

# SimpleLink™ Wi-Fi® AT Command User's Guide

The SimpleLink™ Wi-Fi® Internet-on-a chip™ family of devices from Texas Instruments™ provides a suite of integrated protocols for Wi-Fi and internet connectivity to dramatically simplify the implementation of internet-enabled devices and applications.

This document describes the AT command protocol for SimpleLink, which is a widely used method to configure and control embedded networking systems due to its simplicity, textual parameter representation, and inherent flexibility.

## Contents

1	Supported Platforms .....	3
2	Architecture Overview .....	3
3	Getting Started .....	3
4	Commands Summary .....	4
5	Protocol Syntax .....	6
6	Command Description.....	7

## List of Tables

1	Device Commands .....	4
2	Socket Commands .....	4
3	WLAN Commands.....	4
4	File System Commands .....	5
5	Network Application Commands.....	5
6	Network Configuration Commands.....	5
7	Network Utility Commands .....	5
8	Asynchronous Events .....	6
9	<i>AT+Start</i> Starts the NWP .....	7
10	<i>AT+Stop</i> Stops the NWP .....	7
11	<i>AT+Get</i> Getting Device Configurations .....	8
12	<i>AT+Set</i> Setting Device Configurations.....	9
13	<i>AT+Test</i> Test Command.....	9
14	<i>AT+Socket</i> Create an End-Point for Communication .....	9
15	<i>AT+Close</i> Close Socket .....	9
16	<i>AT+Accept</i> Accept a Connection on a Socket .....	10
17	<i>AT+Bind</i> Assign a Name to a Socket.....	10
18	<i>AT+Listen</i> Listen for Connections on a Socket.....	10
19	<i>AT+Connect</i> Initiate a Connection on a Socket.....	10
20	<i>AT+Select</i> Monitor Socket Activity .....	11
21	<i>AT+SetSockOpt</i> Set Socket Options .....	12
22	<i>AT+GetSockOpt</i> Get Socket Options.....	14
23	<i>AT+Recv</i> Read Data From TCP Socket .....	14
24	<i>AT+RecvFrom</i> Read Data From Socket .....	15
25	<i>AT+Send</i> Write Data to TCP Socket .....	15
26	<i>AT+SendTo</i> Write Data to Socket .....	15

27	<i>AT+WlanConnect</i> Connect to WLAN Network as a Station .....	16
28	<i>AT+WlanDisconnect</i> Disconnect the Connection .....	16
29	<i>AT+WlanProfileAdd</i> Add Profile .....	17
30	<i>AT+WlanProfileGet</i> Get Profile .....	17
31	<i>AT+WlanProfileDel</i> Delete Profile .....	17
32	<i>AT+WlanPolicySet</i> Set Policy Values .....	18
33	<i>AT+WlanPolicyGet</i> Get Policy Values .....	19
34	<i>AT+WlanScan</i> Gets the WLAN Scan Operation Results .....	19
35	<i>AT+WlanProvisioning</i> Start Provisioning .....	20
36	<i>AT+WlanSetMode</i> WLAN Set Mode .....	20
37	<i>AT+WlanSet</i> Setting WLAN Configurations .....	21
38	<i>AT+WlanGet</i> Getting WLAN Configurations .....	23
39	<i>AT+FileOpen</i> Open File in Storage Device .....	25
40	<i>AT+FileClose</i> Close File in Storage Device .....	25
41	<i>AT+FileCtl</i> Controls Various File System Operations .....	25
42	<i>AT+FileDel</i> Delete File From Storage Device .....	26
43	<i>AT+FileGetFilelist</i> Get a List of Files .....	26
44	<i>AT+FileGetInfo</i> Get Information About a File .....	27
45	<i>AT+FileRead</i> Read a Block of Data From a File in Storage Device .....	27
46	<i>AT+FileWrite</i> Write Block of Data to a File in Storage Device .....	27
47	<i>AT+NetAPPStart</i> Starts a Network Application .....	28
48	<i>AT+NetAPPStop</i> Stops a Network Application .....	28
49	<i>AT+NetAPPGetHostByName</i> Get Host IP by Name .....	29
50	<i>AT+NetAPPGetHostByService</i> Get Host IP by Service .....	29
51	<i>AT+NetAPPSet</i> Setting Network Application Configurations .....	30
52	<i>AT+NetAPPGet</i> Getting Network Applications Configurations .....	32
53	<i>AT+NetAPPSend</i> Sends Network Application Response or Data Following a Network Application Request Event .....	32
54	<i>AT+NetAPPRecv</i> Receives Data From the Network Processor Following a Network Application Response Event .....	33
55	<i>AT+NetAPPPing</i> Send Ping to Network Hosts .....	33
56	<i>AT+NetAPPGetServiceList</i> Get Service List .....	34
57	<i>AT+NetAPPRegisterService</i> Register a New mDNS Service .....	34
58	<i>AT+NetAPPUnRegisterService</i> Unregister mDNS Service .....	34
59	<i>AT+NetCfgSet</i> Setting Network Configurations .....	35
60	<i>AT+NetCfgGet</i> Getting Network Configurations .....	36
61	<i>AT+NetUtilGet</i> Getting Utilities Configurations .....	38
62	<i>AT+NetUtilCmd</i> Performing Utilities-Related Commands .....	39
63	<i>+EventFatalError</i> Fatal Error Event for Inspecting Fatal Error .....	40
64	<i>+EventGeneral</i> General Asynchronous Event for Inspecting General Events .....	40
65	<i>+EventWlan</i> WLAN Asynchronous Event .....	40
66	<i>+EventNetApp</i> Network Application Asynchronous Event .....	41
67	<i>+EventSock</i> Socket Asynchronous Event .....	42

## Trademarks

SimpleLink, Internet-on-a-chip, Texas Instruments, SmartConfig are trademarks of Texas Instruments. Wi-Fi, Wi-Fi Direct are registered trademarks of Wi-Fi Alliance. All other trademarks are the property of their respective owners.

## 1 Supported Platforms

Hardware platforms that support the AT command library are:

- CC3220R
- CC3220S
- CC3220SF

## 2 Architecture Overview

SimpleLink Wi-Fi AT Command consists of two main modules:

- AT Commands Application
  - The application is one of the following application demos:
    - The AT\_Commands application provides control by the AT Commands on the local device.
    - The Serial\_wifi application provides control by the AT Commands on the local and the remote device.
    - The user-customized application is based on the two previous applications.
- AT Command Core
  - The core includes the command parser, execution, and return status.
  - The AT Command Core should already be compiled into the library.

The following API communicate between the two modules:

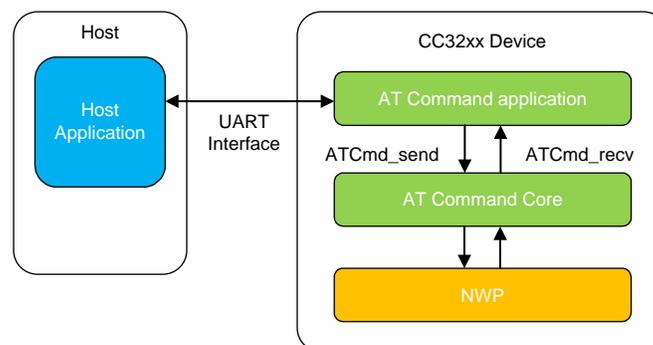
- *ATCmd\_create* creates the AT Command core task and initializes the RX event queue.
- *ATCmd\_send* transmits string from the AT Command application to the AT Command Core. The function takes one parameter, *Buffer*, which stores the sent string.
- *ATCmd\_rcv* transmits a string from the AT Command Core to the AT Command application.

The function takes two parameters:

- *Buffer* stores the received string.
- *Nonblock variant* set to 0 for *waits forever* on the RX queue, otherwise set to 1.

All send and receive buffers should be allocated by the AT Commands application.

Figure 1 shows the basic architecture.



Copyright © 2017, Texas Instruments Incorporated

**Figure 1. Basic Architecture Scheme**

## 3 Getting Started

The following describes the procedure to build the AT Command Core. For building and executing the application binary file, refer to the *README.html* file that is located in each AT Command application. Ensure that the AT Command library includes in the application linking list.

The AT Command Core is prebuilt into the library “atcmd.a” per two OS (TI-RTOS and FreeRTOS) and per three compilers (CCS, GCC, and IAR). In the case where changes must be made to the core and you need to recompile it, there are two ways to build it:

- For CCS (TI-RTOS or FreeRTOS), import the CCS project located under `{SDK ROOT}\source\ti\net\atcmd\ccs` and build the library.

---

**NOTE:** Pay attention to choose the appropriate product number.

- For all other favorites (including CCS), open the command prompt line under the directory `{SDK ROOT}\source\ti\net\atcmd`, and execute `gmake` from the XDC tool root directory. To clean all outputs, execute `gmake clean`.

## 4 Commands Summary

**Table 1. Device Commands**

Command	Definition
AT+Start	Starts the network processor (NWP)
AT+Stop	Stops the NWP
AT+Get	Gets device configurations
AT+Set	Sets device configurations
AT+Test	Test command

**Table 2. Socket Commands**

Command	Definition
AT+Socket	Create an endpoint for communication
AT+Close	Close socket
AT+Accept	Accept a connection on a socket
AT+Bind	Assign a name to a socket
AT+Listen	Listen for connections on a socket
AT+Connect	Initiate a connection on a socket
AT+Select	Monitor socket activity
AT+SetSockOpt	Set socket options
AT+GetSockOpt	Get socket options
AT+Recv	Read data from TCP socket
AT+RecvFrom	Read data from socket
AT+Send	Write data to TCP socket
AT+SendTo	Write data to socket

**Table 3. WLAN Commands**

Command	Definition
AT+WlanConnect	Connect to WLAN network as a station
AT+WlanDisconnect	Disconnect connection
AT+WlanProfileAdd	Add profile
AT+WlanProfileGet	Get profile
AT+WlanProfileDel	Delete profile
AT+WlanPolicySet	Set policy values
AT+WlanPolicyGet	Get policy values
AT+WlanScan	Gets the WLAN scan operation results

**Table 3. WLAN Commands (continued)**

Command	Definition
AT+WlanProvisioning	Start provisioning
AT+WlanSetMode	WLAN set mode
AT+WlanSet	Setting WLAN configurations
AT+ WlanGet	Getting WLAN configurations

**Table 4. File System Commands**

Command	Definition
AT+FileOpen	Open file in storage device
AT+FileClose	Close file in storage device
AT+FileCtl	Controls various file system operations
AT+FileDel	Delete file from storage device
AT+FileGetFilelist	Get list of a files
AT+FileGetInfo	Get information of a file
AT+FileRead	Read block of data from a file in storage device
AT+FileWrite	Write block of data to a file in storage device

**Table 5. Network Application Commands**

Command	Definition
AT+NetAPPStart	Starts a network application
AT+NetAPPStop	Stops a network application
AT+NetAPPGetHostByName	Get host IP by name
AT+NetAPPGetHostByService	Host IP by service
AT+NetAPPSet	Setting network applications configurations
AT+NetAPPGet	Getting network applications configurations
AT+NetAPPSend	Sends Network Application response or data following a Network Application request event
AT+NetAPPRecv	Receives data from the network processor following a Network Application response event
AT+NetAPPping	Send ping to network hosts
AT+NetAPPGetServiceList	Get service list
AT+NetAPPRegisterService	Register a new mDNS service
AT+NetAPPUnRegisterService	Unregister mDNS service

**Table 6. Network Configuration Commands**

Command	Definition
AT+NetCfgSet	Setting network configurations
AT+NetCfgGet	Getting network configurations

**Table 7. Network Utility Commands**

Command	Definition
AT+NetUtilGet	Getting utilities configurations
AT+NetUtilCmd	Performing utilities-related commands

**Table 8. Asynchronous Events**

Command	Definition
+EventFatalError	Fatal Error event for inspecting fatal error
+EventGeneral	General asynchronous event for inspecting general events
+EventWlan	WLAN asynchronous event
+EventNetApp	Network Application asynchronous event
+EventSock	Socket asynchronous event

## 5 Protocol Syntax

### 5.1 Commands

Syntax:

AT<command name>=<param1>, <param2>, ..., <paramX>

- Commands that contain parameters should include an equal sign (=) between the command name and the first parameter.
- Commands that contain parameters should include a comma mark (,) as a delimiter between them—comma delimiters are mandatory.
- In case the parameter is defined as "ignore" or "optional", it could be left empty but the comma delimiter should be mentioned—it looks like two conjunction delimiters (,,).
- Parameters that are left empty must be treated as 0 or NULL (according to the parameter type), and in case it was not defined as "ignore" or "optional", an error should be raised.
- String parameters containing spaces must be enclosed with quotes (" ").
- String parameters containing a comma delimiter (,) must be enclosed with quotes (" ").
- Numeric value parameters could be one of the following:
  - Decimal
  - Hexadecimal—must have a prefix of zero x notation (0x)
- Numeric array parameters could be enclosed with square brackets ([ ]).
- Numeric array parameters could be one of the following:
  - IPv4 address—contains four numeric values (8 bits each) with a point mark (.) as a delimiter between them enclosed with or without square brackets—x.x.x.x or [x.x.x.x]
  - IPv6 address—contains four numeric values (32 bit each) with a colon mark (:) as a delimiter between them enclosed with or without square brackets—x:x:x:x or [x:x:x:x]
  - MAC address—contains six numeric values (8 bit each) with a colon mark (:) as a delimiter between them enclosed with or without square brackets—x:x:x:x:x:x or [x:x:x:x:x:x]
- Bitmask parameters should contain values with a vertical bar ( | ) as delimiter between them enclosed with or without square brackets—x|x|x or [x|x|x]
- The AT command handler allows for the AT commands to be entered in uppercase or lowercase with spaces between the arguments.
- Data parameter should be one of the following formats:
  - Binary format
  - Base64 format—binary to text encoding

## 5.2 Command Return Status

Command return status could be one of the following cases:

- Command that returns values:

```
<command name>: <value1>, ..., <valueX>
```

- Command that returns success:

```
OK
```

- Command that returns failure:

```
ERROR:<error description>, <error code>
```

Command return status should include a colon mark (:) between the command name and the first value.

Command return status that contains list values should include a semicolon mark (;) as a delimiter between the list members.

## 5.3 Asynchronous Event

The events may arrive at any time. Asynchronous events are always built in the following format:

```
<event name>: <event ID>, <value1>, ..., <valueX>
```

The event should include a colon mark (:) between the event name and the event ID.

# 6 Command Description

## 6.1 Device Commands

**Table 9. AT+Start Starts the NWP**

Request:	Response:
AT+Start	OK
Arguments: none	Arguments: none

**Table 10. AT+Stop Stops the NWP**

Request:	Response:
AT+Stop =[Timeout]	OK
Arguments: Timeout: Stop timeout in milliseconds should be used to give the device time to finish any transmission or reception that is not completed when the function was called <ul style="list-style-type: none"> <li>• 0 Enter to hibernate immediately</li> <li>• 0xFFFF Host waits for the response from the device before hibernating, without timeout protection</li> <li>• 0 &lt;Timeout[msec] &lt;0xFFFF Host waits for the response from the device before hibernating, with a defined timeout protection This timeout defines the maximum time to wait. The NWP response can be sent earlier than this timeout.</li> </ul>	Arguments: none

**Table 11. AT+Get Getting Device Configurations**

Request:		Response:
AT+Get= [ID],[Option]		+Get:[Value1],...,[ValueX] OK
Arguments:		Arguments:
ID	Option	Return Values
Status	<i>Device</i>	Value1: bitmask: General error
	<i>WLAN</i>	Value1: bitmask: <ul style="list-style-type: none"> <li>• WLANASYNCONNECTEDRESPONSE</li> <li>• WLANASYNCDISCONNECTEDRESPONSE</li> <li>• STA_CONNECTED</li> <li>• STA_DISCONNECTED</li> <li>• P2P_DEV_FOUND</li> <li>• CONNECTION_FAILED</li> <li>• P2P_NEG_REQ_RECEIVED</li> <li>• RX_FILTERS</li> <li>• WLAN_STA_CONNECTED</li> </ul>
	<i>BSD</i>	Value1: bitmask: <ul style="list-style-type: none"> <li>• TX_FAILED</li> </ul>
	<i>NETAPP</i>	Value1: bitmask: <ul style="list-style-type: none"> <li>• IPACQUIRED</li> <li>• IPACQUIRED_V6</li> <li>• IP_LEASED</li> <li>• IP_RELEASED</li> <li>• IPV4_LOST</li> <li>• DHCP_ACQUIRE_TIMEOUT</li> <li>• IP_COLLISION</li> <li>• IPV6_LOST</li> </ul>
General	<i>Version</i>	Value1: Chip Id Value2: FW Version (x.x.x.x) Value3: PHY Version (x.x.x.x) Value4: NWP Version (x.x.x.x) Value5: ROM Version
	<i>Time</i>	Value1: Hour = Current hours Value2: Minute = Current minutes Value3: Second = Current seconds Value4: Day = Current Date 1–31 Value5: Month = Current Month 1–12 Value6: Year = Current year
	<i>Persistent</i>	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>
<i>IOT</i>	<i>UDID</i>	16 bytes

**Table 12. AT+Set Setting Device Configurations**

Request:			Response:
AT+Set= [ID],[Option],[Value1],...,[ValueX]			OK
Arguments:			
ID	Option	Value	
General	<i>Persistent</i> sets the default system-wide configuration persistence mode. In case true, all APIs that follow <i>system configured</i> persistence (see persistence attribute noted per API) shall maintain the configured settings. In case false, all calls to APIs that follow <i>system configured</i> persistence shall be volatile. Configuration should revert to default after reset or power recycle.	Value1: <ul style="list-style-type: none"> <li>1: Enable</li> <li>0: Disable</li> </ul>	
	<i>Time</i> sets the device time and date	Value1: Hour = Current hours Value2: Minute = Current minutes Value3: Second = Current seconds Value4: Day = Current Date 1–31 Value5: Month = Current Month 1–12 Value6: Year = Current year	

**Table 13. AT+Test Test Command**

Request:	Response:
AT+Test	OK
Arguments: none	Arguments: none

## 6.2 Socket Commands

**Table 14. AT+Socket Create an End-Point for Communication**

Request:	Response:
AT+Socket= [Domain],[Type],[Protocol]	+Socket: [socket] OK
Arguments: <ul style="list-style-type: none"> <li>Domain: Specifies the protocol family of the created socket: <ul style="list-style-type: none"> <li>INET: For network protocol IPv4</li> <li>INET6: For network protocol IPv6</li> <li>RF: For starting transceiver mode</li> </ul> </li> <li>Type: Specifies the communication semantic: <ul style="list-style-type: none"> <li>STREAM: Reliable stream-oriented service or Stream Sockets</li> <li>DGRAM: Datagram service or Datagram Sockets</li> <li>RAW: Raw protocols atop the network layer</li> </ul> </li> <li>Protocol: Specifies a particular transport to be used with the socket: <ul style="list-style-type: none"> <li>TCP</li> <li>UDP</li> <li>RAW</li> <li>SEC</li> </ul> </li> </ul>	Arguments: socket: Socket descriptor that will be used in the socket commands described in <a href="#">Table 15</a> through <a href="#">Table 26</a>

**Table 15. AT+Close Close Socket**

Request:	Response:
AT+Close= [socket]	+Close: [socket] OK
Arguments: socket: Socket descriptor received from AT+Socket command	

**Table 16. AT+Accept Accept a Connection on a Socket**

Request:	Response:
AT+Accept = [socket],[family]	OK +Accept: [New Socket],[Family],[Port],[Address]
Arguments: <ul style="list-style-type: none"> <li>• socket: Socket descriptor received from AT+Socket command</li> <li>• family: Specifies the protocol family of the created socket:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• NewSocket: New connected socket</li> <li>• Family: internet protocol (AF_INET)</li> <li>• Port: Address port</li> <li>• Address: Peer socket address</li> </ul>

**Table 17. AT+Bind Assign a Name to a Socket**

Request:	Response:
AT+Bind = [Socket],[Family],[Port],[Address]	OK
Arguments: <ul style="list-style-type: none"> <li>• Socket: Socket descriptor received from AT+Socket command</li> <li>• Family: Specifies the protocol family of the created socket:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> <li>• Port: Address port</li> <li>• Address – Local socket address</li> </ul>	

**Table 18. AT+Listen Listen for Connections on a Socket**

Request:	Response:
AT+Listen = [socket],[backlog]	OK
Arguments: <ul style="list-style-type: none"> <li>• socket: Received from AT+Socket command</li> <li>• backlog: Listen</li> </ul>	

**Table 19. AT+Connect Initiate a Connection on a Socket**

Request:	Response:
AT+Connect = [Socket],[Family],[Port],[Address]	OK +Connect : [Port], [Address]
Arguments: <ul style="list-style-type: none"> <li>• Socket: Received from AT+Socket command</li> <li>• Family: internet protocol:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> <li>• Port: Address port</li> <li>• Address – Peer socket address (“x.x.x.x”)</li> </ul>	

**Table 20. AT+Select Monitor Socket Activity**

Request:	Response:
AT+Select = [nfds],[readsds],[timeout sec],[timeout usec]	OK +Select: [readsds]
Arguments: <ul style="list-style-type: none"> <li>• nfds: The highest-numbered file descriptor in any of the three sets (read, write, and except)</li> <li>• readsds: Socket descriptors as bit list (for example, 0 2 for monitoring socket 0 and socket 2)</li> <li>• timeout sec: Time in seconds is an upper bound on the amount of time elapsed before select() returns. 0 means return immediately.</li> <li>• timeout usec: Time in microseconds</li> </ul>	Arguments: readsds: Socket descriptors list for read monitoring and accept monitoring

**Table 21. AT+SetSockOpt Set Socket Options**

Request:		Response:
AT+SetSockOpt = [sd],[Level],[Option],[Value1],...,[ValueX]		OK
Arguments: sd: Socket descriptor		
Level: Defines the protocol level for this option	Option	Value
SOCKET	<b>KEEPALIVE</b> Enable or disable periodic keep alive. Keeps TCP connections active by enabling the periodic transmission of messages	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>
	<b>KEEPALIVETIME</b> Set keep alive timeout	Value1: Timeout in seconds
	<b>RX_NO_IP_BOUNDARY</b> Enable or disable RX IP boundary	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>
	<b>RCVTIMEO</b> Sets the timeout value that specifies the maximum amount of time an input function waits until it completes	Value1: Seconds Value2: Microseconds. 10000 microseconds resolution
	<b>RCVBUF</b> Sets TCP maximum receive window size	Value1: Size in bytes
	<b>NONBLOCKING</b> Sets socket to nonblocking	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>
	<b>SECMETHOD</b> Sets method to TCP secured socket	Value1 security method: <ul style="list-style-type: none"> <li>• SSLV3: Security method SSL v3</li> <li>• TLSV1: Security method TLS v1</li> <li>• TLSV1_1: Security method TLS v1_1</li> <li>• TLSV1_2: Security method TLS v1_2</li> <li>• SSLV3_TLSV1_2: Use highest possible version from SSLv3–TLS 1.2</li> <li>• DLSV1: Security method DTL v1</li> </ul>
<b>SECURE_MASK</b> Sets specific cipher to TCP secured socket	Value1: Cipher type: <ul style="list-style-type: none"> <li>• SSL_RSA_WITH_RC4_128_SHA</li> <li>• SSL_RSA_WITH_RC4_128_MD5</li> <li>• TLS_RSA_WITH_AES_256_CBC_SHA</li> <li>• TLS_DHE_RSA_WITH_AES_256_CBC_SHA</li> <li>• TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</li> <li>• TLS_ECDHE_RSA_WITH_RC4_128_SHA</li> <li>• TLS_RSA_WITH_AES_128_CBC_SHA256</li> <li>• TLS_RSA_WITH_AES_256_CBC_SHA256</li> <li>• TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</li> <li>• TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</li> <li>• TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</li> <li>• TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</li> </ul>	

**Table 21. AT+SetSockOpt Set Socket Options (continued)**

Request:		Response:	
SOCKET (continued)	<p><i>SECURE_MASK</i> (continued)</p>	<ul style="list-style-type: none"> <li>• TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA</li> <li>• TLS_RSA_WITH_AES_128_GCM_SHA256</li> <li>• TLS_RSA_WITH_AES_256_GCM_SHA384</li> <li>• TLS_DHE_RSA_WITH_AES_128_GCM_SHA256</li> <li>• TLS_DHE_RSA_WITH_AES_256_GCM_SHA384</li> <li>• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</li> <li>• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</li> <li>• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</li> <li>• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</li> <li>• TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256</li> <li>• TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256</li> <li>• TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256</li> </ul>	
	<p><i>SECURE_FILES_CA_FILE_NAME</i> Map secured socket to CA file by name</p>	Value1: File name	
	<p><i>SECURE_FILES_PRIVATE_KEY_FILE_NAME</i> Map secured socket to private key by name</p>	Value1: File name	
	<p><i>SECURE_FILES_CERTIFICATE_FILE_NAME</i> Map secured socket to certificate file by name</p>	Value1: File name	
	<p><i>SECURE_FILES_DH_KEY_FILE_NAME</i> Map secured socket to Diffie Hellman file by name</p>	Value1: File name	
	<p><i>CHANGE_CHANNEL</i> Sets channel in transceiver mode</p>	Value1: Channel number (range is 1–13)	
	<p><i>SECURE_ALPN</i> Sets the ALPN list</p>	Value1: The parameter is a bit map consist of or of the following values: H1   H2   H2C   H2_14   H2_16   FULL_LIST	
	<p><i>LINGER</i> Socket lingers on close pending remaining send and receive packets</p>	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul> Value2: Linger time in seconds	
	<p><i>SECURE_EXT_CLIENT_CHLNG_RESP</i> Set with no parameter to indicate that the client uses external signature using Network Application request</p>	Value1: Ignore	
	<p><i>SECURE_DOMAIN_NAME_VERIFICATION</i> Set a domain name, to check in SSL client connection</p>	Value1: Domain name	
IP	<p><i>MULTICAST_TTL</i> Set the time-to-live value of outgoing multicast packets for this socket</p>	Value1: Number of hops	
	<p><i>ADD_MEMBERSHIP</i> UDP socket, join a multicast group</p>	Value1: IPv4 multicast address to join Value2: Multicast interface address	
	<p><i>DROP_MEMBERSHIP</i> UDP socket, leave a multicast group</p>	Value1: IPv4 multicast address to join Value2: Multicast interface address	
	<p><i>RAW_RX_NO_HEADER</i> Raw socket remove IP header from received data</p>	Value1: <ul style="list-style-type: none"> <li>• 1: Remove header</li> <li>• 0: Keep header</li> </ul>	
	<p><i>HDRINCL</i> RAW socket only, the IPv4 layer generates an IP header when sending a packet unless this option is enabled on the socket</p>	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>	
	<p><i>RAW_IPV6_HDRINCL</i> RAW socket only, the IPv6 layer generates an IP header when sending a packet unless this option is enabled on the socket</p>	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>	

**Table 21. AT+SetSockOpt Set Socket Options (continued)**

Request:		Response:
PHY	<b>PHY_RATE</b> Set WLAN PHY transmit rate on RAW socket	Value1: Rate
	<b>PHY_TX_POWER</b> RAW socket, set WLAN PHY TX power	Value1: Power range is 1–15
	<b>PHY_NUM_FRAMES_TO_TX</b> RAW socket, set number of frames to transmit in transceiver mode	Value1: Number of frames
	<b>PHY_PREAMBLE</b> RAW socket, set WLAN PHY preamble for long or short	Value1: Preamble value
	<b>PHY_TX_INHIBIT_THRESHOLD</b> RAW socket, set WLAN TX inhibit threshold (CCA).	Value1: Threshold value: <ul style="list-style-type: none"> <li>• MIN</li> <li>• LOW</li> <li>• DEFAULT</li> <li>• MED</li> <li>• HIGH</li> <li>• MAX</li> </ul>
	<b>PHY_TX_TIMEOUT</b> RAW socket, changes the TX timeout (lifetime) of transceiver frames	Value1: Time in milliseconds, maximum value is 10 ms
<b>PHY_ALLOW_ACKS</b> RAW socket, enable sending ACKs in transceiver mode	Value1: <ul style="list-style-type: none"> <li>• 1: Enable</li> <li>• 0: Disable</li> </ul>	

**Table 22. AT+GetSockOpt Get Socket Options**

Request:	Response:
AT+GetSockOpt = [sd],[level],[option]	+GetSockOpt: [value1],...,[valueX] OK
Arguments: <ul style="list-style-type: none"> <li>• sd: Socket handle</li> <li>• level: Defines the protocol level for this option (see <a href="#">Table 21</a>)</li> <li>• option: Defines the option name to interrogate (see <a href="#">Table 21</a>)</li> </ul>	Arguments: value1,...,valueX (see the AT+SetSockOpt command in <a href="#">Table 21</a> )

**Table 23. AT+Recv Read Data From TCP Socket**

Request:	Response:
AT+Recv = [sd],[format],[length]	OK +Recv: [sd],[length],[data]
Arguments: <ul style="list-style-type: none"> <li>• sd: Socket handle</li> <li>• format: Data format:                             <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• length: Maximum number of bytes to receive</li> </ul>	

**Table 24. AT+RecvFrom Read Data From Socket**

Request:	Response:
AT+RecvFrom = [sd],[family],[port],[addr],[format],[length]	OK +RecvFrom: [sd],[length],[data]
Arguments: <ul style="list-style-type: none"> <li>• sd: Socket handle</li> <li>• family: internet protocol               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> <li>• port: Address port (16 bits)</li> <li>• addr: internet address (32 bits)</li> <li>• format: Data format:               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• length: Maximum number of bytes to receive</li> </ul>	

**Table 25. AT+Send Write Data to TCP Socket**

Request:	Response:
AT+Send = [sd],[format],[length],[data]	OK
Arguments: <ul style="list-style-type: none"> <li>• sd: Socket handle</li> <li>• format: Data format:               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• length: Number of bytes to send</li> <li>• data: Data to send</li> </ul>	

**Table 26. AT+SendTo Write Data to Socket**

Request:	Response:
AT+SendTo = [sd],[family],[port],[addr],[format],[length],[data]	OK
Arguments: <ul style="list-style-type: none"> <li>• sd: Socket handle</li> <li>• family: internet protocol:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> <li>• port: Address port (16 bits)</li> <li>• addr: internet address (32 bits)</li> <li>• format: Data format:               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• length: Maximum number of bytes to receive</li> <li>• data: Data to send</li> </ul>	

### 6.3 WLAN Commands

**Table 27. AT+WlanConnect Connect to WLAN Network as a Station**

Request:	Response:
AT+WlanConnect = [SSID],[BSSID],[SecurityType],[SecurityKey],[SecurityExtUser],[SecurityExtAnonUser],[SecurityExtEapMethod]	OK
Arguments: <ul style="list-style-type: none"> <li>• SSID: Name of the Access Point</li> <li>• BSSID: Access Point MAC address (Optional)</li> <li>• SecurityType: Security type:               <ul style="list-style-type: none"> <li>– OPEN</li> <li>– WEP</li> <li>– WEP_SHARED</li> <li>– WPA_WPA2</li> <li>– WPA_ENT</li> <li>– WPS_PBC</li> <li>– WPS_PIN</li> </ul> </li> <li>• SecurityKey: Password (Optional in case it is not needed)</li> <li>• SecurityExtUser: Enterprise user name parameters (Ignored in case <b>WPA_ENT</b> was not selected)</li> <li>• SecurityExtAnonUser: Enterprise anonymous user name parameters (Ignored in case <b>WPA_ENT</b> was not selected)</li> <li>• SecurityExtEapMethod: Extensible Authentication Protocol (Ignored in case <b>WPA_ENT</b> was not selected):               <ul style="list-style-type: none"> <li>– TLS</li> <li>– TTLS_TLS</li> <li>– TTLS_MSCHAPv2</li> <li>– TTLS_PSK</li> <li>– PEAP0_TLS</li> <li>– PEAP0_MSCHAPv2</li> <li>– PEAP0_PSK</li> <li>– PEAP1_TLS</li> <li>– PEAP1_PSK</li> </ul> </li> </ul>	

**Table 28. AT+WlanDisconnect Disconnect the Connection**

Request:	Response:
AT+WlanDisconnect	OK
Arguments: none	

**Table 29. AT+WlanProfileAdd Add Profile**

Request:	Response:
AT+WlanProfileAdd = [SSID],[BSSID],[SecurityType],[SecurityKey],[SecurityExtUser],[SecurityExtAnonUser],[SecurityExtEapMethod],[Priority]	+WlanProfileAdd: [index] OK
Arguments: <ul style="list-style-type: none"> <li>• SSID: Name of the Access Point</li> <li>• BSSID: Access Point MAC address (Optional)</li> <li>• SecurityType: Security type:               <ul style="list-style-type: none"> <li>– OPEN</li> <li>– WEP</li> <li>– WEP_SHARED</li> <li>– WPA_WPA2</li> <li>– WPA_ENT</li> <li>– WPS_PBC</li> <li>– WPS_PIN</li> </ul> </li> <li>• SecurityKey: Password (Optional in case it is not needed)</li> <li>• SecurityExtUser: Enterprise user name parameters (Ignored in case <b>WPA_ENT</b> was not selected)</li> <li>• SecurityExtAnonUser: Enterprise anonymous user name parameters (Ignored in case <b>WPA_ENT</b> was not selected)</li> <li>• SecurityExtEapMethod: Extensible Authentication Protocol (Ignored in case <b>WPA_ENT</b> was not selected):               <ul style="list-style-type: none"> <li>– TLS</li> <li>– TTLS_TLS</li> <li>– TTLS_MSCHAPv2</li> <li>– TTLS_PSK</li> <li>– PEAP0_TLS</li> <li>– PEAP0_MSCHAPv2</li> <li>– PEAP0_PSK</li> <li>– PEAP1_TLS</li> <li>– PEAP1_PSK</li> </ul> </li> <li>• Priority: Profile priority:               <ul style="list-style-type: none"> <li>– Lowest priority: 0</li> <li>– Highest priority: 15</li> </ul> </li> </ul>	Arguments: index: Profile stored index

**Table 30. AT+WlanProfileGet Get Profile**

Request:	Response:
AT+WlanProfileGet = [index]	+WlanProfileGet: [SSID],[BSSID],[SecurityType],[SecurityExtUser],[SecurityExtAnonUser],[SecurityExtEapMethod],[priority] OK
Arguments: index: Profile stored index received from +WlanProfileAdd	Arguments: See the AT+WlanProfileAdd command in <a href="#">Table 29</a>

**Table 31. AT+WlanProfileDel Delete Profile**

Request:	Response:
AT+ WlanProfileDel = [index]	OK
Arguments: index: Number of profile to delete received from +WlanProfileAdd To delete all profiles, use index = 0xFF	

**Table 32. AT+WlanPolicySet Set Policy Values**

Request:		Response:
AT+WlanPolicySet = [Type],[Option],[Value]		OK
Type	Option	Value
<b>CONNECTION</b> Defines options available to connect to the AP (Options could be set as bit masked). No option selected = disable all	<i>Auto</i> Reconnect to one of the stored profiles each time the connection fails or the device is rebooted	Ignore
	<i>Fast</i> Establish a fast connection to AP	Ignore
	<i>P2P</i> Automatically connect to the first P2P device available	Ignore
	<i>Auto_Provisioning</i> Start the provisioning process after a long period of disconnection when profiles exist	Ignore
<b>SCAN</b> Defines system scan time interval. An interval is 10 minutes. After settings scan interval, an immediate scan is activated	<i>Hidden_SSID</i>	Scan interval in seconds
	<i>No_Hidden_SSID</i>	Scan interval in seconds
	<i>Disable_Scan</i>	Ignore
<b>PM</b> Defines a power management policy for Station mode	<i>Normal</i>	Ignore
	<i>Low_Latency</i>	Ignore
	<i>Low_Power</i>	Ignore
	<i>Always_On</i>	Ignore
	<i>Long_Sleep</i>	Maximum sleep time in milliseconds
<b>P2P</b> Defines P2P negotiation policy parameters for P2P role	<ul style="list-style-type: none"> <li>• <b>CLIENT</b> Indicates that the device is forced to be CLIENT</li> <li>• <b>GROUP_OWNER</b> Indicates that the device is forced to be P2P GO</li> <li>• <b>NEGOTIATE</b> Indicates that the device can be either CLIENT or GO, depending on the Wi-Fi Direct® negotiation tiebreaker</li> </ul>	<ul style="list-style-type: none"> <li>• <b>ACTIVE</b> When the remote peer is found after the discovery process, the device immediately sends the negotiation request to the peer device.</li> <li>• <b>PASSIVE</b> When the remote peer is found after the discovery process, the device passively waits for the peer to start the negotiation, and only responds after.</li> <li>• <b>RAND_BACKOFF</b> When the remote peer is found after the discovery process, the device triggers a random timer (from 1 to 6 seconds). During this period, the device passively waits for the peer to start the negotiation. If the timer expires without negotiation, the device immediately sends the negotiation request to the peer device.</li> </ul>

**Table 33. AT+WlanPolicyGet Get Policy Values**

Request:	Response:
AT+WlanPolicyGet = [Type]	+WlanPolicyGet: [Option],[Value] OK
Arguments: <ul style="list-style-type: none"> <li>• Type: Type of policy. The Options are:               <ul style="list-style-type: none"> <li>– CONNECTION Get connection policy</li> <li>– SCAN Get scan policy</li> <li>– PM Get power management policy</li> <li>– P2P Get P2P policy</li> </ul> </li> </ul>	Arguments: <ul style="list-style-type: none"> <li>• Option: See the AT+WlanPolicySet command in <a href="#">Table 32</a></li> <li>• Value: See the AT+WlanPolicySet command in <a href="#">Table 32</a></li> </ul>

**Table 34. AT+WlanScan Gets the WLAN Scan Operation Results**

Request:	Response:
AT+WlanScan = [Index],[Count]	+WlanScan: [SSID],[BSSID],[RSSI],[Channel],[Security_Type], [Hidden_SSID],[Cipher],[Key_Mgmt]; OK
Arguments: <ul style="list-style-type: none"> <li>• Index: Starting index identifier (range 0–29) for getting scan results.</li> <li>• Count: How many entries to fetch; maximum is 30</li> </ul>	Arguments: <ul style="list-style-type: none"> <li>• SSID: Wireless LAN identifier</li> <li>• BSSID: MAC address of the wireless access point</li> <li>• Channel</li> <li>• RSSI: Relative received signal strength in a wireless environment</li> <li>• Security_Type:               <ul style="list-style-type: none"> <li>– OPEN</li> <li>– WEP</li> <li>– WPA</li> <li>– WPA2</li> <li>– WPA_WPA2</li> </ul> </li> <li>• Hidden_SSID:               <ul style="list-style-type: none"> <li>– 1: Hidden</li> <li>– 0: Not hidden</li> </ul> </li> <li>• Cipher:               <ul style="list-style-type: none"> <li>– None</li> <li>– WEP40</li> <li>– WEP104</li> <li>– TKIP</li> <li>– CCMP</li> </ul> </li> <li>• Key_Mgmt:               <ul style="list-style-type: none"> <li>– None</li> <li>– 802_1_X</li> <li>– PSK</li> </ul> </li> </ul>

**Table 35. AT+WlanProvisioning Start Provisioning**

Request:	Response:
AT+WlanProvisioning = [Cmd],[Role],[Period],[Key],[Flag]	OK
Arguments: <ul style="list-style-type: none"> <li>• <b>Cmd:</b> Provisioning mode requested:               <ul style="list-style-type: none"> <li>– AP: Start AP provisioning (AP role)</li> <li>– SC: Start SmartConfig™ technology provisioning (STA role)</li> <li>– APSC: Start AP + SmartConfig provisioning (AP role)</li> <li>– APSC_EXT_CFG: Start AP + SmartConfig + External configuration (AP role)</li> <li>– STOP: Stop provisioning</li> <li>– ABORT_EXT_CFG</li> </ul> </li> <li>• <b>Role:</b> The role that the device will switch to in case of successful provisioning:               <ul style="list-style-type: none"> <li>– STA: Station</li> <li>– AP: Access point</li> </ul> </li> <li>• <b>Period:</b> The period of time (in seconds) the system waits before it automatically stops the provisioning process when no activity is detected</li> <li>• <b>Key:</b> SmartConfig key: public key for SmartConfig process (Optional: relevant for SmartConfig only)</li> <li>• <b>Flag:</b> (Optional)               <ul style="list-style-type: none"> <li>– None: (default)</li> <li>– EXT_CONFIRM: Confirmation phase will be completed externally by host (for example, through cloud assist)</li> </ul> </li> </ul>	

**Table 36. AT+WlanSetMode WLAN Set Mode**

Request:	Response:
AT+WlanSetMode = [Mode]	OK
Arguments: <ul style="list-style-type: none"> <li>• <b>Mode:</b> WLAN mode to start the device:               <ul style="list-style-type: none"> <li>– STA: For WLAN station mode</li> <li>– AP: For WLAN Access Point mode</li> <li>– P2P: For WLAN P2P mode</li> </ul> </li> </ul>	

**Table 37. AT+WlanSet Setting WLAN Configurations**

Request:			Response:
AT+WlanSet = [ID],[Option],[Value1],...,[ValueX]			OK
ID	Option	Value	
AP	<i>SSID</i> Set SSID for AP mode	String up to 32 characters	
	<i>CHANNEL</i> Set channel for AP mode	Channel in the range of [1–11]	
	<i>HIDDEN_SSID</i> Set Hidden SSID Mode for AP mode	<ul style="list-style-type: none"> <li>0: Disabled</li> <li>1: Send empty (length = 0) SSID in beacon and ignore probe request for broadcast SSID</li> <li>2: Clear SSID (ASCII 0), but keep the original length (this may be required with some clients that do not support empty SSID) and ignore probe requests for broadcast SSID</li> </ul>	
	<i>SECURITY</i> Set Security type for AP mode	<ul style="list-style-type: none"> <li>OPEN: Open security</li> <li>WEP: WEP security</li> <li>WPA_WPA2: WPA security</li> </ul>	
	<i>PASSWORD</i> Set Password for AP mode (for WEP or for WPA)	Password for WPA: 8–63 characters Password for WEP: 5 or 13 characters (ASCII)	
	<i>MAX_STATIONS</i> Set Max AP stations	1...4 Note: can be less than the number of currently connected stations	
	<i>MAX_STA_AGING</i> Set Max station aging time	Number of seconds	
	<i>ACCESS_LIST_MODE</i> Set AP access list mode	<ul style="list-style-type: none"> <li>DISABLE</li> <li>DENY_LIST: Set Black List Mode</li> </ul>	
	<i>ACCESS_LIST_ADD_MAC</i> Add MAC address to the AP access list	MAC address: 6 characters	
	<i>ACCESS_LIST_DEL_MAC</i> Delete MAC address from the AP access list	MAC address: 6 characters	
<i>ACCESS_LIST_DEL_IDX</i> Delete MAC address from index in the AP access list	Index		

**Table 37. AT+WlanSet Setting WLAN Configurations (continued)**

Request:		Response:
GENERAL	<i>COUNTRY_CODE</i> Set Country Code for AP mode	Two characters country code
	<i>STA_TX_POWER</i> Set STA mode TX power level	Number between 0–15, as dB offset from maximum power (0 sets maximum power)
	<i>AP_TX_POWER</i> Set AP mode TX power level	Number between 0–15, as dB offset from maximum power (0 sets maximum power)
	<i>INFO_ELEMENT</i> Set Info Element for AP mode	<ul style="list-style-type: none"> <li>• Value1: Index of the info element</li> <li>• Value2: Role:               <ul style="list-style-type: none"> <li>– AP</li> <li>– P2P</li> </ul> </li> <li>• Value3: Info element ID</li> <li>• Value4: Organization unique ID first Byte</li> <li>• Value5: Organization unique ID second Byte</li> <li>• Value6: Organization unique ID third Byte</li> <li>• Value7: Info element (maximum 252 chars)</li> </ul>
	<i>SCAN_PARAMS</i> Set scan parameters	<ul style="list-style-type: none"> <li>• Value1: Channel mask</li> <li>• Value2: RSSI threshold</li> </ul>
	<i>SUSPEND_PROFILES</i> Set suspended profiles mask	Suspended bitmask
	<i>DISABLE_ENT_SERVER_AUTH</i> This option enables to skip server authentication and is valid for one use, when manually connection to an enterprise network	<ul style="list-style-type: none"> <li>• 1: Disable the server authentication</li> <li>• 0: Enable</li> </ul>
P2P	<i>DEV_TYPE</i> Set P2P Device type	Device type is published under P2P I.E (maximum length of 17 characters)
	<i>CHANNEL_N_REGS</i> Set P2P Channels	<ul style="list-style-type: none"> <li>• Value1: Listen channel (either 1/6/11 for 2.4 GHz)</li> <li>• Value2: Listen regulatory class (81 for 2.4 GHz)</li> <li>• Value3: Operating channel (channel 1, 6, or 11 for 2.4 GHz)</li> <li>• Value4: Operating regulatory class (81 for 2.4 GHz)</li> </ul>
RX_FILTER	<i>STATE</i> Enable or disable filters	Filter Bitmap array (16 bytes in format xx:xx)
	<i>SYS_STATE</i> Enable or disable system filters	Filter Bitmap array (4 bytes in format xx:xx)
	<i>REMOVE</i> Remove filters	Filter Bitmap array (16 bytes in format xx:xx)
	<i>STORE</i> Save the filters as persistent	null

**Table 38. AT+ WlanGet Getting WLAN Configurations**

Request:		Response:
AT+WlanGet = [ID],[Option]		+WlanGet: [Value1],...,[ValueX] OK
Arguments:		Arguments: See the AT+WlanSet command in <a href="#">Table 37</a>
ID	Option	
AP	<i>SSID</i> Get SSID for AP mode	
	<i>CHANNEL</i> Get channel for AP mode	
	<i>HIDDEN_SSID</i> Get Hidden SSID Mode for AP mode	
	<i>SECURITY</i> Get Security type for AP mode	
	<i>PASSWORD</i> Get Password for AP mode (for WEP or for WPA)	
	<i>MAX_STATIONS</i> Get Max AP allowed stations	
	<i>MAX_STA_AGING</i> Get AP aging time in seconds	
	<i>ACCESS_LIST_NUM_ENTRIES</i> Get AP access list number of entries	
<i>ACCESS_LIST</i> Get the AP access list from start index	The start index in the access list	
GENERAL	<i>COUNTRY_CODE</i> Get Country Code for AP mode	
	<i>STA_TX_POWER</i> Get STA mode TX power level	
	<i>AP_TX_POWER</i> Get AP mode TX power level	
	<i>SCAN_PARAMS</i> Get scan parameters	
P2P	<i>CHANNEL_N_REGS</i> Get P2P Channels	
RX_FILTER	<i>STATE</i> Retrieves the filters enable/disable status	
	<i>SYS_STATE</i> Retrieves the system filters enable or disable status	

**Table 38. AT+ WlanGet Getting WLAN Configurations (continued)**

Request:		Response:
Connection	Ignore	<ul style="list-style-type: none"> <li>• Value1: Role               <ul style="list-style-type: none"> <li>– sta</li> <li>– ap</li> <li>– p2p</li> </ul> </li> <li>•</li> <li>• Value2: Status               <ul style="list-style-type: none"> <li>– disconnected</li> <li>– station_connected</li> <li>– p2pcl_connected</li> <li>– p2pgo_connected</li> <li>– ap_connected_stations</li> </ul> </li> <li>• Value3: Security               <ul style="list-style-type: none"> <li>– open</li> <li>– wep</li> <li>– wpa_wpa2</li> <li>– wps_pbc</li> <li>– wps_pin</li> <li>– wpa_ent</li> <li>– wep_shared</li> </ul> </li> <li>• Value4: SSID Name</li> <li>• Value5: BSSID</li> <li>• Value6: Device name (relevant to P2P Client only)</li> </ul>

## 6.4 File System Commands

**Table 39. AT+FileOpen Open File in Storage Device**

Request:	Response:
AT+FileOpen= [Filename], [Options],[File size]	+FileOpen:[FileID],[Secure Token] OK
Arguments: <ul style="list-style-type: none"> <li>• filename: Full path File Name</li> <li>• Options: Bitmask depend in option:               <ul style="list-style-type: none"> <li>– READ: Read a file (no bitmask)</li> <li>– WRITE: Open for write for an existing file (optionally bitmask with CREATE)</li> <li>– CREATE: Open for creating a new file (optionally bitmask with WRITE or OVERWRITE)</li> <li>– OVERWRITE: Opens an existing file (optionally bitmask with CREATE) /* Creation flags bitmask with CREATE */</li> <li>– CREATE_FAILSAFE: Fail safe</li> <li>– CREATE_SECURE: Secure file</li> <li>– CREATE_NOSIGNATURE : Relevant to secure file only</li> <li>– CREATE_STATIC_TOKEN: Relevant to secure file only</li> <li>– CREATE_VENDOR_TOKEN: Relevant to secure file only</li> <li>– CREATE_PUBLIC_WRITE: Relevant to secure file only, the file can be opened for write without Token</li> <li>– CREATE_PUBLIC_READ: Relevant to secure file only, the file can be opened for read without Token</li> </ul> </li> <li>• File size: Maximum file size is defined in bytes (mandatory only for the CREATE option and is ignored for other options)</li> </ul>	

**Table 40. AT+FileClose Close File in Storage Device**

Request:	Response:
AT+FileClose= [FileID],[CertificateFileName],[Signature]	OK
Arguments: <ul style="list-style-type: none"> <li>• FileID: Assigned from AT+FileOpen</li> <li>• CertificateFileName: Certificate file with full path (Optional)</li> <li>• Signature: The signature is SHA-1, the certificate chain may include SHA-256 (Optional)</li> </ul>	

**Table 41. AT+FileCtl Controls Various File System Operations**

Request:				Response:	
AT+FileCtl= [Command],[Secure_Token],[Filename],[Data]				+FileCtl:[NewSecureToken],[OutputData] OK	
Arguments:				Arguments:	
Command	Token	Filename	Data	Token	Output Data
<i>RESTORE</i> Return to factory default	Ignore	Ignore	<i>FACTORY_IMAGE</i> The system will be back to the production image. <i>FACTORY_DEFAULT</i> Return to factory default	Ignore	Ignore
<i>ROLLBACK</i> Roll-back file	Token assigned from AT+FileOpen	Filename to roll back	Ignore	New secure token	Ignore
<i>COMMIT</i> Commit file	Token assigned from AT+FileOpen	Filename to commit	Ignore	New secure token	Ignore
<i>RENAME</i> Rename file	Token assigned from AT+FileOpen	Filename to rename	New file name	Ignore	Ignore

**Table 41. AT+FileCtl Controls Various File System Operations (continued)**

Request:				Response:	
<i>GET_STORAGE_INFO</i> Get storage information	Ignore	Ignore	Ignore	Ignore	<ul style="list-style-type: none"> <li>DeviceBlockSize</li> <li>DeviceBlocksCapacity</li> <li>NumOfAllocatedBlocks</li> <li>NumOfReservedBlocks</li> <li>NumOfReservedBlocksForSystemfiles</li> <li>LargestAllocatedGapInBlocks</li> <li>NumOfAvailableBlocks</li> <li>ForUserFiles</li> <li>MaxFsFiles</li> <li>IsDevelopmentFormatType</li> <li>Bundlestate</li> <li>MaxFsFilesReservedForSysFiles</li> <li>ActualNumOfUserFiles</li> <li>ActualNumOfSysFiles</li> <li>NumOfAlerts</li> <li>NumOfAlertsThreshold</li> <li>FATWriteCounter</li> </ul>
<i>BUNDLE_ROLLBACK</i> Roll back a bundle	Ignore	Ignore	Ignore	Ignore	Ignore
<i>BUNDLE_COMMIT</i> Commit a bundle	Ignore	Ignore	Ignore	Ignore	Ignore

**Table 42. AT+FileDel Delete File From Storage Device**

Request:	Response:
AT+FileDel= [FileName], [SecureToken]	OK
Arguments: <ul style="list-style-type: none"> <li>FileName: Full path File Name</li> <li>SecureToken: Token assigned from AT+FileOpen (optional)</li> </ul>	

**Table 43. AT+FileGetFilelist Get a List of Files**

Request:	Response:
AT+FileGetFileList	+FileGetFileList: [FileName],[FileMaxSize],[Properties],[FileAllocatedBlocks] OK
Arguments:	Arguments: <ul style="list-style-type: none"> <li>FileName: File name</li> <li>FileMaxSize: Maximum file size</li> <li>Properties: Info flag bitmask</li> <li>FileAllocatedBlocks: Allocated blocks</li> </ul>

**Table 44. AT+FileGetInfo Get Information About a File**

Request:	Response:
AT+FileGetInfo= [FileName],[SecureToken]	+FileGetInfo: [Flags],[File Size],[Allocated Size],[Tokens],[Storage Size],[Write Counter] OK
Arguments: <ul style="list-style-type: none"> <li>• FileName: Full path file name</li> <li>• SecureToken: token assigned from AT+FileOpen (optional)</li> </ul>	

**Table 45. AT+FileRead Read a Block of Data From a File in Storage Device**

Request:	Response:
AT+FileRead= [FileID], [Offset],[Format],[Length]	+FileRead: [NumberOfReadBytes],[ReceivedData] OK
Arguments: <ul style="list-style-type: none"> <li>• FileID: Assigned from AT+FileOpen</li> <li>• Offset: Offset to specific read block</li> <li>• Format: Data format: <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• Length: Number of bytes to read</li> </ul>	

**Table 46. AT+FileWrite Write Block of Data to a File in Storage Device**

Request:	Response:
AT+FileWrite= [FileID], [Offset],[Format],[Length],[Data]	+FileWrite:[NumberOfWrittenBytes] OK
Arguments: <ul style="list-style-type: none"> <li>• FileID: Assigned from AT+FileOpen</li> <li>• Offset: Offset to specific block to be written</li> <li>• Format: Data format: <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• Length: Number of bytes to write</li> <li>• Data: Transmitted data to the storage device</li> </ul>	

## 6.5 Network Application Commands

Activate networking applications, such as:

- HTTP Server
- DHCP Server
- Ping
- DNS
- mDNS

**Table 47. AT+NetAPPStart Starts a Network Application**

Request:	Response:
AT+NetAPPStart = [APP Bitmap]	OK
Arguments: <ul style="list-style-type: none"> <li>• APP Bitmap: Application bitmap, could be one or a combination of the following with OR (" ") between them:               <ul style="list-style-type: none"> <li>– HTTP_SERVER</li> <li>– DHCP_SERVER</li> <li>– MDNS</li> <li>– DNS_SERVER</li> </ul> </li> </ul>	

**Table 48. AT+NetAPPStop Stops a Network Application**

Request:	Response:
AT+NetAPPStop = [APP Bitmap]	OK
Arguments: <ul style="list-style-type: none"> <li>• APP Bitmap: Application bitmap, could be one or a combination of the following with OR (" ") between them:               <ul style="list-style-type: none"> <li>– HTTP_SERVER</li> <li>– DHCP_SERVER</li> <li>– MDNS</li> <li>– DNS_SERVER</li> </ul> </li> </ul>	

**Table 49. AT+NetAPPGetHostByName Get Host IP by Name**

Request:	Response:
AT+NetAPPGetHostByName = [HostName],[Family]	OK +NetAPPGetHostByName: [HostName], [Host IP address]
Arguments: <ul style="list-style-type: none"> <li>• HostName</li> <li>• Family: Protocol Family:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> </ul>	Arguments: <ul style="list-style-type: none"> <li>• HostName</li> <li>• Host IP address: IP address according to the family (IPv4 or IPv6)</li> </ul>

**Table 50. AT+NetAPPGetHostByService Get Host IP by Service**

Request:	Response:
AT+NetAPPGetHostByService = [ServiceName],[Family]	OK +NetAPPGetHostByService: [ServiceName],[Port],[HostIPAddress],[Text ]
Arguments: <ul style="list-style-type: none"> <li>• ServiceName: Service name can be full or partial</li> <li>• Family: Protocol Family:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> </ul>	Arguments: <ul style="list-style-type: none"> <li>• ServiceName</li> <li>• Port: Service port</li> <li>• HostIPAddress: Host IP address (IPv4 or IPv6)</li> <li>• Text: Text of the service full or partial</li> </ul>

**Table 51. AT+NetAPPSet Setting Network Application Configurations**

Request:			Response:
AT+NetAPPSet = [App ID],[Option],[Value1],...,[ValueX]			OK
Arguments:			
App ID	Option	Values	
DHCP_SERVER	BASIC	<ul style="list-style-type: none"> <li>Value1: Lease time (in seconds) of the IP Address</li> <li>Value2: First IP Address for allocation</li> <li>Value3: Last IP Address for allocation</li> </ul>	
	PRIM_PORT_NUM	Value1: port number	
HTTP_SERVER	AUTH_CHECK	Value1: <ul style="list-style-type: none"> <li>1: Authentication enable</li> <li>0: Authentication disable</li> </ul>	
	AUTH_NAME	Value1: Authentication name (maximum length is 20 bytes)	
	AUTH_PASSWORD	Value1: Authentication password (maximum length is 20 bytes)	
	AUTH_REALM	Value1: Authorization realm (maximum length is 20 bytes)	
	ROM_PAGES_ACCESS	Value1: <ul style="list-style-type: none"> <li>1: Access enable</li> <li>0: Access disable</li> </ul>	
	SECOND_PORT_NUM	Value1: port number	
	SECOND_PORT_EN	Value1: <ul style="list-style-type: none"> <li>1: Enable</li> <li>0: Disable</li> </ul>	
	PRIM_PORT_SEC_EN	Value1: <ul style="list-style-type: none"> <li>1: Enable</li> <li>0: Disable</li> </ul>	
	PRIV_KEY_FILE	Value1: File name (maximum length is 96 bytes)	
	DEV_CERT_FILE	Value1: File name (maximum length is 96 bytes)	
	CA_CERT_FILE	Value1: File name (maximum length is 96 bytes)	
	TMP_REGISTER_SERVICE	Value1: Service name for MDNS (maximum length is 80 bytes)	
	TMP_UNREGISTER_SERVICE	Value1: Service name for MDNS (maximum length is 80 bytes)	

**Table 51. AT+NetAPPSet Setting Network Application Configurations (continued)**

Request:		Response:
<i>MDNS</i>	<i>CONT_QUERY</i>	Value1: Service name (maximum length is 80 bytes)
	<i>QEVETN_MASK</i>	Value1: Event mask: <ul style="list-style-type: none"> <li>• ipp</li> <li>• deviceinfo</li> <li>• http</li> <li>• https</li> <li>• workstation</li> <li>• guid</li> <li>• h323</li> <li>• ntp</li> <li>• objective</li> <li>• rdp</li> <li>• remote</li> <li>• rtsp</li> <li>• sip</li> <li>• smb</li> <li>• soap</li> <li>• ssh</li> <li>• telnet</li> <li>• tftp</li> <li>• xmpp</li> <li>• raop</li> </ul>
	<i>TIMING_PARAMS</i>	<ul style="list-style-type: none"> <li>• Value1: Period in ticks (100 ticks = 1 second)</li> <li>• Value2: Repetitions</li> <li>• Value3: Telescopic factor</li> <li>• Value4: Retransmission interval</li> <li>• Value5: Maximum period interval</li> <li>• Value6: Maximum time</li> </ul>
<i>DEVICE</i>	<i>URN</i>	Value1: device name (maximum length is 33 bytes)
	<i>DOMAIN</i>	Value1: domain name (maximum length is 63 bytes)
<i>DNS_CLIENT</i>	<i>TIME</i>	<ul style="list-style-type: none"> <li>• Value1: Maximum response time in milliseconds</li> <li>• Value2: Number of retries</li> </ul>

**Table 52. AT+NetAPPGet Getting Network Applications Configurations**

Request:		Response:
AT+NetAPPGet = [App ID],Option]		+NetAPPGet: [return values] OK
Arguments:		Arguments: See AT+NetAPPSet command values
App ID	Option	
DHCP_SERVER	BASIC	
HTTP_SERVER	PRIM_PORT_NUM	
	AUTH_CHECK	
	AUTH_NAME	
	AUTH_PASSWORD	
	AUTH_REALM	
	ROM_PAGES_ACCESS	
	SECOND_PORT_NUM	
	SECOND_PORT_EN	
	PRIM_PORT_SEC_EN	
MDNS	CONT_QUERY	
	QEVETN_MASK	
	TIMING_PARAMS	
DEVICE	URN	
	DOMAIN	
DNS_CLIENT	TIME	

**Table 53. AT+NetAPPSend Sends Network Application Response or Data Following a Network Application Request Event**

Request:	Response:
AT+NetAPPSend = [Handle],[Flags],[Format],[Length],[Data]	OK
Arguments: <ul style="list-style-type: none"> <li>• Handle: Handle to send the data to. Should match the handle received in the Network Application request event</li> <li>• Flags: Bit mask:               <ul style="list-style-type: none"> <li>– CONTINUATION: More data will arrive in subsequent calls to AT+NetAPPSend</li> <li>– METADATA: Define data as metadata, otherwise data is payload</li> <li>– ACCUMULATION: The network processor should accumulate the data chunks and will process it when it is completely received</li> </ul> </li> <li>• Format: Data format:               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• Length: Number of bytes to send</li> <li>• Data: Data to send. Can be just data payload or metadata (depends on flags)</li> </ul>	

**Table 54. AT+NetAPPRecv Receives Data From the Network Processor Following a Network Application Response Event**

Request:	Response:
AT+NetAPPRecv = [Handle],[Format],[Length]	OK + NetAPPRecv:[Handle],[Flags],[Length],[Data]
Arguments: <ul style="list-style-type: none"> <li>• Handle: Handle to receive data from. Should match the handle receive in the Network Application request event</li> <li>• Format: Data format:               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• Length: Number of bytes to receive</li> </ul>	Arguments: <ul style="list-style-type: none"> <li>• Handle</li> <li>• Flags: Can have the following value:</li> <li>• <b>CONTINUATION</b>: More data is pending in the network processor. Application should continue reading the data by calling AT+NetAPPRecv again</li> <li>• Length: Number of bytes received</li> <li>• Data: Data received</li> </ul>

**Table 55. AT+NetAPPPing Send Ping to Network Hosts**

Request:	Response:
AT+NetAPPPing = [Family], [Destination], [Size], [Delay], [Timeout], [Max], [Flags]	OK +NetAPPPing: [PacketsSent],[PacketsReceived],[RoundTime]
Arguments: <ul style="list-style-type: none"> <li>• Family:               <ul style="list-style-type: none"> <li>– <b>INET</b>: For network protocol IPv4</li> <li>– <b>INET6</b>: For network protocol IPv6</li> </ul> </li> <li>• Destination: Destination IP address. For stopping an ongoing ping activity, set destination to 0</li> <li>• Size: Size of ping, in bytes</li> <li>• Delay: Delay between pings, in milliseconds</li> <li>• Timeout: Timeout for every ping in milliseconds</li> <li>• Max: Maximum number of ping requests               <ul style="list-style-type: none"> <li>– 0: Forever</li> </ul> </li> <li>• Flags:               <ul style="list-style-type: none"> <li>– Set to 0: Ping reports back once all requested pings are done</li> <li>– Set to 1: Ping reports back after every ping</li> <li>– Set to 2: Ping stops after the first successful ping and reports back for the successful ping, as well as any preceding failed pings</li> </ul> </li> </ul>	

**Table 56. AT+NetAPPGetServiceList Get Service List**

Request:	Response:
AT+NetAPPGetServiceList = [IndexOffset],[MaxServiceCount],[Flags]	+NetAPPGetServiceList:[ServiceInfo1];...:[ServiceInfoX] OK
Arguments: <ul style="list-style-type: none"> <li>• IndexOffset: The start index in the peer cache that from it the first service is returned</li> <li>• MaxServiceCount: The maximum services that can be returned if existed or if not exceed the maximum index in the peer cache</li> <li>• Flags: Which service to use (means which types of service to fill)                             <ul style="list-style-type: none"> <li>– FULL_IPV4_WITH_TEXT</li> <li>– FULL_IPV4</li> <li>– SHORT_IPV4</li> <li>– FULL_IPV6_WITH_TEXT</li> <li>– FULL_IPV6</li> <li>– SHORT_IPV6</li> </ul> </li> </ul>	Arguments: ServiceInfo: Depends on flag type: <ul style="list-style-type: none"> <li>• SHORT_IPV4                             <ul style="list-style-type: none"> <li>– ip</li> <li>– port</li> </ul> </li> <li>• FULL_IPV4</li> <li>• FULL_IPV6                             <ul style="list-style-type: none"> <li>– ip</li> <li>– port</li> <li>– service name</li> <li>– service host name</li> </ul> </li> <li>• FULL_IPV4_WITH_TEXT</li> <li>• FULL_IPV6_WITH_TEXT                             <ul style="list-style-type: none"> <li>– ip</li> <li>– port</li> <li>– service name</li> <li>– service host name</li> <li>– service text</li> </ul> </li> </ul>

**Table 57. AT+NetAPPRegisterService Register a New mDNS Service**

Request:	Response:
AT+NetAPPRegisterService= [ServiceName], [Text], [Port], [TTL], [Options]	OK
Arguments: <ul style="list-style-type: none"> <li>• ServiceName: The service name</li> <li>• Text: The description of the service</li> <li>• Port: The port on this target host port</li> <li>• TTL: The TTL of the service</li> <li>• Options: Bitwise parameters:                             <ul style="list-style-type: none"> <li>– <i>IS_UNIQUE_BIT</i>: Service is unique per interface (means that the service needs to be unique)</li> <li>– <i>IPV6_IPV4_SERVICE</i>: Add this service to IPv6 interface, if exist (default is IPv4 service only)</li> <li>– <i>IPV6_ONLY_SERVICE</i>: Add this service to IPv6 interface, but remove it from IPv4 (only IPv6 is available)</li> <li>– <i>UPDATE_TEXT</i>: For update text fields (without reregistering the service)</li> <li>– <i>IS_NOT_PERSISTENT</i>: For setting a nonpersistent service</li> </ul> </li> </ul>	

**Table 58. AT+NetAPPUnRegisterService Unregister mDNS Service**

Request:	Response:
AT+NetAPPUnRegisterService= [ServiceName], [Options]	OK
Arguments: <ul style="list-style-type: none"> <li>• ServiceName: Full service name</li> <li>• Options: Bitwise parameters:                             <ul style="list-style-type: none"> <li>– <i>IS_UNIQUE_BIT</i>: Service is unique per interface (means that the service needs to be unique)</li> <li>– <i>IPV6_IPV4_SERVICE</i>: Add this service to IPv6 interface, if exist (default is IPv4 service only)</li> <li>– <i>IPV6_ONLY_SERVICE</i>: Add this service to IPv6 interface, but remove it from IPv4 (only IPv6 is available)</li> <li>– <i>UPDATE_TEXT</i>: For update text fields (without reregistering the service)</li> <li>– <i>IS_NOT_PERSISTENT</i>: For setting a nonpersistent service</li> </ul> </li> </ul>	

## 6.6 Network Configuration Commands

The Network Configuration Commands control the configuration of the device addresses (that is, IP and MAC addresses).

**Table 59. AT+NetCfgSet Setting Network Configurations**

Request:			Response:
AT+NetCfgSet = [ConfigId],[ConfigOpt],[Value1],...,[ValueX]			OK
Arguments:			
ConfigId	ConfigOpt	Value	
<i>IF</i>	<i>STATE</i> Enable or disable modes (bitmask)	<ul style="list-style-type: none"> <li>• <i>IPV6_STA_LOCAL</i>: Enable ipv6 local</li> <li>• <i>IPV6_STA_GLOBAL</i>: Enable ipv6 global</li> <li>• <i>DISABLE_IPV4_DHCP</i>: Disable ipv4 DHCP</li> <li>• <i>IPV6_LOCAL_STATIC</i>: Enable ipv6 local static</li> <li>• <i>IPV6_LOCAL_STATELESS</i>: Enable ipv6 local stateless</li> <li>• <i>IPV6_LOCAL_STATEFUL</i>: Enable ipv6 local stateful</li> <li>• <i>IPV6_GLOBAL_STATIC</i>: Enable ipv6 global static</li> <li>• <i>IPV6_GLOBAL_STATEFUL</i>: Enable ipv6 global stateful</li> <li>• <i>DISABLE_IPV4_LLA</i>: Disable LLA feature</li> <li>• <i>ENABLE_DHCP_RELEASE</i>: Enables DHCP release</li> <li>• <i>IPV6_GLOBAL_STATELESS</i>: Enable ipv6 global stateless</li> <li>• <i>DISABLE_FAST_RENEW</i>: Fast renew disabled</li> </ul>	
<i>SET_MAC_ADDR</i> Setting MAC address to the Device	Ignore value	New MAC address	
<i>IPV4_STA_ADDR</i> Setting IP address	<i>STATIC</i> Setting a static IP address	<ul style="list-style-type: none"> <li>• Value1: IP address</li> <li>• Value2: Subnet mask</li> <li>• Value3: Default gateway address</li> <li>• Value4: DNS server address</li> </ul>	
	<i>DHCP</i> Setting IP address by DHCP	Ignore value	
	<i>DHCP_LLA</i> Setting DHCP LLA	Ignore value	
	<i>RELEASE_IP_SET</i> Setting release IP before disconnect enables sending a DHCP release frame to the server	Ignore value	
	<i>RELEASE_IP_OFF</i> Setting release IP before disconnect disables sending a DHCP release frame to the server	Ignore value	
<i>IPV4_AP_ADDR</i> Setting a static IP address to the device working in AP mode	<i>STATIC</i> Setting a static IP address	<ul style="list-style-type: none"> <li>• Value1: IP address</li> <li>• Value2: Subnet mask</li> <li>• Value3: Default gateway address</li> <li>• Value4: DNS server address</li> </ul>	

**Table 59. AT+NetCfgSet Setting Network Configurations (continued)**

Request:			Response:
IPV6_ADDR_LOCAL	STATIC Setting a IPv6 Local static address	IP address	
	STATELESS Setting a IPv6 Local stateless address	Ignore value	
	STATEFUL Setting a IPv6 Local stateful address	Ignore value	
IPV6_ADDR_GLOBAL	STATIC Setting a IPv6 Global static address Value1 : IP address Value2: DNS Server IP STATEFUL	<ul style="list-style-type: none"> <li>Value1: IP address</li> <li>Value2: DNS Server IP</li> </ul>	
	STATEFUL Setting a IPv6 Global stateful address	Ignore value	
AP_STATION_DISCONNECT Disconnect AP station by MAC address	Ignore value	AP MAC address	
IPV4_DNS_CLIENT Set secondary DNS address	Ignore value	Secondary DNS Server address	

**Table 60. AT+NetCfgGet Getting Network Configurations**

Request:	Response:
AT+NetCfgGet = [ConfigId]	+NetCfgGet:[Value1],...,[ValueX] OK
Arguments: ConfigId: Configuration ID:	Arguments:
GET_MAC_ADDR Get the device MAC address	Value1: MAC address
IPV4_STA_ADDR Get IP address from WLAN station or P2P client	<ul style="list-style-type: none"> <li>Value1: Address option:               <ul style="list-style-type: none"> <li>- DHCP</li> <li>- DHCP_LLA</li> <li>- STATIC</li> </ul> </li> <li>Value2: Address</li> <li>Value3: Subnet mask</li> <li>Value4: Gateway</li> <li>Value5: DNS</li> </ul>
IPV4_AP_ADDR Get static IP address for AP or P2P go	
IF Get interface bitmap	Value1: State (bitmask): <ul style="list-style-type: none"> <li>ipv6_sta_local</li> <li>ipv6_sta_global</li> <li>disable_ipv4_dhcp</li> <li>ipv6_local_static</li> <li>ipv6_local_stateless</li> <li>ipv6_local_stateful</li> <li>ipv6_global_static</li> <li>ipv6_global_stateful</li> <li>disable_ipv4_lls</li> <li>enable_dhcp_release</li> <li>ipv6_global_stateless</li> <li>disable_fast_renew</li> </ul>

**Table 60. AT+NetCfgGet Getting Network Configurations (continued)**

Request:	Response:
<i>IPV6_ADDR_LOCAL</i> Get IPV6 Local address	<ul style="list-style-type: none"> <li>• Value1: Address option:               <ul style="list-style-type: none"> <li>– stateless</li> <li>– stateful</li> <li>– STATIC</li> </ul> </li> <li>• Value2: Address</li> </ul>
<i>IPV6_ADDR_GLOBAL</i> Get IPV6 Global address	
<i>AP_STATIONS_CONNECTED</i> Get AP number of connected stations	Value1: Number of connected stations
<i>AP_STATIONS_INFO</i> Get AP full list of connected stations	[address1],[MAC address1],[name1]; ...; [addressX],[MAC addressX],[nameX]
<i>IPV4_DNS_CLIENT</i> Set secondary DNS address	Value1: DNS second server address
<i>IPV4_DHCP_CLIENT</i> Get DHCP Client info	<ul style="list-style-type: none"> <li>• Value1: Address</li> <li>• Value2: Subnet mask</li> <li>• Value3: Gateway</li> <li>• Value4: DNS 1</li> <li>• Value5: DNS 2</li> <li>• Value6: DHCP server</li> <li>• Value7: Lease time</li> <li>• Value8: Time to renew</li> <li>• Value9: DHCP State:               <ul style="list-style-type: none"> <li>– unknown</li> <li>– disabled</li> <li>– enabled</li> <li>– bound</li> <li>– renew</li> <li>– rebind</li> </ul> </li> </ul>

## 6.7 Network Utility Commands

Networking related commands and configuration follow.

**Table 61. AT+NetUtilGet Getting Utilities Configurations**

Request:		Response:
AT+NetUtilGet =[ID], [Option]		+NetUtilGet: [Value] OK
Arguments:		Arguments:
ID Identifier of the specific "get" operation to perform	Option	Value
<i>public_key</i>	<ul style="list-style-type: none"> <li>• 0: Binary data format</li> <li>• 1: Base64 data format (binary to text encoding)</li> </ul>	Key (maximum length is 255 bytes or 370 bytes in base64 format)
<i>true_random</i>	Number of random numbers (maximum is 172 numbers)	List of random numbers

**Table 62. AT+NetUtilCmd Performing Utilities-Related Commands**

Request:		Response:
AT+NetUtilCmd = [Cmd],[Value1],...[ValueX]		+NetUtilCmd:[Value1],...[ValueX] OK
Arguments:		Arguments:
Cmd	Value	
Identifiers of the specific command to perform		
<i>sign_msg</i> Create a digital signature using the ECDSA algorithm	<ul style="list-style-type: none"> <li>• Value1: Key index</li> <li>• Value2: Data format               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• Value3: Data length</li> <li>• Value4: Data (maximum length is 1500 bytes)</li> </ul>	<ul style="list-style-type: none"> <li>• Value1: Signature length (maximum is 255 bytes)</li> <li>• Value2: Signature</li> </ul>
<i>verify_msg</i> verify a digital signature using the ECDSA algorithm	<ul style="list-style-type: none"> <li>• Value1: Key index</li> <li>• Value2: Data and signature format               <ul style="list-style-type: none"> <li>– 0: Binary data format</li> <li>– 1: Base64 data format (binary to text encoding)</li> </ul> </li> <li>• Value3: Data length (maximum length is 1500 bytes)</li> <li>• Value4: Signature length</li> <li>• Value5: Data and signature (signature concatenate to end of data)</li> </ul>	Value1: Success or failure
<i>temp_keys</i> Create or remove a temporary ECC key pair with the SECP256R1 curve	<ul style="list-style-type: none"> <li>• Value1: Key index</li> <li>• Value2: Action:               <ul style="list-style-type: none"> <li>– <i>create</i></li> <li>– <i>remove</i></li> </ul> </li> </ul>	
<i>install_op</i> Install or uninstall a key pair in one of the crypto utilities key pair management mechanism	<ul style="list-style-type: none"> <li>• Value1: Key index</li> <li>• Value2: Action:               <ul style="list-style-type: none"> <li>– <i>install</i></li> <li>– <i>uninstall</i></li> </ul> </li> <li>• Value3: Key Algorithm (ignored for uninstall action)               <ul style="list-style-type: none"> <li>– <i>none</i></li> <li>– <i>ec</i></li> </ul> </li> <li>• Value4: EC Named Curve identifier (optional for Key Algorithm none) (ignored for uninstall action)               <ul style="list-style-type: none"> <li>– <i>none</i></li> <li>– <i>secp256r1</i></li> </ul> </li> <li>• Value5: Certification file name (ignored for uninstall action)</li> <li>• Value6: Key file name (ignored for uninstall action)</li> </ul>	

## 6.8 Asynchronous Events

**Table 63. +EventFatalError Fatal Error Event for Inspecting Fatal Error**

Response:	
+EventFatalError:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>DEVICE_ABORT</i> Indicates a severe error occurred and the device stopped	<ul style="list-style-type: none"> <li>Value1: An indication of the abort type</li> <li>Value2: The abort data</li> </ul>
<i>NO_CMD_ACK</i> Indicates that the command sent to the device had no ACK	Value1: An indication of the CMD opcode
<i>CMD_TIMEOUT</i> Indicates that the command got a timeout while waiting for its asynchronous response	Value1: An indication of the asynchronous event opcode
<i>DRIVER_ABORT</i> Indicates a severe error occurred in the driver	null
<i>SYNC_LOSS</i> Indicates a sync loss with the device	null

**Table 64. +EventGeneral General Asynchronous Event for Inspecting General Events**

Response:	
+EventGeneral:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>RESET_REQUEST</i>	<ul style="list-style-type: none"> <li>Value1: An error code indication from the device</li> <li>Value2: The sender originator:               <ul style="list-style-type: none"> <li>WLAN</li> <li>NETCFG</li> <li>NETAPP</li> <li>SECURITY</li> <li>OTHER</li> </ul> </li> </ul>
<i>ERROR</i>	<ul style="list-style-type: none"> <li>Value1: An error code indication from the device</li> <li>Value2: The sender originator</li> </ul>

**Table 65. +EventWlan WLAN Asynchronous Event**

Response:	
+EventWlan:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>CONNECT</i> STA connection indication event	<ul style="list-style-type: none"> <li>Value1: SSID name</li> <li>Value2: BSSID</li> </ul>
<i>P2P_CONNECT</i> P2P client connection indication event	<ul style="list-style-type: none"> <li>Value1: SSID name</li> <li>Value2: BSSID</li> <li>Value3: Go Device Name</li> </ul>
<i>DISCONNECT</i> STA client disconnection event	<ul style="list-style-type: none"> <li>Value1: SSID name</li> <li>Value2: BSSID</li> <li>Value3: Reason</li> </ul>

**Table 65. +EventWlan WLAN Asynchronous Event (continued)**

<b>Response:</b>	
<i>P2P_DISCONNECT</i> P2P client disconnection event	<ul style="list-style-type: none"> <li>Value1: SSID name</li> <li>Value2: BSSID</li> <li>Value3: Reason</li> <li>Value4: Go Device Name</li> </ul>
<i>STA_ADDED</i> AP connected STA	Value1: MAC address
<i>STA_REMOVED</i> AP disconnected STA	Value1: MAC address
<i>P2P_CLIENT_ADDED</i> P2P(Go) connected P2P(Client)	<ul style="list-style-type: none"> <li>Value1: MAC address</li> <li>Value2: Go Device Name</li> <li>Value3: Own SSID</li> </ul>
<i>P2P_CLIENT_REMOVED</i> P2P(Go) disconnected P2P(Client)	<ul style="list-style-type: none"> <li>Value1: MAC address</li> <li>Value2: Go Device Name</li> <li>Value3: Own SSID</li> </ul>
<i>P2P_DEVFOUND</i>	<ul style="list-style-type: none"> <li>Value1: Go Device Name</li> <li>Value2: MAC address</li> <li>Value3: WPS Method</li> </ul>
<i>P2P_REQUEST</i>	<ul style="list-style-type: none"> <li>Value1: Go Device Name</li> <li>Value2: MAC address</li> <li>Value3: WPS Method</li> </ul>
<i>P2P_CONNECTFAIL</i> P2P only	Value1: Status
<i>PROVISIONING_STATUS</i>	Value1: Status
<i>PROVISIONING_PROFILE_ADDED</i>	<ul style="list-style-type: none"> <li>Value1: Status</li> <li>Value2: SSID name</li> </ul>

**Table 66. +EventNetApp Network Application Asynchronous Event**

<b>Response:</b>	
+EventNetApp:[EventID],[Value1],...,[ValueX]	
Arguments:	
<b>EventID</b>	<b>Value</b>
<i>IPV4_ACQUIRED</i>	<ul style="list-style-type: none"> <li>Value1: IP address</li> <li>Value2: Gateway</li> <li>Value3: DNS</li> </ul>
<i>IPV6_ACQUIRED</i>	<ul style="list-style-type: none"> <li>Value1: IP address</li> <li>Value2: DNS</li> </ul>
<i>ip_collision</i>	<ul style="list-style-type: none"> <li>Value1: IP address</li> <li>Value2: DHCP MAC</li> <li>Value3: DNS</li> </ul>
<i>IP_LEASED</i> AP or P2P go DHCP lease event	<ul style="list-style-type: none"> <li>Value1: IP address</li> <li>Value2: Lease time</li> <li>Value3: MAC</li> </ul>
<i>IP_RELEASED</i> AP or P2P go DHCP IP release event	<ul style="list-style-type: none"> <li>Value1: IP address</li> <li>Value2: MAC</li> <li>Value3: Reason</li> </ul>
<i>IPV4_LOST</i>	Value1: Status
<i>dhcp_ipv4_acquire_timeout</i>	Value1: Status
<i>IPV6_LOST</i>	Value1: IP lost

**Table 67. +EventSock Socket Asynchronous Event**

Response:	
+EventSock:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
TX_FAILED	<ul style="list-style-type: none"> <li>• Value1: sd</li> <li>• Value2: Status</li> </ul>
ASYNC_EVENT	<ul style="list-style-type: none"> <li>• Value1: sd</li> <li>• Value2: Type:               <ul style="list-style-type: none"> <li>– SSL_ACCEPT</li> <li>– RX_FRAG_TOO_BIG</li> <li>– OTHER_SIDE_CLOSE_SSL</li> <li>– CONNECTED_SECURED</li> <li>– WRONG_ROOT_CA</li> </ul> </li> <li>• Value3: Error value</li> </ul>

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2017, Texas Instruments Incorporated