tx) **AT! AVNS**	Off
AGC, AVC (Rx) **AT!AVRXAGC**	Off
Echo Cancellation **AT!AVEC**	Headset
Tx gain **AT!AVTXVOL**	0 dB
Sidetone gain **AT!AVCODECSTG**	n/a

Table 7-2: MSM6290 audio profile default settings

Setting	Profile ID							
	0	1	2	3	4	5	6	7
Automatic Gain Control (Tx) **AT! AVTXAGC**	Off	Off	Off	Off	Off	Off	Off	Off
Noise Suppression (Tx) **AT! AVNS**	On	On	Off	Off	On	On	Off	Off
AGC, AVC (Rx) **AT!AVRXAGC**	Off	Off	Off	Off	Off	Off	Off	Off
Echo Cancellation **AT!AVEC**	ESEC	Headset	AEC	Speaker phone	Headset	ESEC	Headset	Headset
Tx gain **AT!AVTXVOL**	0 dB	0 dB	0 dB	0 dB	0 dB	0 dB	0 dB	0 dB
Microphone gain **AT!AVMICGAIN**	24.0 dB	24.0 dB	24.0 dB	24.0 dB	24.0 dB	0 dB	n/a	n/a

Table 7-2: MSM6290 audio profile default settings (Continued)

Setting	Profile ID							
	0	1	2	3	4	5	6	7
Sidetone gain **AT!AVCODECSTG**	-24.0 dB	-24.0 dB	-24.0 dB	-24.0 dB	-24.0 dB	Mute	n/a	n/a
Codec Tx gain **AT!AVCODECTXG**	4 dB	4 dB	4 dB	4 dB	4 dB	0 dB	n/a	n/a
Codec Rx gain **AT!AVCODECRXG**	3.5 dB	3.5 dB	3.5 dB	3.5 dB	3.5 dB	0 dB	n/a	n/a

Table 7-3: QSC6270 audio profile default settings

Setting	Profile ID							
	0	1	2	3	4	5	6	7
Automatic Gain Control (Tx) **AT! AVTXAGC**	Off	Off	Off	Off	Off	Off	Off	Off
AGC, AVC (Rx) **AT!AVRXAGC**	Off	Off	Off	Off	Off	Off	Off	Off
Echo Cancellation **AT!AVEC**	ESEC	Headset	AEC	Speaker phone	Headset	ESEC	Headset	Headset
Tx gain **AT!AVTXVOL**	0 dB	0 dB	0 dB	0 dB	0 dB	0 dB	0 dB	0 dB
Microphone gain **AT!AVMICGAIN**	24.0 dB	24.0 dB	24.0 dB	24.0 dB	24.0 dB	0 dB	n/a	n/a
Sidetone gain **AT!AVCODECSTG**	-24.0 dB	-24.0 dB	-24.0 dB	-24.0 dB	-24.0 dB	Mute	n/a	n/a
Codec Tx gain **AT!AVCODECTXG**	4 dB	4 dB	4 dB	4 dB	4 dB	0 dB	n/a	n/a
Codec Rx gain **AT!AVCODECRXG**	3.5 dB	3.5 dB	3.5 dB	3.5 dB	3.5 dB	0 dB	n/a	n/a

Profile activation

(MSM6290/QSC6270)

Profile 0 (Handset) is the default profile. Unless you activate a different profile prior to establishing a circuit-switched call, the default profile is used in establishing the call. To use a profile other than Profile 0, use the command AT!AVSETPROFILE (described on page 151) to activate the profile prior to establishing the call.

Command summary

The table below lists the commands described in this chapter:

Table 7-4: Voice commands

Command	Description	Page
!AVAUDIOLPBK	Enable/disable an audio loopback	132
!AVCODECRXG	Set/report CODEC Rx gain	132
!AVCODECSTG	Set/report CODEC sidetone gain	134
!AVCODECTXG	Set/report CODEC Tx gain	135
!AVDEF	Set audio settings to default values	136
!AVDTMFTXG	Set/report the DTMF Tx gain	137
!AVDTMFVOLDB	Set/report volume for each DTMF volume level in Rx direction	138
!AVEC	Set/report the echo cancellation setting	139
!AVEXTPCMCFG	Configure external PCM interface	140
!AVEXTPCMSTOPCLKOFF	Prevent/allow external PCM interface clock from turning off	141
!AVINBANDRANGE	Specify Progress Descriptor value range for in-band signaling	142
!AVMICGAIN	Set/report microphone gain	143
!AVNS	Enable/disable noise suppression	144
!AVRXAGC	Set/report Rx AVC/AGC configuration	145
!AVRXPCMFLTR	Set/report the Rx PCM filter tap	146
!AVRXPCMIIRFLTR	Set/report the Rx PCM IIR filter parameters	148
!AVRXVOLDB	Set/report volume for each voice volume level in Rx direction	150
!AVSETPROFILE	Configure and activate profile	151
!AVSN	Set/report audio revision number	152
!AVTONEPLAY	Play DTMF tone	152
!AVTONESETTINGS	Enable/disable playing of locally-generated DTMF tones	153
!AVTXAGC	Set Tx AGC	154
!AVTXPCMFLTR	Set/report Tx PCM filter tap	155
!AVTXPCMIIRFLTR	Set/report the Tx PCM IIR filter parameters	156
!AVTXVOL	Set Tx volume	158

Command reference

Table 7-5: Voice command details

Command	Description
!AVAUDIOLPBK Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K1_0_1_0ap) • QSC6270 (min f/w rev: S2.0)	**Enable/disable an audio loopback** Configure an audio loopback. The loopback occurs at the vocoder interface and tests the audio front end with the EFR (Enhanced Full Rate) vocoder. **Usage:** • Execution: **AT!AVAUDIOLPBK=<enable>** Response: OK Purpose: Enable or disable audio loopback. **Parameters:** <enable> (Enable/disable the loopback) • 0=Loopback off • 1=Loopback on
!AVCODECRXG Supporting chipsets (voice-enabled devices only): • MSM6290 (min f/w rev: K1_0_2_0ap) • QSC6270 (min f/w rev: S2.0)	**Set/report CODEC Rx gain** Set the CODEC Rx gain for the specified audio profile. The CODEC Rx gain is applied to the digital signal prior to its conversion to the analog domain to provide additional gain range from -84dB to +12dB in the receive direction. This setting is stored in non-volatile memory and persists across power cycles. *Note: This command has no effect when PCM audio is being used.* **Device-specific details:** • MSM6290 (Firmware revisions K2_0_7_8ap and lower): • The change takes effect the next time the modem restarts. • All other chipsets/firmware revisions: • The change takes effect immediately for the specified profile. **Usage:** • Execution: **AT!AVCODECRXG=<profile>, <value>** Response: OK Purpose: Set the CODEC Rx gain for the specified <profile>. • Query: **AT!AVCODECRXG?<profile>** Response: <value> OK Purpose: Return the current CODEC Rx gain (<value>) for the specified <profile>. (Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description

!AVCODECRXG (continued)	**Set/report CODEC Rx gain (continued)**
	Parameters:
	<profile> (Audio profile number)
	• Valid values:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	<value> (Gain value—selected <profile>)
	• The value is entered/returned in hexadecimal format:
	  • Min=0x0000 (mute)
	  • Max=0xFFFF
	• To calculate the gain in dB, convert <value> to decimal (<decvalue>) and use the following formula:
	Gain = 20 $\log_{10}$ (<decvalue> / 16384)
	• Unity gain=0x4000

Table 7-5: Voice command details (Continued)

Command	Description

!AVCODECSTG
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set/report CODEC sidetone gain
	Set the CODEC sidetone gain for the specified audio profile. Sidetone gain is the portion of audio from the microphone that gets routed back to the user's speaker. This prevents the user from speaking too loudly by making them aware of the volume of their own voice.
	The gain range is chipset-dependent:
	• QSC6270: -96dB to 0dB
	• Other chipsets: -84dB to +12dB
	This setting is stored in non-volatile memory and persists across power cycles.
	Note: (MSM6290, QSC6270) This command has no effect when PCM audio is being used.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVCODECSTG=<profile>, <value>**
	Response: OK
	Purpose: Set the CODEC sidetone gain for the specified <profile>.
	• Query: **AT!AVCODECSTG?<profile>**
	Response: <value>
	OK
	Purpose: Return the current CODEC sidetone gain (<value>) for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	(Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description

!AVCODECSTG (continued)	**Set/report CODEC sidetone gain (continued)**
	<value> (Gain value for selected <profile>)
	• The value is entered/returned in hexadecimal format:
	  • Min=0x0000 (mute)
	  • Max=0xFFFF
	• To calculate the gain in dB, convert <value> to decimal (<decvalue>) and use the following formula:
	Gain = 20 $\log_{10}$ (<decvalue> / 16384) - 12
	• Unity gain=0xFECA

!AVCODECTXG
Supporting chipsets (voice-enabled devices only):
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set/report CODEC Tx gain
	Set the CODEC Tx gain for the specified audio profile. The CODEC Tx gain is applied to the digital signal after its conversion from the analog domain to provide additional gain range in the transmit direction.
	The gain range is chipset-dependent:
	• QSC6270: -66dB to +30dB
	• Other chipsets: -84dB to +12dB
	This setting is stored in non-volatile memory and persists across power cycles.
	Note: This command has no effect when PCM audio is being used.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVCODECTXG=<profile>, <value>**
	Response: OK
	Purpose: Set the CODEC Tx gain for the specified <profile>.
	• Query: **AT!AVCODECTXG?<profile>**
	Response: <value>
	OK
	Purpose: Return the current CODEC Tx gain (<value>) for the specified <profile>.
	(Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description

!AVCODECTXG (continued)	**Set/return the Tx gain (continued)**
	Parameters:
	<profile> (Audio profile number)
	• Valid values:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	<value> (Gain value for selected <profile>)
	• The value is entered/returned in hexadecimal format:
	  • Min=0x0000 (mute)
	  • Max=0xFFFF
	• QSC6270:
	  • To calculate the gain in dB, convert <value> to decimal (<decvalue>) and use the following formula:
	  Gain = 20 $\log_{10}$ (<decvalue> / 2048)
	  • Unity gain=0x0800
	• Other chipsets:
	  • To calculate the gain in dB, convert <value> to decimal (<decvalue>) and use the following formula:
	  Gain = 20 $\log_{10}$ (<decvalue> / 16384)
	  • Unity gain=0x4000

!AVDEF
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set audio settings to default values
	Set all the configurable audio parameters to default values. The default values are also loaded into non-volatile memory.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVDEF**
	Response: OK
	Purpose: Set audio settings to default values.

Table 7-5: Voice command details (Continued)

Command	Description
!AVDTMFTXG Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K1_0_2_0ap) • QSC6270 (min f/w rev: S2.0)	**Set/report the DTMF Tx gain** Set the DTMF Tx gain for the specified audio profile. The DTMF Tx gain determines the gain, from -84dB to +12dB, for the DTMF tone that is transmitted over the air. This setting is stored in non-volatile memory and persists across power cycles. The change to the audio system takes effect when a phone call is made or received. **Usage:** • Execution: **AT!AVDTMFTXG=<profile>, <value>** Response: OK Purpose: Set the DTMF Tx gain for the specified <profile>. • Query: **AT!AVDTMFTXG?<profile>** Response: <value> OK Purpose: Return the current DTMF Tx gain (<value>) for the specified <profile>. **Parameters:** <profile> (Audio profile number) • MSM6290, QSC6270: • 0=Handset • 1=Headset • 2=Car kit • 3=Speaker phone • 4=Auxiliary • 5=TTY • 6=Auxiliary external PCM (128 kHz clock) • 7=Primary external PCM (2 MHz clock) • MDM6200: • 7=Primary external PCM (2 MHz clock) <value> (Gain value for selected <profile>) • The value is entered/returned in hexadecimal format. Valid ranges: • 0x0000–0x4B0 (0x0000=mute) • 0xEC78–0xFFFF • To calculate the gain in dB, convert <value> to decimal (<decvalue>) and use the following formula: Gain = 20 $\log_{10}$ (<decvalue> / 16384) • Unity gain=0x4000

Table 7-5: Voice command details (Continued)

Command	Description
!AVDTMFVOLDB Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K1_0_2_0ap) • QSC6270 (min f/w rev: S2.0)	**Set/report volume for each DTMF volume level in Rx direction** Set the volume for each DTMF voice 'volume level' in the Rx direction. The setting is stored in non-volatile memory and persists across power cycles. The change to the audio system takes effect immediately if the specified path is active and all the volumes have been initialized. **Usage:** • Execution: **AT!AVDTMFVOLDB=<profile>, <generator>, <level>, <value>** Response: OK Purpose: Set the audio and DTMF volumes for the specified <profile>. • Query: **AT!AVDTMFVOLDB?<profile>, <generator>, <level>** Response: <value> OK Purpose: Return the current volume (<value>) for the specified <profile><generator><level> combination. **Parameters:** <profile> (Audio profile number) • MSM6290, QSC6270: • 0=Handset • 1=Headset • 2=Car kit • 3=Speaker phone • 4=Auxiliary • 5=TTY • 6=Auxiliary external PCM (128 kHz clock) • 7=Primary external PCM (2 MHz clock) • MDM6200: • 7=Primary external PCM (2 MHz clock) <generator> (Audio type) • 0=Voice <level> (Volume level) • 0=Level 0 • 1=Level 1 • 2=Level 2 • 3=Level 3 • 4=Level 4 • 5=Level 5 • 6=Level 6 • 7=Level 7 <value> (Volume for the specified <level>) • Valid ranges: • 0x0000–0x04B0 (for positive gains) • 0xFFFF–0xEC78 (for negative gains) • The volume in dB is equal to the <value>/100.

Table 7-5: Voice command details (Continued)

Command	Description

!AVEC
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
 	
Note: For MDM8200A, see !AVEC on page 168.
	Set/report the echo cancellation setting
	Set the echo cancellation mode for the specified profile. The echo canceller detects and removes audio that echoes back from the far end of the voice conversation.
	Several settings are available:
	• Handset mode for mild echo with short delay
	• Headset mode for moderate echo with short delay
	• Car kit for loud echo with long delay
	• Speakerphone mode for loud echo with extreme acoustic distortion
	This setting is stored in non-volatile memory and persists across power cycles.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVEC=<profile>, <value>**
	Response: OK
	Purpose: Set the echo cancellation mode for the specified <profile>
	• Query: **AT!AVEC?<profile>**
	Response: <value>
	OK
	Purpose: Return the echo cancellation mode (<value>) for the specified <profile>
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<value> (Echo cancellation mode)
	• 0=Echo cancellation off
	• 1=Handset echo cancellation mode (ESEC—Ear Seal Echo Cancellation)
	• 2=Headset echo cancellation mode
	• 3=Car kit echo cancellation mode (AEC—Acoustic Echo Cancellation)
	• 4=Echo cancellation speaker

Table 7-5: Voice command details (Continued)

Command	Description

!AVEXTPCMCFG
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Configure external PCM interface
	Configure the external PCM interface by specifying the clock speed, the format, and enabling/disabling padding.
	These settings are stored in non-volatile memory and persist across power cycles.
	Note: MDM6200 supports only PCM <clock> speed 0 (2.048 MHz).
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect immediately if the current profile uses the external PCM interface.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVEXTPCMCFG=<clock>, <format>, <padding>**
	Response: OK
	Purpose: Configure the external PCM interface.
	• Query: **AT!AVEXTPCMCFG?**
	Response: <clock> <format> <padding>
	OK
	Purpose: Display current PCM interface configuration settings.
	Parameters:
	<clock> (PCM clock speed)
	• 0=2.048 MHz (short sync)
	• 1=128 kHz (long sync)
	<format> (PCM format type)
	• 0=8-bit μ-law
	• 1=8-bit a-law
	• 2=16-bit linear
	<padding> (Enable/disable padding)
	• 0=Disable
	• 1=Enable
	Note: Padding is typically disabled (padding bits are used to control the volume level for some external codecs).

Table 7-5: Voice command details (Continued)

Command	Description

!AVEXTPCMSTOPCLKOFF
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Prevent/allow external PCM interface clock from turning off
	Prevent (or allow) the external PCM interface clock from being turned off if the current audio profile uses the external PCM interface.
	Note: The external PCM interface for the current audio profile must be enabled before using this command.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately.
	Usage:
	• Execution: **AT!AVEXTPCMSTOPCLKOFF=<value>**
	Response: OK
	Purpose: Enable or disable (<value>) the ability to turn off the external PCM interface clock
	• Query: **AT!AVEXTPCMSTOPCLKOFF?**
	Response: <value>
	OK
	Purpose: Return the current status (<value>) of this option.
	Parameters:
	<value> (Ability to prevent PCM clock from being turned off)
	• 0=Disable
	• 1=Enable
	  • At startup, if audio profile 0 (default) uses the external PCM interface, the modem enables the clock.
	  • At startup, if audio profile 0 (default) does not use the external PCM interface, the user must switch to a different profile that does use the external PCM interface to enable the clock.
	Note: If the user switches from a profile that uses the external PCM interface to one that does not, the PCM clock is lost.
	Note: Depending on the external CODEC configuration, OEMs using this command might prevent the audio from being muted, so the device could be more prone to noise from the RF subsystem.

Table 7-5: Voice command details (Continued)

Command	Description
!AVINBANDRANGE Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K2_0_7_17ap) • QSC6270 (min f/w rev: S2.0)	**Specify Progress Descriptor value range for in-band signaling** During call establishment, several OTA messages may include a Progress Indicator information element that indicates whether the network uses in-band DTMF signaling. 3GPP TS24.008 (section 5.5.1 and section 10.5.4.21) indicates the valid range of Progress Descriptor values. This command can be used to specify one of two possible ranges to accommodate differing interpretations of the specification. This setting takes immediate effect (does not require a restart), and persists across power cycles. **Usage:** • Execution: **AT!AVINBANDRANGE=<range>** Response: OK *or* ERROR Purpose: Specify the range of possible Progress Descriptor values. • Query: **AT!AVINBANDRANGE?** Response: <range> OK Purpose: Return the range of possible Progress Descriptor values. **Parameters:** <range> (Progress Descriptor value range) • 0=Default (1,2,3,6,..,20) • 1=Alternate (1,2,3,6,..,0x20)

Table 7-5: Voice command details (Continued)

Command	Description

!AVMICGAIN
Supporting chipsets (voice-enabled devices only):
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set/report microphone gain
	Set the microphone gain for the specified audio profile. This setting provides a set of twenty-six gain levels from -6 dB to +49.5 dB (steps of 1.5 dB). The gain is applied to the analog audio input prior to its conversion into the digital domain.
	This setting is stored in non-volatile memory and persists across power cycles.
	Note: This command has no effect when PCM audio is being used.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVMICGAIN=<profile>, <value>**
	Response: OK
	Purpose: Set the microphone gain for the specified <profile>.
	• Query: **AT!AVMICGAIN?<profile>**
	Response: <value>
	OK
	Purpose: Return the microphone gain (<value>) for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• 0=Handset
	• 1=Headset
	• 2=Car kit
	• 3=Speaker phone
	• 4=Auxiliary
	• 5=TTY
	<value> (Gain value)
	• The value is entered/returned in hexadecimal format:
	  • QSC6270
	  • Valid values: 0x04 (0 dB), 0x14 (24 dB)
	  • Other chipsets
	  • Valid range: 0x00 (-6 dB) to 0x25 (49.5 dB) in 1.5 dB steps
	Examples:
	• QSC6270—To set the microphone gain for the car kit to 24 dB: AT!AVMICGAIN=2,14
	• MSM6290—To set the microphone gain for the speakerphone to 4.5 dB: AT!AVMICGAIN=3,7

Table 7-5: Voice command details (Continued)

Command	Description

!AVNS
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
	Enable/disable noise suppression
	Set the noise suppression mode for a specified profile. The noise suppressor reduces or eliminates continuous background noise, providing a clearer Rx audio signal.
	This setting is stored in non-volatile memory and persists across power cycles.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVNS=<profile>, <value>**
	Response: OK
	Purpose: Set the noise suppression mode for the specified <profile>.
	• Query: **AT!AVNS?<profile>**
	Response: <value>
	OK
	Purpose: Return the noise suppression mode (<value>) for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• MSM6290:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<value> (Turn noise suppression mode on/off)
	• 0=Noise suppression mode off
	• 1=Noise suppression mode on

Table 7-5: Voice command details (Continued)

Command	Description

!AVRXAGC
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set/report Rx AVC/AGC configuration
	Set the Rx AVC/AGC (Automatic Volume Control/Automatic Gain Control) configuration for the specified profile. The Rx AGC compensates for variations in audio gains from the land line side, while the Rx AVC tracks the ambient audio noise on the mobile side and compensates accordingly. Both controls allow for a constant audio level in the Rx direction.
	The setting is stored in non-volatile memory and persists across power cycles.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVRXAGC=<profile>, <value>**
	Response: OK
	Purpose: Set the Rx AVC/AGC configuration for the specified <profile>.
	• Query: **AT!AVRXAGC?<profile>**
	Response: <value>
	OK
	Purpose: Return the current Rx AVC/AGC configuration (<value>) for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<value> (AVC/AGC configuration)
	• 0=AGC off, AVC off
	• 1=AGC on, AVC on
	• 2=AGC on, AVC off
	• MDM6200/QSC6270:
	  • 3=AGC off, AVC on
	  • 4=AGC on, RVE on
	  • 5=AGC off, RVE on

Table 7-5: Voice command details (Continued)

Command	Description

!AVRXPCMFLTR
Supporting chipsets (voice-enabled devices only):
• MDM6200 (min f/w rev: P1_0_0_4)
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set/report the Rx PCM filter tap
	The modem has a 7-tap PCM (Pulse Code Modulation) FIR (Finite Impulse Response) filter. Use this command to set the Rx PCM filter tap for the specified profile.
	(This filter will be overridden if the PCM IIR (Infinite Impulse Response) filter is enabled—see !AVRXPCMIIRFLTR on page 148.)
	Note: This command is only useful when embedding the modem in a handset.
	Mobile phones, PDAs or other handheld transmitters and receivers that incorporate a GSM module are required to comply with the GSM 11.10 3GPP TS51.010 or 3GPP TS26.132 standard, or with national standards or government regulations. To conform to the relevant standard you may need to tune certain audio characteristics. This command allows you to tune the receive PCM filter to alter audio characteristics.
	Settings are stored in non-volatile memory and persist across power cycles.
	The process of tuning the receive audio characteristics generally involves these steps:
	1. Turn off the PCM filter (set <value> parameter to 0x0000 on Tap 6).
	2. Use test equipment to obtain a frequency response curve with the passing mask.
	3. Identify the frequency bands that need correction to bring the overall response within the bounds specified in the test case.
	4. Use a filter design tool to determine the filter coefficients, convert to signed Q14 format, and enter the appropriate tap settings using the **!AVRXPCMFLTR** command.
	5. Repeat the process until the specifications are met.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVRXPCMFLTR=<profile>, <tap>, <value>**
	Response: OK
	Purpose: Set the Rx PCM filter tap for the specified <profile>.
	• Query: **AT!AVRXPCMFLTR?<profile>, <tap>**
	Response: <value>
	OK
	Purpose: Return the filter setting (<value>) for the specified <profile>.
	(Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description

!AVRXPCMFLTR (continued)	**Set/report the Rx PCM IIR filter parameters (continued)**
	Parameters:
	<profile> (Audio profile number)
	• Valid values:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	<tap> (Filter tap in use)
	• 0=Tap 0
	• 1=Tap 1
	• 2=Tap 2
	• 3=Tap 3
	• 4=Tap 4
	• 5=Tap 5
	• 6=Tap 6
	<value> (Rx PCM filter tap value)
	• Valid range: 0x0000-0xFFFF
	• Calculated using the formula:
	<value> = ROUND (filter coefficient x 2^{14})
	• 0x0000 on Tap6 causes the PCM filter to be bypassed

Table 7-5: Voice command details (Continued)

Command	Description
!AVRXPCMIIRFLTR Supporting chipsets (voice-enabled devices only): • MDM6200 (min f/w rev: P1_0_0_4) • MSM6290 (min f/w rev: K2_0_7_51ap) • QSC6270: (min f/w rev: S2.0) *Note: For MDM8200A, see !AVRXPCMIIRFLTR on page 170.*	**Set/report the Rx PCM IIR filter parameters** The modem has a multi-stage PCM (Pulse Code Modulation) IIR (Infinite Impulse Response) filter. Use this command to set parameters for each stage of the specified profile. (Enabling this filter disables the PCM FIR (Finite Impulse Response) filter—see !AVRXPCMFLTR on page 146.) *Note: This command is only useful when embedding the modem in a handset.* Mobile phones, PDAs or other handheld transmitters and receivers that incorporate a GSM module are required to comply with the GSM 11.10 3GPP TS51.010 or 3GPP TS26.132 standard, or with national standards or government regulations. To conform to the relevant standard you may need to tune certain audio characteristics. This command lets you tune the receive PCM filter to alter audio characteristics. Settings are stored in non-volatile memory and persist across power cycles. To enable the PCM IIR filter (and override the PCM FIR filter): 1. Set the number of stages > 0. To disable the PCM IIR filter: 1. Set the number of states = 0. To tune the receive or transmit audio characteristics, follow a procedure similar to the following: 1. Disable the PCM IIR and FIR filters. 2. Use test equipment to obtain the initial Tx/Rx frequency response (uncorrected). 3. Make sure your filter design tool is configured to generate filter coefficients in signed Q30 format. 4. Identify frequency bands that must be corrected (boosted or attenuated) to bring the overall response within the bounds specified in the test case, considering the following points: • Fit the conformance mask (3GPP specification). • Minimize overall gain introduced by the PCM filter. • Modify the filter if necessary to improve quality (higher frequencies are more legible; lower frequencies will sound muffled). 5. Use **!AVRXPCMIIRFLTR** with an appropriate number of stages for the speech codec being used: • Wideband codecs (e.g. AMR-WB)—Five stages required • Narrowband codecs (e.g. AMR-NB)—Fewer stages required (for example, 3) 6. Repeat steps 4–5 until the specifications are met. **Device-specific details:** • All chipsets/firmware revisions: • The change takes effect immediately for the specified profile. (Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description
!AVRXPCMIIRFLTR (continued)	**Set/report the Rx PCM IIR filter parameters (continued)** **Usage:** • Execution: **AT!AVRXPCMIIRFLTR=<profile>, <param>, <stages>** *or* **AT!AVRXPCMIIRFLTR=<profile>, <param>, <a1>, <a2>, <b0>, <b1>, <b2>** Response: OK Purpose: Set the number of stages for the filter, or set the parameters for a specific stage. • Query: **AT!AVRXPCMIIRFLTR?<profile>, <param>** Response: <stages> *or* <a1>,<a2>,<b0>,<b1>,<b2> OK Purpose: Return the number of IIR filter stages, or the parameters for a specific stage. **Parameters:** <profile> (Audio profile number) • MSM6290, QSC6270: • 0=Handset • 1=Headset • 2=Car kit • 3=Speaker phone • 4=Auxiliary • 5=TTY • 6=Auxiliary external PCM (128 kHz clock) • 7=Primary external PCM (2 MHz clock) • MDM6200: • 7=Primary external PCM (2 MHz clock) <param> (Stage number): • 0=Configure the number of <stages> • 1–5 <stages> (Number of stages) • 0–5 <a1> (IIR filter design parameter a1) <a2> (IIR filter design parameter a2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF <b0> (IIR filter design parameter b0) <b1> (IIR filter design parameter b1) <b2> (IIR filter design parameter b2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF

Table 7-5: Voice command details (Continued)

Command	Description

!AVRXVOLDB
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290: (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
 	
Note: For MDM8200A, see !AVRXVOLDB on page 172.
	Set/report volume for each voice volume level in Rx direction
	Set the volume for each OTA (over the air) voice 'volume level' in the Rx direction.
	Volumes range from -50 dB to 12 dB and are applied to PCM voice packets after they have been decoded by the vocoder.
	The setting is stored in non-volatile memory and persists across power cycles.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!RXVOLDB=<profile>, <generator>, <level>, <value>**
	Response: OK
	Purpose: Set the Rx volume (in dB) for the specified <profile>.
	• Query: **AT!RXVOLDB?<profile>,<generator>,<level>**
	Response: <value>
	OK
	Purpose: Return the Rx volume (in dB) for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<generator> (Audio type)
	• 0=Voice
	<level> (Volume level)
	• 0=Level 0
	• 1=Level 1
	• 2=Level 2
	• 3=Level 3
	• 4=Level 4
	• 5=Level 5
	• 6=Level 6
	• 7=Level 7
	<value> (Rx volume (signed) in dB = value/100)
	• 0x0000–0x04B0, FFFF–EC78

Table 7-5: Voice command details (Continued)

Command	Description

!AVSETPROFILE
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_1_0ap)
• QSC6270 (min f/w rev: S2.0)
 	
Note: For MDM8200A, see !AVSETPROFILE on page 174.
	Configure and activate profile
	Select a profile with which to establish a circuit-switched call. (See "Profile activation" on page 130.) This command also enables / disables muting on the earpiece and microphone and sets the volume level.
	The profile you select remains active until the modem is reset or powered down and up again. Following a reset or power up, Profile 0 (the default profile) is active.
	If desired, you can run the command !AVRXVOLDB on page 150 to assign specific volume levels to each of the predefined volume levels, 1 through 7.
	Usage:
	• Execution: **AT!SETPROFILE=<profile>, <earmute>, <micmute>, <generator>, <volume>[, <cwtmute>]**
	Response: OK
	Purpose: Set the audio characteristics for the specified <profile>.
	• Query: **AT!AVSETPROFILE?<generator>**
	Response: <profile>, <earmute>, <micmute>, <volume>
	OK
	Purpose: Return the audio profile characteristics for the specified audio type (<generator>).
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<earmute> (Enable/disable earpiece muting)
	• 0=Unmuted
	• 1=Muted
	<micmute> (Enable/disable microphone muting)
	• 0=Unmuted
	• 1=Muted
	<generator> (Audio type)
	• 0=Voice
	(Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description
!AVSETPROFILE (continued)	**Activate a profile (continued)** <level> (Volume level) • 0=Level 0 • 1=Level 1 • 2=Level 2 • 3=Level 3 • 4=Level 4 • 5=Level 5 • 6=Level 6 • 7=Level 7 <cwtmute> (Enable/disable call waiting tone muting) • 0=Unmuted (default) • 1=Muted
!AVSN Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K1_0_2_0ap) • QSC6270 (min f/w rev: S2.0)	**Set/report audio revision number** Store and retrieve a revision number for your audio configuration. The modem does NOT associate this number with any settings and this command does not provide a means of restoring a particular configuration. The command only provides a means of storing and retrieving a number. **Usage:** • Execution: **AT!AVSN=<value>** Response: OK Purpose: Set the audio configuration revision number. • Query: **AT!AVSN?** Response: <value> OK Purpose: Return the current audio configuration revision number. **Parameters:** <value> (Revision number) • Valid range: 0x00000000–0xFFFFFFFF
!AVTONEPLAY Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K1_0_2_0ap) • QSC6270 (min f/w rev: S2.0)	**Play DTMF tone** Play a specified DTMF tone with the current active audio profile. This is intended for testing purposes (not for normal operation). **Usage:** • Execution: **AT!AVTONEPLAY=<generator>, <value> [, <duration>]** Response: OK Purpose: Play a specific tone. **Parameters:** <generator> (Audio type) • 0=Voice <value> (Tone value) • Valid range: 0x00–0x39 • For details, see “Tone values for AT!AVTONEPLAY command” on page 159.

Table 7-5: Voice command details (Continued)

Command	Description
!AVTONESETTINGS Supporting chipsets (voice-enabled devices only): • MDM6200 • MSM6290 (min f/w rev: K1_0_2_25ap, K1_1_1_16ap, K2_0_6_10ap) • QSC6270 (min f/w rev: S2.0)	**Enable/disable playing of locally-generated DTMF tones** Block locally-generated DTMF tones from playing, while leaving voice unaffected. The setting is stored in non-volatile memory and persists across power cycles. *Note: This does not block in-band DTMF tones.* **Usage:** • Execution: **AT!AVTONESETTINGS=<group>,<setting>[,<group>,<setting>][,...]** Response: <group 1> TONES: <Off | On> ... <group N> TONES: <Off | On> OK or ERROR Purpose: Enable or disable local playing of specific tone groups. • Query: **AT!AVTONESETTINGS?** Response: <group 1> TONES: <Off | On> ... <group N> TONES: <Off | On> OK Purpose: Return the current state for each supported tone group. • Query list: **AT!AVTONESETTINGS=?** Purpose: Return the command format, and supported <group> and <setting> values. **Parameters:** <group> (Tone group affected) • ASCII string • Supported value: “ALL” <setting> (Enable/disable specified tone group) • 0=Disable • 1=Enable

Table 7-5: Voice command details (Continued)

Command	Description

!AVTXAGC
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set Tx AGC
	Set the Tx AGC (Automatic Gain Control) for the specified profile. The Tx AGC compensates for variations in audio gains from the mobile side to allow for a constant audio level in the Tx direction.
	The setting is stored in non-volatile memory and persists across power cycles.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVTXAGC=<profile>, <value>**
	Response: OK
	Purpose: Set the Tx AGC for the specified profile.
	• Query: **AT!AVTXAGC?<profile>**
	Response: <value>
	OK
	Purpose: Return the current Tx AGC (<value>) for the specified profile.
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<value> (Enable/disable Tx AGC)
	• 0=AGC off
	• 1=AGC on

Table 7-5: Voice command details (Continued)

Command	Description

!AVTXPCMFLTR
Supporting chipsets (voice-enabled devices only):
• MDM6200 (min f/w rev: P1_0_0_4)
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
	Set/report Tx PCM filter tap
	The modem has a 7-tap PCM (Pulse Code Modulation) FIR (Finite Impulse Response) filter. Use this command to set the Tx PCM filter tap for the specified profile.
	See "!AVRXPCMFLTR" on page 146 for a description of using the filters.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect the next time the modem restarts.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVTXPCMFLTR=<profile>, <tap>, <value>**
	Response: OK
	Purpose: Set the Tx PCM filter tap for the specified <profile>.
	• Query: **AT!AVTXPCMFLTR?<profile>, <tap>**
	Response: <value>
	OK
	Purpose: Return the filter setting (<value>) for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• Valid values:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	<tap> (Filter tap in use)
	• Valid values:
	  • 0=Tap0
	  • 1=Tap1
	  • 2=Tap2
	  • 3=Tap3
	  • 4=Tap4
	  • 5=Tap5
	  • 6=Tap6
	<value> (Tx PCM filter tap value)
	• Calculated using the formula:
	<value>=ROUND (filter coefficient x 2^{14})
	• Valid range: 0x0000–0xFFFF
	• 0x0000 on Tap6 causes the PCM filter to be bypassed.

Table 7-5: Voice command details (Continued)

Command	Description

!AVTXPCMIIRFLTR
Supporting chipsets (voice-enabled devices only):
• MDM6200 (min f/w rev: P1_0_0_4)
• MSM6290 (min f/w rev: K2_0_7_51ap)
• QSC6270 (min f/w rev: S2.0)
 	
Note: For MDM8200A, see !AVTXPCMIIRFLTR on page 177.
	Set/report the Tx PCM IIR filter parameters
	The modem has a multi-stage PCM (Pulse Code Modulation) IIR (Infinite Impulse Response) filter. Use this command to set parameters for each stage of the specified profile.
	See "!AVRXPCMIIRFLTR" on page 148 for a description of using the filters.
	Device-specific details:
	• All chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVTXPCMIIRFLTR=<profile>, <param>, <stages>**
	or **AT!AVTXPCMIIRFLTR=<profile>, <param>, <a1>, <a2>, <b0>, <b1>, <b2>**
	Response: OK
	Purpose: Set the number of stages for the filter, or set the parameters for a specific stage.
	• Query: **AT!AVTXPCMIIRFLTR?<profile>, <param>**
	Response: <stages>
	or <a1>,<a2>,<b0>,<b1>,<b2>
	OK
	Purpose: Return the number of IIR filter stages, or the parameters for a specific stage.
	Parameters:
	<profile> (Account profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<param> (Stage number)
	• 0=Configure the number of <stages>
	• 1–5
	<stages> (Number of stages)
	• 0–5
	<a1> (IIR filter design parameter a1)
	• Signed hexadecimal
	• 0x00000000–0xFFFFFFFF
	<a2> (IIR filter design parameter a2)
	• Signed hexadecimal
	• 0x00000000–0xFFFFFFFF
	(Continued on next page)

Table 7-5: Voice command details (Continued)

Command	Description
!AVTXPCMIIRFLTR (continued)	**Set/report the Tx PCM IIR filter parameters (continued)** <b01> (IIR filter design parameter b0) • Signed hexadecimal • 0x00000000–0xFFFFFFFF <b1> (IIR filter design parameter b1) • Signed hexadecimal • 0x00000000–0xFFFFFFFF <b2> (IIR filter design parameter b2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF

Table 7-5: Voice command details (Continued)

Command	Description

!AVTXVOL
Supporting chipsets (voice-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K1_0_2_0ap)
• QSC6270 (min f/w rev: S2.0)
 	
Note: For MDM8200A, see !AVTXVOL on page 178.
	Set Tx volume
	Set the Tx volume gain for the specified audio profile. The Tx volume gain determines the gain, from -84 dB to +12 dB, for the voice that is transmitted over the air. This gain is applied to the PCM voice packets prior to feeding them into the vocoder, which encodes the PCM packets into a more efficient format for over the air transmission.
	This setting is stored in non-volatile memory and persists across power cycles.
	Device-specific details:
	• MSM6290 (Firmware revisions K2_0_7_8ap and lower):
	  • The change takes effect when a phone call is made or received.
	• All other chipsets/firmware revisions:
	  • The change takes effect immediately for the specified profile.
	Usage:
	• Execution: **AT!AVTXVOL=<profile>, <value>**
	Response: OK
	Purpose: Set the Tx volume gain for the specified <profile>.
	• Query: **AT!AVTXVOL?<profile>**
	Response: <value>
	OK
	Purpose: Display the Tx volume gain for the specified <profile>.
	Parameters:
	<profile> (Audio profile number)
	• MSM6290, QSC6270:
	  • 0=Handset
	  • 1=Headset
	  • 2=Car kit
	  • 3=Speaker phone
	  • 4=Auxiliary
	  • 5=TTY
	  • 6=Auxiliary external PCM (128 kHz clock)
	  • 7=Primary external PCM (2 MHz clock)
	• MDM6200:
	  • 7=Primary external PCM (2 MHz clock)
	<value> (Tx volume gain)
	• The value is entered/returned in hexadecimal format:
	  • Min=0x0000 (mute)
	  • Max=0xFFFF
	• To calculate the gain in dB, convert <value> to decimal (<decvalue>) and use the following formula:
	Gain = 20 $\log_{10}$ (<decvalue> / 16384)
	• Unity gain value is 0x4000

Table 7-6: Tone values for AT!AVTONEPLAY command

<Value> parameter setting	Tone	Description
0x00	SND_0	DTMF for 0 key
0x01	SND_1	DTMF for 1 key
0x02	SND_2	DTMF for 2 key
0x03	SND_3	DTMF for 3 key
0x04	SND_4	DTMF for 4 key
0x05	SND_5	DTMF for 5 key
0x06	SND_6	DTMF for 6 key
0x07	SND_7	DTMF for 7 key
0x08	SND_8	DTMF for 8 key
0x09	SND_9	DTMF for 9 key
0x0A	SND_A	DTMF for A key
0x0B	SND_B	DTMF for B key
0x0C	SND_C	DTMF for C key
0x0D	SND_D	DTMF for D key
0x0E	SND_POUND	DTMF for # key
0x0F	SND_STAR	DTMF for * key
0x10	SND_CTRL	Tone for a control key
0x11	SND_2ND	Tone for secondary function on a key
0x12	SND_WARN	Warning tone (e.g. overwriting user phone# slot)
0x13	SND_ERR	Tone to indicate an error
0x14	SND_TIME	Time marker tone
0x15	SND_RING_A	1st Ringer tone
0x16	SND_RING_B	2nd Ringer tone
0x17	SND_RING_C	3rd Ringer tone
0x18	SND_RING_D	4th Ringer tone
0x19	SND_RING_A4	440.0 Hz (Piano Notes)
0x1A	SND_RING_AS4	466.1 Hz
0x1B	SND_RING_B4	493.8 Hz
0x1C	SND_RING_C4	523.2 Hz
0x1D	SND_RING_CS4	554.3 Hz

Table 7-6: Tone values for AT!AVTONEPLAY command (Continued)

<Value> parameter setting	Tone	Description
0x1E	SND_RING_D4	587.3 Hz
0x1F	SND_RING_DS4	622.2 Hz
0x20	SND_RING_E4	659.2 Hz
0x21	SND_RING_F4	698.5 Hz
0x22	SND_RING_FS4	739.9 Hz
0x23	SND_RING_G4	784.0 Hz
0x24	SND_RING_GS4	830.6 Hz
0x25	SND_RING_A5	880.0 Hz
0x26	SND_RING_AS5	932.2 Hz
0x27	SND_RING_B5	987.7 Hz
0x28	SND_RING_C5	1046.5 Hz
0x29	SND_RING_CS5	1108.7 Hz
0x2A	SND_RING_D5	1174.6 Hz
0x2B	SND_RING_DS5	1244.3 Hz
0x2C	SND_RING_E5	1318.5 Hz
0x2D	SND_RING_F5	1397.0 Hz
0x2E	SND_RING_FS5	1479.9 Hz
0x2F	SND_RING_G5	1568.0 Hz
0x30	SND_RING_GS5	1661.2 Hz
0x31	SND_RING_A6	1760.0 Hz
0x32	SND_RING_AS6	1864.7 Hz
0x33	SND_RING_B6	1975.5 Hz
0x34	SND_RING_C6	2093.1 Hz
0x35	SND_RING_CS6	2217.4 Hz
0x36	SND_RING_D6	2349.3 Hz
0x37	SND_RING_DS6	2489.1 Hz
0x38	SND_RING_E6	2637.0 Hz
0x39	SND_RING_F6	2793.7 Hz
0x3A	SND_RING_FS6	2959.9 Hz
0x3B	SND_RING_G6	3135.9 Hz

Table 7-6: Tone values for AT!AVTONEPLAY command (Continued)

<Value> parameter setting	Tone	Description
0x3C	SND_RING_GS6	3322.4 Hz
0x3D	SND_RING_A7	3520.0 Hz
0x3E	SND_RBACK	Ring back (audible ring)
0x3F	SND_BUSY	Busy tone
0x40	SND_INTERCEPT_A	First tone of an intercept
0x41	SND_INTERCEPT_B	Second tone of an intercept
0x42	SND_REORDER_TONE	Reorder
0x43	SND_PWRUP	Power-up tone
0x44	SND_OFF_HOOK_TONE	Off-hook tone, IS-95 (CAI 7.7.5.5)
0x45	SND_CALL_WT_TONE	Call-waiting tone
0x46	SND_DIAL_TONE_TONE	Dial tone
0x47	SND_ANSWER_TONE	Answer tone
0x48	SND_HIGH_PITCH_A	1st High pitch for IS-54B alerting
0x49	SND_HIGH_PITCH_B	2nd High pitch for IS-54B alerting
0x4A	SND_MED_PITCH_A	1st Medium pitch for IS-54B alerting
0x4B	SND_MED_PITCH_B	2nd Medium pitch for IS-54B alerting
0x4C	SND_LOW_PITCH_A	1st Low pitch for IS-54B alerting
0x4D	SND_LOW_PITCH_B	2nd Low pitch for IS-54B alerting
0x4E	SND_TEST_ON	Test tone on
0x4F	SND_MSG_WAITING	Message Waiting Tone
0x50	SND_PIP_TONE_TONE	Used for Pip-Pip-Pip-Pip (Vocoder) Tone
0x51	SND_SPC_DT_INDIA	Used for India's Special Dial Tone
0x52	SND_SIGNAL_INDIA	Used in Various India Signalling Tones
0x53	SND_DT_TONE_INDIA	Used for India's Normal Dial Tone (and others)
0x54	SND_DT_TONE_BRAZIL	Used for Brazil's Dial Tone
0x55	SND_DT_DTACO_TONE	Used for DTACO's single tone (350 Hz,350 Hz)
0x56	SND_HFK_TONE1	These two tones are used for Voice Activation and Incoming Call Answer in phone VR-HFK
0x57	SND_HFK_TONE2	

8: I2S Audio Commands

Introduction

Note: This chapter applies to modems supporting I2S audio. For modems supporting analog and PCM digital audio interfaces, see Voice Commands on page 127.

Some Sierra Wireless Mini Card modems support I2S audio. Host devices may use either the modem or an MCU to control an I2S audio codec chip in one of two modes—test mode (using a WM8904 audio codec) or commercial mode (using a different audio codec).

The commands in this chapter are used to select the mode, configure the codec, set and read codec registers, and set the audio sampling rate.

For more detailed information about I2S audio support, refer to the modem's product specification document. For assistance with testing the I2S audio interface using Sierra Wireless' I2S Audio Board, refer to *Sierra Wireless I2S Audio Board User Guide*.

Command summary

The table below lists the commands described in this chapter:

Table 8-1: I2S audio commands

Command	Description	Page
!AVCODECBRG	Configure codec registers to make call	165
!AVCODECCFG	Configure codec register	165
!AVCODECRED	Read a codec register	166
!AVCODECRST	Configure codec registers for reset	166
!AVCUSTI2CCFG	Configure external codec I2C details	167
!AVEC	Set/report the echo cancellation setting	168
!AVMODESET	Select codec mode	168
!AVREGVALWID	Set codec register bit width	169
!AVRXDECGAIN	Set/report voice decoder gain	169
!AVRXPCMIIRFLTR	Set/report the Rx PCM IIR filter parameters	170
!AVRXSPKRGAIN	Set/report audio profile speaker gain	172
!AVRXVOLDB	Set/report Rx voice volume	172

Table 8-1: I2S audio commands (Continued)

Command	Description	Page
!AVSETDEV	Set audio profile Rx and Tx mute states	173
!AVSETPROFILE	Configure and activate profile	174
!AVSETSAMP	Set I2S sample rate	175
!AVSETVOL	Set audio profile default volume level	175
!AVTXENCGAIN	Set/report audio profile encoder gain	176
!AVTXMICGAIN	Set/report audio profile microphone gain	176
!AVTXPCMIIRFLTR	Set/report the Tx PCM IIR filter parameters	177
!AVTXVOL	Set Tx volume	178
!AVUSEMCU	Select codec controller	178

Command reference

Table 8-2: I2S audio command details

Command	Description
!AVCODECBRG Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Configure codec registers to make call** (This command is used only when the modem is controlling a codec in commercial mode—[AT!AVUSEMCU=0 and AT!AVMODESET=1].) This command is used to populate a table with up to 511 register values that the device uses to bring up the audio codec when making a call. *Note: AT!AVCODECBRG=0xFFFF,0xFFFF is not allowed.* **Usage:** • Execution: **AT!AVCODECBRG=<register address>, <value>** Response: OK Purpose: Set the specified register to the <value>. • Query list: **AT!AVCODECBRG=?** Purpose: Return the command format and supported <values>. **Parameters:** <register address> (Address of register) • Valid range: 0–0xFFFF • 0xFFFF is used to set the delay <value> (Value to store in the specified register) • Valid range: 0–0xFFFF • 0xFFFF is not valid when <register_address> = 0xFFFF. (This combination identifies the end of the configuration table.)
!AVCODECCFG Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Configure codec register** (This command is used only when the modem is controlling a codec in commercial mode—[AT!AVUSEMCU=0 and AT!AVMODESET=1].) This command is used to store a value directly into a codec register. **Usage:** • Execution: **AT!AVCODECCFG=<register address>, <value>** Response: OK Purpose: Set the specified register to the <value>. • Query list: **AT!AVCODECCFG=?** Purpose: Return the command format and supported <values>. **Parameters:** <register address> (Address of register) • Valid range: 0–0xFFFF • 0xFFFF is used to set the delay <value> (Value to store in the specified register) • Valid range: 0–0xFFFF

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVCODECRED Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Read a codec register** (This command is used only when the modem is controlling a codec in commercial mode—[AT!AVUSEMCU=0 and AT!AVMODESET=1].) This command is used to read the value stored in a codec register. **Usage:** • Execution: **AT!AVCODECRED=<register address>** Response: !AVCODECRED: <value> OK Purpose: Return the <value> stored in the specified register. • Query list: **AT!AVCODECRED=?** Purpose: Return the command format and supported <values>. **Parameters:** <register address> (Address of register) • Valid range: 0–0xFFFF <value> (Value that is stored in the specified register) • Valid range: 0–0xFFFF
!AVCODECRST Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Configure codec registers for reset** (This command is used only when the modem is controlling a codec in commercial mode—[AT!AVUSEMCU=0 and AT!AVMODESET=1].) This command is used to populate a table with up to four (4) register values that the modem will access when AT!RESET is issued. These values are used to stop the codec before the modem resets. **Important:** *If this command is not used to populate the register reset table, the modem will receive I2S signals when it resets and will go into an abnormal (undefined) state.* *Note: AT!AVCODECRST=0xFFFF,0xFFFF is not allowed.* **Usage:** • Execution: **AT!AVCODECRST=<register address>,<value>** Response: OK Purpose: Set the specified register to the <value>. • Query list: **AT!AVCODECRST=?** Purpose: Return the command format and supported <values>. **Parameters:** <register address> (Address of register) • Valid range: 0–0xFFFF • 0xFFFF is used to set the delay <value> (Value to store in the specified register) • Valid range: 0–0xFFFF • 0xFFFF is not valid when <register_address> = 0xFFFF. (This combination identifies the end of the configuration table.)

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVCUSTI2CCFG Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Configure external codec I2C details** (This command is used only when the modem is controlling a codec in commercial mode—[AT!AVUSEMCU=0 and AT!AVMODESET=1].) Configure the external codec's I2C interface. The change takes effect immediately. This setting is stored in non-volatile memory and persists across power cycles. **Usage:** • Execution: **AT!AVCUSTI2CCFG=<bus frequency>, <slave ID>, <address type>, <device type>, <read option>** Response: OK Purpose: Set I2C interface options. • Query: **AT!AVCUSTI2CCFG?** Response: !AVCUSTI2CCFG: <bus frequency>, <slave ID>, <address_type>, <device_type>, <read_option> OK Purpose: Return the current I2C interface configuration. • Query list: **AT!AVCUSTI2CCFG=?** Purpose: Return the command format and supported <values>. **Parameters:** <bus frequency> (I2C bus frequency) • 100=100 KHz • 400=400 KHz • 3400=3400 KHz <slave ID> (I2C slave ID) • 0x00–0x7F (0–127) • In the execution command, the value may be entered in hexadecimal or decimal format. • In the query response, the value is shown in hexadecimal format. <address type> (Slave address type) • 0=7-bit slave address • 1=10-bit slave address <device type> (I2C address device type) • 1=I2C memory address device • 2=I2C register address device <read option> (Master-generated signals required on bus before read) • 1=Start signal required before read • 2=Stop and start signals required before read

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVEC Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5) *Note: This command is not password protected.* *Note: For MDM6200, MSM6290, and QSC6270, see !AVEC on page 139.*	**Set/report the echo cancellation setting** Set the echo cancellation mode for the specified voice profile. The echo canceller detects and removes audio that echoes back from the far end of the voice conversation. This setting is stored in non-volatile memory and persists across power cycles. The change takes effect immediately if used on the active profile, or after the modem restarts if used on a different profile. **Usage:** • Execution: **AT!AVEC=<profile>, <value>** Response: OK Purpose: Set the echo cancellation mode for the specified <profile> • Query: **AT!AVEC?<profile>** Response: <value> OK Purpose: Return the echo cancellation mode (<value>) for the specified <profile> • Query list: **AT!AVEC=?** Purpose: Return a list of supported modes (<value>). **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <value> (Echo cancellation (EC) mode) • 0=EC Default • 1=EC I2S mode • 2=EC handset mode
!AVMODESET Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Select codec mode** Select the I2S audio codec mode—test mode (WM8904 only), or commercial mode (any codec). **Usage:** • Execution: **AT!AVMODESET=<value>** Response: OK Purpose: Select the I2S audio codec mode. • Query: **AT!AVMODESET?** Response: !AVMODESET: <value> OK Purpose: Return the current I2S audio codec mode. • Query list: **AT!AVMODESET=?** Purpose: Return the command format and supported <values>. **Parameters:** <value> (Codec mode) • 0=WM8904 test mode (Default value) • 1=Commercial mode (Any codec)

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVREGVALWID Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Set codec register bit width** (This command is used only when the modem is controlling a codec in commercial mode—[AT!AVUSEMCU=0 and AT!AVMODESET=1].) Set the register bit width of the customer's I2S audio codec. The change takes effect immediately but is non-persistent (the bit width reverts to default when the device power cycles). **Usage:** • Execution: **AT!AVREGVALWID=<value>** Response: OK Purpose: Set the audio codec's register bit width. • Query: **AT!AVREGVALWID?** Response: !AVREGVALWID: <value> OK Purpose: Return the register bit width. • Query list: **AT!AVREGVALWID=?** Purpose: Return the command format, and supported <values>. **Parameters:** <value> (Bit width of codec register) • 8=8 bits • 16=16 bits (Default)

!AVRXDECGAIN
Supporting chipsets (voice-enabled devices only):
• MDM8200A (min f/w rev: T3_0_1_5)
	Set/report voice decoder gain
	Set/report the voice decoder gain for a specific audio profile.
	Usage:
	• Execution: **AT!AVRXDECGAIN=<profile>,<value>**
	Response: OK
	Purpose: Set the decoder gain for the specified profile.
	• Query: **AT!AVRXDECGAIN?<profile>**
	Response: <value>
	OK
	Purpose: Return the decoder gain for the specified profile.
	Parameters:
	<profile> (Voice profiles)
	• Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.)
	<value> (QCT Rx decoder gain)
	• The value is entered/returned in hexadecimal format.
	  • Valid range: 0x2000 (0 dB)–0xFFFF (18.06 dB)
	  • Typical value: 0x2000

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVRXPCMIIRFLTR Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5) *Note: For MDM6200, MSM6290, and QSC6270, see !AVRXPCMIIRFLTR on page 148.*	**Set/report the Rx PCM IIR filter parameters** The modem has a multi-stage PCM (Pulse Code Modulation) IIR (Infinite Impulse Response) filter. Use this command to set parameters for each stage of the specified profile. (Enabling this filter disables the PCM FIR (Finite Impulse Response) filter—see !AVRXPCMFLTR on page 147.) *Note: This command is only useful when embedding the modem in a handset.* Mobile phones, PDAs or other handheld transmitters and receivers that incorporate a GSM module are required to comply with the GSM 11.10 3GPP TS51.010 or 3GPP TS26.132 standard, or with national standards or government regulations. To conform to the relevant standard, and to improve voice quality, you may need to tune certain audio characteristics. This command lets you tune the receive PCM filter to alter audio characteristics. Settings take effect immediately for the specified profile, are stored in non-volatile memory, and persist across power cycles. To enable the PCM IIR filter (and override the PCM FIR filter): **1.** Set the number of stages > 0. To disable the PCM IIR filter: **1.** Set the number of stages = 0. To tune the receive or transmit audio characteristics, follow a procedure similar to the following: **1.** Disable the PCM IIR and FIR filters. **2.** Use test equipment to obtain the initial Tx/Rx frequency response (uncorrected). **3.** Make sure your filter design tool is configured to generate filter coefficients in signed Q30 format. **4.** Identify frequency bands that must be corrected (boosted or attenuated) to bring the overall response within the bounds specified in the test case, considering the following points: • Fit the conformance mask (3GPP specification). • Minimize overall gain introduced by the PCM filter. • Modify the filter if necessary to improve quality (higher frequencies are more legible; lower frequencies will sound muffled). **5.** Use **!AVRXPCMIIRFLTR** with an appropriate number of stages for the speech codec being used: • Wideband codecs (e.g. AMR-WB)—Five stages required • Narrowband codecs (e.g. AMR-NB)—Fewer stages required (for example, 3) **6.** Repeat steps 4–5 until the specifications are met. (Continued on next page)

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVRXPCMIIRFLTR (continued)	**Set/report the Rx PCM IIR filter parameters (continued)** **Usage:** • Execution (Enable/disable IIR): **AT!AVRXPCMIIRFLTR=<profile>, 0, <IIR_stages>** Response: OK Purpose: Set the number of stages for the filter. • Execution (Set stage coefficients): **AT!AVRXPCMIIRFLTR=<profile>, <stage>, <b0>, <b1>, <b2>, <a1>, <a2>** Response: OK Purpose: Set the parameters for a specific stage. • Query (IIR state): **AT!AVRXPCMIIRFLTR?<profile>, 0** Response: <enabled>,<IIR_stages> OK Purpose: Indicate whether IIR is enabled, and (if enabled) the number of IIR filter stages. • Query: **AT!AVRXPCMIIRFLTR?<profile>, <stage>** Response: <b0>,<b1>,<b2>,<a1>,<a2> OK *or* ERROR *(if <stage> is greater than the number of stages defined for the profile)* Purpose: Return the filter coefficients for a specific stage. (Note: The coefficients are returned even if IIR is currently disabled.) **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <IIR_stages> (Enable IIR and set number of stages, or disable IIR): • 0=Disable IIR • 1–10=Enable IIR with this number of stages <stage> (IIR stage) • 1–10=Stage to be configured <a1> (IIR filter coefficient a1) <a2> (IIR filter coefficient a2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF <b0> (IIR filter coefficient b0) <b1> (IIR filter coefficient b1) <b2> (IIR filter coefficient b2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVRXSPKRGAIN Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5)	**Set/report audio profile speaker gain** Set/report the speaker gain for a specific audio profile. **Usage:** • Execution: **AT!AVRXSPKRGAIN=<profile>,<value>** Response: OK Purpose: Set the specified audio profile's speaker gain. • Query: **AT!AVRXSPKRGAIN?<profile>** Response: <value> OK Purpose: Return the specified audio profile's speaker gain. **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <value> (QCT Rx speaker gain) • The value is entered/returned in hexadecimal format. • Valid range: 0x2000 (0 dB)–0xFFFF (18.06 dB)
!AVRXVOLDB Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5) *Note: This command is not password protected.* *Note: For MDM6200, MSM6290, and QSC6270, see !AVRXVOLDB on page 150.*	**Set/report Rx voice volume** Set/report the actual volumes associated with Rx volume 'levels'. **Usage:** • Execution: **AT!AVRXVOLDB=<level>,<value>** Response: OK Purpose: Set the actual volume (<value>) to associate with the specified volume level. • Query: **AT!AVRXVOLDB?<level>** Response: <value> OK Purpose: Return the actual volume associated with the specified volume level. **Parameters:** <level> (Voice volume level) • Valid range: 0–5 (levels 0–5) <value> (Actual Rx volume in dB) • The value is entered/returned in hexadecimal format. • Valid range: 0x0000–0xFFFF

Table 8-2: I2S audio command details (Continued)

Command	Description

!AVSETDEV
Supporting chipsets (voice-enabled devices only):
• MDM8200A (min f/w rev: T3_0_1_5)
 	
Note: This command is not password protected.
	Set audio profile Rx and Tx mute states
	Mute or unmute a profile's Rx and Tx audio paths (earphone/microphone).
	Usage:
	• Execution: **AT!AVSETDEV=<profile>,<earmute>,<micmute>**
	Response: OK
	Purpose: Set the audio codec's register bit width.
	• Query: **AT!AVSETDEV?**
	Response: <profile>,<earmute>,<micmute>
	OK
	Purpose: Return the register bit width.
	• Query list: **AT!AVSETDEV=?**
	Purpose: Return list of available profiles.
	Parameters:
	<profile> (Voice profiles)
	• Valid values:
	  • 0=Default mode
	  • 1=I2S mode
	  • 2=Handset mode
	<earmute> (Enable/disable earpiece muting)
	• Valid values:
	  • 0=Unmuted
	  • 1=Muted
	<micmute> (Enable/disable microphone muting)
	• Valid values:
	  • 0=Unmuted
	  • 1=Muted

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVSETPROFILE Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5) *Note: For MDM6200, MSM6290, and QSC6270, see !AVSET-PROFILE on page 151.*	**Configure and activate profile** Select a profile with which to establish a circuit-switched call. This command also enables / disables muting on the earpiece and microphone and sets the volume level. This setting is stored in non-volatile memory and persists across power cycles. If desired, you can run the command !AVRXVOLDB on page 172 to assign specific volume levels to each of the predefined volume levels, 0 through 5. **Usage:** • Execution: **AT!SETPROFILE=<profile>, <earmute>, <micmute>, <volume>** Response: OK Purpose: Set the audio characteristics for the specified <profile>. • Query: **AT!AVSETPROFILE?** Response: <earmute>, <micmute>, <volume> OK Purpose: Return the audio profile characteristics of the current profile. **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <earmute> (Enable/disable earpiece muting) • 0=Unmuted • 1=Muted <micmute> (Enable/disable microphone muting) • 0=Unmuted • 1=Muted <volume> (Voice volume level) • 0=Level 0 • 1=Level 1 • 2=Level 2 • 3=Level 3 • 4=Level 4 • 5=Level 5

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVSETSAMP Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T2.0.2.1) *Note: This command is not password protected.*	**Set I2S sample rate** Set the I2S sample rate. **Usage:** • Execution: **AT!AVSETSAMP=<value>** Response: OK Purpose: Set the I2S sample rate. • Query: **AT!AVSETSAMP?** Response: !AVSETSAMP: <value> OK Purpose: Return the current I2S sample rate. • Query list: **AT!AVSETSAMP=?** Purpose: Return the command format and supported <values>. **Parameters:** <value> (I2S sample rate) • 8=8 KHz • 16=16 KHz (Default value) • 48=48 KHz
!AVSETVOL Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: R0.2.15.0) *Note: This command is not password protected.*	**Set audio profile default volume level** Set a voice audio profile's default volume level. This setting is stored in non-volatile memory and persists across power cycles. Use this command with !AVSETDEV on page 173 to set the default values for each voice audio profile. **Usage:** • Execution: **AT!AVSETVOL=<profile>,<value>** Response: OK Purpose: Set the audio profile's default volume level. • Query: **AT!AVSETVOL?** Response: <value> OK Purpose: Return the default volume level for the current profile. • Query list: **AT!AVSETVOL=?** Purpose: Return the supported volume levels. **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <value> (Voice volume level) • Valid range: 0–5 (levels 0–5)

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVTXENCGAIN Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5)	**Set/report audio profile encoder gain** Set/report the encoder gain for a specific audio profile. **Usage:** • Execution: **AT!AVTXENCGAIN=<profile>,<value>** Response: OK Purpose: Set the specified audio profile's encoder gain. • Query: **AT!AVTXENCGAIN?<profile>** Response: <value> OK Purpose: Return the specified audio profile's encoder gain. **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <value> (QCT Tx encoder gain) • The value is entered/returned in hexadecimal format. • Valid range: 0x2000 (0 dB)–0xFFFF (18.06 dB)
!AVTXMICGAIN Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5)	**Set/report audio profile microphone gain** Set/report the microphone gain for a specific audio profile. **Usage:** • Execution: **AT!AVTXMICGAIN=<profile>,<value>** Response: OK Purpose: Set the specified audio profile's microphone gain. • Query: **AT!AVTXMICGAIN?<profile>** Response: <value> OK Purpose: Return the specified audio profile's microphone gain. **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <value> (QCT Tx microphone gain) • The value is entered/returned in hexadecimal format. • Valid range: 0x2000 (0 dB)–0xFFFF (18.06 dB)

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVTXPCMIIRFLTR Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5) *Note: For MDM6200, MSM6290, and QSC6270, see !AVTXPCMI-IRFLTR on page 156.*	**Set/report the Tx PCM IIR filter parameters** The modem has a multi-stage PCM (Pulse Code Modulation) IIR (Infinite Impulse Response) filter. Use this command to set parameters for each stage of the specified profile. See "!AVRXPCMIIRFLTR" on page 170 for a description of using the filters. The change takes effect immediately for the specified profile. **Usage:** • Execution (Enable/disable IIR): **AT!AVTXPCMIIRFLTR=<profile>, 0, <IIR_stages>** Response: OK Purpose: Set the number of stages for the filter. • Execution (Set stage coefficients): **AT!AVTXPCMIIRFLTR=<profile>, <stage>, <b0>, <b1>, <b2>, <a1>, <a2>** Response: OK Purpose: Set the parameters for a specific stage. • Query (IIR state): **AT!AVTXPCMIIRFLTR?<profile>, 0** Response: <enabled>,<IIR_stages> OK Purpose: Indicate whether IIR is enabled, and (if enabled) the number of IIR filter stages. • Query: **AT!AVTXPCMIIRFLTR?<profile>, <stage>** Response: <b0>,<b1>,<b2>,<a1>,<a2> OK *or* ERROR *(if <stage> is greater than the number of stages defined for the profile)* Purpose: Return the filter coefficients for a specific stage. (Note: The coefficients are returned even if IIR is currently disabled.) **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <IIR_stages> (Enable IIR and set number of stages, or disable IIR): • 0=Disable IIR • 1–10=Enable IIR with this number of stages <stage> (IIR stage) • 1–10=Stage to be configured <a1> (IIR filter coefficient a1) <a2> (IIR filter coefficient a2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF <b0> (IIR filter coefficient b0) <b1> (IIR filter coefficient b1) <b2> (IIR filter coefficient b2) • Signed hexadecimal • 0x00000000–0xFFFFFFFF

Table 8-2: I2S audio command details (Continued)

Command	Description
!AVTXVOL Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T3_0_1_5) *Note: For MDM6200, MSM6290, and QSC6270, see !AVTXVOL on page 158.*	**Set Tx volume** Set the Tx volume gain for the specified audio profile. The Tx volume gain determines the gain, from -84 dB to +12 dB, for the voice that is transmitted over the air. This gain is applied to the PCM voice packets prior to feeding them into the vocoder, which encodes the PCM packets into a more efficient format for over the air transmission. This setting is stored in non-volatile memory and persists across power cycles. • The change takes effect when a phone call is made or received. • The change takes effect immediately for the specified profile. **Usage:** • Execution: **AT!AVTXVOL=<profile>, <value>** Response: OK Purpose: Set the Tx volume gain for the specified <profile>. • Query: **AT!AVTXVOL?<profile>** Response: <value> OK Purpose: Display the Tx volume gain for the specified <profile>. **Parameters:** <profile> (Voice profiles) • Valid range: 0–2 (See !AVSETDEV on page 173 for available profiles.) <value> (Actual Rx volume in dB) • The value is entered/returned in hexadecimal format. • Valid range: 0x0000–0xFFFF
!AVUSEMCU Supporting chipsets (voice-enabled devices only): • MDM8200A (min f/w rev: T2.0.2.1) *Note: This command is not password protected.*	**Select codec controller** Select the device that will control the codec—the modem or the MCU. The change takes effect immediately. This setting is stored in non-volatile memory and persists across power cycles. **Usage:** • Execution: **AT!AVUSEMCU=<value>** Response: OK Purpose: Select the device that controls the codec. • Query: **AT!AVUSEMCU?** Response: !AVUSEMCU: <value> OK Purpose: Identify the device that is controlling the codec. • Query list: **AT!AVUSEMCU=?** Purpose: Return the command format and supported <values>. **Parameters:** <value> (Codec controller) • 0=Modem controls codec • 1=MCU controls codec

9: GPS Commands

Introduction

This chapter describes commands used to access GPS functionality in supporting modules.

When using these commands, the following considerations apply:

- GPS is typically enabled by default; however, it may be disabled by default for some SKUs. If so, enable GPS using AT!CUSTOM="GPSENABLE"
- If supported by the modem, gpsOneXTRA is enabled (over the NDIS interface) by default when GPS is enabled, and it generates data traffic.

Command summary

The table below lists the commands described in this chapter.

Table 9-1: GPS commands

Command	Description	Page
!GPS3RDPARTYXFER	Initiate Location Service (LCS) third party transfer location request	181
!GPSAUTOSTART	Configure GPS auto-start features	182
!GPSCLRASSIST	Clear specific GPS assistance data	183
!GPSCOLDSTART	Clear all GPS assistance data	184
!GPSEND	End an active session	184
!GPSFIX	Initiate GPS position fix	185
!GPSIPADDR	Set/report IP address to use over TCP/IP	186
!GPSKEEPWARM	Configure Keep Warm functionality	186
!GPSLBSAPN	Set GPS LBS APNs	187
!GPSLBSSETTINGS	Set default GPS location fix options	188
!GPSLOC	Return last known location of the modem	189
!GPSMTLRSETTINGS	Set/report MT location request settings	190
!GPSNIQOSTIME	Set/report GPS QoS timeout period for network-initialized fixes	191
!GPSNMEACONFIG	Enable and set NMEA data output rate	191
!GPSNMEASENTENCE	Set/report NMEA sentence type	192

Table 9-1: GPS commands (Continued)

Command	Description	Page
!GPSPORTID	Set/report port ID to use over TCP/IP	193
!GPSPOSMODE	Configure support for GPS positioning modes	194
!GPSPROTOSEL	Control GPS protocol selection	195
!GPSSATINFO	Request satellite information	196
!GPSSTATUS	Request current status of a position fix session	197
!GPSSUPLPID	Set/report supplementary channel connection profile ID	198
!GPSSUPLURL	Set/report SUPL server URL	198
!GPSSUPLVER	Set/report SUPL server version	199
!GPSTRACK	Initiate local tracking (multiple fix) session	200
!GPSTRANSSEC	Control GPS transport security	201
!GPSXTRAAPN	Set GPS XTRA APNs	202
!GPSXTRADATAENABLE	Set/report GPS XTRA settings	203
!GPSXTRADATAURL	Set/report GPS XTRA data server URLs	204
!GPSXTRAINITDNLD	Initiate gpsOneXTRA data download and inject operation	204
!GPSXTRASTATUS	Return current status of gpsOneXTRA	205
!GPSXTRATIME	Inject GPS or UTC time into gpsOneXTRA system	206
!GPSXTRATIMEENABLE	Set/report GPS XTRA time settings	207
!GPSXTRATIMEURL	Set/report GPS XTRA SNTP server URLs	208

Command reference

Table 9-2: GPS command details

Command	Description
!GPS3RDPARTYXFER Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Initiate Location Service (LCS) third party transfer location request** Initiate a location fix, directing the location information to a third party at a specified ISDN address. **Usage:** • Execution: **AT!GPS3RDPARTYXFER=<External Client ID - ISDN address>[, <MLC number - ISDN address>]** Response: OK *or* ERROR Purpose: Initiate a location fix and send the information to the specified address. • Query List: **AT!GPS3RDPARTYXFER=?** Purpose: Return the expected command format. **Parameters:** <External Client ID> (ISDN address to which fix information is to be sent) • Format: <Extension flag>,<Nature of address>,<Numbering plan>, <Number string> <MLC number> (ISDN address of the Mobile Location Centre) • Format: <Extension flag>,<Nature of address>,<Numbering plan>, <Number string> <Extension flag> • 0=Extension • 1=No extension <Nature of address> (See AddressString definition in *3GPP TS 29.002*) • Valid range: 0–7 <Numbering plan> (See AddressString definition in *3GPP TS 29.002*) • Valid range: 0–15 <Number string> (ISDN address—See AddressString definition in *3GPP TS 29.002*) • Maximum length: 20 characters • Valid characters: ‘0’–’9’, ‘*’, ‘#’, ‘a’, ‘b’, ‘c’

Table 9-2: GPS command details (Continued)

Command	Description

!GPSAUTOSTART
Supporting chipsets (GPS-enabled devices only):
• MDM6200
• MDM8200A
• MDM8220
• MDM9200
• MDM9600
• MSM6290 (min f/w rev: K1_0_2_1ap, L1_0_2_1ap)
• QSC6270: (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Configure GPS auto-start features
	Configure the GPS auto-start features. Any changes take effect the next time the modem is reset.
	
	Note: If auto-start is enabled, another GPS session cannot be started.
	
	Usage:
	• Execution: **AT!GPSAUTOSTART=<enable>[, <fixtype>, <maxtime>, <maxdist>, <fixrate>]**
	Response: OK
	or ERROR
	Purpose: Assign start values for various GPS settings
	• Query: **AT!GPSAUTOSTART?**
	Response: !GPSAUTOSTART
	enable: <enable>
	fixtype: <fixtype>
	maxtime: <maxtime> seconds
	maxdist: <maxdist> meters
	fixrate: <fixrate> seconds
	OK
	Purpose: Display the current values for auto-start features
	• Query List: **AT!GPSAUTOSTART=?**
	Purpose: Return the expected command format.
	
	Parameters:
	<enable> (Enable/disable the feature)
	• 0=Disabled
	• 1=Enabled (GPS tracking session starts automatically when modem is reset)
	
	<fixtype> (Type of fix to establish)
	• 1=Standalone (not supported by a mobile station)
	• 2=MS-based only
	• 3=MS-assisted only
	
	<maxtime> (Maximum time to wait for a position fix)
	• Valid range: 0–255—Number of seconds to wait
	
	<maxdist> (Requested accuracy of fix)
	• Entered in decimal format
	• Valid range:
	  • 0–4294967279 meters
	  • 4294967280=No preference
	
	<fixrate> (Time to wait between fixes)
	• Valid range: 1–65535 seconds

Table 9-2: GPS command details (Continued)

Command	Description
!GPSCLRASSIST Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Clear specific GPS assistance data** Clear one or more types of assistance data from the modem. This forces a cold start for GPS acquisition the next time a session starts. The command is only available when there is no active GPS session—the GPS receiver is off and no position fix is being calculated. This command is equivalent to **!GPSCOLDSTART** when all four parameters are set to '1'. **Usage:** • Execution: **AT!GPSCLRASSIST=<eph>, <alm>, <pos>, <time>, <iono>** Response: OK *or* Command ignored OK Purpose: Clear each assistance data type that is flagged as '1'. • Query List: **AT!GPSCLRASSIST=?** Purpose: Return the expected command format and supported values. **Parameters:** <eph> (Ephemeris assistance data) • 0=Ignore (Do not clear the ephemeris assistance data) • 1=Clear this assistance data type Note: MDM9200 (min fw rev: SWI9200X_3.0 Release 2, SWI9200X_3.5-Beta3)—Clears GPS, GLONASS, and SBAS ephemeris assistance data. <alm> (Almanac assistance data) • 0=Ignore (Do not clear the almanac assistance data) • 1=Clear this assistance data type Note: MDM9200 (min fw rev: SWI9200X_3.0 Release 2, SWI9200X_3.5-Beta3)—Clears GPS, GLONASS, and SBAS almanac assistance data. <pos> (Position assistance data) • 0=Ignore (Do not clear the position assistance data) • 1=Clear this assistance data type <time> (Time reference) • 0=Ignore (Do not clear the time reference) • 1=Clear the time reference <iono> (Ionosphere assistance data) • 0=Ignore (Do not clear the ionosphere assistance data) • 1=Clear this assistance data type

Table 9-2: GPS command details (Continued)

Command	Description
!GPSCOLDSTART Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Clear all GPS assistance data** Clear all GPS assistance details from the modem and put the modem into a coldstart state. Data cleared includes Almanac, Ephemeris, Previous Position, Ionosphere, and GPS time. This forces a cold start for GPS acquisition the next time a session starts. The command is only available when there is no active GPS session—the GPS receiver is off and no position fix is being calculated. This command is equivalent to **!GPSCLRASSIST** when all four of that command's parameters are set to '1'. **Usage:** • Execution: **AT!GPSCOLDSTART** Response: OK Purpose: Clear the modem's GPS details **Parameters:** None
!GPSEND Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**End an active session** End an active position fix session. **Usage:** • Execution: **AT!GPSEND=<sessType>** Response: ERRCODE = <value> OK *or* OK Purpose: End the current session. **Parameters:** <sessType> (Type of session to end) • 0=Position fix session <value> (Error code returned when command fails for any reason) • See Table 9-3 on page 208 for a list of possible error codes.

Table 9-2: GPS command details (Continued)

Command	Description

!GPSFIX
Supporting chipsets (GPS-enabled devices only):
• MDM6200
• MDM8200a
• MDM8220
• MDM9200
• MDM9600
• MSM6290
• QSC6270 (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Initiate GPS position fix
	Initiate a GPS position fix.
	Usage:
	• Execution: **AT!GPSFIX=<fixType>, <maxTime>, <maxDist>**
	Response: Fix initiated
	OK
	or ERROR CODE = <value>
	OK
	Purpose: Initiate a time-limited position fix with a specified accuracy.
	• Query List: **AT!GPSFIX=?**
	Purpose: Return supported <fixType>, <maxTime>, and <maxDist> values.
	Parameters:
	<fixType> (Type of fix to establish)
	• 1=Standalone (not supported by a mobile station)
	• 2=MS-based only
	• 3=MS-assisted only
	<maxTime> (Maximum time to wait for a position fix)
	• Valid range: 0–255 seconds
	<maxDist> (Requested accuracy of fix)
	• Entered in decimal format
	• Valid range:
	  • 0–4294967279 meters
	  • 4294967280=No preference
	<value> (Error code returned when command fails for any reason)
	• See Table 9-3 on page 208 for a list of possible error codes.
	Example:
	AT!GPSFIX=1, 15, 10 requests a standalone position fix to 10 meters accuracy. The request will fail (timeout) if the modem cannot determine a position fix within 15 seconds.
	Related commands:
	• **!GPSSTATUS** (page 197)—Use this command while the tracking session is in progress.
	• **!GPSLOC** (page 189)—Use this command after the session completes to obtain the result.

Table 9-2: GPS command details (Continued)

Command	Description
!GPSIPADDR Supporting chipsets (GPS-enabled devices only): • MDM8200A • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report IP address to use over TCP/IP** *Note: Deprecated. Use !GPSSUPLURL instead.* Sets or report the IP address of the SUPL server to use when using TCP/IP as the transport mechanism for SUPL. **Usage:** • Execution: **AT!GPSIPADDR=<ipaddr>** Response: OK *or* ERROR Purpose: Queue the request to set the IP address. • Query: **AT!GPSIPADDR?** Response: <ipaddr> OK Purpose: Return the address currently being used. **Parameters:** <ipaddr> (IP address to use) • Standard IP address format. For example, AT!GPSIPADDR=63.162.134.132 **Related commands** • !GPSPORTID (p.193)—Set/query the port ID to use over TCP/IP
!GPSKEEPWARM Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Configure Keep Warm functionality** Set, clear, or report the modem's 'keep warm' functionality. This functionality downloads GPS assistance data from the GPS server. **Usage:** • Execution: **AT!GPSKEEPWARM=<enableFlag>** Response: OK Purpose: Enable/disable the keep warm functionality. • Query: **AT!GPSKEEPWARM?** Response: KeepWarm Enabled: <enableFlag> Warm Status: <warmStatus> Purpose: Display the current status (<enableFlag>) of the keep warm functionality and indicate if GPS is in 'warm' state (<warmStatus>). • Query List: **AT!GPSKEEPWARM=?** Purpose: Display valid <enableFlag> options. **Parameters:** <enableFlag> (Enable/disable keep warm functionality) • 0=Disable • 1=Enable <warmStatus> (GPS is in Warm state) • 0=No • 1=Yes **Example:** AT!GPSKEEPWARM? returns: KeepWarm Enabled: 1 Warm Status: 1 In this example, KeepWarm is enabled, and GPS is in Warm state.

Table 9-2: GPS command details (Continued)

Command	Description
!GPSLBSAPN Supporting chipsets (GPS-enabled devices only): • MDM9200 (min f/w rev: SWI9200X_03.00.10.00; SWI9200X_03.05.04.01)	**Set GPS LBS APNs** Set the GPS LBS APNs to be used for various RATs (Radio Access Technologies). **Usage:** • Execution (Add): **AT!GPSLBSAPN=<operation>,<ratmask>,<IPtype>,<APN>** Execution (Delete one): **AT!GPSLBSAPN=<operation>,<ratmask>** Execution (Delete all): **AT!GPSLBSAPN=<operation>** Response: OK *or* ERROR Purpose: Set the APN to be used for the specified <ratmask>, or delete the APN for a single <ratmask> or all RATs. • Query: **AT!GPSLBSAPN?** Response: <operation>, <ratmask>, <IPType>, <APN> <operation>, <ratmask>, <IPType>, <APN> ... OK *or* OK *(if no ID has been set)* Purpose: Display the APNs currently assigned for each RAT. • Query List: **AT!GPSLBSAPN=?** Purpose: Display valid parameter options. **Parameters:** <operation> (Add or delete APNs) • 1=Add an APN for a specific <ratmask> and <IPtype>. Note: All paramters are required. *Note: To change an APN that has been set for a RAT, you must first delete the current APN, then add the new APN.* • 2=Delete the APN for a specific <ratmask> Note: Only <ratmask> is required. • 3=Delete all APNs Note: No other parameters are required. <ratmask> (Radio access technology) • Valid values (values shown are in hexadecimal format): • 01=CDMA • 02=HDR • 04=GSM • 08=WCDMA • 10=LTE (Continued on next page)

Table 9-2: GPS command details (Continued)

Command	Description

!GPSLBSAPN (continued)	**Set GPS LBS APNs (continued)**
	<IPtype> (Internet Protocol version)
	• Character string, entered without quotation marks
	• Valid values:
	  • IPV4
	  • IPV6
	  • IPV4V6
	<APN> (Access Point Name)
	• Character string, entered with quotation marks
	• Examples: “mycompany.mnc987.mcc123.gprs”, “ourinternet”

!GPSLBSSETTINGS
Supporting chipsets (GPS-enabled devices only):
• MDM6200
• MSM6290 (min f/w rev: K2_0_7_40ap/K2_2_0_7ap/L2_2_1_7ap)
• QSC6270 (min f/w rev: S2.0)
	Set default GPS location fix options
	Set default GPS LBS (location based service) values.
	
	Usage:
	• Execution: **AT!GPSLBSSETTINGS=<fixType>,<maxTime>,<maxDist>,<fixCount>,<fixRate>**
	Response: OK
	Purpose: Set specific LBS default values.
	• Query: **AT!GPSLBSSETTINGS?**
	Response: FIX TYPE: <fixType>
	MAX TIME: <maxTime>
	MAX DIST: <maxDist>
	FIX COUNT: <fixCount>
	FIX RATE: <fixRate>
	Purpose: Display the current GPS location fix default values.
	• Query List: **AT!GPSLBSSETTINGS=?**
	Purpose: Display valid parameter options.
	Parameters:
	<fixType> (Type of fix to establish)
	• 1=Standalone (not supported by a mobile station)
	• 2=MS-based only
	• 3=MS-assisted only
	<maxTime> (Maximum time to wait for a position fix)
	• Valid range: 0–255 seconds
	<maxDist> (Requested accuracy of fix)
	• Entered in decimal format
	• Valid range:
	  • 0–4294967279 meters
	  • 4294967280=No preference
	<fixCount> (Number of position fixes to take)
	• Valid range: 1–1000 (1000—Take a continuous series of position fixes)
	<fixrate> (Time to wait between fixes)
	• Valid range: 1–65535 seconds

Table 9-2: GPS command details (Continued)

Command	Description
!GPSLOC Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Return last known location of the modem** Return the details obtained during the most recent position location session, if available. **Usage:** • Query: **AT!GPSLOC?** Response: Unknown *(No information is available)* OK *or* Not Available *(No information is available)* OK *or* Lat: <latitude> Lon: <longitude> Time: <time> LocUncAngle: <luAngle> LocUncA: <luA> LocUncP: <luP> HEPE: <hepe> <fixType> Altitude: <altitude> LocUncVe: <luV> Heading: <heading> VelHoriz: <vH> VelVert: <vV> OK *(Altitude and heading only appear if data was collected as part of the most recent fix.)* Purpose: Return last position location details. **Parameters:** <latitude> (Latitude at last position fix) • Example: “49 Deg 10 Min 21.49 Sec N (0x008BDE6C)” <longitude> (Longitude at last position fix) • Example: “123 Deg 4 Min 14.76 Sec W (0xFEA1EE9A)” <time> (Time at which last position fix was taken) • Example: “2009 01 30 4 20:27:18 (GPS)” <luAngle> (Location uncertainty angle of returned position) • Example: “11.2 deg” <luA> (Standard deviation of axis along <luAngle>) • Example: “6.0 m” <luP> (Standard deviation of axis perpendicular to <luAngle>) • Example: “6.0 m” <hepe> (Horizontal Estimated Positional Error) • Example: “8.485 m” <fixType> (2D or 3D fix) • Example: “2D Fix” or “3D Fix” <altitude> (Altitude in meters at which last position fix was taken) • Only present if <fixType> is 3D • Example: “-1 m” <luV> (Vertical uncertainty in meters) • Only present if <fixType> is 3D • Example: “3.0 m” (Continued on next page)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSLOC (continued)	**Return last known location of the modem (continued)** <heading> (Direction of MS) • Example: “0.0 deg” <vH> (Horizontal velocity) • Example: “0.0 m/s” <vV> (Vertical velocity) • Example: “0.0 m/s”
!GPSMTLRSETTINGS Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report MT location request settings** Set or report the current MT (mobile-terminated) Location Request settings. *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSMTLRSETTINGS=<notifyResp>** Response: OK *or* ERROR Purpose: Indicate how MT location request will be handled. • Query: **AT!GPSMTLRSETTINGS?** Response: Notification Response Setting: <notifResp> OK Purpose: Return the current <notifResp> setting. • Query List: **AT!GPSMTLRSETTINGS=?** Purpose: Return valid <notifResp> values. **Parameters:** <notifResp> (Notification response setting) • 0=Default setting as defined in *3GPP specification 29.002*, ‘NotificationToMSUser’ enumeration. • 1=Accept all MT location requests. • 2=Reject all MT location requests. • 3=Verify all—User will be asked to accept or reject every MT location request.

Table 9-2: GPS command details (Continued)

Command	Description
!GPSNIQOSTIME Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report GPS QoS timeout period for network-initialized fixes** Set or report the current GPS QoS timeout period for network-initiated fixes. *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSNIQOSTIME=<timeout>** Response: OK *or* ERROR Purpose: Set the new timeout period. • Query: **AT!GPSNIQOSTIME?** Response: QoS time: <timeout> OK Purpose: Return the current <timeout> period. **Parameters:** <timeout> (GPS QoS timeout period) • Timeout period (in seconds)
!GPSNMEACONFIG Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Enable and set NMEA data output rate** Enable or disable NMEA data output, and set the output rate for use with !GPSTRACK. **Usage:** • Execution: **AT!GPSNMEACONFIG=<enable>[,<outputRate>]** Response: OK *or* ERROR Purpose: Enable or disable NMEA output and set rate. • Query: **AT!GPSNMEACONFIG?** Response: Enabled: 0 Output Rate: <outputRate> *or* Enabled Output Rate: <outputRate> OK Purpose: Return the current <timeout> period. • Query List: **AT!GPSNMEACONFIG=?** Purpose: Return valid parameter values. **Parameters:** <enable> (Enable/disable NMEA data output) • 0=Disable. (Note: <outputRate> is ignored) • 1=Enable. (Note: <outputRate> is required) <outputRate> (NMEA data output rate—time between outputs) • Valid range: 1–255 seconds

Table 9-2: GPS command details (Continued)

Command	Description
!GPSNMEASENTENCE Supporting chipsets (GPS-enabled devices only): • MDM9200 (min f/w rev: SWI9200X_3.0-Release 2 or SWI9200X_3.5-Beta 3)	**Set/report NMEA sentence type** Set or report the current GPS NMEA sentence types. **Usage:** • Execution: **AT!GPSNMEASENTENCE=<nmea type>** Response: OK *or* ERROR Purpose: Enable or disable NMEA sentence types. • Query: **AT!GPSNMEASENTENCE?** Response: !GPSNMEASENTENCE: <nmea type> OK Purpose: Indicate the currently enabled GPS NMEA sentence types. • Query List: **AT!GPSNMEASENTENCE=?** Purpose: Return valid parameter values. **Parameters:** <nmea type> (NMEA sentence types) • 2-byte hex format mask (Note: In the execution format, do not include '0x' before the mask value) • Each bit: 0=Disabled; 1=Enabled • Bit 0: GGA (Fix information) • Bit 1: RMC (Recommended minimum data for GPS) • Bit 2: GSV (Detailed satellite data) • Bit 3: GSA (Overall satellite data) • Bit 4: VTG (Vector track and speed over the ground) • Bit 5: PQXFI (Proprietary Qualcomm eXtended Fix Information) • Bit 6: GLGSV (GLONASS GSV) • Bit 7: GNGSA (GLONASS GSA) • Bit 8: GNGNS (Time, position, and fixed related data for GLONASS receiver) • Bit 13: PSTIS (GPS session start indication)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSPORTID Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report port ID to use over TCP/IP** Set or report the port ID of the SUPL server to use when using TCP/IP as the transport mechanism for SUPL. The command can also be used when the FQDN is auto-generated from the IMSI. **Usage:** • Execution: **AT!GPSPORTID=<portid>** Response: OK *or* ERROR Purpose: Queue the request to set the port ID. • Query: **AT!GPSPORTID?** Response: <portid> OK Purpose: Return the port ID currently being used **Parameters:** <port ID> (Port ID to use over TCP/IP) • Valid range: 0–65535 **Related commands** • !GPSSUPLURL (p.198)—Set/return SUPL server URL used for TCP/IP • !GPSIPADDR (p.186)—Set/query the IP address to use over TCP/IP

Table 9-2: GPS command details (Continued)

Command	Description
!GPSPOSMODE Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Configure support for GPS positioning modes** Enable or disable support for several GPS positioning modes. *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSPOSMODE=<mask>** Response: OK *or* ERROR Purpose: Use a single byte hexadecimal format mask to indicate which GPS positioning modes are to be supported. • Query: **AT!GPSPOSMODE?** Response: MASK: <mask> OK Purpose: Return a <mask> value indicating which GPS positioning modes are currently supported. • Query List: **AT!GPSPOSMODE=?** Purpose: Return supported <mask> values. **Parameters:** <mask> (Bitmap value representing supported GPS positioning modes) • 1-byte hex format mask (do not include '0x' before the mask value) • 'On' bits identify modes that are supported • Bit 0: Standalone • Bit 1: UP MS-based • Bit 2: UP MS-assisted • Bit 3: CP MS-based (2G) • Bit 4: CP MS-assisted (2G) • Bit 5: CP UE-based (3G) • Bit 6: CP UE-assisted (3G) • Bit 7: Unused **Example:** AT!GPSPOSMODE=2a enables support for Bit 5 (CP UE-based), Bit 3 (CP MS-based), and Bit 1 (UP MS-based)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSPROTOSEL Supporting chipsets (GPS-enabled devices only): • MSM6290 • (Deprecated) QSC6270 (min f/w rev: S2.0)	**Control GPS protocol selection** Set or report the current GPS protocol selection method for User Plane GPS. *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSPROTOSEL=<protocol>** Response: OK *or* ERROR Purpose: Indicate the protocol selection method to use. • Query: **AT!GPSPROTOSEL?** Response: Protocol selection: <protocol> OK Purpose: Return the current <protocol> selection method. • Query List: **AT!GPSPROTOSEL=?** Purpose: Return supported <protocol> values. **Parameters:** <protocol> (Protocol selection method) • 0=None • 1=PreSUPL • 2=X1 • 3=SUPL (Secure User Plane Location)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSSATINFO Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Request satellite information** Return the following information for up to twelve satellites in view (including those used in the latest position fix): satellite vehicle number (SV), elevation (ELEV), azimuth (AZI), and signal to noise ratio (SNR). The information returned is valid regardless of the current fix mode or whether the PDE or the modem performs the fix calculations. **Usage:** • Query: **AT!GPSSATINFO?** Response: NO SAT INFO OK *or* Satellites in view: <numSats> * SV: <SV 1> ELEV:<ELEV 1> AZI:<AZI 1> SNR:<SNR 1> ... * SV: <SV n> ELEV:<ELEV n> AZI:<AZI n> SNR:<SNR n> OK Purpose: Return the number of satellites in view (including those used in the latest position fix) and details for each satellite (or return an error message). *Note: An asterisk (*) at the beginning of a line indicates the satellite was used in the fix location calculation.* **Parameters:** <numSats> (Number of satellites in view) • Valid range: 1–12 <SV n> (Satellite vehicle number for the nth satellite in the list) • Valid range: 1–32 <ELEV n> (Satellite elevation relative to modem location, in degrees) • Valid range: 0–90 <AZI n> (Satellite azimuth relative to modem location, in degrees) • Valid range: 0–360 <SNR n> (Signal to noise ratio, in dB) • Valid range: 0–99

Table 9-2: GPS command details (Continued)

Command	Description
!GPSSTATUS Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Request current status of a position fix session** Return the current status of a position fix session. **Usage:** • Query: **AT!GPSSTATUS?** Response: <year> <month> <day> <day of week> <time of day> Last Fix Status = <status> <year> <month> <day> <day of week> <time of day> Fix Session Status = <status> Purpose: Return timestamps and status of a position fix session. **Timestamp parameters:** <year> • Example: “2007” <month> • 01–12 (Jan–Dec) <day> • 01–31 <day of week> • 0–6 (0=Monday) <time of day> • 24-hour clock format • Example: “13:25:48” **Status parameters:** <status> (Session status) • “NONE”: No session of this type has occurred since the modem powered up. • The timestamp is the current time. • “ACTIVE”: A session of this type is currently active. • The timestamp is the time when the session entered this state. • “SUCCESS”: The most recent session of this type succeeded. • The timestamp is the time when the previous session completed successfully. • “FAIL”: The most recent session of this type failed. • The timestamp is the time when the previous session failed. • An error code is displayed with the “FAIL” string. See Table 9-3 on page 208 for a list of error codes. **Example:** AT!GPSSTATUS? returns: 2007 01 06 6 00:25:01 Last Fix Status = SUCCESS 2007 01 06 6 00:25:02 Fix Session Status = ACTIVE

Table 9-2: GPS command details (Continued)

Command	Description
!GPSSUPLPID Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200 (min f/w rev: M3.0 Alpha3) • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Set/report supplementary channel connection profile ID** Set or return the connection profile ID that GPS uses for its supplementary channel. **Usage:** • Execution: **AT!GPSSUPLPID=<pid>** Response: OK or ERROR Purpose: Set the PID used for the GPS supplementary channel. • Query: **AT!GPSSUPLPID?** Response: !GPSSUPLPID: <pid> OK Purpose: Return the PID used for the GPS supplementary channel. • Query List: **AT!GPSSUPLPID=?** Purpose: Return a list of valid <pid> values. **Parameters:** <pid> (Profile ID) • 0—Use connected or default profile ID • 1–16—Profile ID
!GPSSUPLURL Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report SUPL server URL** Set or return the URL of the SUPL server to be used when TCP/IP is used as the transport mechanism for location processing. Use !GPSPORTID to set the port ID. *Note: !GPSSUPLURL should be used in place of !GPSIPADDR.* **Usage:** • Execution: **AT!GPSSUPLURL="<suplURL>"** Response: OK or ERROR Purpose: Identify the SUPL server URL. • Query: **AT!GPSSUPLURL?** Response: <suplURL> OK Purpose: Return the SUPL server's URL.. • Query List: **AT!GPSSUPLURL=?** Purpose: Return the execution command format. **Parameters:** <suplURL> (SUPL server URL) • Must be a fully qualified domain name (FQDN) or address • Examples: "supl.url.net", "123.123.123.123" • The <suplURL> is not checked for correctness—if the string is invalid, the modem will not be able to perform MS-assisted GPS fixes. **Examples:** AT!GPSSUPLURL="supl.url.net" AT!GPSSUPLURL="123.123.123.123"

Table 9-2: GPS command details (Continued)

Command	Description
!GPSSUPLVER Supporting chipsets (GPS-enabled devices only): • MDM9200	**Set/report SUPL server version** Set or return the version of the SUPL server. **Usage:** • Execution: **AT!GPSSUPLURL=<supl ver>"** Response: OK or ERROR Purpose: Identify the SUPL server version. • Query: **AT!GPSSUPLVER?** Response: <supl ver> OK Purpose: Return the SUPL server's version. • Query List: **AT!GPSSUPLVER=?** Purpose: Return the execution command format. **Parameters:** <supl ver> (SUPL server version) • 1—SUPL version 1 • 2—SUPL version 2

Table 9-2: GPS command details (Continued)

Command	Description

!GPSTRACK
Supporting chipsets (GPS-enabled devices only):
• MDM6200
• MDM8200A
• MDM8220
• MDM9200
• MDM9600
• MSM6290
• QSC6270 (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Initiate local tracking (multiple fix) session
	Initiate a local tracking session comprising a specific number of position fixes taken at regular time intervals.
	Usage:
	• Execution: **AT!GPSTRACK = <fixType>, <maxTime>, <maxDist>, <fixCount>, <fixRate>**
	Response: Fix initiated
	OK
	or ERROR CODE = <value>
	OK
	Purpose: Initiate a series of time-limited position fixes.
	• Query List: **AT!GPSTRACK=?**
	Purpose: Return supported <fixType>, <maxTime>, <maxDist>, <fixCount>, and <fixRate> values.
	Parameters:
	<fixType> (Type of fix to establish)
	• 1=Standalone (not supported by a mobile station)
	• 2=MS-based only
	• 3=MS-assisted only
	<maxTime> (Maximum time to wait for satellite information)
	• Valid range: 0–255 seconds
	<maxDist> (Requested accuracy of fix)
	• Entered in decimal format
	• Valid range:
	  • 0–4294967279 meters
	  • 4294967280=No preference
	<fixCount> (Number of position fixes requested)
	• Valid range: 1–1000 (1000—Take a continuous series of position fixes)
	<fixrate> (Amount of time to wait between fix attempts)
	• Valid range: 0–1799999 seconds
	Failure conditions:
	The request fails if the tracking session fails to initiate.
	If the request fails, the message ERROR CODE = <value> is returned. See Table 9-3 on page 208 for a list of error codes.
	
	*Note: The 'time to first fix' may require more time than subsequent fixes, if almanac, ephemeris, or location data needs to be updated. (Almanac data is valid for 3–4 days, ephemeris for 30–120 minutes, and location data for 4 minutes). To avoid a timeout error (time spent > <maxtime>), your application could precede the **!GPSTRACK** call with a single position fix (**AGPSFIX**) with a greater <maxTime> value.*
	
	(Continued on next page)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSTRACK (continued)	**Initiate local tracking (multiple fix) session (continued)** **Example:** AT!GPSTRACK=1, 15, 10, 20, 60 requests a series of 20 standalone position fixes to 10 meters accuracy—fixes are taken every 60 seconds. One of the following responses will be received: • "OK" if the request is successful, or • "ERROR CODE = <value>" if the request fails for any reason. See Table 9-3 on page 208 for a list of error codes. **Related commands:** • **!GPSSTATUS**—Use this command while the tracking session is in progress. • **!GPSLOC**—Use this command after the session completes to obtain the result.

!GPSTRANSSEC
Supporting chipsets (GPS-enabled devices only):
• MDM6200
• MDM8200A
• MDM8220
• MDM9200
• MDM9600
• MSM6290
• QSC6270 (min f/w rev: S2.0)
	Control GPS transport security
	Enable or disable GPS transport security for SUPL GPS fixes.
	*Note: **!RESET** must be issued after this command is used.*
	Usage:
	• Execution: **AT!GPSTRANSSEC=<security>**
	Response: OK
	or ERROR
	Purpose: Indicate if transport security is used.
	• Query: **AT!GPSTRANSSEC?**
	Response: Transport security: <security>
	OK
	Purpose: Return the current <security> setting.
	• Query list: **AT!GPSTRANSSEC=?**
	Purpose: Display valid values for <security> parameter.
	Parameters:
	<security> (Transport security state)
	• MDM9200 (min f/w rev: SWI9200X_3.0 Release 2, SWI9200X_3.5 Beta 3)
	  • Bit 0: 0=Disabled (No security); 1=Enabled (Security)
	  • Bit 1: 0=SSL Version TLS 1.1; 1=SSL Version TLS 1.0
	  • Bit 2: 0=SHA256; 1=SHA1
	• All other chipsets (including earlier MDM9200 f/w revisions):
	  • 0=Disable (Secure TCP/IP is not used)
	  • 1=Enable (Secure TCP/IP is used)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSXTRAAPN Supporting chipsets (GPS-enabled devices only): • MDM9200 (min f/w rev: SWI9200X_03.00.10.00; SWI9200X_03.05.04.01)	**Set GPS XTRA APNs** Set the GPS XTRA APNs to be used for various RATs (Radio Access Technologies). **Usage:** • Execution (Add): **AT!GPSXTRAAPN=<operation>,<ratmask>,<IPtype>,<APN>** Execution (Delete one): **AT!GPSXTRAAPN=<operation>,<ratmask>** Execution (Delete all): **AT!GPSXTRAAPN=<operation>** Response: OK *or* ERROR Purpose: Set the APN to be used for the specified <ratmask>, or delete the APN for a single <ratmask> or all RATs. • Query: **AT!GPSXTRAAPN?** Response: <operation>, <ratmask>, <IPType>, <APN> <operation>, <ratmask>, <IPType>, <APN> ... OK *or* OK *(if no ID has been set)* Purpose: Display the APNs currently assigned for each RAT. • Query List: **AT!GPSXTRAAPN=?** Purpose: Display valid parameter options. **Parameters:** <operation> (Add or delete APNs) • 1=Add an APN for a specific <ratmask> and <IPtype> Note: All paramters are required. *Note: To change an APN that has been set for a RAT, you must first delete the current APN, then add the new APN.* • 2=Delete the APN for a specific <ratmask> Note: Only <ratmask> is required. • 3=Delete all APNs Note: No other parameters are required. <ratmask> (Radio access technology) • Valid values (hexadecimal format): • 01=CDMA • 02=HDR • 04=GSM • 08=WCDMA • 10=LTE <IPtype> (Internet Protocol version) • Character string, entered without quotation marks • Valid values: • IPV4 • IPV6 • IPV4V6 (Continued on next page)

Table 9-2: GPS command details (Continued)

Command	Description
!GPSXTRAAPN (continued)	**Set GPS XTRA APNs (continued)** <APN> (Access Point Name) • Character string, entered with quotation marks • Examples: “mycompany.mnc987.mcc123.gprs”, “ourinternet”
!GPSXTRADATAENABLE Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report GPS XTRA settings** Enable or disable GPS XTRA data and set or report GPS XTRA data configuration settings. *Note: These settings are persistent across power cycles.* *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSXTRADATAENABLE=<enable>[,<retries>,<retryInt>[,<dload>,<dloadInt>]]** Response: OK *or* ERROR Purpose: Enable or disable GPS XTRA data. You can only set the retry parameters if <enable> = 1, and you can only set the download parameters if the retry parameters are set. • Query: **AT!GPSXTRADATAENABLE?** Response: XTRA Data Enabled: <enable> XTRA Data Retry Number: <retries> XTRA Data Retry Interval: <retryInt> XTRA Data Autodownload Enabled: <dload> XTRA Data Autodownload Interval: <dloadInt> Purpose: Return the current GPS XTRA data settings. • Query List: **AT!GPSXTRADATAENABLE=?** Purpose: Return supported <enable>, <retries>, <retryInt>, <dload>, and <dloadInt> values. **Parameters:** <enable> (Enable or disable gpsOneXTRA functionality) • 0=Disable. To fully disable gpsOneXTRA, you must also call **!GPSXTRATIMEENABLE=0** to disable gpsOneXTRA time functionality. • 1=Enable <retries> (Number of download retries) • Valid range: 0–10 <retryInt> (Interval between download retries, in minutes) • Valid range: 1–120 <dload> (Enable or disable automatic downloads) • 0=Disable • 1=Enable <dloadInt> (Interval between automatic downloads, in hours) • Valid range: 1–168

Table 9-2: GPS command details (Continued)

Command	Description
!GPSXTRADATAURL Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report GPS XTRA data server URLs** Set or report the URLs of up to three GPS XTRA data servers. **Usage:** • Execution: **AT!GPSXTRADATAURL=<urlIndex>,<url>** Response: OK *or* ERROR Purpose: Set the URL used for the primary, secondary, or tertiary data server. • Query: **AT!GPSXTRADATAURL?** Response: XTRA Primary Server: <url1> XTRA Secondary Server: <url2> XTRA Tertiary Server: <url3> OK Purpose: Return the URLs of the primary, secondary, and tertiary data servers. **Parameters:** <urlIndex> (Server index) • 1=Primary server • 2=Secondary server • 3=Tertiary server <url> (Server URL) • URL string includes quotes • Example: “http://xtra1.gpsoneextra.net/xtra.bin” • URL must be complete, including the “http://” • Maximum string length: 128 characters
!GPSXTRAINITDNLD Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Initiate gpsOneXTRA data download and inject operation** Initiate a gpsOneXTRA data download and inject operation using the data server specified in the **!GPSXTRADATAURL** command. **Usage:** • Execution: **AT!GPSXTRAINITDNLD** Response: Xtra command sent successfully OK *or* Error code = <err> OK Purpose: Initiate the download and inject operation. If the command fails, it returns “Error code = <err>”. **Parameters:** <err> (Error code returned if command fails) • 3=Bad CRC for XTRA data file • 4=Old XTRA data file • 7=GPS subsystem busy • 8=GPS time reference entered is invalid • 9=Unknown error

Table 9-2: GPS command details (Continued)

Command	Description

!GPSXTRASTATUS
Supporting chipsets (GPS-enabled devices only):
• MDM6200
• MDM8200A
• MDM8220
• MDM9200
• MDM9600
• MSM6290
• QSC6270 (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Return current status of gpsOneXTRA
	Return the status of the most recent time and data injection operations.
	Usage:
	• Query: **AT!GPSXTRASTATUS?**
	Response: Xtra Time status = <timeStatus>
	Xtra Data status = <dataStatus>
	Validity Start = <timeStamp>
	Validity End = <timeStamp>
	OK
	Purpose: Return the status of the most recent time and data injection operations.
	Parameters:
	<timeStatus>
	• Returned string does not include quotes (they are used in this description for clarity).
	  • "Unknown": Default value if time injection operation has not been performed yet, or if operation was incomplete
	  • "Valid": GPS time injection succeeded
	  • "Invalid": GPS time injection failed
	
	<dataStatus>
	• Returned string does not include quotes (they are used in this description for clarity).
	  • "Unknown": Default value if data injection operation has not been performed yet, or if operation was incomplete
	  • "Valid": GPS data injection succeeded
	  • "Invalid": GPS data injection failed
	  • "xtra.bin file has bad crc"
	  • "GPS Busy, end current session first"
	  • "error reading xtra.bin file"
	  • "bad TOA in xtra.bin file": The XTRA data retrieved from the XTRA server is too old (exceeds the Time Of Applicability).
	
	<timeStamp> (GPS time stamp)
	• Format: <year> <month> <day> <dayOfWeek> <time>
	  • <year>: 4 digits (Example: 2008)
	  • <month>: 2 digits (01–12)
	  • <day>: 2 digits (01–31)
	  • <dayOfWeek>: 1 digit (0–6) where 0=Monday
	  • <time>: time of day (Example: 13:15:45)
	• Example: 2008 02 28 5 13:15:45 represents Thursday 28 Feb 2008 at 1:15:45 PM

Table 9-2: GPS command details (Continued)

Command	Description
!GPSXTRATIME Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Inject GPS or UTC time into gpsOneXTRA system** Inject the GPS or UTC time into the gpsOneXTRA system. **Usage:** • Execution: **AT!GPSXTRATIME=<YYYY>, <MM>, <DD>, <hh>, <mm>, <ss>, <utc>, <force>, <uncrtn>** Response: Xtra command sent successfully OK *or* Error code = <err> OK Purpose: Inject the specfied date and time into the gpsOneXTRA system. If the command fails, it returns “Error code = <err>”. • Query List: **AT!GPSXTRATIME=?** Purpose: Return supported parameter values. **Parameters:** <YYYY> (Year) • 4 digits required <MM> (Month) • Valid range: 1–12 <DD> (Day) • Valid range: 1–31 <hh> (Hour) • Valid range: 0-23 <mm> (Minute) • Valid range: 0–59 <ss> (Second) • Valid range: 0–59 <utc> (Flag indicating time type) • 0=GPS time • 1=UTC time <force> (Force or allow GPS subsystem to decide to accept the time entered) • 0=Do not force acceptance • 1=Force acceptance <err> (Error code returned if command fails) • 3=Bad CRC for XTRA data file • 4=Old XTRA data file • 7=GPS subsystem busy • 8=GPS time reference entered is invalid • 9=Unknown error

Table 9-2: GPS command details (Continued)

Command	Description
!GPSXTRATIMEENABLE Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report GPS XTRA time settings** Enable or disable GPS XTRA time information, and set or report specific GPS XTRA time settings. *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSXTRATIMEENABLE=<enable> [,<thresh>, <delay>]** Response: OK *or* ERROR Purpose: Enable or disable time information. If enabled, sets the uncertainty threshold and delay time to retry with a backup server. • Query: **AT!GPSXTRATIMEENABLE?** Response: XTRA Time Info Enabled: <enable> XTRA Time Uncertainty Threshold: <thresh> XTRA Time Delay Threshold: <delay> Purpose: Return the current values of GPS XTRA time parameters. • Query List: **AT!GPSXTRATIMEENABLE=?** Purpose: Return supported execution parameter values. **Parameters:** <enable> (Enable or disable time information) • 0=Disable. To fully disable gpsOneXTRA, you must also call **!GPSXTRADATAENABLE=0** to disable gpsOneXTRA data functionality. • 1=Enable <thresh> (XTRA time uncertainty threshold, in ms) • Valid range: 100–30000 <delay> (Time to delay before retrying with backup server, in ms) • Valid range: 100–10000

Table 9-2: GPS command details (Continued)

Command	Description
!GPSXTRATIMEURL Supporting chipsets (GPS-enabled devices only): • MDM6200 • MDM8200A • MDM8220 • MDM9200 • MDM9600 • MSM6290 • QSC6270 (min f/w rev: S2.0)	**Set/report GPS XTRA SNTP server URLs** Set or report the URLs of up to three GPS XTRA SNTP (Simple Network Time Protocol) servers. *Note: **!RESET** must be issued after this command is used.* **Usage:** • Execution: **AT!GPSXTRATIMEURL=<urlIndex>,<url>** Response: OK *or* ERROR Purpose: Set the URL used for the primary, secondary, or tertiary data server. • Query: **AT!GPSXTRATIMEURL?** Response: XTRA SNTP Primary Server: <url 1> XTRA SNTP Secondary Server: <url 2> XTRA SNTP Tertiary Server: <url 3> Purpose: Return the URLs of the primary, secondary, and tertiary SNTP servers. **Parameters:** <urlIndex> (Server index) • 1=Primary server • 2=Secondary server • 3=Tertiary server <url> (Server URL) • URL string includes quotes • Example: “xtra1.gpsoneextra.net” • Maximum string length=128 characters

Error codes

Table 9-3 describes error codes that can be returned by **!GPSEND** (page 184), **!GPSSTATUS** (page 197), and **!GPSTRACK** (page 200).

Table 9-4 on page 210 describes error codes that can be returned by **!GPSFIX** (page 185)

Table 9-3: AT command error codes (!GPSEND, !GPSSTATUS, !GPSTRACK)

Error code	Description
0	Phone is offline
1	No service
2	No connection with PDE (Position Determining Entity)
3	No data available
4	Session Manager is busy
5	Reserved

Table 9-3: AT command error codes (!GPSEND, !GPSSTATUS, !GPSTRACK) (Continued)

Error code	Description
6	Phone is GPS-locked
7	Connection failure with PDE
8	Session ended because of error condition
9	User ended the session
10	End key pressed from UI
11	Network session was ended
12	Timeout (for GPS search)
13	Conflicting request for session and level of privacy
14	Could not connect to the network
15	Error in fix
16	Reject from PDE
17	GPS is disabled
18	Ending session due to E911 call
19	Server error
20	Reserved
21	Reserved
22	Unknown system error
23	Unsupported service
24	Subscription violation
25	Desired fix method failed
26	Reserved
27	No fix reported because no Tx confirmation was received
28	Network indicated normal end of session
29	No error specified by the network
30	No resources left on the network
31	Position server not available
32	Network reported an unsupported version of protocol

Table 9-4: AT command error codes (!GPSFIX)

Error code	Description
0	No error
1	Invalid client ID
2	Bad service parameter
3	Bad session type parameter
4	Incorrect privacy parameter
5	Incorrect download parameter
6	Incorrect network access parameter
7	Incorrect operation parameter
8	Incorrect number of fixes parameter
9	Incorrect server information parameter
10	Error in timeout parameter
11	Error in QOS accuracy threshold parameter
12	No active session to terminate
13	Session is active
14	Session is busy
15	Phone is offline
16	Phone is CDMA locked
17	GPS is locked
18	Command is invalid in current state
19	Connection failure with PDE
20	PDSM command buffer unavailable to queue command
21	Search communication problem
22	Temporary problem reporting position determination results
23	Error mode not supported
24	Periodic NI in progress
25	Unknown error
26	Unknown error

10: STK Commands

Introduction

This chapter describes commands used to configure host support for STK features, monitor and respond to unsolicited proactive SIM commands, and present STK option menus provided by the SIM. These commands implement a subset of the functions defined in 3GPP TS 11.14.

STK interactions

The host STK support profile informs the SIM about the STK features the host supports. This profile is downloaded to the SIM each time the modem resets. At any time, the profile can be updated by using AT!STKPD and then resetting the modem.

Processing unsolicited SIM commands

To process commands issued by the SIM:

1. Enable the AT interface for STK functionality—issue the command **AT!CUSTOM="STKUIEN", 2**.
2. Monitor the command prompt for unsolicited proactive SIM commands:
   - "AT!STKC: <cmdId>"—This is a command that requires a response. See Table 10-2 on page 213 for supported commands.
   - "AT!STKN: <cmdId>, <data>"—This is a notification that does not require a response. See Table 10-18 on page 240 and Table 10-19 on page 241 for supported notifications.
3. If the unsolicited SIM command is "AT!STKC: <cmdId>, follow the appropriate procedure below:
   - If <cmdId> = "81" (End of proactive session), no response is required.
   - If <cmdId> = "25" (Set Up Menu):
     i. Respond with **AT!STKGC=<cmdId>** to retrieve the menu structure.
     ii. Respond with **AT!STKCR=25,0** to indicate success.
     iii. Determine the menu item to select (for example, present the menu to the user and get their input) and send it to the SIM using **AT!STKMS=<item>**.
     iv. The SIM will take action on the selected item and will send a new unsolicited command to the host for the next operation to be performed.

- For any other <cmdId>:
  i. Respond with **AT!STKGC=<cmdId>** to retrieve the data to use for <cmdId>.
  i. Execute the requested command (<cmdId>) and respond with: **AT!STKCR=<cmdId>,<result>, ...**
  or, optionally (for MSM6290 only, when <cmdId> = “11” (Send SS), “12” (Send USSD), or “13” (Send SMS)), **AT!STKAUTOCR=<cmdId>**

Configuring the host’s profile

To configure the host’s profile for STK support:

1. At any time, issue the command **AT!STKPD=<bitmask>** where the <bitmask> indicates supported features.
2. Reset the modem. When the modem restarts, the profile automatically downloads to the SIM.

Notifying SIM of host-monitored STK events

(Devices supporting **!STKEVENT** and **!STKEVENTLIST** only)

If the SIM has been configured (typically, by the carrier) to request that the host monitor specific events:

- Use **!STKEVENTLIST** to determine which events are to be monitored.
- Use **!STKEVENT** to notify the SIM whenever a monitored event occurs.

Command summary

The table below lists the commands described in this chapter.

Table 10-1: STK commands

Command	Description	Page
!STKAUTOCR	Configure host responses to SIM commands	213
!STKC	Receive unsolicited SIM command	214
!STKCR	Respond to unsolicited SIM command	215
!STKDTMF	Send DTMF string on active call	216
!STKEVENT	Notify SIM when monitored STK event occurs	216
!STKEVENTLIST	Return list of host-monitored STK events	217
!STKGC	Retrieve data for unsolicited SIM command	218
!STKMS	Request menu item selection or help from SIM	219
!STKPD	Update STK supported features profile	220
!STKPLI	Record local provisioning information	222
!STKVER	Display STK version	222

Command reference

Table 10-2: STK command details

Command	Description
!STKAUTOCR Supporting chipsets: • MSM6290: (min f/w rev: K2_0_7_49ap) *Note: This command is not password-protected.*	**Configure host responses to SIM commands** This command is used by the host to send 'command data' in response to specific proactive SIM commands. When the host receives one of the proactive SIM commands listed in the <cmdId> parameter description, it should issue this command to automatically send the appropriate command data. **Usage:** • Execution: **!STKAUTOCR=<cmdId>** Response: OK *or* ERROR *(Error is returned if <cmdId> does not match the previously received proactive SIM command.)* Purpose: Send a response to a specific proactive command. • Query List: **AT!STKAUTOCR=?** Purpose: Return the expected command format. **Parameters:** <cmdId> (STK command issued by the SIM) • Valid values (hexadecimal format)—no other values are supported at this time: • 11=Command data for Send SS command • 12=Command data for Send USSD command • 13=Command data for Send SMS command

Table 10-2: STK command details (Continued)

Command	Description

!STKC
Supporting chipsets:
• All, with these exceptions:
  • MSM6290 (min f/w rev: K1_0_2_3ap)
 	
Note: This command is not password-protected.
	Receive unsolicited SIM command
	This command is used by the SIM to send unsolicited notifications to the host, and is used by the host to query the last command received.
	Usage:
	• Unsolicited SIM command:
	!STKC: <cmdId>
	Purpose: Unsolicited notification sent by the SIM to the host.
	• Query: **AT!STKC?**
	Response: Outstanding Proactive Command: <cmdId>
	OK
	or
	OK
	Purpose: Return the <cmdId> received in the last unsolicited SIM command notification.
	Parameters:
	<cmdId> (STK command issued by the SIM)
	• Valid values (hexadecimal format):
	  • 05=Set up Event List command
	  • 10=Set up Call command
	  • 11=Send SS command
	  • 12=Send USSD command
	  • 13=Send SMS command
	  • 14=Send DTMF command
	  • 15=Launch Browser command
	  • 20=Play Tone command
	  • 21=Display Text command
	  • 22=Get Inkey command
	  • 23=Get Input command
	  • 24=Select Item command
	  • 25=Set Up menu command
	  • 28=Set Up Idle Mode Text command
	  • 35=Language Notification command
	  • 81=End of proactive session

Table 10-2: STK command details (Continued)

Command	Description

!STKCR
Supporting chipsets:
• All, with these exceptions:
  • MSM6290 (min f/w rev: K1_0_2_3ap)
 	
Note: This command is not password-protected.
	Respond to unsolicited SIM command
	Respond to the last received unsolicited SIM command, sending the results of the command's execution and any associated data (dependent on type of command executed).
	
	Note: No response is required for <cmdId = 81>.
	
	Usage:
	• Execution: **AT!STKCR=<cmdId>, <result>[, <data>]**
	Response: OK
	or ERROR
	Purpose: Send the result of the command that was just executed and any associated data to the SIM.
	• Query List: **AT!STKCR=?**
	Purpose: Return the expected command format.
	Parameters:
	<cmdId> (The STK command identified by the last received unsolicited SIM command)
	• Valid values (hexadecimal format):
	  • 05=Set up Event List command
	  • 10=Set up Call command
	  • 11=Send SS command
	  • 12=Send USSD command
	  • 13=Send SMS command
	  • 14=Send DTMF command
	  • 15=Launch Browser command
	  • 20=Play Tone command
	  • 21=Display Text command
	  • 22=Get Inkey command
	  • 23=Get Input command
	  • 24=Select Item command
	  • 25=Set Up menu command
	  • 28=Set Up Idle Mode Text command
	  • 35=Language Notification command
	
	<result> (Result of host's attempt to process the <cmdId>)
	• Format is <cmdId>-dependent. See STK command (<cmdId>) parameters on page 223 for details.
	
	<data> (Information obtained when <cmdId> was processed on the host)
	• Format is <cmdId>-dependent. See STK command (<cmdId>) parameters on page 223 for details.

Table 10-2: STK command details (Continued)

Command	Description

!STKDTMF
Supporting chipsets:
• MSM6290 (min f/w rev: K2_0_7_52)
 	
Note: This command is not password-protected.
	Send DTMF string on active call
	Command used by the host to send a DTMF (Dual Tone Multi-Frequency) string on an active call.
	Usage:
	• Execution: **AT!STKDTMF=<DTMF>**
	Response: OK *(returned if string is completely sent)*
	or ERROR *(returned if string is interrupted)*
	Purpose: Send
	Parameters:
	<DTMF> (Dual Tone Multi Frequency string)
	• ASCII string surrounded by quote marks. (e.g. “5551212C1A”)
	• Valid characters:
	  • ‘0’–’9’
	  • ‘A’=‘*’
	  • ‘B’=‘#’
	  • ‘C’=3-second pause
	Example:
	• AT!STKDTMF = “5551212C1A”
	The number 5551212 is sent, followed by a three second pause, the number ‘1’, and a ‘*’.

!STKEVENT
Supporting chipsets:
• MSM6290 (min f/w rev: K2_0_7_49ap)
• QSC6270 (min f/w rev: S2_0_0_11ap)
 	
Note: This command is not password-protected.
	Notify SIM when monitored STK event occurs
	Notify SIM when a monitored SDK event occurs.
	
	Note: To identify all monitored events, see !STKEVENTLIST on page 217.
	
	Usage:
	• Execution: **AT!STKEVENT=<event>**
	Response: OK
	or ERROR
	Purpose: Send message to SIM indicating <event> has occurred.
	• Query List: **AT!STKEVENT=?**
	Purpose: Display the execution format and parameter values
	Parameters:
	<event> (The STK event detected by the host.)
	• Valid values (hexadecimal format):
	  • 04=User activity
	  • 05=Idle screen

Table 10-2: STK command details (Continued)

Command	Description
!STKEVENTLIST Supporting chipsets: • MSM6290 (min f/w rev: K2_0_7_49ap) • QSC6270 (min f/w rev: S2_0_0_11ap) *Note: This command is not password-protected.*	**Return list of host-monitored STK events** Display a list (hexadecimal string) of the STK events monitored by the host. (This SIM is configured with this list by the SIM provider (the carrier, usually).) *Note: When a monitored event occurs, use !STKEVENT on page 216 to notify the SIM.* **Usage:** • Query: **AT!STKEVENTLIST?** Response: !STKEVENTLIST: <event>[<event>]] OK Example: !STKEVENTLIST: 0405 Purpose: Display a list of host-monitored STK events. **Parameters:** <event> (An STK event being monitored by the host.) • Valid values (hexadecimal format): • 04=User activity • 05=Idle screen

Table 10-2: STK command details (Continued)

Command	Description

!STKGC
Supporting chipsets:
• All, with these exceptions:
  • MSM6290 (min f/w rev: K1_0_2_3ap)
 	
Note: This command is not password-protected.
	Retrieve data for unsolicited SIM command
	Retrieve the data associated with the last received unsolicited SIM command.
	Usage:
	• Execution: **AT!STKGC=<cmdId>**
	Response: OK
	or ERROR
	Purpose: Retrieve the data for <cmdId> identified in the last unsolicited SIM command. If the <cmdId> doesn't match the last command, an error is returned.
	• Query List: **AT!STKGC=?**
	Purpose: Return the expected command format.
	Parameters:
	<cmdId> (The STK command identified by the last received unsolicited SIM command)
	• Valid values (hexadecimal format):
	  • 05=Set up Event List command
	  • 10=Set up Call command
	  • 11=Send SS command
	  • 12=Send USSD command
	  • 13=Send SMS command
	  • 14=Send DTMF command
	  • 15=Launch Browser command
	  • 20=Play Tone command
	  • 21=Display Text command
	  • 22=Get Inkey command
	  • 23=Get Input command
	  • 24=Select Item command
	  • 25=Set Up menu command
	  • 28=Set Up Idle Mode Text command
	  • 35=Language Notification command
	<data> (Information needed to be able to execute the specified <cmdId>)
	• Format is <cmdId>-dependent. See STK command (<cmdId>) parameters on page 223 for details.

Table 10-2: STK command details (Continued)

Command	Description
!STKMS Supporting chipsets: • All, with these exceptions: • MSM6290 (min f/w rev: K1_0_2_3ap) *Note: This command is not password-protected.*	**Request menu item selection or help from SIM** Instruct the SIM to select a menu item or to respond with help information for the menu item via a Display Text command (<cmdId = 21>). **Usage:** • Execution: **AT!STKMS=<item>[, <help>]** Response: OK *or* ERROR *or* Error code: <error> OK Purpose: Instruct the SIM to select the specified menu item, or to respond with help information for the specified menu item. • Query List: **AT!STKMS=?** Purpose: Return the expected command format. **Parameters:** <item> (Menu item) • Integer value obtained from the previously processed Set Up Menu command (<cmdId=25>). <help> (Menu item) • 0=SIM should select the menu <item>. This is the default behavior if <help> is not specified. • 1=Provide help information for this menu <item> using a Display Text command (<cmdId = 21>). <error> • 0=Card is busy • 1=General failure

Table 10-2: STK command details (Continued)

Command	Description

!STKPD
Supporting chipsets:
• All, with these exceptions:
  • MSM6290 (min f/w rev: K1_0_2_3ap)
 	
Note: This command is not password-protected.
	Update STK supported features profile
	Indicate to the SIM which STK features are supported by the host.
	
	Note: The modem must be restarted before the new profile information takes effect. (The modem downloads the profile to the SIM automatically each time it resets.)
	
	Usage:
	• Execution: **AT!STKPD=<bitmask>**
	Response: OK
	or ERROR
	Purpose: Store the new profile on the host. The profile will be downloaded the next time the modem resets.
	• Query: **AT!STKPD?**
	Response: Profile config=<bitmask>
	OK
	Purpose: Return the current profile <bitmask>.
	• Query List: **AT!STKPD=?**
	Purpose: Return the expected command format.
	
	Parameters:
	<bitmask> (Supported STK features)
	• Format: Seven hexadecimal bytes
	  • Byte order: 07 06 05 04 03 02 01
	  • Bit order: 76543210
	• Example:
	  • “0A000000000001”—“Number of character support across ME” = 10 and “Menu selection support” = enabled. All other values are 0.
	• In the ‘bit’ references below, the code in brackets—“(BXbY)”—refers to the bit position in the terminal profile message defined in 3GPP TS 11.14, where “BX” = byte X, and “bY” = bit Y.
	• Byte 01:
	  • Bit 0: Menu selection support (B1b4)
	  • Bit 1: Support for alpha in call control (B2b5)
	  • Bit 2: UCS2 entry support (B2b6)
	  • Bit 3: UCS2 display support (B2b7)
	  • Bit 4: Display Text command support (B3b1)
	  • Bit 5: Get Inkey command support (B3b2)
	  • Bit 6: Get Input command support (B3b3)
	  • Bit 7: Play Tone command support (B3b5)
	(Continued on next page)

Table 10-2: STK command details (Continued)

Command	Description

!STKPD (continued)	**Update STK supported features profile (continued)**
	• Byte 02:
	  • Bit 0: Select Item command support (B4b1)
	  • Bit 1: Send SMS command support (B4b2)
	  • Bit 2: Send SS command support (B4b3)
	  • Bit 3: Send USSD command support (B4b4)
	  • Bit 4: Set Up Call command support (B4b5)
	  • Bit 5: Set Up Menu command support (B4b6)
	  • Bit 6: Set Up Idle Mode Text command support (B8b5)
	  • Bit 7: Second alpha in setup call support (B8b7)
	• Byte 03:
	  • Bit 0: Second capability configuration parameter support (B8b8)
	  • Bit 1: Sustained display text support (B9b1)
	  • Bit 2: Send DTMF command support (B9b2)
	  • Bit 3: Language notification command support (B9b6)
	  • Bit 4: Launch Browser command support (B9b7)
	  • Bit 5: Softkey support in select item command (B10b1)
	  • Bit 6: Softkey support in setup menu command (B10b2)
	  • Bit 7: Screen size support (B14b8)
	• Byte 04:
	  • Bit 0: Variable font size support (B15b8)
	  • Bit 1: Display resized support (B16b1)
	  • Bit 2: Text wrapping support (B16b2)
	  • Bit 3: Text scrolling support (B16b3)
	  • Bit 4–Bit 7: Not used
	• Byte 05:
	  • Bit 0–Bit 7: Maximum softkey size (B11b1–B11b8)
	• Byte 06:
	  • Bit 0–Bit 4: Number of character support down ME (B14b1–B14b5)
	  • Bit 5–Bit 7: Reduce width of menu support (B16b6–B16b8)
	• Byte 07:
	  • Bit 0–Bit 6: Number of character support across ME (B15b1–B15b7)
	  • Bit 7: Not used

Table 10-2: STK command details (Continued)

Command	Description
!STKPLI Supporting chipsets: • All, with these exceptions: • MSM6290 (min f/w rev: K1_0_2_7ap)	**Record local provisioning information** This command, used when provisioning the modem, records a vendor-defined value for a specific command qualifier in NV memory. This value is then sent to the SIM automatically when the SIM issues a Provide Local Information proactive command—no action is required by the host. **Usage:** • Execution: **AT!STKPLI=<qualifier>, <value>** Response: OK *or* ERROR Purpose: Store <value> in NV memory. • Query List: **AT!STKPLI=?<qualifier>** Purpose: Return the stored <value>. **Parameters:** <qualifier> (Command qualifier for Provide Local Information proactive command) • Represented as hexadecimal ASCII • 0xFE: This is a reserved qualifier that is being overloaded to store the vendor-defined <value> <value> (Single byte vendor-defined value) • Represented as hexadecimal ASCII • Valid values: '00'–'FF'
!STKVER Supporting chipsets: • MDM8220: (min f/w rev: N2.0 Release 5) • MDM9200 • MDM9600	**Display STK version** Display the STK version. **Usage:** • Query: **AT!STKVER?** Response: <stkVersion> OK *or* ERROR *(ERROR appears for any firmware revision that does not support the command.)* Purpose: Return the current STK version. **Parameters:** <stkVersion> (Version of currently running STK) • Numeric value (e.g. 1, 2, etc.)

STK command (<cmdId>) parameters

Set Up Event List (<cmdId=05>)

Table 10-3: Setup Event List parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <event>[<event>]		

<event>	Hex	Supported event list
		• 04—User activity
		• 05—Idle screen available
		• Examples:
		  • “05”
		  • “0405”

!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully

Set Up Call (<cmdId=10>)

Table 10-4: Setup Call parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <method>, <TON>, <NPI>, <address>, <subaddress>, <ccp>, [<DCS1>,] <alphaId1>, <iconId>, <dispMode>, [<DCS2>,] <alphaId2>, <iconId>, <dispMode>, <redial>, <timeout>		
<method>	Integer	Call setup method • 0—Only if there are no other calls • 1—Put all other calls on hold • 2—Disconnect all other calls
<TON>	Integer	Type of number • 0—Unknown • 1—International • 2—National • 3—Network specific
<NPI>	Integer	Numbering plan identifier • 0—Unknown • 1—ISDN telephony • 3—Data • 4—Telex • 9—Private
<address>	Hex string	Dialing address

Table 10-4: Setup Call parameters (Continued)

Parameter	Format	Description
<subaddress>	Hex string	Dialing subaddress
<ccp>	Hex string	Capability configuration parameters
<DCS1>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId1> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId1>	Hex string	Alpha identifier for user confirmation
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <alphaId> or text string
<DCS2>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId2> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId2>	Hex string	Alpha identifier for call setup display
<redial>	integer	Redial flag • Redial not required • Redial required
<timeout>	Integer	Timeout period (in ms)

Table 10-4: Setup Call parameters (Continued)

Parameter	Format	Description
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Command beyond ME's capabilities • 2—Currently busy on call • 3—Currently busy with SS transaction • 4—Terminated by user • 5—SS returned Result Error Code • 6—Network currently unable to process command • 7—Call setup not accepted • 8—User cleared down call before connection or network release

Send SS (<cmdId=11>)

Table 10-5: Send SS parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <TON>, <NPI>, <address>, [<DCS>,] <alphaId>, <iconId>, <dispMode>		
<TON>	Integer	Type of number • 0—Unknown • 1—International • 2—National • 3—Network specific
<NPI>	Integer	Numbering plan identifier • 0—Unknown • 1—ISDN telephony • 3—Data • 4—Telex • 9—Private
<address>	Hex string	SS address
<DCS>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation

Table 10-5: Send SS parameters (Continued)

Parameter	Format	Description
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <alphaId> or text string
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Command beyond ME's capabilities • 2—Currently busy with USSD transaction • 3—Currently busy with SS transaction • 4—Terminated by user • 5—SS returned Result Error Code • 6—Network currently unable to process command

Send USSD (<cmdId=12>)

Table 10-6: Send USSD parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <dcs1>, <ussd>, [<DCS2>,] <alphaId>, <iconId>, <dispMode>		
<dcs1>	Integer	Data coding scheme for <ussd> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<ussd>	Hex string	USSD string
<DCS2>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation

Table 10-6: Send USSD parameters (Continued)

Parameter	Format	Description
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <alphaId> or text string
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Command beyond ME's capabilities • 2—Currently busy with USSD transaction • 3—Currently busy with SS transaction • 4—Terminated by user • 5—SS returned Result Error Code • 6—Network currently unable to process command

Send SMS (<cmdId=13>)

Table 10-7: Send SMS parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <pack>, <tpdu>, <TON>, <NPI>, <address>, [<DCS>,] <alphaId>, <iconId>, <dispMode>		
<pack>	Integer	Packing flag • 0—Packing not required • 1—Packing required
<tpdu>	Hex string	TPDU string
<TON>	Integer	Type of number • 0—Unknown • 1—International • 2—National • 3—Network specific

Table 10-7: Send SMS parameters (Continued)

Parameter	Format	Description
<NPI>	Integer	Numbering plan identifier • 0—Unknown • 1—ISDN telephony • 3—Data • 4—Telex • 9—Private
<address>	Hex string	Destination address
<DCS>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <alphaId> or text string
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Command beyond ME's capabilities • 2—SMS RP error

Send DTMF (<cmdId=14>)

Table 10-8: Send DTMF parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <dtmf>, [<DCS>,] <alphaId>, <iconId>, <dispMode>		
<dtmf>	Hex string	DTMF string

Table 10-8: Send DTMF parameters (Continued)

Parameter	Format	Description
<DCS>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <alphaId> or text string
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Command beyond ME's capabilities • 2—Not in speech call • 3—Terminate proactive session

Launch browser (<cmdId=15>)

Table 10-9: Launch browser parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <comQual>, <url>, <browserId>, <bearer>, <dcs1>, <gateway>, [<DCS2>,] <alphaId>, <iconId>, <dispMode>, <numFiles>[, <provfiles>, [...]]		
<comQual>	Integer	Command qualifier • 0—Launch browser if not already launched • 2—Use existing browser • 3—Close existing browser and launch new browser
<url>	Hex string	Initial browser URL • 8-bit data using the GSM default 7-bit alphabet • If the <url> is null (""), use the browser's default <url>
<browserId>	Hex string	Browser Id to use • "00"—Use the default browser

Table 10-9: Launch browser parameters (Continued)

Parameter	Format	Description
<bearer>	Hex string	List of one or more allowed bearers, sorted in priority order • "00"—SMS • "01"—CSD • "02"—USSD • "03"—GPRS • Example: "010200"—CSD, USSD, and SMS support; CSD is highest priority
<dcs1>	Integer	Data coding scheme for <gateway> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<gateway>	Hex string	Gateway text string in <dcs> format
<DCS2>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <alphaId> or text string
<numFiles>	Integer	Number of provisioning files following this parameter
<provFiles>	Hex string	List of zero or more provisioning files separated by commas. Each file includes its full path.

Table 10-9: Launch browser parameters (Continued)

Parameter	Format	Description
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Command performed—partially completed • 2—Command performed—missing information • 3—Error—no specific cause given • 4—Bearer unavailable • 5—Browser unavailable • 6—ME cannot process command • 7—Network cannot process command • 8—Command beyond ME's capabilities

Play tone (<cmdId=20>)

Table 10-10: Play tone parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: [<DCS>,] <alphaId>, <tone>, <duration>		
<DCS>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation

Table 10-10: Play tone parameters (Continued)

Parameter	Format	Description
<tone>	Hex string	Tone type. (SST—Standard Supervisory Tone; MPT—ME Proprietary Tone) • If no tone is specified, ME defaults to General beep (“10”) • “01”—Dial (SST) • “02”—Called subscriber busy (SST) • “03”—Congestion (SST) • “04”—Radio path acknowledge (SST) • “05”—Radio path not available/Call dropped (SST) • “06”—Error/Special information (SST) • “07”—Call waiting (SST) • “08”—Ringing tone (SST) • “10”—General beep (MPT) • “11”—Positive ack (MPT) • “12”—Negative ack or Error (MPT)
<duration>	Integer	Duration of tone to be played (in ms) • If <duration> = 0, use a host-defined default value
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully • 1—Terminate proactive session • 2—Specified tone not supported

Display text (<cmdId=21>)

Table 10-11: Display text parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <dcs>, <text>, <priority>, <clear>, <iconId>, <dispMode>, <response>		
<dcs>	Integer	Data coding scheme for <text> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<text>	Hex string	Text string in <dcs> format
<priority>	Integer	Priority information flag • 0—Do not display priority information • 1—Display priority information

Table 10-11: Display text parameters (Continued)

Parameter	Format	Description
<clear>	Integer	Clear message flag • 0—Do not allow user to clear message • 1—Allow user to clear message
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace any text string or <alphaId>) • 1—Display with <text> string
<response>	Integer	Response flag • 0—Normal response expected • 1—Immediate response expected
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Message displayed OK • 1—Terminate proactive session • 2—Screen is busy • 3—Backward move requested • 4—No response from user

Get Inkey (<cmdId=22>)

Table 10-12: Get Inkey parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <dcs>, <text>, <response>, <helpInfo>, <iconId>, <dispMode>		
<dcs>	Integer	Data coding scheme for <text> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<text>	Hex string	Text string in <dcs> format
<response>	Integer	Expected response character format • 0—SMS default alphabet • 1—Yes/No response only • 2—Digits only (0–9, *, #, +) • 3—UCS2 alphabet

Table 10-12: Get Inkey parameters (Continued)

Parameter	Format	Description
<helpInfo>	Integer	Help information flag • 0—No help information available • 1—Help information available
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replace the <text> string) • 1—Display with <text> string
!STKCR parameters		
<data>	[<dcs>, <text>]	Required for <result = 0>. (The SIM expects a single character to be provided in a Text String Data Object in the Terminal Response SIM command when data has been input.) • Format: <dcs>, <text>
<dcs>	Integer	Data coding scheme • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet

<text>	Hex string	Text string in <dcs> format
		• For Yes/No responses, use:
		  • "00"—No
		  • "01"—Yes

<result>	Integer	• 0—Data entered OK • 1—Terminate proactive session • 2—Help information requested • 3—Backward move requested • 4—No response from user

Get Input (<cmdId=23>)

Table 10-13: Get Input parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <dcs>, <text>, <response>, <echo>, <helpInfo>, <minLgth>, <maxLgth>, <dcs>, <default>, <iconId>, <dispMode>		
<dcs>	Integer	Data coding scheme for <text> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet

Table 10-13: Get Input parameters (Continued)

Parameter	Format	Description
<text>	Hex string	Text string in <dcs> format
<response>	Integer	Expected response character format • 0—SMS default alphabet • 1—Yes/No response only • 2—Digits only (0–9, *, #, +) • 3—UCS2 alphabet
<echo>	Integer	Echo flag • 0—No echo allowed. Actual input string can be hidden, or can be masked to indicate key entry using the following characters: 0–9, *, #. • 1—Echo input to display
<helpInfo>	Integer	Help information flag • 0—No help information available • 1—Help information available
<minLgth>	Integer	Minimum length of expected response • 0—No minimum length requirement • 1..255
<maxLgth>	Integer	Maximum length of expected response • 0..254 • 1..255—No maximum length requirement
<default>	Hex string	Text string in <dcs> format
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replaces any <text> string) • 1—Display with <text> string
!STKCR parameters		
<data>	[<dcs>, <text>]	If the <dcs> is present, but <text> is an empty string, then a null text string data object must be sent to the SIM. This is caused by the user making an 'empty' input. • Format: <dcs>, <text>
<dcs>	Integer	Data coding scheme • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet

Table 10-13: Get Input parameters (Continued)

Parameter	Format	Description
<text>	Hex string	Text string in <dcs> format
<result>	Integer	• 0—Data entered OK • 1—Terminate proactive session • 2—Help information requested • 3—Backward move requested • 4—No response from user

Select Item (<cmdId=24>)

Table 10-14: Select Item parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <numItems>, <selection>, [<default>,] <helpInfo>, [<DCS>,] <alphaId>, <iconId>, <dispMode>		
<numItems>	Integer	Number of items that are accessible in the menu structure • 0—Remove existing menu from the ME's menu structure • 1 or higher—Number of menu items
<selection>	Integer	Preferred user selection method • 0—No selection preference • 1—Soft key selection preferred
<default>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Default selection item
<helpInfo>	Integer	Help information flag • 0—No help information available • 1—Help information available
<DCS>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag

Table 10-14: Select Item parameters (Continued)

Parameter	Format	Description
<dispMode>	Integer	Icon usage • 0—Display icon only (replaces <alphaId>) • 1—Display with <alphaId> string
!STKCR parameters		
<data>	[<itemId>]	ID of item selected, or for which help is requested
<itemId>	Integer	Identifier of the item that was selected
<result>	Integer	• 0—Item selected OK • 1—Terminate proactive session • 2—Help information requested • 3—Backward move requested • 4—No response given

Set Up Menu (<cmdId=25>)

Table 10-15: Set Up Menu parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <numItems>, <selection>, [<default>,] <helpInfo>, [<DCS>,] <alphaId>, <iconId>, <dispMode> [<itemId>, <itemText>, <iconId>, <dispMode>, <nai> […]		
<numItems>	Integer	Number of items that are accessible in the menu structure • 0—Remove existing menu from the ME's menu structure • 1 or higher—Number of menu items
<selection>	Integer	Preferred user selection method • 0—No selection preference • 1—Soft key selection preferred
<default>	Integer	*Note: Only returned if AT!STKVER? returns ERROR.* Default selection item
<helpInfo>	Integer	Help information flag • 0—No help information available • 1—Help information available

Table 10-15: Set Up Menu parameters (Continued)

Parameter	Format	Description
<DCS>	Integer	*Note: Only returned if AT!STKVER? returns 1.* Data coding scheme for <alphaId> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<alphaId>	Hex string	Alpha identifier for user confirmation
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replaces <alphaId> or <text> strings) • 1—Display with <alphaId> or <text> strings
<itemId>	Integer	Menu item identifier
<itemText>	Hex string	Menu item text
<nai>	Hex string	Next action indicator (the next action the SIM will request when this menu item is selected)
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Menu successfully added/removed • 1—Problem with menu operation

Set Up Idle Mode Text (<cmdId=28>)

Table 10-16: Set Up Idle Mode parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <dcs>, <text>, <iconId>, <dispMode>		
<dcs>	Integer	Data coding scheme for <text> • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet
<text>	Hex string	Text string in <dcs> format

Table 10-16: Set Up Idle Mode parameters (Continued)

Parameter	Format	Description
<iconId>	Integer	Numeric tag of the icon to display (matches the index in the SIM's image file) • 0—No icon • 1..255—Icon tag
<dispMode>	Integer	Icon usage • 0—Display icon only (replaces <text> string) • 1—Display with <text> string
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Text successfully added/removed • 1—Problem performing operation

Language Notification (<cmdId=35>)

Table 10-17: Language Notification parameters

Parameter	Format	Description
!STKGC <data> parameter—Format: <spec>, <lang>		
<spec>	Integer	Language notification type • 0—Non-specific language notification • 1—Specific language notification
<lang>	Hex string	List of language codes
!STKCR parameters		
<data>		Not used
<result>	Integer	• 0—Command performed successfully

Response notification to Mobile Originating Call Control request (<cmdId=D4>)

Table 10-18: Response notification to MO Call Control request parameters

Parameter	Format	Description
!STKN <data> parameter—Format (depends on call type): Voice: <result>, <repeatind>, <alphaId>, 0, <TON>, <NPI>, <address>, <subaddress>, <ccp1>, <ccp2> SS: <result>, <repeatind>, <alphaId>, 1, <TON>, <NPI>, <address> USSD: <result>, <repeatind>, <alphaId>, 2, <dcs>, <ussd> PDP context: <result>, <repeatind>, <alphaId>, 6, <pdp> None: <result>, <repeatind>, <alphaId>, 7		
<result>	Integer	Call control result • 0—Allowed with no modifications • 1—Not allowed • 2—Allowed with modifications
<repeatind>	Integer	BC repeat indicator • 1—Alternate mode • 3—Sequential mode
<alphaId>	Hex string	Alpha identifier
<TON>	Integer	Type of number • 0—Unknown • 1—International • 2—National • 3—Network specific
<NPI>	Integer	Numbering plan identifier • 0—Unknown • 1—ISDN telephony • 3—Data • 4—Telex • 9—Private
<address>	Hex string	New dialing address
<subaddress>	Hex string	New dialing subaddress
<ccp1>	Hex string	First capability configuration parameters
<ccp2>	Hex string	Second capability configuration parameters
<dcs>	Integer	Data coding scheme • 0—7-bit GSM default alphabet (packed) • 4—8-bit GSM default alphabet (unpacked) • 8—UCS2 alphabet

Table 10-18: Response notification to MO Call Control request parameters

Parameter	Format	Description
<ussd>	Hex string	USSD control string
<pdp>	Hex string	PDP control string

Response notification to Mobile Originating SMS Control request (<cmdId=D5>)

Table 10-19: Response notification to MO SMS Control request parameters

Parameter	Format	Description
!STKN <data> parameter—Format: <result>, <alphaId>, <TON>, <NPI>, <rpaddress>, <TON>, <NPI>, <tpaddress>		
<result>	Integer	SMS control result • 0—Allowed with no modifications • 1—Not allowed • 2—Allowed with modifications
<alphaId>	Hex string	Alpha identifier
<TON>	Integer	Type of number • 0—Unknown • 1—International • 2—National • 3—Network specific
<NPI>	Integer	Numbering plan identifier • 0—Unknown • 1—ISDN telephony • 3—Data • 4—Telex • 9—Private
<rpaddress>	Hex string	RP (Relay Layer Protocol) address
<tpaddress>	Hex string	TP (Transport Layer Protocol) address

11: PAD Commands

Introduction

This chapter describes commands used to configure, initiate, and disconnect single- and multi-PAD (Packet Assembler/Disassembler) client and server connections. For detailed descriptions of PAD functionality, see *EMConnect Guide (Document 2131177)*.

Managing PAD sessions

- Before initiating a PAD session, use !PADCONF, !PADSETUP, and !PADFILTER to configure PAD profile settings.
- To initiate a PAD session, use !PADCONN (for a client connection) or !PADLISTEN (for a server connection).
- To switch between active PAD sessions, use !PADSWITCH.
- To disconnect a PAD session, use !PADDISCONN.
- To stop a PAD server, use !PADENDLISTEN.

Command summary

The table below lists the commands described in this chapter.

Table 11-1: PAD commands

Command	Description	Page
!PADCONF	Configure profile options	244
!PADCONN	Initiate PAD client connection	245
!PADDISCONN	Disconnect PAD connection	246
!PADENDLISTEN	Disable PAD server	246
!PADFILTER	Set IP address filters for TCP PAD server	247
!PADLISTEN	Initiate PAD server connection	248
!PADSETUP	Set/query PAD (Packet Assembler/Disassembler) profile connection parameters	249
!PADSWITCH	Switch active PAD session	250

Command reference

Table 11-2: PAD command details

Command	Description

!PADCONF
Supporting chipsets:
• MDM6200 (min f/w rev: P1_0_0_0)
• MDM6270 (min f/w rev: S2.0)
• MSM6290 (min f/w rev: K2.0 Release 2)
• QSC6270 (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Configure profile options
	Configure the PAD (Packet Assembler/Disassembler) session options and trigger conditions for packetization on the transmit side.
	
	Note: Multiple triggers can be active simultaneously. If any trigger condition is met, packetization occurs.
	
	This is a persistent setting (stored in NVRAM).
	For PAD server profiles, this configuration applies to all PAD sessions connected to the server.
	Usage:
	• Execution: **AT!PADCONF=<PADprofile>, <idle>, <interchar>, <pktlen>, <fwdopt>, <fwdchar>**
	Response: OK
	or ERROR
	Purpose: Configure trigger conditions for the specified <PADprofile>.
	• Query: **AT!PADCONF?<PADprofile>**
	Response: AT!PADCONF:<PADprofile>, <idle>, <interchar>, <pktlen>, <fwdopt>,<fwdchar>
	OK
	Purpose: Return the current timer conditions for the specified <PADprofile>.
	Parameters:
	<PADprofile> (PAD profile ID number)
	• Valid range depends on multipad support:
	  • Multipad not supported: 1–3
	  • Multipad supported: 1–10
	<idle> (Idle disconnect timeout trigger period)
	• Session disconnects if there is no data activity for this period of time.
	• Valid range: 0–65535 seconds (0—Trigger disabled)
	<interchar> (Inter-character timeout trigger period)
	• Packetization begins if the time interval between successive characters is greater than this period of time.
	• Valid range: 0–65535 ms (0—Trigger disabled)
	<pktlen> (Packet length trigger)
	• Packetization begins when this many bytes have been collected.
	• Valid range: 0–1460 bytes (0—Trigger disabled)
	
	(Continued on next page)

Table 11-2: PAD command details (Continued)

Command	Description
!PADCONF (continued)	**Configure profile trigger conditions for packetization (continued)** <fwdopt> (Forwarding character trigger) • Packetization begins when the <fwdchar> character appears in the data stream. • 0=Trigger disabled • 1=Trigger enabled. Character is included in message. • 2=Trigger enabled. Character is not included in message. <fwdchar> (Character that forces packetization to occur.) • Used in combination with <fwdopt>. • Valid range: 0–255—ASCII value of forwarding character
!PADCONN Supporting chipsets: • MDM6200 • MDM6270: (min f/w rev: S2.0) • MSM6290 (min f/w rev: K2.0 Release 2) • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Initiate PAD client connection** Initiate a PAD (Packet Assembler/Disassembler) client connection for a specific profile ID. *Note: The modem must have a data connection on the network with an open TCP or UDP socket on the remote agent, and at least one trigger condition must be defined for the specified profile ID. See !PADCONF on page 244.* **Usage:** • Execution: **AT!PADCONN=<PADprofile>** Response (multipad not supported): CONNECT PAD *or* NO CARRIER *or* ERROR Response (multipad supported): OK CONNECT PAD: <PADprofile>,<connectionState> *or* ERROR Purpose: Initiate a connection for the specified profile. For multipad, the "CONNECT PAD" response indicates the start of the PAD connection. • Query: **AT!PADCONN?<PADprofile>** Response: AT!PADCONN:<PADprofile>,<connectionState> OK Purpose: Display the current connection state of the specified profile. **Parameters:** <PADprofile> (PAD profile ID number) • Valid range depends on multipad support: • Multipad not supported: 1–3 • Multipad supported: 1–10 <connectionState> (Current connection state of the <PADprofile> session) • 0=Disconnected • 1=Connected

Table 11-2: PAD command details (Continued)

Command	Description
!PADDISCONN Supporting chipsets: • MDM6200 • MDM6270 (min f/w rev: S2.0) • MSM6290 (min f/w rev: K2.0 Release 2) • QSC6270 (min f/w rev: S2.0) *Note: This command is not password-protected.*	**Disconnect PAD connection** Disconnect the currently active PAD (Packet Assembler/Disassembler) session. *Note: This command is equivalent to using **ATH**.* **Usage:** • Execution: **AT!PADDISCONN [=<PADprofile>,<session>]** Response (multipad not supported): DISCONNECT PAD *or* OK Response (multipad supported): If PAD session has been disconnected (DCD line is deasserted as per AT&C setting): OK DISCONNECT PAD: <PADprofile>,<connectionState> *or* If data connection is already disconnected: OK Purpose: Disconnect the currently active profile. For multipad, the "DISCONNECT PAD" response appears after the session disconnects. **Parameters:** <PADprofile> (PAD profile ID number) • Only used when multipad is supported. • Valid range: 1–10 <session> (Session index) • Only used when multipad is supported. • Valid range: 0–3
!PADENDLISTEN Supporting chipsets: • MDM6200 (min f/w rev: P1_0_0_0) • MDM6270 (min f/w rev: S2.0.0.10) • QSC6270 (min f/w rev: S2.0.0.10) *Note: This command is not password-protected.*	**Disable PAD server** Disable a PAD server (and disconnect all PAD sessions related to the PAD server). If remote clients were connected, the following notification will be received after the OK response: DISCONNECT PAD: <padprofile>, <session> **Usage:** • Execution: **AT!PADENDLISTEN=<PADprofile>** Response: OK *or* ERROR Purpose: Disable the indicated PAD server. **Parameters:** <PADprofile> (PAD profile ID number) • Valid range: 1–10 <session> (Session index) • Valid range: 0–3

Table 11-2: PAD command details (Continued)

Command	Description

!PADFILTER
Supporting chipsets:
• MDM6200
• MDM6270 (min f/w rev: S2.0)
• MSM6290 (min f/w rev: K2.0 Release 2)
• QSC6270 (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Set IP address filters for TCP PAD server
	Set up to two IP address filters for a specific PAD (Packet Assembler/Disassembler) profile when configured as a TCP server.
	When a remote PAD client attempts to connect to the server, the server compares the client's IP addresses to the filter(s). If the address is in one of the ranges, the server accepts the client's connection.
	Filter setup:
	• Low and High IP addresses specified—Defines a range of allowed IP addresses.
	• One address (Low or High) specified—Defines a single allowed IP address.
	• No addresses specified for either filter—Filtering is disabled, all IP addresses are allowed.
	• If Low and High IP addresses are specified for a filter, the High address must be greater than the Low address or the execution command returns ERROR.
	
	Note: These settings are ignored if the specified profile is configured as a client.
	
	Usage:
	• Execution: **AT!PADFILTER=<PADprofile>[, <ip1Low>[, <ip1High>[, <ip2Low>[, <ip2High>]]]]**
	Response: OK
	or ERROR
	Purpose: Set up to two ranges of PAD client IP addresses that the TCP PAD server will accept.
	• Query: **AT!PADFILTER?<PADprofile>**
	Response: AT!PADFILTER: <PADprofile>, <ip1Low>, <ip1High>, <ip2Low>, <ip2High>
	OK
	Purpose: Show up to two ranges of PAD client IP addresses that the TCP PAD server will accept.
	Parameters:
	<PADprofile> (PAD profile ID number)
	• Valid range depends on multipad support:
	  • Multipad not supported: 1–3
	  • Multipad supported: 1–10
	<ip1Low> (Starting value of first IP range)
	• Standard IP address format. For example, 63.162.134.132.
	<ip1High> (Ending value of first IP range)
	• Standard IP address format. For example, 63.162.134.150.
	<ip2Low> (Starting value of second IP range)
	• Standard IP address format. For example, 63.162.178.110.
	<ip2High> (Ending value of second IP range)
	• Standard IP address format. For example, 63.162.178.130.

Table 11-2: PAD command details (Continued)

Command	Description

!PADLISTEN
Supporting chipsets:
• MDM6200
• MDM6270 (min f/w rev: S2.0)
• MSM6290 (min f/w rev: K2.0 Release 2)
• QSC6270 (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Initiate PAD server connection
	Initiate a TCP PAD (Packet Assembler/Disassembler) server connection for a specific profile ID.
	
	Note: At least one trigger condition must be defined for the specified profile ID. See !PADCONF on page 244.
	
	Usage:
	• Query: **AT!PADLISTEN?<PADprofile>**
	Response: AT!PADLISTEN:<PADprofile>,<connectionstate>
	OK
	Purpose: Display the current connection state of the specified profile.
	• Execution: **AT!PADLISTEN=<PADprofile>[,<manual>]**
	Response: OK *(Connected, waiting for remote TCP client to connect. Serial port remains in AT command mode.)*
	or CONNECT *(PAD connection established over TCP socket. Modem in PAD mode, and DCD line is asserted.)*
	or CLIENT-UP *(Remote TCP client is connected. Modem is in AT command mode, and DCD line remains deasserted.)*
	or NO CARRIER *(Connection failed. DCD line is deasserted.)*
	or ERROR *(Error activating session, or all triggers are disabled.)*
	Purpose: Initiate a server connection for the specified profile.
	Parameters:
	<PADprofile> (PAD profile ID number)
	• Valid range depends on multipad support:
	  • Multipad not supported: 1–3
	  • Multipad supported: 1–10
	
	<manual> (Action to take when remote client connects)
	• 0=Modem sends CONNECT to the host and switches port to PAD mode.
	• 1=Modem sends CLIENT-UP to host. It then waits for an ATO command before sending CONNECT to the host and switches port to PAD mode.
	
	<connectionstate> (Current connection state of the <PADprofile> PAD session)
	• 0=Disconnected
	• 1=Listening
	• 2=Connected (This option is not supported for multipad.)

Table 11-2: PAD command details (Continued)

Command	Description

!PADSETUP
Supporting chipsets:
• MDM6200
• MDM6270: (min f/w rev: S2.0)
• MSM6290: (min f/w rev: K2.0 Release 2)
• QSC6270: (min f/w rev: S2.0)
 	
Note: This command is not password-protected.
	Set/query PAD (Packet Assembler/Disassembler) profile connection parameters
	Set the connection parameters for a specific PAD profile, including its IP address and connection type, TCP/UDP port numbers, and auto-start configuration.
	Usage:
	• Execution: **AT!PADSETUP=<PADprofile>, <conntype>, <addrtype>, <ipaddr>, <localport>, <remoteport>, <autostart>[, <connprofile>]**
	Response: OK
	or ERROR
	Purpose: Configure the connection parameters for the specified <PADprofile>. The command returns ERROR if <autostart> is enabled on more than one PAD profile.
	• Query: **AT!PADSETUP?<PADprofile>**
	Response: AT!PADSETUP:<PADprofile>, <conntype>, <addrtype>, <ipaddr>, <localport>, <remoteport>, <autostart>, <connprofile>
	OK
	Purpose: Return the current connection setup parameters for the specified <PADprofile>.
	Parameters:
	<PADprofile> (PAD profile ID number)
	• Valid range depends on multipad support:
	  • Multipad not supported: 1–3
	  • Multipad supported: 1–10
	<conntype> (IP connection type)
	• 0=PAD profile disabled
	• 1=UDP client
	• 2=TCP client
	• 3=TCP server
	<addrtype> (IP address type)
	• 0=IPv4 address
	• (MDM6200 only, min f/w rev: P1_0_0_8) 1=IPv6 address
	<ipaddr> (Destination IP address)
	• Client mode only (<conntype = 1 or 2 only>). Parameter is ignored in server mode.
	• Standard IP address format. For example:
	  • IPv4: 63.162.134.132.
	  • (MDM6200 only, min f/w rev: P1_0_0_8) IPv6: 1234:5678:9012:3456:7890:1234:5678:9012
	<localport> (Modem port number)
	• 0—Port number is assigned by the modem (in Client mode only). In serial mode, must be non-zero.
	• 1–65535—Port number used by the modem.
	(Continued on next page)

Table 11-2: PAD command details (Continued)

Command	Description
!PADSETUP (continued)	**Set/query PAD (Packet Assembler/Disassembler) profile connection parameters (continued)** <remoteport> (Remote client port number) • 1–65535—Port number of remote client (in client mode only). Parameter is ignored in server mode. <autostart> (State of auto-start feature) • 0=Disabled (Default) • 1=Enabled <connprofile> (Connection profile ID) • Valid range: 1–16 • Default: 1 if not specified
!PADSWITCH Supporting chipsets: • MDM6200: (min f/w rev: P1_0_0_0) • MDM6270 (min f/w rev: S2.0.0.10) • QSC6270 (min f/w rev: S2.0.0.10) *Note: This command is not password-protected.*	**Switch active PAD session** Switch the active PAD session to another session (uniquely identified by the combination of <PADprofile> and <session>). After switching sessions, ATO can be used to enter PAD mode and then send and receive data for the active session, or to query available PAD sessions. **Usage:** • Execution: **AT!PADSWITCH=<PADprofile>,<session>** Response: OK *or* ERROR Purpose: Switch the active PAD session to the PAD session uniquely identified by <PADprofile> and <session>. • Query: **AT!PADSWITCH?<qualifier>** Response: !PADSWITCH: Active session: <PADprofile>, <session> !PADSWITCH: <padprofile>,<session>,<dst_ip>,<dst_port>,<unread> [...] *(repeats for each available PAD session]* OK Purpose: Show the active session (1st line of response) and all other available sessions (lines 2–*n*). **Parameters:** <PADprofile> (PAD profile ID number) • Valid range: 1–10 <session> (Session index) • Valid range: 0–3 <dst_ip> (Destination port address) • Standard IP address format. For example: • IPv4: 63.162.134.132. • (MDM6200 only, min f/w rev: P1_0_0_8) IPv6: 1234:5678:9012:3456:7890:1234:5678:9012 <dst_port> (Destination port) • Valid range: 0–65535 <unread> (Unread data flag) • 0=No unread data • 1=Unread data available

12: OMA-DM Commands

Introduction

This chapter describes commands used to configure DM (Device Management) accounts, sessions, and host–device–server interactions.

Command summary

The table below lists the commands described in this chapter.

Table 12-1: OMA-DM commands

Command	Description	Page
!IDSAUTOFOTA	Configure automatic settings for FOTA updates	252
!IDSAUTOSDM	Configure Subscriber Device Management response to server request	253
!IDSCONFIGACC	Configure DM account authentication mode and XML format	254
!IDSCREATEACC	Enter DM account credentials	255
!IDSDFLTACC	Set DM account to use for device-initiated sessions	256
!IDSFUMOROOT	Set DM Tree root path for FUMO node	256
!IDSPID	Set profile ID for DM data connection types	257
!IDSROAM	Configure DM client roaming support	257
!IDSSUPPORT	Configure DM sessions	258

Command reference

Table 12-2: OMA-DM command details

Command	Description
!IDSAUTOFOTA Supporting chipsets: • MDM9200 • MDM9600 *Note: This command is not password-protected.*	**Configure automatic settings for FOTA updates** Configure the automatic download, automatic update, and automatic check flags for over-the-air firmware updates: • Automatic check—If enabled, the device initiates a FOTA session on every startup to check if the server has a firmware update available. • Automatic download/Automatic upload—If enabled, the device does not request user permission before proceeding with the download or update. **Usage:** • Execution: **AT!IDSAUTOFOTA=<autodownload>, <autoupdate>, <autocheck>** Response: OK *or* ERROR Purpose: Set 'automatic' options. • Query: **AT!IDSAUTOFOTA?** Response: !IDSAUTOFOTA:<autodownload>, <autoupdate>, <autocheck> OK Purpose: Display current 'automatic' options. • Query List: **AT!IDSAUTOFOTA=?** Purpose: Display the execution command format and parameter values. **Parameters:** <autodownload> (Check for user permission before downloading firmware update) • 0=Permission required • 1=Permission not required <autoupdate> (Check for user permission before updating firmware) • 0=Permission required • 1=Permission not required. Auto update when download finishes. • 2=Auto update only on power up. • Note: The device will reboot when the update completes. <autocheck> (Check for firmware update on startup) • 0=Disabled. Do not check for firmware updates on startup. • 1=Enabled. Check for firmware updates on startup.

Table 12-2: OMA-DM command details (Continued)

Command	Description
!IDSAUTOSDM Supporting chipsets: • MDM9200 (min f/w rev: SWI9200X_00.07.02.01) • MDM9600 (min f/w rev: SWI9600M_01.00.06.00) *Note: This command is not password-protected.*	**Configure Subscriber Device Management response to server request** Configure the Subscriber Device Management (SDM) response to DM server requests. DM requests can be always accepted, always rejected, or presented to the host (user) application for a decision. **Usage:** • Execution: **AT!IDSAUTOSDM=<autosdm>** Response: OK *or* ERROR Purpose: Set the response behavior. • Query: **AT!IDSAUTOSDM?** Response: !IDSAUTOSDM:<autosdm> OK Purpose: Display the current response behavior setting. • Query List: **AT!IDSAUTOSDM=?** Purpose: Display the execution command format and parameter values. **Parameters:** <autosdm> (How to proceed with DM session) • 0=Permission required from host application before proceeding with session. (The user must choose whether to accept or reject requests as appropriate.) • 1=Always accept (proceed with) the DM session. Do not notify the host application. (Choose this behavior for standalone devices that do not present a GUI.) • 2=Always reject (do not proceed with) the DM session. Do not notify the host application.

Table 12-2: OMA-DM command details (Continued)

Command	Description

!IDSCONFIGACC
Supporting chipsets:
• MDM9200
 	
Note: This command is not password-protected.
	Configure DM account authentication mode and XML format
	Configure the preferred authentication mode and XML format for a DM account.
	Usage:
	• Execution: **AT!IDSCONFIGACC=<AccountIndex>, <authentication>, <xml_mode>**
	Response: OK
	or ERROR
	Purpose: Set the authentication mode and XML format for a specific DM account.
	• Query: **AT!IDSCONFIGACC?**
	Response: !IDSCONFIGACC:<AccountIndex>, <authentication>, <xml_mode>
	OK
	Purpose: Show the authentication mode and XML format for a specific DM account.
	• Query List: **AT!IDSCONFIGACC=?**
	Purpose: Display the execution command format and parameter values.
	Parameters:
	<AccountIndex> (DM account number)
	• Valid values: 1–2
	<authentication> (Account authentication mode)
	• Valid values
	  • “NONE”—No authentication
	  • “BASIC”—Basic authentication
	  • “DIGEST”—MD5 authentication
	  • “HMAC”—HMAC authentication
	<xml_mode> (XML format)
	• Valid values:
	  • “XML”—XML format
	  • “WBXML”—WBXML format

Table 12-2: OMA-DM command details (Continued)

Command	Description
!IDSCREATEACC Supporting chipsets: • MDM9200 • MDM9600 *Note: This command is not password-protected.*	**Enter DM account credentials** Enter the credentials for a DM account. **Usage:** • Execution: **AT!IDSCREATEACC=<AccountIndex>, <ServerAddress>, <ServerID>, <ServerPassword>, <ClientUsername>, <ClientPassword>** Response: OK *or* ERROR Purpose: Set the account credentials for a specific DM account. • Query: **AT!IDSCREATEACC?** Response: !IDSCREATEACC:<AccountIndex>, <ServerAddress>, <ServerID>,<ServerPassword>, <ClientUsername>, <Client-Password> OK Purpose: Show the account credentials for a specific DM account. • Query List: **AT!IDSCREATEACC=?** Purpose: Display the execution command format. **Parameters:** <AccountIndex> (DM account number) • Valid values: 1–2 <ServerAddress> (URL of DM server) • Maximum length—121 characters • This parameter configures the following DM tree node: • ./DMAcc/AppAddr/1/Addr <ServerID> (DM Server ID and Username) • Maximum length—32 characters • This parameter configures the following DM tree nodes: • ./DMAcc/ServerID • ./DMAcc/AppAuth/Server/AAuthName <ServerPassword> (DM Server Password) • Maximum length—32 characters • This parameter configures the following DM tree node: • ./DMAcc/AppAuth/Server/AAuthSecret <ClientUsername> (DM Client Username) • Maximum length—32 characters • This parameter configures the following DM tree node: • ./DMAcc/AppAuth/Client/AAuthName <ClientPassword> (DM Client Password) • Maximum length—32 characters • This parameter configures the following DM tree node: • ./DMAcc/AppAuth/Client/AAuthSecret

Table 12-2: OMA-DM command details (Continued)

Command	Description
!IDSDFLTACC Supporting chipsets: • MDM9200 *Note: This command is not password-protected.*	**Set DM account to use for device-initiated sessions** Indicate which DM account to use for device-initiated sessions. **Usage:** • Execution: **AT!IDSDFLTACC=<AccountIndex>** Response: OK *or* ERROR Purpose: Indicate the DM account to use for device-initiated sessions. • Query: **AT!IDSDFLTACC?** Response: !IDSDFLTACC:<AccountIndex> OK Purpose: Show which DM account is currently used for device-initiated sessions. • Query List: **AT!IDSDFLTACC=?** Purpose: Display the execution command format and parameter values. **Parameters:** <AccountIndex> (DM account number) • Valid values: 1–2
!IDSFUMOROOT Supporting chipsets: • MDM9200 • MDM9600	**Set DM Tree root path for FUMO node** Set the DM Tree root path for the FUMO node. **Usage:** • Execution: **AT!IDSFUMOROOT=<root path str>** Response: OK *or* ERROR Purpose: Set the FUMO node path. • Query: **AT!IDSFUMOROOT?** Response: !IDSFUMOROOT:<root path str> OK Purpose: Show the FUM node path • Query List: **AT!IDSFUMOROOT=?** Purpose: Display the execution command format and parameter values. **Parameters:** <enable> (Roaming support state) • Valid values: • 0=Disabled. The DM client will not attempt to connect to a DM server when roaming. • 1=Enabled. (Default) The DM client will attempt to connect to a DM server when roaming.

Table 12-2: OMA-DM command details (Continued)

Command	Description
!IDSPID Supporting chipsets: • MDM9200 (min f/w rev: SWI9200X_00.07.01.05) • MDM9600 (min f/w rev: SWI9600M_01.00.04.06) *Note: This command is not password-protected.*	**Set profile ID for DM data connection types** Set the connection profile ID that OMA-DM will use for its data connection. **Usage:** • Execution: **AT!IDSPID=<lte_profile>,<3GPP_legacy_profile>,<eHRPD profile>** Response: OK *or* ERROR Purpose: Set the profile ID for each data connection type. • Query: **AT!IDSPID?** Response: !IDSPID:<lte_profile>, <3GPP_legacy_profile>, <eHRPD profile> OK Purpose: Show the current profile IDs used for each data connection type. • Query List: **AT!IDSPID=?** Purpose: Display the execution command format and parameter values. **Parameters:** <lte_profile> (Profile ID to use for LTE data connections) • Valid values: • 0=Use connected or default profile ID. (Default) • 1–16=Profile ID <3GPP_legacy_profile> (Profile ID to use for non-LTE 3GPP data connections) • Valid values: • 0=Use connected or default profile ID. (Default) • 1–16=Profile ID <eHRPD profile> (Profile ID to use for eHRPD data connections) • Valid values: • 0=Use connected or default profile ID. (Default) • 101–150=Profile ID
!IDSROAM Supporting chipsets: • MDM9200	**Configure DM client roaming support** Configure the OMA DM client roaming option. **Usage:** • Execution: **AT!IDSROAM=<enable>** Response: OK *or* ERROR Purpose: Enable/disable DM roaming support. • Query: **AT!IDSROAM?** Response: !IDSROAM:<enable> OK Purpose: Show current DM roaming support state. • Query List: **AT!IDSROAM=?** Purpose: Display the execution command format and parameter values. **Parameters:** <enable> (Roaming support state) • Valid values: • 0=Disabled. The DM client will not attempt to connect to a DM server when roaming. • 1=Enabled. (Default) The DM client will attempt to connect to a DM server when roaming.

Table 12-2: OMA-DM command details (Continued)

Command	Description
!IDSSUPPORT Supporting chipsets: • MDM9200 • MDM9600	**Configure DM sessions** Enable/disable client-initiated and network-initiated DM device configuration and FOTA sessions. **Usage:** • Execution: **AT!IDSSUPPORT=<CI Config session>, <NI Config session>, <CI FOTA session>, <NI FOTA session>** Response: OK *or* ERROR Purpose: Enable/disable device configuration sessions and FOTA sessions. • Query: **AT!IDSSUPPORT?** Response: !IDSSUPPORT:<CI Config session>, <NI Config session>, <CI FOTA session>,<NI FOTA session> OK Purpose: Show current state of device configuration sessions and FOTA sessions. • Query List: **AT!IDSSUPPORTI=?** Purpose: Display the execution command format and allowed parameter values. **Parameters:** <CI Config session> (Client-initiated DM configuration session state) • 0=Disabled (Default) • 1=Enabled <NI Config session> (Network-initiated DM configuration session state) • 0=Disabled (Default) • 1=Enabled <CI FOTA session> (Client-initiated DM FOTA session state) • 0=Disabled (Default) • 1=Enabled <NI Config session> (Network-initiated DM FOTA session state) • 0=Disabled (Default) • 1=Enabled

13: SAR Backoff and Thermal Control Commands

Introduction

This chapter describes:

- SAR-related commands (Specific Absorption Rate)—SAR commands are used to meet regulatory requirements for the OEM host device by managing the modem's SAR backoff state. OEMs should carefully evaluate their use of these commands and their impact on device operation.

Note: Operators may require OEMs to disclose SAR settings and theory of operation for applicable certifications.

- Thermal mitigation-related commands—These commands may affect the host device's performance. OEMs should carefully evalute their use of these commands to ensure that the device meets performance expectations.

Command summary

The table below lists the commands described in this chapter.

Table 13-1: SAR backoff and thermal control commands

Command	Description	Page
!MAXPWR	Set/report maximum Tx power	260
!SARBACKOFF	Set/report maximum Tx power limit	261
!SARSTATE	Set/report SAR backoff state	262
!SARSTATEDFLT	Set/report default SAR backoff state	262
!THERMCONFIG	Set/report thermal mitigation configuration options	263
!THERMDELTATX	Set/report amount to reduce maximum Tx power	264
!THERMDELTATXTEMP	Set/report amount power backoff temperature threshold	265
!THERMENABLE	Enable/disable thermal mitigation	265
!THERMINFO	Display thermal mitigation information	266
!THERMTHRESHOLD	Set/report thermal threshold, mitigation threshold, and hysteresis	267
!THERMTIMERS	Set/report thermal mitigation algorithm timer details	268

Command reference

Table 13-2: Thermal mitigation command details

Command	Description
!MAXPWR Supporting chipsets: • MDM6200 (min f/w rev: P1_0_0_8) • MDM9200 • MDM9600	**Set/report maximum Tx power** Set or report the maximum Tx power for a specific band. **Caution:** *Any adjustments of Tx power may impact regulatory certification of the module in the host platform. The OEM is responsible for ensuring that the final module configuration in the host platform meets all regulatory requirements.* *Note: Increasing the Tx power affects the module's current consumption and thermal performance.* **Usage:** • Execution: **AT!MAXPWR=<band>,<tech>,<maxpwr>** Response: OK Purpose: Set the maximum Tx power for the specified band/technology combination. • Query: **AT!MAXPWR?<band>,<tech>** Response: <maxpwr> dBm OK Purpose: Indicate the maximum Tx power for the specified band/technology combination. • Query list: **AT!MAXPWR=?** Purpose: Display valid execution format and parameter values. **Parameters:** <band> (RF band) • 3GPP band number. For a full listing of 3GPP band numbers, see Table 4-2 on page 280. • Band support is product specific—see the device's Product Specification or Product Technical Specification document for details. • Valid range: 0–43. <tech> (Network technology) • 0=WCDMA • 1=CDMA (Note: not supported by MDM6200) • 2=LTE (Note: not supported by MDM6200) <maxpwr> (Maximum Tx power in dBm) • Integer value, ASCII format

Table 13-2: Thermal mitigation command details (Continued)

Command	Description

!SARBACKOFF
Supporting chipsets:
• MDM6200 (min f/w rev: P1_0_0_8)
• MDM9200
• MDM9600
	Set/report maximum Tx power limit
	Set or report the maximum Tx power limit for a specific band/technology/state combination.
	Usage:
	• Execution (WCDMA/CDMA/LTE): **AT!SARBACKOFF=<tech>,<band>,<state>,<offset>**
	Execution (GSM): **AT!SARBACKOFF=<tech>,<band>,<slot>,<state>,<modulation>,<offset>**
	Response: OK
	Purpose: Set the SAR backoff values for a specific band/technology combination.
	• Query: **AT!SARBACKOFF?<tech>,<band>,<state>**
	Response: <backoff> dBm
	OK
	Purpose: Indicate the maximum Tx power limit for the specified band/technology/state combination.
	• Query list: **AT!SARBACKOFF=?<tech>**
	Purpose: Display the execution and query formats with valid parameter values for the requested technology.
	Parameters:
	<tech> (Network technology)
	• 0=WCDMA
	• 1=CDMA (Note: Not supported by MDM6200)
	• 2=LTE (Note: Not supported by MDM6200)
	• 3=GSM
	<band> (RF band)
	• 3GPP band number. For a full listing of 3GPP band numbers, see Table 4-2 on page 280.
	• Band support is product specific—see the device's Product Specification or Product Technical Specification document for details.
	• Valid range: 0–40
	<state> (SAR backoff state setting)
	• 0=No backoff
	• 1–8=Backoff state 1 to 8
	<offset> (Offset from maximum Tx power, in dBm)
	• Range is technology-dependent. Use Query list command format to display valid values.
	• Values may be integer or decimal (for example, "4", "6.8")
	• Valid values—Execute the Query List command format to view valid values.
	  • (MDM6200) 0–81 (WCDMA); 0–8 (GSM)
	<slot> (GSM only—Tx slot (GPRS/EDGE))
	• Valid values:
	  • (MDM6200) 1–4
	  • (All others) 1–5
	<modulation> (GSM only—Modulation method)
	• 0=GMSK (GPRS)
	• 1=8PSK (EDGE)

Table 13-2: Thermal mitigation command details (Continued)

Command	Description
!SARSTATE Supporting chipsets: • MDM6200 (min f/w rev: P1_0_0_8) • MDM9200 • MDM9600	**Set/report SAR backoff state** Set or report the current SAR (Specific Absorption Rate) backoff state. *Note: This is a non-persistent setting. Use !SARSTATEDFLT to change the default backoff state.* **Usage:** • Execution: **AT!SARSTATE=<state>** Response: OK Purpose: Set the SAR backoff state. • Query: **AT!SARSTATE?** Response: !SARSTATE: <state> OK Purpose: Indicate the current <state> setting. • Query list: **AT!SARSTATE=?** Purpose: Display valid execution format and parameter values. **Parameters:** <state> (SAR backoff state setting) • 0=No backoff • 1–8=Backoff state 1 to 8
!SARSTATEDFLT Supporting chipsets: • MDM6200 (min f/w rev: P1_0_0_8) • MDM9200 • MDM9600	**Set/report default SAR backoff state** Set or report the default SAR (Specific Absorption Rate) backoff state used when the device powers up. By choosing an appropriate default backoff state, the device is made to start in a low-exposure state, which is important if there is any delay in the host proximity or position detection measurement and control algorithms. *Note: To temporarily change the SAR backoff state, use !SARSTATE. The change will last until the command is repeated or the modem resets.* **Usage:** • Execution: **AT!SARSTATEDFLT=<state>** Response: OK Purpose: Set the default SAR backoff state. • Query: **AT!SARSTATEDFLT?** Response: <state> OK *or* ERROR Purpose: Indicate the default state. • Query list: **AT!SARSTATEDFLT=?** Purpose: Display valid execution format and parameter values. **Parameters:** <state> (Default SAR backoff state setting) • 0=No backoff • 1–8=Backoff state 1 to 8

Table 13-2: Thermal mitigation command details (Continued)

Command	Description
!THERMCONFIG Supporting chipsets: • MDM8200 • MDM8220 (min f/w rev: N2.0 Release 6) • MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6)	**Set/report thermal mitigation configuration options** Set or report thermal mitigation configuration options. **Usage:** • Execution (MDM8200 only): **AT!THERMCONFIG=<MTP enable>, <RxD enable>, <QICE enable>** Execution (MDM8220/MDM9200 only): **AT!THERMCONFIG=<MTP enable>, <RxD enable>, <QICE enable>, <DL RLC throttling enable>, <UL RLC throttling enable>** Response: OK Purpose: Set thermal configuration options. • Query: **AT!THERMCONFIG?** Response: Thermal mitigation configuration: Adjust maximum Tx power (0-1) <MTP enable>-Enabled *(or Disabled)* Disable Rx Diversity (0-1) <RxD enable>-Enabled *(or Disabled)* Disable Q-ICE at mitigation state (0-1) <QICE enable>-Enabled *(or Disabled)* DL RLC throttling control (0-1) <DL RLC throttling enable>-Enabled *(or Disabled)* *(MDM8220/MDM9200 only)* UL RLC throttling control (0-1) <UL RLC throttling enable>-Enabled *(or Disabled)* *(MDM8220/MDM9200 only)* OK Purpose: Display current configuration options. • Query List: **AT!THERMCONFIG=?** Purpose: Display the execution command format and parameter values. **Parameters:** <MTP enable> (Maximum Tx power adjustment feature) • 0=Disabled—adjustments not allowed • 1=Enabled (Default)—adjustments allowed <RxD enable> (Rx diversity disabling feature) • 0=Disabled (Default)—Rx diversity cannot be disabled • 1=Enabled—Rx diversity can be disabled <QICE enable> (Q-ICE disabling feature) • 0=Disabled (Default)—Q-ICE cannot be disabled • 1=Enabled—Q-ICE can be disabled <DL RLC throttling enable> (Enable/disable downlink RLC throttling) • 0=Disabled (Default) • 1=Enabled <UL RLC throttling enable> (Enable/disable uplink RLC throttling) • 0=Disabled (Default) • 1=Enabled

Table 13-2: Thermal mitigation command details (Continued)

Command	Description
!THERMDELTATX Supporting chipsets: • MDM8200 • MDM8220 (min f/w rev: N2.0 Release 6) • MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6)	**Set/report amount to reduce maximum Tx power** Set or report the amount of power to subtract from maximum Tx power when the device is in mitigation state. The Tx power will repeatedly be stepped-down by this amount as long as the temperature remains above the mitigation threshold. See !THERMTIMERS on page 268 for details. **Usage:** • Execution (MDM8200 only): **AT!THERMDELTATX=<delta tx>** Execution (MDM8220/MDM9200 only): **AT!THERMDELTATX=<delta tx>[, <max backoff>, <time for non backoff>, <time for backoff>, <step timer>]** Response: OK Purpose: Set the amount of power to subtract from maximum Tx power. • Query: **AT!THERMDELTATX?** Response (MDM8200 only): Delta Tx power (1/12 dB)=<delta tx> OK Response (MDM8220/MDM9200 only): Delta Tx power (dB)=<delta tx> Max backoff (dB) = <max backoff> time for non-backoff value of power (ms) = <time for non backoff> time for backoff value of power (ms) = <time for backoff> timer for each step of backoff (ms) = <step timer> OK Purpose: Display the amount to subtract from maximum Tx power, and, depending on chipset, backoff time values. • Query List: **AT!THERMDELTATX=?** Purpose: Display the execution command format and parameter values. **Parameters:** <delta tx> (Power subtracted from maximum Tx power) • Units: • MDM8200—1/12 dB • MDM8220/MDM9200—dB • Valid range: • MDM8200—0–120 (corresponds to 0–10 dB) • MDM8220/MDM9200—0–<max backoff> <max backoff> (MDM8220/MDM9200 only—Maximum value that can be used to set <delta tx>) • Units: dB <time for non-backoff> (MDM8220/MDM9200 only—Length of time that device runs at regular power level before switching to the reduced level of the current backoff step.) • Units: ms <time for backoff> (MDM8220/MDM9200 only—Length of time that device runs at the reduced power level of the current backoff step, before switching to regular power level.) • Units: ms <step timer> (MDM8220/MDM9200 only—Length of time for each backoff step.) • Units: ms

Table 13-2: Thermal mitigation command details (Continued)

Command	Description
!THERMDELTATXTEMP Supporting chipsets: • MDM8220 (min f/w rev: N2.0 Release 6) • MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6)	**Set/report amount power backoff temperature threshold** Set or report the temperature above which the second stage of thermal mitigation begins. (See !THERMTHRESHOLD on page 267 for the first stage of thermal mitigation.) These chipsets support a two-stage thermal mitigation phase. In stage 1, data throughput flow control is used. If this is unsuccessful and the temperature rises above the power backoff temperature threshold, the second stage of mitigation begins and the Tx power backoff algorithm is used. (See !THERMTIMERS on page 268 for details.) **Usage:** • Execution: **AT!THERMDELTATXTEMP=<Tx backoff threshold>** Response: OK *or* ERROR *(if value is not in range)* Purpose: Set the temperature at which stage 2 thermal mitigation occurs. • Query: **AT!THERMDELTATXTEMP?** Response: at!thermdeltatxtemp? Delta Tx power temp=<Tx backoff threshold> OK Purpose: Display the current stage 2 threshold temperature. • Query List: **AT!THERMDELTATXTEMP=?** Purpose: Display the execution command format and parameter values. **Parameters:** <Tx backoff threshold> (Temperature above which stage 2 mitigation (Tx power backoff state) occurs) • Units: °C • Valid range: Value between mitigation threshold and emergency threshold (see !THERMTHRESHOLD on page 267 to identify threshold values)
!THERMENABLE Supporting chipsets: • MDM8200 • MDM8220 (min f/w rev: N2.0 Release 6) • MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6)	**Enable/disable thermal mitigation** Enable/disable the thermal mitigation feature. **Usage:** • Execution: **AT!THERMENABLE=<enable>** Response: OK Purpose: Enable/disable thermal mitigation. • Query: **AT!THERMENABLE?** Response: Thermal mitigation version=1 Thermal mitigation enable (0-1) <enable>-Enable *(or Disable)* OK Purpose: Show the current state of the thermal mitigation feature. • Query List: **AT!THERMENABLE=?** Purpose: Display the execution command format and parameter values. **Parameters:** <enable> (Thermal mitigation feature state) • 0=Disable (Default) • 1=Enable

Table 13-2: Thermal mitigation command details (Continued)

Command	Description
!THERMINFO Supporting chipsets: • MDM8200 • MDM8220 (min f/w rev: N2.0 Release 6) • MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6) *Note: This command is not password-protected.*	**Display thermal mitigation information** Display all thermal mitigation-related information. (This returns the same information that is returned using the Query formats of the other thermal mitigation commands. See those commands for parameter definitions.) **Usage:** • Execution: **AT!THERMINFO** Response: Mitigation threshold=*xx* *(See !THERMTHRESHOLD on page 267)* Emergency threshold=*xx* Hysteresis=*xx* Basic loop(ms)=*xxxx* *(See !THERMTIMERS on page 268)* Mitigation loop(ms)=*xxxxx* T_down(ms)=*xxxxx* Misc time(ms)=*xxxxx* Pa_change_holdoff=*xx* Delta Tx power(1/12 dB)=*xx* *(MDM8200 only)* *(See !THERM-DELTATX on page 264)* Delta Tx power(dB)=*xx* *(MDM8220/MDM9200 only)* Max backoff (dB) = *xx* *(MDM8220/MDM9200 only)* time for non-backoff value of power (ms) = *xxxxx* *(MDM8220/MDM9200 only)* time for backoff value of power (ms) = *xxxxx* *(MDM8220/MDM9200 only)* timer for each step of backoff (ms) = *xxxxx* *(MDM8220/MDM9200 only)* Thermal mitigation version=1 *(See !THERMENABLE on page 265)* Thermal mitigation enable (0-1) 1-Enabled *(or 0-Disabled)* Thermal mitigation configuration: *(See !THERMCONFIG on page 263)* Adjust maximum Tx power (0-1) 1-Enabled *(or 0-Disabled)* Disable Rx Diversity (0-1) 0-Disabled *(or 1-Enabled)* Disable Q-ICE at mitigation state (0-1) 0-Disabled DL RLC throttling control (0-1) 0-Disabled *(or 1-Enabled)* *(MDM8220/MDM9200 only)* UL RLC throttling control (0-1) 0-Disabled *(or 1-Enabled)* *(MDM8220/MDM9200 only)* Delta Tx power temp = *xx* *(MDM8220/MDM9200 only)* OK Purpose: Display all thermal mitigation-related information. **Parameters:** See the Parameter sections for the thermal mitigation commands referenced in the Response format.

Table 13-2: Thermal mitigation command details (Continued)

Command	Description

!THERMTHRESHOLD
Supporting chipsets:
• MDM8200
• MDM8220 (min f/w rev: N2.0 Release 6)
• MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6)
	Set/report thermal threshold, mitigation threshold, and hysteresis
	Set/report the threshold values for entering mitigation and emergency states, and the hysteresis value for causing a state change:
	• Mitigation state
	  • MDM8200—Tx power is reduced.
	  • MDM8220/MDM9200—First stage of mitgation (data throughput flow control) occurs. If temperature continues to rise past the second stage mitigation threshold (see !THERMDELTATXTEMP on page 265), Tx power is reduced.
	• Emergency state—Tx power is stopped.
	• Hysteresis—Temperature reduction required before moving from emergency to mitigation state, or from mitigation to normal operating state. For example, if the device is in mitigation state, it will not return to normal state until the temperature drops below (<threshold1> - <hysteresis>).
	Usage:
	• Execution: **AT!THERMTHRESHOLD=<threshold1>,<threshold2>,<hysterisis>**
	Response: OK
	Purpose: Set the threshold and hysteresis values. (All three values must be specified.)
	• Query: **AT!THERMTHRESHOLD?**
	Response: Mitigation threshold=<threshold1>
	Emergency threshold=<threshold2>
	Hysteresis=<hysteresis>
	OK
	Purpose: Show current threshold and hysteresis values.
	• Query List: **AT!THERMTHRESHOLD=?**
	Purpose: Display the execution command format and parameter values.
	Parameters:
	<threshold1> (Threshold to enter mitigation state)
	• 0–255 (degrees Celsius)
	<threshold2> (Threshold to enter emergency state)
	• 0–255 (degrees Celsius)
	<hysteresis> (Temperature reduction below threshold required to return to normal state (from mitigation) or mitigation (from emergency))
	• 0–255 (degrees Celsius)

Table 13-2: Thermal mitigation command details (Continued)

Command	Description

!THERMTIMERS
Supporting chipsets:
• MDM8200
• MDM8220 (min f/w rev: N2.0 Release 6)
• MDM9200 (min f/w rev: M1.0 Release 3, X3.0 Beta 6)
	Set/report thermal mitigation algorithm timer details
	Set/report the timers and related values used in the thermal mitigation algorithm.
	Usage:
	• Execution: **AT!THERMTIMERS=<basic loop>,<mitigation loop>,<T_down>, <Misc time>,<Pa_change_holdoff>**
	Response: OK
	Purpose: Set the timers and related values for the thermal mitigation algorithm.
	• Query: **AT!THERMTIMERS?**
	Response: Basic loop(ms)=<basic loop>
	Mitigation loop(ms)=<mitigation loop>
	T_down(ms)=<T_down>
	Misc time(ms)=<Misc time>
	Pa_change_holdoff=<Pa_change_holdoff>
	OK
	Purpose: Show the timers and related values for the thermal mitigation algorithm.
	• Query List: **AT!THERMTIMERS=?**
	Purpose: Display the execution command format and parameter values.
	Parameters:
	<basic loop> (Normal state—time period to wait between temperature measurements)
	• Units—ms
	<mitigation loop> (Mitigate and emergency states—time period to wait between temperature measurements)
	• Units—ms
	<T_down> (Mitigation state timer—time period to use reduced Tx power during mitigation state)
	• Units—ms
	<Misc time> (Mitigation state timer—time period to use max Tx power during mitigation state)
	• Units—ms
	<Pa_change_h> (Number of mitigation loops between Tx power reductions)
	• 0–255
	• Example: If Max Tx power is 25 dB, the delta Tx power (amount to reduce Tx power) is 3 dB, and <Pa_change_h> is 4, then:
	  • When device enters mitigation state from normal state, Tx power decreases to 22 dB.
	  • After <Pa_change_h> * <mitigation loop> ms, if the temperature is still higher than <threshold1>, the Tx power reduces to 19 dB.
	Note: To set or report the delta Tx power value, see !THERMDELTATX on page 264.

A: Module Customization

Various modem features can be customized to meet individual carrier or OEM needs.

The following procedure describes how to prepare the modem for customization, maintain profiles, and customize various features.

Note: Some customizations described in this section may only be available on specific modules (for example, GPS customizations) or for specific minimum firmware versions. Also, this is only a representative sample of available customizations—all AT commands can be used for customizations depending on carrier requirements.

Note: Contact your Sierra Wireless account representative to get the password.

(Prepare the modem for customization)

1. Unlock the command to use extended commands:
   AT!ENTERCND="<password>" (page 28)
2. Put the modem into offline mode to perform customizations:
   AT!DAOFFLINE (page 109)
3. Back up the modem's RF calibration data as a precaution:
   AT!NVBACKUP=0 (See *UMTS Modems Supported AT Command Reference, Document 2130617*)

(Maintain Profiles)

4. If the modem was provisioned with profiles that you do not want to keep, you can remove them. To remove (clear) a profile (profile number <pid>):
   a. Mark the profile as non write-protected:
      AT!SCROPROF=<pid>,0 (page 74)
   b. Set the profile for manual activation, don't prompt for password, don't auto-launch application, and disable the PDP linger timer:
      AT!SCPROF=<pid>,"",0,0,0,0 (See *UMTS Modems Supported AT Command Reference, Document 2130617*)
   c. Initialize the primary and secondary DNS addresses:
      AT!SCDNS=<pid>,"0.0.0.0","0.0.0.0" (See *UMTS Modems Supported AT Command Reference, Document 2130617*)
   d. Clear PDP-IP connection authentication parameters:
      AT$QCPDPP=<pid> (page 89)

*Note: Make sure a SIM is inserted in the module before using the **+CGTFT** command.*

   e. Set all packet filters in the Traffic Flow Template as undefined:
      AT+CGTFT=<pid> (See *3GPP TS 27.007*)
   f. Mark this Quality of Service Profile as undefined:
      AT+CGQREQ=<pid> (See *3GPP TS 27.007*)
   g. Mark the minimum acceptable Quality of Service profile as undefined:
      AT+CGQMIN=<pid> (See *3GPP TS 27.007*)

h. Mark the 3G Quality of Service profile as undefined:
AT+CGEQREQ=<pid> (See *3GPP TS 27.007*)

i. Mark the minimum acceptable 3G Quality of Service profile as undefined:
AT+CGEQMIN=<pid> (See *3GPP TS 27.007*)

j. Mark PDP context parameter values as undefined:
AT+CGDCONT=<pid> (See *3GPP TS 27.007*)

k. Initialize the software option byte in the profile details:
AT!SCPROFSWOPT=<pid>,0 (page 74)

(Customize configuration details)

5. If desired, assign a custom PRI part number or revision: **AT!PRIID=<priPn>, <priRev>** (page 71)
6. Clear all old customizations from non-volatile memory and restore factory defaults:
AT!NVDEF (page 126)
AT!NVRESTORE=0 (page 126)
7. If desired, customize non-MUX mode port mappings:
AT!NVPORTMAP=<normMode> [, <diagMode>] (page 65)
8. Set GPRS MS Class and EDGE MS Class to match network requirements (default is Class 10, example below switches to Class 12):
AT!NVOEM=GMSCLASS,0C (page 63)
AT!NVOEM=EMSCLASS,0C (page 63)

Note: Most carriers enable both ciphering and integrity (<setting> = 2).

9. Match the modem's ciphering and integrity settings to the live UMTS network's settings:
AT!GCIPHER=<setting> (See *UMTS Modems Supported AT Command Reference, Document 2130617*)
10. Enable GERAN FP #1 functionality if required by carrier:
AT!NVOEM=GERANFP1,01 (page 63)
11. Configure the modem to use a specific service domain (circuit-switched, packet-switched, both):
AT!SELMODE=<sdInd> (See *UMTS Modems Supported AT Command Reference, Document 2130617*)
where <sdInd> = 0 (CS), 1 (PS), or 2 (both)

*Note: The **!BAND** command shows only the bands of the current band group (0 or 1).*

12. Choose an appropriate region-specific scanning algorithm:
AT!CUSTOM="PRLREGION", <value> (page 40).
13. Set carrier-specific configuration values.The following are examples of some parameters that you may wish to customize:
- Enable/disable ENS functionality:
AT!ENSEN=<value>
where <value> = 0 (disable) or 1 (enable)
- Enable/disable A5/2 encryption:
AT!NVOEM=GSMA5ALG,<value> (page 63)
- Enable/disable FDN check for PS data calls:
AT!CUSTOM="DISFDNPDPCHK",<value> (page 40)

Note: The customizations in this step affect the interfaces between the modem and the user (drivers, API, CnS, Watcher)—they do not affect the modem directly.

14. If desired, configure Watcher 3G and Windows driver functionality:

a. Enable/disable roaming indicator display:
AT!CUSTOM="NOROAM", <value> (page 40)
(If enabled, the firmware will always report "No Roam" to the API in the CNS_ICON_ROAM object.)

b. Enable/disable GPRS/EDGE indicator display:
AT!CUSTOM="NOGPRS", <value> (page 40)
(If enabled, the firmware clears the display icon flag in the CNS_SERVICE_ICON object.)

c. Enable/disable STK UI:
AT!CUSTOM="STKUIEN", <value> (page 40)
(If enabled for CnS interface, CNS_STK_CMD notifications are sent to the API/Watcher.)

d. Enable/disable GPRS attach on power-up using the !SELMODE command:
AT!SELMODE=<value>
(<value> = 0 (disable — use CS only)
<value> = 1 or 2 (enable — use PS only, or CS and PS))
(See *UMTS Modems Supported AT Command Reference, Document 2130617*)

e. Enable/disable CSD setting from Watcher, and prevent MO CSD call from being set up using AT commands:
AT!CUSTOM="CSDOFF", <value> (page 40)

f. Enable/disable prompt for SIM's PUK code when CHV1 is blocked (the prompt is always forced when CHV2 is blocked):
AT!CUSTOM="PUKPRMPT", <value> (page 40)
(If disabled, the firmware sets required operation to "No Operation" in the CNS_SIM_STATUS_EXP object. If enabled, the required operation is set to "Enter PUK". In either case, the SIM status in the object is set as PIN 1 blocked.)

g. Enable/disable voice support
AT!CUSTOM="ISVOICEN", <value> (page 40)
(If disabled (<value> = 0 or 2), the CNS_AVAILABLE_FEATURES object is set as "no voice", and headset indications and other voice-related CnS notifications are not sent to the host. If enabled (<value> =1), the object is set as "voice", and headset indications and other voice-related CnS notifications are sent to the host. If <value> is 0 or 1, voice calls can be made using the AT command interface.)

h. Enable /disable GPS — causes Windows driver to open an NMEA port:
AT!CUSTOM="GPSENABLE", <value> (page 40)

Note: The host device needs to include a facility for restoring these settings after the modem power cycles.

15. Customize non-persistent modem settings.

a. Set the band using one of the following commands:
- **AT!BAND = <bandsetInd>** (*See UMTS Modems Supported AT Command Reference, Document 2130617*)
- **AT!GBAND = <bandmask>** (page 88)

*Note: **!GBAND** allows for finer tuning than **!BAND**.*

b. Indicate the communications protocol to be used:
AT!REL=<wcdmarrc>[, <sgsnr>, <mscr>] (page 72)
(Example: HSDPA is !REL=1,1,1; HSUPA is !REL=2,1,1)

16. Define up to sixteen profiles (profile number <pid>):

a. Indicate the PDP context:
AT+CGDCONT = <pid>, <pdp_type>, <apn> [, <ipaddr>] (See *3GPP TS 27.007*)

b. Indicate the user authentication method to use:
AT$QCPDPP = <pid>, <auth_type>, <password>, <username> (page 89)

c. Define the profile:
AT!SCPROF=<pid>, <label>, <autoconn>, <promptpwd>, <autolaunchapp>, <rffu> (See *UMTS Modems Supported AT Command Reference, Document 2130617*)

d. (optional) Indicate requested Quality of Service profile:
AT+CGQREQ=<pid> (See *3GPP TS 27.007*)

e. (optional) Indicate minimum acceptable Quality of Service profile:
AT+CGQMIN=<pid> (See *3GPP TS 27.007*)

f. (optional) Indicate requested 3G Quality of Service Profile:
AT+CGEQREQ=<pid>, <traffic class> (See *3GPP TS 27.007*)

g. (optional) Indicate minimum acceptable 3G Quality of Service Profile:
AT+CGEQMIN=<pid> (See *3GPP TS 27.007*)

h. Indicate if profile is to be read-only:
AT!SCROPROF=<readonly>, <pid> (page 74)

17. Identify the default profile:
AT!SCDFTPROF=<pid> (See *UMTS Modems Supported AT Command Reference, Document 2130617*)

18. Enable or disable scan profile (try all profiles configured on card until a successful connection is found):
AT!CUSTOM="SCANPROF",<value> (page 40)

19. Enable/disable slow clocking mode:
AT!SLEEP = <state> (page 78)

20. Clear the error log:
AT!ERR=0 (page 87)

21. Back up data:

Note: Profiles (PDP contexts) are not restored from these backups. Host application software is responsible for this task if profiles become corrupted.

a. Back up provisioning data:
AT!NVBACKUP = 1 (See *UMTS Modems Supported AT Command Reference, Document 2130617*)

b. Back up user data:
AT!NVBACKUP = 2 (See *UMTS Modems Supported AT Command Reference, Document 2130617*)

c. Backup RF calibration data:
AT!NVBACKUP = 0 (See *UMTS Modems Supported AT Command Reference, Document 2130617*)

22. Set up MEP/MCC/MNC:
    a. Enable/disable prompt for MEP code when incorrect SIM is inserted:
    AT!CUSTOM="MEPCODE",<value> (page 40)
    b. Set MEP locking status (Only enable if MEP code prompt is also enabled.)
    AT!CUSTOM="MEPLOCK",<value> (page 40)

Note: <mcc> is three digits in North America, or two digits elsewhere.

    c. If MEP locking is enabled:
        i. Populate the PLMN list with up to sixty-four MCC/MNC pairs. Up to six pairs can be added at a time using the following command:
        AT!NVPLMN=<mcc1>, <mnc1>, ... , <mccn>, <mncn> (page 64)

Note: The OEM must provide a random and unique MEP lock code. This is used by the end-user to unlock the modem.

        ii. Lock the PLMN list:
        AT+CLCK="PN", 1 [, <password> [, <class>]] (See *3GPP TS 27.007*)

B: Test Frequencies/Channel Lists

B

The following tables describe the frequencies and channels typically used when checking receiver and transmit paths for Sierra Wireless AirPrime MC8xxx intelligent embedded modules.

Table B-1: GSM/EDGE channels[a]

Band	Channel	Frequency (MHz)	
		Tx	Rx
GSM850	128	824.20	869.20
	190	836.60	881.60
	251	848.80	893.80
EGSM900	975	880.20	925.20
	979	881.00	926.00
	62	902.40	947.40
	65	903.00	948.00
	120	914.00	959.00
	124	914.80	959.80
DCS1800	512	1710.20	1805.20
	520	1711.80	1806.80
	697	1747.20	1842.20
	880	1783.80	1878.80
	885	1784.80	1879.80
PCS1900	512	1850.20	1930.20
	520	1851.80	1931.80
	661	1880.00	1960.00
	804	1908.60	1988.60
	810	1909.80	1989.80

a. When testing, use the Rx frequencies plus a 67 kHz offset. For example, to test GSM850, channel 190, use a signal generator setting of 881.667 MHz

Table B-2: WCDMA Channels[a]

Band	Tx Channel	UE Tx (MHz)	Rx Channel	UE Rx (MHz)
UMTS Band I (2100)	9612	1922.40	10562	2112.40
	9750	1950.00	10700	2140.00
	9888	1977.60	10838	2167.60
UMTS Band II (1900)	9262	1852.40	9662	1932.40
	9400	1880.00	9800	1960.00
	9538	1907.60	9938	1987.60
UMTS Band V (850)	4132	826.40	4357	871.40
	4182	836.40	4407	881.40
	4233	846.60	4458	891.60
UMTS Band VIII (900)	2712	882.40	2937	927.40
	2787	897.40	3012	942.40
	2863	912.60	3088	957.60

a. When testing, use the UE Rx frequencies plus a 1.2 MHz offset. For example, to test Band V (850 MHz), channel 4407, use a signal generator setting of 882.60 MHz.

C: HSDPA/ HSUPA Categories

The following tables describe standard HSDPA and HSUPA categories.

Table C-1: HSDPA-capable terminals

Category	Maximum number of supported HS-DSCH codes	Minimum inter-TTI interval	Number of soft values in terminal's hybrid ARQ buffer	Theoretical download maximum (L1 peak rate [Mbps])	Modulation
Category 1	5	3	19,200	1.2	16QAM, QPSK
Category 2	5	3	28,800	1.2	16QAM, QPSK
Category 3	5	2	28,800	1.8	16QAM, QPSK
Category 4	5	2	38,400	1.8	16QAM, QPSK
Category 5	5	1	57,600	3.6	16QAM, QPSK
Category 6	5	1	67,200	3.6	16QAM, QPSK
Category 7	10	1	115,200	7.2	16QAM, QPSK
Category 8[a]	10	1	134,400	7.2	16QAM, QPSK
Category 9	15	1	172,800	10.0	16QAM, QPSK
Category 10	15	1	172,800	14.0	16QAM, QPSK
Category 11	5	2	14,400	0.9	QPSK
Category 12	5	1	28,800	1.8	QPSK

Table C-2: HSUPA-capable terminals

E-DCH Category	Maximum number of E-DCH codes transmitted	Minimum spreading factor	Support for 10 ms; 2 ms TTI E-DCH	Maximum data rate with 10 ms TTI	Maximum data rate with 2 ms TTI
Category 1	1	SF4	10 ms only	0.72 Mbps	N/A
Category 2	2	SF4	10 ms and 2 ms	1.45 Mbps	1.45 Mbps
Category 3	2	SF4	10 ms only	1.45 Mbps	N/A
Category 4	2	SF2	10 ms and 2 ms	2.0 Mbps	2.91 Mbps
Category 5	2	SF2	10 ms only	2.0 Mbps	N/A
Category 6	4	SF2	10 ms and 2 ms	2.0 Mbps	5.76 Mbps

D: Band Definitions

Some commands described in this document include input and/or output ‘band’ parameters, where the band value is one of the following:

- An enumerated value representing a network technology and band (Table D-1)
- A 3GPP band number (Table 4-2 on page 280)

Note: Band support is product-specific—see the device’s Product Specification Document or Product Technical Specification for details.

Table D-1: Band/technology enumerations[a]

<band>	Description	<band>	Description	<band>	Description	<band>	Description
0	CDMA	22	WCDMA 800	42	LTE B4	60	LTE B24
2	Sleep	25	WCDMA BC3	43	LTE B2	61	LTE B25
5	CDMA 800	26	CDMA BC14	44	LTE B3	62	LTE B26
6	CDMA 1900	27	CDMA BC11	45	LTE B5	63	LTE B27
7	HDR	28	WCDMA BC4	46	LTE B6	64	LTE B28
8	CDMA 1800	29	WCDMA BC8	47	LTE B8	65	LTE B29
9	WCDMA IMT	30	MF 700	48	LTE B9	66	LTE B30
10	GSM 900	31	WCDMA BC9	49	LTE B10	67	LTE B31
11	GSM 1800	32	CDMA BC15	50	LTE B12	68	LTE B32
12	GSM 1900	33	CDMA BC10	51	LTE B14	69	LTE B33
14	JCDMA	34	LTE B1	52	LTE B15	70	LTE B34
15	WCDMA 1900A	35	LTE B7	53	LTE B16	71	LTE B35
16	WCDMA 1900B	36	LTE B13	54	LTE B18	72	LTE B36
17	CDMA 450	37	LTE B17	55	LTE B19	73	LTE B37
18	GSM 850	38	LTE B38	56	LTE B20	74	LTE B39
19	IMT	39	LTE B40	57	LTE B21	75	WCDMA BC19
20	HDR 800	40	WCDMA BC11	58	LTE B22	76	LTE B41
21	HDR 1900	41	LTE B11	59	LTE B23		

a. Band values not listed (e.g. 1, 3, 4) are reserved.

Table 4-2: 3GPP bands

Band	Frequency bands (MHz)		Band	Frequency bands (MHz)	
	Rx	Tx		Rx	Tx
1	1920–1980	2110–2170	**20**	832–862	791–821
2	1850–1910	1930–1990	**21**	1447.9–1462.9	1495.9–1510.9
3	1710–1785	1805–1880	**22**	Reserved	Reserved
4	1710–1755	2110–2155	**23**	2000–2020	2180–2200
5	824–849	869–894	**24**	1626.5–1660.5	1525–1559
6	830–840	875–885	**25**	1850–1915	1930–1995
7	2500–2570	2620–2690	**26–32**	Reserved	Reserved
8	880–915	925–960	**33**	1900–1920	1900–1920
9	1749.9–1784.9	1844.9–1879.9	**34**	2010–2025	2010–2025
10	1710–1770	2110–2170	**35**	1850–1910	1850–1910
11	1427.9–1447.9	1475.9–1495.9	**36**	1930–1990	1930–1990
12	699–716	729–746	**37**	1910–1930	1910–1930
13	777–787	746–756	**38**	2570–2620	2570–2620
14	788–798	758–768	**39**	1880–1920	1880–1920
15	Reserved	Reserved	**40**	2300–2400	2300–2400
16	Reserved	Reserved	**41**	2496–2690	2496–2690
17	704–716	734–746	**42**	3400–3600	3400–3600
18	815–830	860–875	**43**	3600–3800	3600–3800
19	830–845	875–890			

E: ASCII Table

Table E-1: ASCII values

Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex
NUL	0	00	SP	32	20	@	64	40	‘	96	60
SOH	1	01	!	33	21	A	65	41	a	97	61
STX	2	02	“	34	22	B	66	42	b	98	62
ETX	3	03	#	35	23	C	67	43	c	99	63
EOT	4	04	$	36	24	D	68	44	d	100	94
ENQ	5	05	%	37	25	E	69	45	e	101	95
ACK	6	06	&	38	26	F	70	46	f	102	96
BEL	7	07	’	39	27	G	71	47	g	103	97
BS	8	08	(	40	28	H	72	48	h	104	98
HT	9	09	)	41	29	I	73	49	i	105	99
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	-	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
XON	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
XOFF	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B	{	123	7B
FS	28	1C	<	60	3C	\	92	5C	|	124	7C
GS	29	1D	=	61	3D	]	93	5D	}	125	7D
RS	30	1E	>	62	3E	^	94	5E	~	126	7E
US	31	1F	?	63	3F	_	95	5F	DEL	127	7F

Index (AT commands)

D

E

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

W

Index

Symbols

A

B

C

D

E

F

G

H

O

P

Q

R

S

T

U

V

W


SIERRA
WIRELESS