



WE866E4-P AT Command Reference Guide

80572ST10887A Rev. 0.0 – 2018-09-07

TELIT
TECHNICAL
DOCUMENTATION

APPLICABILITY TABLE

PRODUCT
WE866E4-P

Software Release
7.0.0A

TELIT MAY MAKE CHANGES TO SPECIFICATIONS AND PRODUCT DESCRIPTIONS AT ANY TIME, WITHOUT NOTICE.

NOTICES

LIMITATION OF LIABILITY

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in the content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

Telit assumes no liability whatsoever, and disclaims any express or implied warranty, relating to sale and/or use of Telit products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Telit products are not authorized for use as critical components in medical, lifesaving, or life-sustaining applications.

COPYRIGHT

This Telit manual and contents are owned by Telit and/or its licensors and protected by U.S. and international copyright laws, conventions, and treaties. Your right to use this manual is subject to limitations and restrictions imposed by applicable licenses and copyright laws. Unauthorized reproduction, modification, distribution, display or other use of this manual may result in criminal and civil penalties.

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in Italy and other countries preserve for Telit and its licensors certain exclusive rights to copyright material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

USAGE AND DISCLOSURE RESTRICTIONS

I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit.

III. High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High-Risk Activities.

IV. Trademarks

Telit, Telit and stylized Logos are registered Trademarks. All other product or service names are the property of their respective owners.

V. Third Party Rights

The software may include Third Party Right software. In this case, you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY "OTHER CODE"), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

www.Telit.com

DOCUMENT HISTORY

Revision	Date	Changes
0.0	2018-09-07	First Issue.

CONTENTS

NOTICES.....	3
LIMITATION OF LIABILITY	3
COPYRIGHT	3
USAGE AND DISCLOSURE RESTRICTIONS	4
I. License Agreements	4
II. Copyrighted Materials	4
III. High Risk Materials	4
IV. Trademarks	4
V. Third Party Rights	4
DOCUMENT HISTORY	6
CONTENTS	8
1. INTRODUCTION.....	14
1.1. Scope.....	14
1.2. Audience	14
1.3. Contact Information and Support.....	14
1.4. Text Convention	14
1.5. Related Documents.....	15
2. OVERVIEW.....	17
2.1. AT Commands	17
2.2. Command Definition.....	17
2.3. AT Command Syntax	18
2.4. Information Response and Result Codes	19
3. ARCHITECTURE.....	20
3.1. System Overview	20
3.2. Using SDK Builder	21
3.2.1. S2W Module Information.....	21
3.3. Working Principle	21
3.4. System Initialization	21
4. HOST INTERACTION.....	23
4.1. Interface	23
4.1.1. Configure UART	23
5. GENERAL OPERATIONS	25
5.1. System Settings	25

5.1.1.	Auto Mode.....	25
5.1.2.	Data Mode	26
5.1.3.	Echo Mode.....	26
5.1.4.	Verbose Mode.....	27
5.1.5.	File System	28
5.2.	Profile Settings.....	39
5.2.1.	Read Profile	39
5.2.2.	Save Profile.....	40
5.2.3.	Set Profile	41
5.2.4.	TCP Profile.....	42
5.2.5.	UDP Profile	43
5.3.	RTC Settings.....	44
5.3.1.	Time Settings	44
5.3.2.	Reset	45
5.3.3.	Version.....	45
5.4.	SNTP Settings	46
5.4.1.	Configure SNTP	46
5.4.2.	Start SNTP	47
6.	WIRELESS	49
6.1.	WLAN	49
6.1.1.	Application Information Element.....	49
6.1.2.	Set Beacon Interval.....	50
6.1.3.	Get Country Code	52
6.1.4.	Set Country Code.....	53
6.1.5.	Disconnect	55
6.1.6.	Interface Initilisation	56
6.1.7.	Get MAC Address	57
6.1.8.	Get Physical Mode	58
6.1.9.	Set Physical Mode	60
6.1.10.	Power Save.....	61
6.1.11.	Send Raw Packets.....	63
6.1.12.	Get Regulatory Domain.....	66
6.1.13.	Set Regulatory Domain	67
6.1.14.	Get RSSI.....	69
6.1.15.	Scan.....	70
6.1.16.	Get Power Transmission	72
6.1.17.	Set Power Transmission	73

6.1.18.	Get Transmission Rate.....	74
6.1.19.	Set Transmission Rate.....	76
6.1.20.	WPS.....	77
7.	NETWORKS.....	80
7.1.	CID	80
7.1.1.	Get CID Information	80
7.1.2.	CID Status Information.....	81
7.2.	DNS	83
7.2.1.	NDNS URL.....	83
7.2.2.	NDNS IP Address	84
7.2.3.	NDNS Host Addition.....	85
7.2.4.	NDNS Server Start.....	86
7.2.5.	NDNS Server Stop.....	87
7.2.6.	NDNS Service Discovery	88
7.2.7.	NDNS Target Information.....	89
8.	DATA TRANSFER.....	92
8.1.	WNCM	92
8.1.1.	Configure IP Address	92
8.1.2.	Configure IP Address in AP Mode.....	93
8.1.3.	NCM Connect	95
8.1.4.	Configure AP.....	97
8.1.5.	Create NCM	99
8.1.6.	Initialize NCM.....	102
8.1.7.	Interface Statistics in AP Mode.....	103
8.1.8.	Get Associated Station Information	107
8.1.9.	Interface Configuration Information	109
8.1.10.	Interface Statistics in Station Mode	112
8.1.11.	NCM Disconnect	115
8.2.	Socket.....	117
8.2.1.	Create Socket	117
8.2.2.	Bind Socket.....	118
8.2.3.	Connect Socket.....	119
8.2.4.	Socket Listen	120
8.2.5.	Set Socket Option	122
8.2.6.	Get Socket Option.....	125
8.2.7.	Accept Socket	129
8.2.8.	Send Data to Socket	130

8.2.9.	Socket Receive Ready	132
8.2.10.	Receive Socket	134
8.2.11.	Get Socket Error	136
8.2.12.	Socket Throughput Test	138
8.2.13.	Close Socket	140
8.3.	SSL	141
8.3.1.	Initialize SSL	141
8.3.2.	Configure SSL	142
8.3.3.	SSL Connect	144
8.3.4.	Bind SSL	146
8.3.5.	SSL Listen	147
8.3.6.	Accept SSL	148
8.3.7.	Send Data to SSL	150
8.3.8.	SSL Receive Ready	151
8.3.9.	Read SSL	153
8.3.10.	Write SSL	155
8.3.11.	Close SSL	157
8.3.12.	SSL Certificate List	158
8.3.13.	SSL Certificate Store	159
8.3.14.	SSL Certificate Deletion	161
8.4.	HTTPC (Client)	162
8.4.1.	Initialize HTTP Client	162
8.4.2.	Configure HTTP Client	163
8.4.3.	HTTP Client Connect	165
8.4.4.	HTTP Client Request	166
8.4.5.	Read HTTP Client	168
8.4.6.	Close HTTP Client	169
8.5.	HTTPD (Server)	171
8.5.1.	Initialize HTTP Server	171
8.5.2.	Configure HTTP Server	172
8.5.3.	Start/Stop/Close HTTP Server connection	174
8.5.4.	Configure HTTP Server URI	175
8.5.5.	Read HTTP Server	176
8.5.6.	Send Data to HTTP Server	178
8.5.7.	Send Response to HTTP Server Request	180
8.6.	MQTT	181
8.6.1.	MQTT Client Connect	181

8.6.2.	MQTT Client Publish	184
8.6.3.	Receive Request from MQTT Client.....	186
8.6.4.	MQTT Client Subscribe	188
8.6.5.	MQTT Client Receive	189
8.6.6.	MQTT Client Disconnect	190
8.7.	Web Socket.....	191
8.7.1.	Configure WebSocket	191
8.7.2.	Create WebSocket	193
8.7.3.	WebSocket Connect	196
8.7.4.	Send WebSocket Data.....	197
8.7.5.	WebSocket Ping	198
8.7.6.	WebSocket Pong	200
8.7.7.	Receive Ready WebSocket Data	201
8.7.8.	Receive WebSocket Data	203
8.7.9.	Close WebSocket.....	205
9.	ADVANCED SERVICES.....	207
9.1.	Network.....	207
9.1.1.	MDNS	207
9.1.2.	Ping.....	213
9.2.	Application Features	214
9.2.1.	Firmware Upgrade	214
9.2.2.	Provisioning	218
9.2.3.	TR50.....	220

1. INTRODUCTION

1.1. Scope

This document is aimed in providing a detailed specification and a comprehensive listing as a reference for the whole set of AT command

1.2. Audience

This document is intended for Telit customers, who are integrators, about to implement their applications using our WE866E4-P modules. Readers of this document should be familiar with Telit modules and their ease of

controlling by means of AT Commands.

1.3. Contact Information and Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:

- TS-SRD@telit.com

Alternatively, use:

<http://www.telit.com/support>

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

<http://www.telit.com>

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Text Convention



Danger – This information **MUST** be followed or catastrophic equipment failure or bodily injury may occur.



Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.



Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

1.5. Related Documents

Please refer to <https://www.telit.com/m2m-iot-products/wifi-bluetooth-modules/> for current documentation and downloads.

2. OVERVIEW

This chapter provides the guidelines for using AT command line interface to design, configure and provision WE866E4-P module in a Wi-Fi network using serial commands.

2.1. AT Commands

The Telit wireless module family can be controlled via the serial interface using the standard AT commands.

2.2. Command Definition

This document uses the following syntactical definitions:

- Special text fonts represents particular, commands, keywords, variables, or window sessions
- Color text indicates cross-reference hyperlinks to supplemental information
- Command notation indicates commands, subcommands, or command elements

Following table describes the text conventions used in this manual for software procedures that are explained using the AT command line interface.

Convention Type	Description
[] Square brackets	Enclose optional parameters. Choose none; or select one or more an unlimited number of times each. Do not enter brackets as part of any command. [parm1 parm2 parm3]
? Question mark	Used with the square brackets to limit the immediately following token to one occurrence.
<CR> Carriage return	Each command is terminated by a carriage return.
<LF> Line feed	Each command is terminated by a line feed.
<CR> <LF> Carriage return Line feed	Each response is started with a carriage return and line feed with some exceptions.
< > Angle brackets	Enclose a numeric range, endpoints inclusive. Do not enter angle brackets as part of any command. <SSID>
= Equal sign	Separates the variable from explanatory text. Is entered as part of the command. PROCESSID = <CID>

.	Allows the repetition of the element that immediately follows it multiple times. Do not enter as part of the command.
dot (period)	
	.AA:NN can be expanded to 1:01 1:02 1:03.
A.B.C.D	IPv4-style address.
IP address	
	10.0.11.123
LINE	Indicates user input of any string, including spaces. No other parameters may be entered after input for this token.
End-to-line input token	
	string of words
WORD	Indicates user input of any contiguous string (excluding spaces).
Single token	
	singlewordnospaces

2.3. AT Command Syntax

Following table describes the syntax rules followed by Telit implementation used in this manual for software procedures that are explained using the AT command line interface.

Convention Type	Description
command syntax monospaced font	This monospaced font represents command strings entered on a command line and sample source code. AT XXXX
Proportional font description	Gives specific details about a parameter. <Data> DATA
UPPERCASE Variable parameter	Indicates user input. Enter a value according to the descriptions that follow. Each uppercased token expands into one or more other token.
lowercase Keyword parameter	Indicates keywords. Enter values exactly as shown in the command description.
M/O	Gives the limitation of the parameter as mandatory or optional.

2.4. Information Response and Result Codes

TBD

3. ARCHITECTURE

This chapter provides a brief overview of the system and the architecture of S2W application.

3.1. System Overview

The S2W stack is used to provide Wi-Fi capability to any device having a serial interface. This approach offloads WLAN, TCP/IP stack and network management overhead to the Wi-Fi chip, allowing a small embedded host (for example an MCU) to communicate with other hosts on the network using a Wi-Fi wireless link. The host processor can use serial commands to configure the S2W application and to create wireless and network connections.

The user will have to register on Telit website, sign the NDA and check with the local sales team for any queries during this procedure. This gives access to all the respective documentation according to the product purchased.

The following is the basic application development sequence for a Serial-to-Wireless user.

- 1) Evaluate Telit hardware and firmware
 - Download the software, program, and execute.
 - To download the software, visit Telit website, and download all the latest packages including the binary (Refer WE866E4-P SDK Builder User Guide).
 - Flash the binary using module programming user guide on the custom hardware or on the evaluation board and execute in RUN mode. (Refer “WE866E4-P Module Programming User Guide”).
- 2) Design the custom hardware by following the design guidelines. (Refer WE866E4-P Hardware Design Guidelines)
- 3) Develop Host firmware
 - Interface host application using AT commands. (Refer WE866E4-P AT Command Reference Guide)
 - Configure the serial interface (UART/SPI/SDIO) as required for mode, polarity. For software interface, choose Command & Response, Byte stuffing/de-stuffing as “None” for UART and SDIO options.
 - Issue general, power save, and security related commands as required.
 - Start connection to an Access Point or do provisioning as required.
 - Obtain IP Address and start Data Transfer.
 - Select advanced services if any.
- 4) Debug Host and Telit module
 - Debug using provided AT commands and other options if required. (Refer WE866E4-P AT Command Reference Guide)
 - Analyze using Wire shark over wireless. For more details, refer information about AirPcap Nx in <http://www.riverbed.com>
- 5) Production Process
 - Perform generic recommendations in production line
 - Perform RF tests



NOTE:

Refer “WE866E4-P Use Case Reference Guide document for detailed use cases and examples

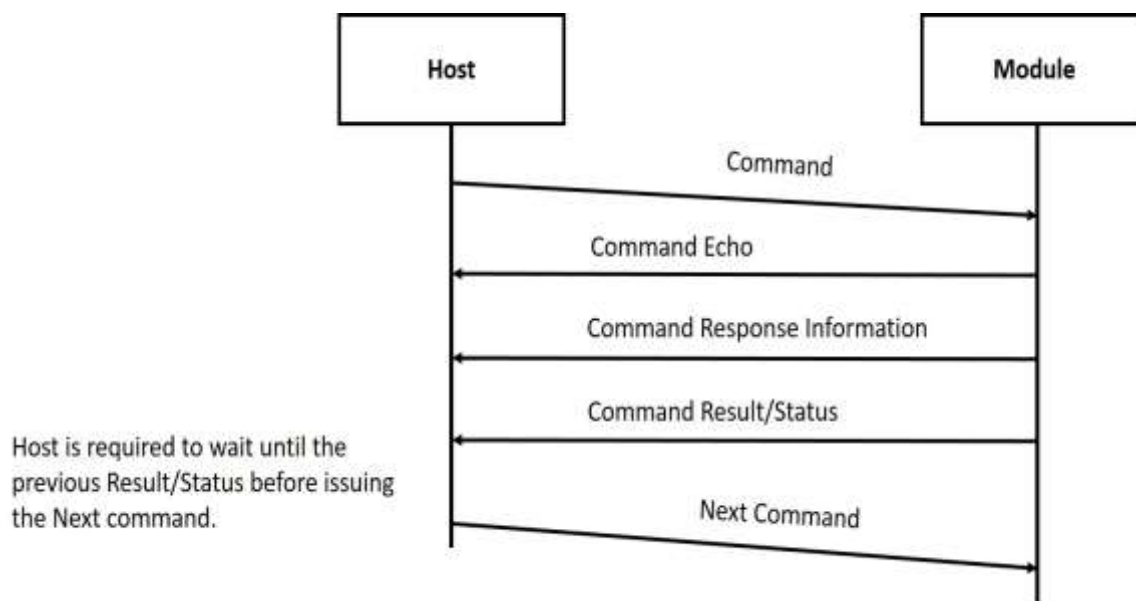
3.2. Using SDK Builder

3.2.1. S2W Module Information

Selecting the S2W (Hosted) under the SDK Builder Configuration screen displays the module information that includes the module selected, firmware version.

3.3. Working Principle

Following representation shows the communication between WE866E4-P Module and the Host:

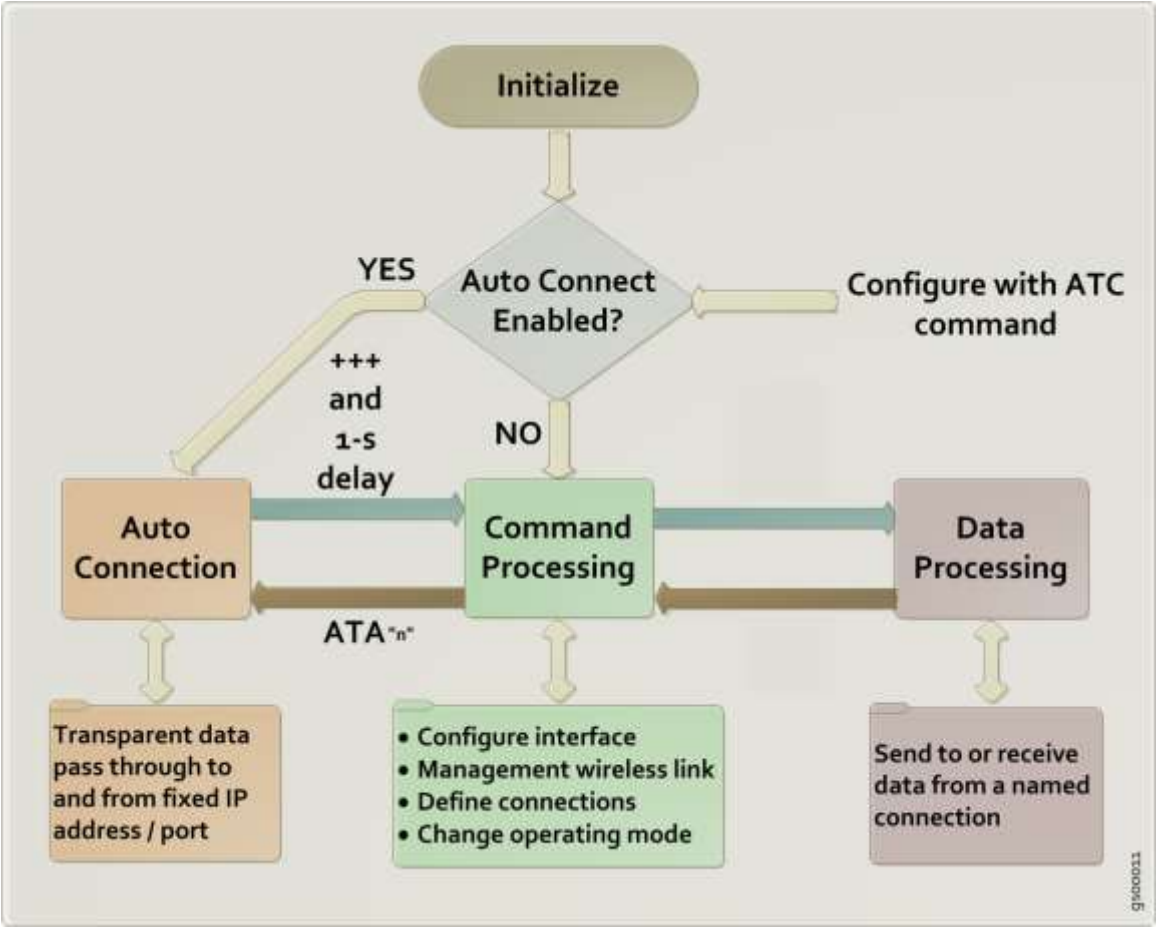


3.4. System Initialization

During the initialization process, the module software tries to fetch the configuration file (also called as profile) from the file system. If the profile file is not found, it sets the default values i.e. factory default values to profile and creates the file in the file system.

As per the default profile, it starts in AP mode initializing the provisioning software block. User can either configure the module through the provisioning mechanism or it can issue commands to start in STA mode and join to the intended AP.

In the profile, configuration related to wireless network, UDP/TCP/DNS/mDNS/HTTP/MQTT are kept. If the auto connection mode is set, then the module will attempt to join the wireless network and will try to open connection based on the UDP/TCP/HTTP/MQTT configuration. It configures the services and fetches based on the mDNS configuration.



The default UART baud rate is 115200 instead of 921600, using 8-bit characters with no parity bits and one stop bit.

4. HOST INTERACTION

4.1. Interface

4.1.1. Configure UART

Command Description:

To configure UART and set UART params

Pre-requisites:

Command Syntax:

ATB=<BaudRate>, [<BitsPerChar>,<ParityMode>,<StopBits>]

Parameter Description:

Parameters	Required?	Value	Description
BaudRate	M	Value Range: 110-921600 Format: UINT Default value: 115200	specifies UART params
BitsPerChar	O	Value Range: 0-3 Format: UINT Default value: 3	specifies bits per character
ParityMode	O	Value Range: 0-3 Format: UINT Default value: 0	specifies parity mode used
StopBits	O	Value Range: 0-3 Format: UINT Default value: 1	specifies number of stop bits

Response:

If the uart configuration is successful, it gives OK else returns FAILURE.

Responses	Remarks
ATB=115200,3,0,1<CR><LF> <CR><LF>OK<CR><LF>	Successful

ERROR

If parameter is not valid

Example:

ATB=115200,3,0,1

OK

5. GENERAL OPERATIONS

5.1. System Settings

5.1.1. Auto Mode

Command Description:

To enable/disable auto mode

Pre-requisites:

Command Syntax:

ATCn<enable>

Parameter Description:

Parameters	Required?	Value	Description
enable	M	Value Range: 0-1 Format: INT Default value: 1	enable/disable auto mode, 1 to enable, 0 to disable

Response:

Success if auto mode change operation is successful else failure.

Responses	Remarks
To disable the auto mode ATC0<CR><LF> <CR><LF>OK<CR><LF> To enable the auto mode ATC1<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To disable the auto mode

ATC0

OK

To enable the auto mode

ATC1

OK

5.1.2. Data Mode

Command Description:

To enable data mode

Pre-requisites:

none.

Command Syntax:

ATAn<CID>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: ""	enable data mode for CID

Response:

Success if verbos mode change operation is successful else failure.

Responses	Remarks
To enable data mode ATAn<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To enable data mode

ATAn

OK

5.1.3. Echo Mode

Command Description:

To enable/disable echo mode

Pre-requisites:

none.

Command Syntax:

```
ATEn<enable>
```

Parameter Description:

Parameters	Required?	Value	Description
enable	M	Value Range: 0-1 Format: INT Default value: 1	enable/disable the echo mode, 1 to enable, 0 to disable

Response:

Success if echo mode change operation is successful else failure.

Responses	Remarks
To disable the echo mode ATE0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To disable the echo mode

```
ATE0
```

```
OK
```

5.1.4. Verbose Mode

Command Description:

To enable/disable verbose mode

Pre-requisites:

none.

Command Syntax:

```
ATVn<enable>
```

Parameter Description:

Parameters	Required?	Value	Description
enable	M	Value Range: 0-1 Format: INT Default value: 1	enable/disable the verbose mode, 1 to enable, 0 to disable.

Response:

Success if verbose mode change operation is successful else failure.

Responses	Remarks
To disable the echo mode ATV1<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To disable the echo mode

ATV1

OK

5.1.5. File System

5.1.5.1. Close File

Command Description:

To close a file specified

Pre-requisites:

File should have been open before

Command Syntax:

AT+YFCL=<fd>

Parameter Description:

Parameters	Required?	Value	Description
fd	M	Value Range: 0X0-	File Descriptor (FD) is the value

0xFFFFFFFF Format: HEX Default value:	returned while opening file
--	-----------------------------

Response:

If file close operation is successful, returns success else failure.

Responses	Remarks
To open a file. AT+YFOP="/abc.txt",0<CR><LF> <CR><LF>+YFOP:0<CR><LF> <CR><LF>OK<CR><LF> To close a file. AT+YFCL=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To open a file.

```
AT+YFOP="/abc.txt",0
+YFOP:0
OK
```

To close a file.

```
AT+YFCL=0
OK
```

5.1.5.2. Delete File

Command Description:

To Delete/Remove specified file

Pre-requisites:

File should be created before

Command Syntax:

```
AT+YFRM=<file name>
```

Parameter Description:

Parameters	Required?	Value	Description
file name	M	Value Range: 1-64 Format: STR Default value:	File name to delete/remove

Response:

If file delete operation is successful, returns success else failure.

Responses	Remarks
To read a file. AT+YFRM="/abc.txt"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To read a file.

AT+YFRM="/abc.txt"

OK

5.1.5.3. File Length

Command Description:

To get actual file length and space occupied in FLASH

Pre-requisites:

File should be present in FLASH

Command Syntax:

AT+YFLN=<name>

Parameter Description:

Parameters	Required?	Value	Description
name	M	Value Range: 0-100 Format: STR Default value:	File name along with path

Command Response

If file write operation is successful it gives the actual length and the space occupied by the file in the flash.

Response Name	Response Type	Response Range
actual file size	INT	0-4096
total file size in flash	INT	0-4096

Response:

If operation is successful, returns success else failure.

Responses	Remarks
To get the length of the file. AT+YFLN="/abc.txt"<CR><LF> <CR><LF>+YFLN:55,1024<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To get the length of the file.

```
AT+YFLN="/abc.txt"
+YFLN:55,1024
OK
```

5.1.5.4. List of Files

Command Description:

To list all files in specified path

Pre-requisites:

File path should be present in FLASH

Command Syntax:

```
AT+YFLS=[<path>]
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

path	O	Value Range: 0-100 path
		Format: STR
		Default value:

Command Response

Lists all the file in the specified path along with their actual size and size occupied in the flash.

Response Name	Response Type	Response Range
file name	STR	0-100
actual file size	INT	0-4096
total file size in flash	INT	0-4096

Response:

If operation is successful, returns success else failure.

Responses	Remarks
To get the length a file. AT+YFLS="/abc"<CR><LF> <CR><LF>+YFLS:/abc/xyz.txt,55,1024<CR><LF> <CR><LF>+YFLS:/abc/def.txt,110,1024<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To get the length a file.

```
AT+YFLS="/abc"
+YFLS:/abc/xyz.txt,55,1024
+YFLS:/abc/def.txt,110,1024
OK
```

5.1.5.5. Open File

Command Description:

To open file specified in file name with option given

Pre-requisites:**Command Syntax:**

AT+YFOP=<name>,<option>,[<pwd>]

Parameter Description:

Parameters	Required?	Value	Description
name	M	Value Range: 1-64 Format: STR Default value:	File name with the path e.g. /sys/abc.txt
option	M	Value Range: "0x0-0x2, 0x41, 0x42, 0xC1, 0xC2, 0x241, 0x242, 0x441, 0x442, 0x4C1, 0x4C2" Format: HEX Default value:	Open flag. Possible values are given below One of following three flags must be used:\b 0x0 to Open in reading only mode. 0x1 to Open in writing only mode. 0x2 to Open in read and write mode. In addition any of following flags may be bitwise ORed:\b 0x40 CREAT:If file exists, this flag has no effect except as noted under EXCL below. Otherwise, file shall be created. 0x80 EXCL:If CREAT and EXCL are set, open fails if file exists. 0x200 TRUNC:If file exists and is a regular file, and file is successfully opened with Read-Write or Write only modes, its length is truncated to 0, and mode is unchanged. 0x400 APPEND:If set, file offset shall be set to end of file prior to each write.
pwd	O	Value Range: 16	File password if it is a secured file.

Format: STR
Default value:

Whenever it has to be opened,
 same password has to be given

Command Response

If file open operation is successful +YFOP:FD is sent.

Response Name	Response Type	Response Range
FD	HEX	0X0-0XFFFFFFFF

Response:

If file open operation is successful, returns success else failure.

Responses	Remarks
To open a file. AT+YFOP="/abc.txt", 0<CR><LF> <CR><LF>+YFOP:0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To open a file.

AT+YFOP="/abc.txt", 0

+YFOP:0

OK

5.1.5.6. Read File

Command Description:

To read specified number of bytes from specified file starting from offset given

Pre-requisites:

File should have been opened before in read mode

Command Syntax:

AT+YFRD=<fd>, [<offset>], <Length>

Parameter Description:

Parameters	Required?	Value	Description
fd	M	Value Range: 0X0-0XFFFFFFFF Format: HEX Default value:	File Descriptor (FD) is the value returned while opening file
offset	O	Value Range: -1-4096 Format: INT Default value:	offset from the starting of file
Length	M	Value Range: 1-4096 Format: INT Default value:	Number of bytes to read

Command Response

If file read operation is successful +YFRD:FD,length,data is sent.

Response Name	Response Type	Response Range
fd	HEX	0X0-0XFFFFFFFF
data length	INT	0-4096
data	BIN	0-4096

Response:

If file read operation is successful, returns success else failure.

Responses	Remarks
To open a file. AT+YFOP="/abc.txt",0<CR><LF> <CR><LF>+YFOP:0<CR><LF> <CR><LF>OK<CR><LF> To read a file. AT+YFRD=0,length<CR><LF>	Successful

```
<CR><LF>+YFRD=0,tlength,flength,data<CR><LF>
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

To open a file.

```
AT+YFOP="/abc.txt",0
+YFOP:0
OK
```

To read a file.

```
AT+YFRD=0,length
+YFRD=0,tlength,flength,data
OK
```

5.1.5.7. Write File

Command Description:

To write to specified file from offset given

Pre-requisites:

File should have been open before in write mode

Command Syntax:

```
AT+YFWR=<fd>,[<offset>],<data length>,<data>
```

Parameter Description:

Parameters	Required?	Value	Description
fd	M	Value Range: 0X0-0XFFFFFFFF Format: HEX Default value:	File Descriptor (FD) is the value returned while opening the file
offset	O	Value Range: -1-4096 Format: INT Default value:	offset from the starting of the file
data length	M	Value Range: 1-4096 Format: INT Default value:	data to be written.

data	M	Value Range: 1-4096	data to be written.
		Format: BIN	
		Default value:	

Response:

If file write operation is successful, returns success else failure.

Responses	Remarks
To open a file. AT+YFOP="/abc.txt",0<CR><LF> <CR><LF>+YFOP:0<CR><LF> <CR><LF>OK<CR><LF>	Successful
To write a file. AT+YFWR=0,offset,length,data<CR><LF> <CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

To open a file.

```
AT+YFOP="/abc.txt",0
+YFOP:0
OK
```

To write a file.

```
AT+YFWR=0,offset,length,data
OK
```

5.1.5.8. Heap Information

Command Description:

To get details on total heap size and memory available for allocation

Pre-requisites:**Command Response**

It gives total heap size and memory available for allocation. The memory available for allocation may be fragmented, so even the memory available suggest one can allocate a particular size of memory, but still malloc would fail.

Response Name	Response Type	Response Range
total heap size	INT	0-204800
free memory	INT	0-204800

Command Syntax:

```
AT+YHD
```

Response:

If operation is successful, returns success else failure.

Responses	Remarks
To get the length a file. AT+YHD<CR><LF> <CR><LF>+YFD:102400,9024<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To get the length a file.

```
AT+YHD
```

```
+YFD:102400,9024
```

```
OK
```

5.1.5.9. Switch Between Legacy and AT Command

Command Description:

To switch between the legacy and new at commands. This command will reset the module.

Pre-requisites:**Command Syntax:**

```
AT+YLC=<enable>
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

enable	M	Value Range: 0-1 1 to switch to legacy AT command. Format: UINT Default value: 0
---------------	---	--

Response:

If operation is successful, returns success else failure.

Responses	Remarks
To run legacy cmd. AT+YLC = 1<CR><LF> <CR><LF>Serial2WiFi APP<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To run legacy cmd.

AT+YLC = 1

Serial2WiFi APP

5.2. [Profile Settings](#)

5.2.1. Read Profile

Command Description:

To read profile from flash

Pre-requisites:**Command Syntax:**

ATZn<profile_number>

Parameter Description:

Parameters	Required?	Value	Description
profile_number	M	Value Range: 0-1 Format: INT Default value: 0	Profile number to be read

Command Response

If profile read operation is successful +ZN:Profile string is sent.

Response Name	Response Type	Response Range
profile string	STR	0-65575

Response:

Success if profile read operation is successfule else failure.

Responses	Remarks
To read the profile from flash ATZ0<CR><LF> <CR><LF>OK<CR><LF> <CR><LF>+Zn:"profile":{"s2w_cfg"....<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To read the profile from flash

ATZ0

OK

+Zn:"profile":{"s2w_cfg"....

5.2.2. Save Profile**Command Description:**

To save profile to file system

Pre-requisites:**Command Syntax:**

AT&WNP=<profile number>

Parameter Description:

Parameters	Required?	Value	Description
profile number	M	Value Range: 0-1	Profile number to be saved

Format: INT
Default value:

Response:

Success if profile save operation is successfule else failure.

Responses	Remarks
To save the profile. AT&W0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To save the profile.

AT&W0

OK

5.2.3. Set Profile

Command Description:

To set profile to specified profile number

Pre-requisites:**Command Syntax:**

AT&Yn<profile number>

Parameter Description:

Parameters	Required?	Value	Description
profile number	M	Value Range: 0-1 Format: INT Default value:	Profile number to set

Response:

Success if profile set operation is successfule else failure.

Responses	Remarks
To set the profile. AT&Y0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To set the profile.

AT&Y0

OK

5.2.4. TCP Profile

Command Description:

To save particular CID profile to the SRAM

Pre-requisites:**Command Syntax:**

AT+YPTCP=<cid>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	cid given by AT+SC command or given by AT+SA command

Response:

Success if CID profile store in SRAM operation is successfull else failure.

Responses	Remarks
To store the CID profile. AT+SPTCP=0<CR><LF>	Successful

<CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

To store the CID profile.

AT+SPTCP=0

OK

5.2.5. UDP Profile

Command Description:

To save particular CID profile to the SRAM

Pre-requisites:**Command Syntax:**

AT+YPUDP=<cid>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	cid given by AT+SC command or given by AT+SA command

Response:

Success if CID profile store in SRAM operation is successful else failure.

Responses	Remarks
To store the CID profile. AT+SPUDP=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To store the CID profile.

AT+SPUDP=0

OK

5.3. RTC Settings

5.3.1. Time Settings

Command Description:

To set/get time to/from RTC module in Julian format. To get system Time, user has to set it once atleast

Pre-requisites:**Command Syntax:**

AT+YTIME=<absolute_time>

Parameter Description:

Parameters	Required?	Value	Description
absolute_time	M	Value Range: 1-32 Format: STR Default value: 16/02/2018,11:15:00+04	Absolute time to be set in Julian format

Response:

If getting or setting time of RTC module is successful, it returns SUCCESS, else returns FAILURE. Once the operation is done, registered API is called with the result.

Responses	Remarks
AT+YTIME="09/01/2018,11:15:00+04"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+YTIME="09/01/2018,11:15:00+04"

OK

5.3.2. Reset

Command Description:

This command does the soft reset.

Pre-requisites:**Response:**

Responses	Remarks
To do soft reset of the module AT+YSR<CR><LF> <CR><LF><CR><LF>	Successful
ERROR	If parameter is not valid

Command Syntax:

AT+YSR

Example:

To do soft reset of the module

AT+YSR

5.3.3. Version

Command Description:

This command gets the version number of the software.

Pre-requisites:**Command Response**

Returns the software version +YVER:VERSION is sent.

Response Name	Response Type	Response Range
Version	STR	5-7

Command Syntax:

AT+YVER

Response:

Responses	Remarks
To get the software version AT+YVER<CR><LF> YVER:"7.0.0"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To get the software version

AT+YVER

YVER:"7.0.0"

OK

5.4. [SNTP Settings](#)

5.4.1. Configure SNTP

Command Description:

To configure server in SNTP module

Pre-requisites:

L2 - L3 connections should be established and AT+NSNTPSTART command should be issued

Command Syntax:

```
AT+NSNTPCFG=<server>,[<id>]
```

Parameter Description:

Parameters	Required?	Value	Description
server	M	Value Range: 1-32 Format: STR Default value:	Server address to be configured
id	O	Value Range: 0x0,0x1,0xFFFF Format: HEX	ID of Server to be configured

Default value: FFFF

Response:

If configuring server is successful, it returns SUCCESS, else returns FAILURE. Once the configuring server is done, registered API is called with the result.

Responses	Remarks
AT+NSNTPCFG="pool.ntp.org",0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSNTPCFG="pool.ntp.org",0
OK
```

5.4.2. Start SNTP

Command Description:

To start SNTP module

Pre-requisites:

L2 - L3 connections should be established

Response:

If starting SNTP module is successful, it returns SUCCESS, else returns FAILURE. Once the starting server is done, registered API is called with the result.

Responses	Remarks
AT+NSNTPSTART<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Command Syntax:

```
AT+NSNTPSTART
```

Example:

```
AT+NSNTPSTART
OK
```

6. WIRELESS

6.1. WLAN

6.1.1. Application Information Element

Command Description:

This command is used to add the application information element to beacon, probe response frames in AP mode and to probe request frame in STA mode. In STA mode after setting the IE, the next probe request will have that IE which was added by the user. In AP mode, firstly the user should create the network, then user should issue the command to add the application IE.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command.

Command Syntax:

```
AT+WAPPIE=<WHANDLE>,<frame_type>,<oui>,[<vendor_content>]
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.
frame_type	M	Value Range: 0-2 Format: INT Default value:	This parameter tells the driver in which management frame the application IE should get added. 0 - BEACON, 1 - PROBE REQUEST, 2 - PROBE RESPONSE.
oui	M	Value Range: 6-6 Format: STR Default value: "506f9a"	This parameter is used to let WLAN driver know the OUI value of vendor specific Information Element. The user must give exact 6 bytes of input. For example if the OUI value is 5 6A 9D then user must give 05 6A 9D.
vendor_content	O	Value	This parameter is used to send the vendor

Range: 0-504 specific content to the
Format: STR Information Element.
Default value: ""

Response:

If AT+WAPPIE is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to send the Application Information Elements to the beacon and probe resp in AP and probe req in STA mode. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WAPPIE=1,1,"506f9a","1010"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to send the Application Information Elements to the beacon and probe resp

in AP and probe req in STA mode.

AT+WI=0

+WI=1

OK

AT+WAPPIE=1,1,"506f9a","1010"

OK

6.1.2. Set Beacon Interval

Command Description:

This API is used in STA mode to set the Beacon Miss Interval. So that the STA will disconnect from the network after synchronization loss occurred.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command.

Command Syntax:

```
AT+WBMISINTS=<WHANDLE>, [<beacon_miss_time_in_ms>,<num_beacons>]
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of the WLAN enabled interface.
beacon_miss_time_in_ms	O	Value Range: 0,1000-10000 Format: UINT Default value: 0	This parameter is used to set the time in milliseconds of beacon missing.
num_beacons	O	Value Range: 0,10-100 Format: UINT Default value: 0	This parameter is used to set number of beacons missing, after which a disconnect is triggered .

Response:

If AT+WBMISINTS is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the no of beacons in beacon miss interval at STA side. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WBMISINTS=1,,10<CR><LF> <CR><LF>OK<CR><LF> Or AT+WI=0<CR><LF>	Successful

```
+WI=1<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
AT+WBMISINTS=1,1000<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

This command is used to set the no of beacons in beacon miss interval at STA side.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WBMISINTS=1,10
```

```
OK
```

Or

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WBMISINTS=1,1000
```

```
OK
```

6.1.3. Get Country Code

Command Description:

This command is used to get the country code from the WLAN interface.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Syntax:

```
AT+WCCG=<WHANDLE>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of the WLAN enabled interface.

Command Response

If getting country code is successful '+WCCG:country code' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
country code	STR	

Response:

If AT+WCCG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get country code from device. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WCCG=1<CR><LF> +WCCG:US<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to get country code from device.

AT+WI=0

+WI=1

OK

AT+WCCG=1

+WCCG:US

OK

6.1.4. Set Country Code

Command Description:

This command is used to set country code to the WLAN interface

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Syntax:

```
AT+WCCS=<WHANDLE>,<country_code>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.
country_code	M	Value Range: 2-3 Format: STR Default value:	Country code to be set on WLAN enabled interface.

Response:

If AT+WCCS is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the country code to WLAN interface. <pre>AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WCCS=1,"IN"<CR><LF> <CR><LF>OK<CR><LF></pre>	Successful
ERROR	If parameter is not valid

Example:

This command is used to set the country code to WLAN interface.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

AT+WCCS=1,"IN"

OK

6.1.5. Disconnect

Command Description:

To disconnect from connected network

Pre-requisites:

AT+WI and AT+WAS commands should be issued

Command Syntax:

AT+WD=<WHANDLE>

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of WLAN enabled interface

Command Response

If disconnection is successful '+WD:disconnect status' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
disconnect status	not disconnected, disconnected	0-1

Response:

If AT+WD is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to disconnect from the network. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WD=1<CR><LF> +WD:DISCONNECTED<CR><LF>	Successful

```
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

This command is used to disconnect from the network.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WD=1
```

```
+WD:DISCONNECTED
```

```
OK
```

6.1.6. Interface Initilisation

Command Description:

To initialize WLAN interface

Pre-requisites:**Command Syntax:**

```
AT+WI=<MODE>
```

Parameter Description:

Parameters	Required?	Value	Description
MODE	M	Value Range: 0-1 Format: INT Default value: 0	Mode to start WLAN interface. 0 - Station, 1 - AP.

Command Response

If connection is successful '+WI:WHANDLE' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
whandle	INT	

Response:

If AT+WI is successful, returns SUCCESS else FAILURE.

Responses	Remarks
User wants the mode as AP , so user has given 1 as input parameter. If user wants the mode as STA, then user should give 0 as input parameter. AT+WI=0<CR><LF> +WI:1<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

User wants the mode as AP , so user has given 1 as input parameter. If user wants the mode as STA, then user should give 0 as input parameter.

AT+WI=0

+WI:1

OK

6.1.7. Get MAC Address

Command Description:

To get the MAC address to WLAN interface

Pre-requisites:

AT+WI={mode}

Command Syntax:

AT+WMACG=<WHANDLE>

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of WLAN enabled interface

Command Response

If getting mac address is successful '+WMACG:MACADDRESS' and 'OK' is sent

else 'ERROR' is sent.

Response Name	Response Type	Response Range
MACADDRESS	MAC	

Response:

If AT+WMACG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get the mac address from WLAN interface. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WMACG=1<CR><LF> +WMACG:00:03:7f:56:04:74<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to get the mac address from WLAN interface.

AT+WI=0

+WI=1

OK

AT+WMACG=1

+WMACG:00:03:7f:56:04:74

OK

6.1.8. Get Physical Mode

Command Description:

This command is used to get PHY mode from WLAN interface.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Syntax:

```
AT+WPHYMODEG=<WHANDLE>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of the WLAN enabled interface.

Command Response

If getting PHY mode is successful '+WPHYMODEG:PHY MODE' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
phy mode	A, B, G, AG, AN, GN, AGN	

Response:

If AT+WPHYMODEG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get the PHY mode from WLAN interface. <pre>AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WPHYMODEG=1<CR><LF> +WPHYMODEG:AGN<CR><LF> <CR><LF>OK<CR><LF></pre>	Successful
ERROR	If parameter is not valid

Example:

This command is used to get the PHY mode from WLAN interface.

AT+WI=0

+WI=1

OK

AT+WPHYMODEG=1

+WPHYMODEG:AGN

OK

6.1.9. Set Physical Mode

Command Description:

This command is used to set the PHY mode of WLAN Interface.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Syntax:

AT+WPHYMODES=<WHANDLE>,<phy_mode>

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of the WLAN enabled interface.
phy_mode	M	Value Range: 1-3 Format: STR Default value:	PHY mode for WLAN enabled interface.

Response:

If setting phy mode is successful then SUCCESS or else FAILURE.

Responses	Remarks
This command is used to set the PHY mode to WLAN interface. AT+WI=0<CR><LF> <CR><LF>+WI=1<CR><LF> <CR><LF>OK<CR><LF>	Successful

```
AT+WPHYMODES=1,"AN"<CR><LF>
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

This command is used to set the PHY mode to WLAN interface.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WPHYMODES=1,"AN"
```

```
OK
```

6.1.10. Power Save

Command Description:

This command is used to set the values related to power save option.

Pre-requisites:

So AT+WI=0 is the prerequisite for this command

Command Syntax:

```
AT+WPOWERSAVE=<WHANDLE>,<power_save_enable>,<idle_time_in_ms>,<ps_poll_num>,<dtim_policy>,<tx_num_to_wake_up>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.
power_save_enable	M	Value Range: 0-1 Format: INT Default value:	This parameter tells the WLAN driver to enable power save mode or to have full performance mode. 1 - enable 0 - disable.
idle_time_in_ms	M	Value Range: 200-65535	This parameter tells the WLAN driver after how much ideal time in

		Format: INT Default value:	milli seconds the device should go to power save after connection.
ps_poll_num	M	Value Range: 1-65535 Format: INT Default value: 1	This parameter tells the number of contiguous PS-Poll frames that the WLAN firmware is to send before sending an 802.11 NULL frame to indicate a power save exit to the access point.
dtim_policy	M	Value Range: 1-3 Format: INT Default value: 1	This parameter tells the dtim policy followed by station when it is in power save mode. 1 for station wakeup at TIM intervals. 2 for station wakeup only at DTIM intervals. 3 for station wakeup at both TIM and DTIM intervals.
tx_num_to_wake_up	M	Value Range: 0-65535 Format: INT Default value: 1	This parameter is used to tell after how many contiguous transmit packets, the device should come out of power save mode.

Response:

If AT+WPOWERSAVE is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the power save parameters in STA mode. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WPOWERSAVE=1,1,1000,3,2,5<CR><LF> <CR><LF>OK<CR><LF>	Successful

ERROR

If parameter is not valid

Example:

This command is used to set the power save parameters in STA mode.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WPOWERSAVE=1,1,1000,3,2,5
```

```
OK
```

6.1.11. Send Raw Packets**Command Description:**

This command is used to send the RAW WLAN packet (Beacon, QOS data, 4 addr data).

The device should be in disassociation state and max performance state when user wants to send RAW

WLAN packets. The following information will tell about addresses 1 to 4 in different WLAN packets.

In BEACON FRAME the addr1 will be receiver and destination address, Since receiver and destination addresses

are broadcast in beacon so user has to give "ff:ff:ff:ff:ff:ff". addr2 will be transmitter and source address. So

addr2 may be MAC address of its own. addr3 will be BSSID of the created network. Its MAC address will be

own MAC address. addr4 is not used in BEACON FRAME. In QOS DATA FRAME :- The addr1 will be BSSID and receiver address of the network(AP).

The addr2 will be source and transmitter address in the network(STA1). addr3 will be destination address (STA2). In 4 ADDR DATA FRAME the

addr1 will be the receiver address. addr2 will be transmitter address. addr3 will be the destination address. addr4 will be source address.

Pre-requisites:

So AT+WI={mode} is the prerequisite for this command

Command Syntax:

```
AT+WRAWPKTS=<WHANDLE>,<rate_index>,<num_tries>,<channel>,<packet_type>,<payload_size>,[<payload>],<addr1>,<addr2>,<addr3>,[<addr4>]
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range:	Device ID of the WLAN enabled interface.

		0-1 Format: UINT Default value:	
rate_index	M	Value Range: 0-20 Format: INT Default value:	This parameter tells the data rate on which the WLAN RAW packet user wants to send.
num_tries	M	Value Range: 1-14 Format: INT Default value:	This parameter tells how many RAW packets should WLAN driver send into air.
channel	M	Value Range: 0-11 Format: INT Default value:	This parameter tells the channel on which WLAN RAW packet user wants to send.
packet_type	M	Value Range: 0-2 Format: INT Default value:	This parameter is used to tell the payload to send in RAW WLAN packet. 0 : beacon, 1 : QOS data, 2 : 4 addr data packet.
payload_size	M	Value Range: 0-1400 Format: INT Default value:	This parameter tells the length of data in WLAN RAW packet user wants to send.
payload	O	Value Range: 0-1400 Format: STR Default value:	This parameter is used to tell the payload to send in RAW WLAN packet. packet may be beacon , QOS data, 4 addr data packet.
addr1	M	Value Range:	This parameter is used to give the MAC

		Format: MAC Default value:	addr1, which is used again to send the RAW WLAN packet in the air by user. The description will give the information about usage of addr1 in different scenarios.
addr2	M	Value Range: Format: MAC Default value:	This parameter is used to give the MAC addr2, which is used again to send the RAW WLAN packet in the air by user. The description will give the information about usage of addr2 in different scenarios.
addr3	M	Value Range: Format: MAC Default value:	This parameter is used to give the MAC addr3, which is used again to send the RAW WLAN packet in the air by user. The description will give the information about usage of addr3 in different scenarios.
addr4	O	Value Range: Format: MAC Default value:	This parameter is used to give the MAC addr4, which is used again to send the RAW WLAN packet in the air by user. The description will give the information about usage of addr4 in different scenarios.

Response:

If AT+WRAWPKTS is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to send the raw pkts. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WRAWPKTS=1,1,0,5,6,0,0,,"11:22:33:44:55:66", "11:22:33:44:55:63", "11:22:33:44:55:65", "11:22:33:44:55:67"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If param

eter is
not
valid

Example:

This command is used to send the raw pkts.

AT+WI=0

+WI=1

OK

AT+WRAWPKTS=1,1,0,5,6,0,0,,,"11:22:33:44:55:66","11:22:33:44:55:63","11:22:33:44:55:65","11:22:33:44:55:67"

OK

6.1.12. Get Regulatory Domain

Command Description:

This command is used to get the Regulatory domain which is set in driver.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Syntax:

AT+WREGDG=<WHANDLE>

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of the WLAN enabled interface.

Command Response

If getting regulatory domain is successful '+WREGDG:Regulatory domain' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
Regulatory domain	ETSI, FCC, APL, TELEC	

Response:

If AT+WREGDG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get the regulatory domain from WLAN interface. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WREGDG=1<CR><LF> +WREGDG:ETSI<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to get the regulatory domain from WLAN interface.

```
AT+WI=0
+WI=1
OK
AT+WREGDG=1
+WREGDG:ETSI
OK
```

6.1.13. Set Regulatory Domain

Command Description:

This command is used to set the regulatory domain.

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Syntax:

```
AT+WREGDS=<WHANDLE>,<regulatory_domain>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value	Device ID of the WLAN enabled

		Range: 0-1 interface. Format: UINT Default value:
regulatory_domain	M	Value Range: 3-5 Regulatory domains to set in WLAN . Format: ETSI, FCC, APL, TELEC are STR supported. Default value: 3

Response:

If AT+WREGDS is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the regulatory domain to WLAN interface. AT+WI=0<CR><LF> +WI:1<CR><LF> <CR><LF>OK<CR><LF> AT+WREGDS=1,"TELEC"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to set the regulatory domain to WLAN interface.

AT+WI=0

+WI:1

OK

AT+WREGDS=1,"TELEC"

OK

6.1.14. Get RSSI

Command Description:

This API is used to get the RSSI value for WLAN interface

Pre-requisites:

AT+WI={mode} is the prerequisite for this command

Command Response

If RSSI get from WLAN is successful then '+WRSSIG=RSSI value' and 'OK'
or else 'ERROR' will sent.

Response Name	Response Type	Response Range
rss value	INT	

Command Syntax:

AT+WRSSIG

Response:

If AT+WRSSIG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get RSSI value from WLAN interface . AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WRSSIG<CR><LF> +WRSSIG:30<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to get RSSI value from WLAN interface .

AT+WI=0

+WI=1

OK

AT+WRSSIG

+WRSSIG:30

OK

6.1.15. Scan

Command Description:

To scan WLAN channels

Pre-requisites:

AT+WI=0 command should be issued

Command Syntax:

AT+WS=<WHANDLE>, [<SSID>,<CHANNEL>]

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Device ID of WLAN interface on which to scan
SSID	O	Value Range: 1-32 Format: STR Default value:	SSID of network to get scanned
CHANNEL	O	Value Range: 1-14,36,40,44,48,52,56,60,64,100,104,108,112,116,132,136,140,149,153,157,161,165 Format: INT Default value: 0	channel to scan on

Command Response

If scan is successful '+WS:bssid,ssid,channel,type,rssi,security' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
BSSID	MAC	

SSID	STR	
CHANNEL	UINT	
TYPE	INFRA, ADHOC	0-1
RSSI	INT	
SECURITY	NONE, WEP, WPA PSK, WPA ENTERPRISE, WPA2 PSK, WPA2 ENTERPRISE	0-5

Response:

If AT+WS=1 successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
This command is used to print the scan results of available networks. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WS=1<CR><LF> +WS:RESULTS<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to print the scan results of available networks.

```
AT+WI=0
+WI=1
OK
AT+WS=1
+WS:RESULTS
OK
```

6.1.16. Get Power Transmission

Command Description:

This API is used to get the TX POWER from WLAN interface.

Pre-requisites:

AT+WI is the prerequisite for this command.

Command Syntax:

```
AT+WTXPOWERG=<WHANDLE>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.

Command Response

If TX power get is successful '+WTXPOWERG:tx power' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
tx power	INT	

Response:

If AT+WTXPOWERG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get the TX power from WLAN interface. <pre>AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WTXPOWERG=1<CR><LF> +WTXPOWERG:63<CR><LF> <CR><LF>OK<CR><LF></pre>	Successful

ERROR

If parameter is not valid

Example:

This command is used to get the TX power from WLAN interface.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WTXPOWERG=1
```

```
+WTXPOWERG:63
```

```
OK
```

6.1.17. Set Power Transmission

Command Description:

This API is used to set the TX power to WLAN interface

Pre-requisites:

AT+WI is the prerequisite for this command.

Command Syntax:

```
AT+WTXPOWERS=<WHANDLE>,<txpower>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.
txpower	M	Value Range: 0-63 Format: UINT Default value: 63	TX power value which user wants to set to WLAN interface.

Response:

If AT+WTXPOWERS is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the TX power in WLAN interface. Successful <pre> AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WTXPOWERS=34<CR><LF> <CR><LF>OK<CR><LF> </pre>	
ERROR	If parameter is not valid

Example:

This command is used to set the TX power in WLAN interface.

```
AT+WI=0
```

```
+WI=1
```

```
OK
```

```
AT+WTXPOWERS=34
```

```
OK
```

6.1.18. Get Transmission Rate

Command Description:

This API is used to get the TX RATE from WLAN interface ,
the default one or which was set by user. User should map the output as follows:-

1-1Mbps, 2-2Mbps, 3-5.5Mbps,

4-6Mbps, 5-9Mbps, 6-11Mbps, 7-12Mbps, 8-18Mbps, 9-24Mbps, 10-36Mbps, 11-48Mbps,
12-54Mbps

13-MCS0, 14-MCS1, 15-MCS2, 16-MCS3, 17-MCS4, 18-MCS5, 19-MCS6, 20-MCS7. For
example if this

AT command returns 12 then user should consider it as 54 Mbps.

Pre-requisites:

Before issuing this command the user should initialize the wlan
interface by using AT+WI command. So AT+WI is the prerequisite for this command.

Command Syntax:

```
AT+WTXRATEG=<WHANDLE>
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

WHANDLE	M	Value Range: 0-1 Device ID of the WLAN enabled interface. Format: UINT Default value: 1
----------------	---	--

Command Response

If TX rate get is successful '+WTXRATEG:tx rate' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
tx rate	INT	

Response:

If AT+WTXRATEG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get the TX rate from WLAN interface. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WTXRATEG=1<CR><LF> +WTXRATEG:12<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to get the TX rate from WLAN interface.

AT+WI=0

+WI=1

OK

AT+WTXRATEG=1

+WTXRATEG:12

OK

6.1.19. Set Transmission Rate

Command Description:

This API is used to set the TX rate to WLAN interface.

Pre-requisites:

Before issuing this command the user should initialize the WLAN interface by using AT+WI command. So AT+WI is the prerequisite for this command.

Command Syntax:

```
AT+WTXRATES=<WHANDLE>,<txrate>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.
txrate	M	Value Range: 1-20 Format: UINT Default value: 1	TX rate value which user wants to set to WLAN interface. In future the packets will transmit with user defined TX rate. 1-1Mbps, 2-2Mbps, 3-5.5Mbps, 4-6Mbps, 5-9Mbps, 6-11Mbps, 7-12Mbps, 8-18Mbps, 9-24Mbps, 10-36Mbps, 11-48Mbps, 12-54Mbps 13-MCS0, 14-MCS1, 15-MCS2, 16-MCS3,

17-MCS4,
18-MCS5,
19-MCS6,
20-MCS7.

Response:

If AT+WTXRATES is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the TX rate in WLAN interface. Successful AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WTXRATES=1,12<CR><LF> <CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

This command is used to set the TX rate in WLAN interface.

AT+WI=0

+WI=1

OK

AT+WTXRATES=1,12

OK

6.1.20. WPS

Command Description:

This command is used to connect to an AP by using WPS (PUSH, PIN) procedure.

Pre-requisites:

AT+WI=0 in STA mode. AT+WNCR in AP mode.

Command Syntax:

AT+WWPS=<WHANDLE>,<method>,<l2connectinfo>,[<pin>,<ssid>]

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 1	Device ID of the WLAN enabled interface.
method	M	Value Range: 1-3 Format: INT Default value: 1	WPS method which user wants to use. 1 - PUSH, 2 - PIN, 3 - DEFAULT PIN.
l2connectinfo	M	Value Range: 0-1 Format: INT Default value: 0	Connection after WPS procedure. 0 - Only WPS procedure. 1 - L2 connection after WPS.
pin	O	Value Range: 8 Format: STR Default value: ""	This parameter is used to take the pin as input from the user which can be used in PIN method.
ssid	O	Value Range: 1-32 Format: STR Default value: ""	This parameter is used to take SSID as input from the user. It is used for default PIN method to pass SSID to the WLAN driver.

Command Response

If connection is successful '+WWPS:connection status,ssid,channel,mac address' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
connection status	WPS FAILED, WPS SUCCESS, NW CONNECT FAIL, NW CONNECT SUCCESS	0-1

ssid	STR
channel	INT
mac_address	MAC

Response:

If AT+WWPS is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to set the country code to WLAN interface. AT+WI=0<CR><LF> +WI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WWPS=1,1,1<CR><LF> ++WWPS:NW CONNECT SUCCESS,"Quartz_AP",6,54:B8:0A:0A:12:9C<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

This command is used to set the country code to WLAN interface.

AT+WI=0

+WI=1

OK

AT+WWPS=1,1,1

++WWPS:NW CONNECT SUCCESS,"Quartz_AP",6,54:B8:0A:0A:12:9C

OK

7. NETWORKS

7.1. CID

7.1.1. Get CID Information

Command Description:

To get CID information

Pre-requisites:

L2 -L3 connections should be established

Command Syntax:

```
AT+NCIDI=<cid>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided socket/network commands

Command Response

If the cid exist, then it gives response as

'+NCIDI:CidType,SocketFamily,SocketType,IpAddr,LocalPort,RemotePort' and 'OK'
otherwise module sends 'ERROR'.

Response Name	Response Type	Response Range
CidType	UNKNOWN, RAW, UDP, TCP, TCP CLIENT, TCP SERVER, TLS CLIENT, TLS SERVER, DTLS CLIENT, DTLS SERVER, HTTP CLIENT, HTTP CLIENT WITH SSL, HTTP SERVER, HTTP SERVER WITH SSL, MQTT, WEBSOCKET CLIENT	
SocketFamily	Unspecified, IPv4, IPv6	
IPAddr	IP	
Lport	UINT	
Rport	UINT	

Response:

Responses	Remarks
AT+SC=2,1,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+NCIDI=0<CR><LF> +NCIDI:TCP- SERVER,IPv4,STREAM,192.168.120.3,8344,8355<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```

AT+SC=2,1,0
+SC:0
OK
AT+NCIDI=0
+NCIDI:TCP-SERVER,IPv4,STREAM,192.168.120.3,8344,8355
OK

```

7.1.2. CID Status Information

Command Description:

To get CID status information

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+NCIDS=<cid>, [<rstFlag>]
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT	cid provided by socket /network commands

		Default value:
rstFlag	0	Value Range: 0-1 Format: UINT Default value: 0

Command Response

If the cid exist, then it gives response as

'+NCIDS:UDP/TCP CONNECTED/TCP DISCONNECTED, Num bytes sent,Num bytes received and 'OK'

else sends 'ERROR'

Response Name	Response Type	Response Range
CidType	UNKNOWN, RAW, UDP, TCP, TCP CLIENT, TCP SERVER, TLS CLIENT, TLS SERVER, DTLS CLIENT, DTLS SERVER, HTTP CLIENT, HTTP CLIENT WITH SSL, HTTP SERVER, HTTP SERVER WITH SSL, MQTT	
CidStatus	NA, Connected, Not Connected	
TtlSentBytes	UINT	
TtlRxdBytes	UINT	

Response:

Responses	Remarks
AT+SC=2,1,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+NCIDI=0<CR><LF> +NCIDI:IPv4,STREAM	Successful

```
socket,server,192.168.120.3,8355<CR><LF>
<CR><LF>OK<CR><LF>
AT+NCIDS=1,0<CR><LF>
+NCIDS:TCP CLIENT, CONNECTED, 1000, 2000<CR><LF>
```

ERROR

If parameter is not valid

Example:

```
AT+SC=2,1,0
+SC:0
OK
AT+NCIDI=0
+NCIDI:IPv4,STREAM socket,server,192.168.120.3,8355
OK
AT+NCIDS=1,0
+NCIDS:TCP CLIENT, CONNECTED, 1000, 2000
```

7.2. DNS**7.2.1. NDNS URL****Command Description:**

To resolve a URL in DNS module

Pre-requisites:

L2 - L3 connections should be done and AT+NDNSCSRVIP command should be issued

Command Syntax:

```
AT+NDNSCRURL=<URL>, [<ip_version>]
```

Parameter Description:

Parameters	Required?	Value	Description
URL	M	Value Range: 1-64 Format: STR Default value:	URL to be resolved.
ip_version	O	Value Range: 4,6 Format: UINT Default value: 4	IP version of the URL to be resolved

Command Response

If URL resolving is successful +NDNSCRURL:IP address is sent

Response Name	Response Type	Response Range
IP	STR	

Response:

If URL resolving is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NDNSCRURL="www.telit.com"<CR><LF> +NDNSCRURL:successful<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NDNSCRURL="www.telit.com"
+NDNSCRURL:successful
OK
```

7.2.2. NDNS IP Address

Command Description:

To set and get server IP in DNS module

Pre-requisites:

L2 - L3 connection should be done

Command Syntax:

```
AT+NDNSCSRVIP=<ip>, [<id>]
```

Parameter Description:

Parameters	Required?	Value	Description
ip	M	Value Range: 7-15 Format: IP	Server IP address to be set

Default value:			
id	O	Value Range: 0x0,0x1,0xFFFF Format: HEX Default value: FFFF	ID of Server IP address to be set

Response:

If setting or getting server IP is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NDNSCSRVIP=8.8.8.8,0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NDNSCSRVIP=8.8.8.8,0
OK
```

7.2.3. NDNS Host Addition

Command Description:

To add a host in DNS module

Pre-requisites:

L2 - L3 connections should be done and AT+NDNSCSRVIP command should be issued

Command Syntax:

```
AT+NDNSSADDDHOST=<host_name>,<ipaddress>,<ttd>
```

Parameter Description:

Parameters	Required?	Value	Description
host_name	M	Value Range: 1-32 Format: STR Default value:	Host name to be added

ipaddress	M	Value Range: 1-64 Format: IP Default value:	IP address to be added
ttl	M	Value Range: 1-1024 Format: UINT Default value:	Time to Live for the host to be added

Response:

If adding host is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NDNSSSTART<CR><LF> <CR><LF>OK<CR><LF> AT+NDNSSADDDHOST="www.qca4020.com",192.192.192.192,600<CR><LF> > <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```

AT+NDNSSSTART
OK
AT+NDNSSADDDHOST="www.qca4020.com",192.192.192.192,600
OK

```

7.2.4. NDNS Server Start

Command Description:

To start server in DNS module

Pre-requisites:

L2 - L3 connections should be done

Response:

If starting server is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NDNSSSTART<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Command Syntax:

AT+NDNSSSTART

Example:

```
AT+NDNSSSTART
OK
```

7.2.5. NDNS Server Stop

Command Description:

To stop server in DNS module

Pre-requisites:

L2 - L3 connections should be done

Response:

If stopping server is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NDNSSSTOP<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Command Syntax:

AT+NDNSSSTOP

Example:

AT+NDNSSSTOP

OK

7.2.6. NDNS Service Discovery

Command Description:

For service discovery in DNS module

Pre-requisites:

L2 - L3 connections should be done

Command Syntax:

AT+NDNSSD=<device_id>,<instance_name>,[<ip_version>,<timeout>]

Parameter Description:

Parameters	Required?	Value	Description
device_id	M	Value Range: 0,1 Format: UINT Default value:	Device ID obtained in AT+WNI command response
instance_name	M	Value Range: 1-64 Format: STR Default value:	service instance name to be discovered
ip_version	O	Value Range: 4,6,46 Format: UINT Default value: 4	IP version to be used in service discovery
timeout	O	Value Range: 0-5000 Format: UINT Default value: 5000	Timeout for the discovery request handling

Command Response

If service discovery is successful +NDNSSD:response values is sent

Response Name	Response Type	Response Range
rsp_type	UINT	

svc_name	STR
tll	UINT
svr_name	STR
priority	UINT
weight	UINT
port	UINT
server_name_in_target_info	STR
ipv4_addr	IPV4
ipv6_addr	IPV6
txt	STR

Response:

If service discovery is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

7.2.7. NDNS Target Information

Command Description:

To get target information in DNS module

Pre-requisites:

L2 - L3 connections should be done

Command Syntax:

```
AT+NDNSSDGETTARGETINFO=<device_id>,<instance_name>,[<ip_version>,<timeout>]
```

Parameter Description:

Parameters	Required?	Value	Description
device_id	M	Value Range: 0,1 Format: UINT Default value:	Device ID obtained in AT+WNl command response
instance_name	M	Value Range: 1-64 Format: STR Default value:	service instance name to be discovered
ip_version	O	Value Range: 4,6,46 Format: UINT Default value: 4	IP version to be used in discovery
timeout	O	Value Range: 0-5000 Format: UINT Default value: 5000	Timeout for the request handling

Command Response

If getting target info is successful +NDNSSDGETTARGETINFO:respose values is sent

Response Name	Response Type	Response Range
rsp_type	UINT	
svc_name	STR	
tll	UINT	
svr_name	STR	
priority	UINT	

weight	UINT
port	UINT
server_name_in_target_info	STR
ipv4_addr	IPV4
ipv6_addr	IPV6
txt	STR

Response:

If getting target info is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

8. DATA TRANSFER

8.1. WNCM

8.1.1. Configure IP Address

Command Description:

To set IP configuration of device

Pre-requisites:

AT+WNI command should be issued

Command Syntax:

```
AT+WNIPC=<WHANDLE>,<IPFLAG>,[<IPADDRESS>,<IPMASK>,<IPGATEWAY>,<Hostname>]
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value: 0	Handle, which user got from AT+WNI command
IPFLAG	M	Value Range: 1-2 Format: UINT Default value: 2	IPFLAG to notify that IP configuration is for STATIC or for DHCP 1 - STATIC 2 - DHCP
IPADDRESS	O	Value Range: Format: IP Default value:	Static IPV4 address for the device. It may be STA mode or AP mode.
IPMASK	O	Value Range: Format: IPV4 Default value:	Net mask for the device.
IPGATEWAY	O	Value Range: Format: IP Default value:	Gate way for the device
Hostname	O	Value Range: 0-32 Format: STR Default value: Telit_Device	Host name for the device

Response:

If AT+WNIPC is successful, returns SUCCESS else FAILURE.

Responses	Remarks
First initialize the NCM by using AT+WNI=1. Set the Static IP configuration for AP by using AT+WNIPC AT+WNI=1<CR><LF> +WNI=0<CR><LF> <CR><LF>OK<CR><LF> AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1","Telit_Device" "<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

First initialize the NCM by using AT+WNI=1.

Set the Static IP configuration for AP by using AT+WNIPC

AT+WNI=1

+WNI=0

OK

AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1","Telit_Device"

OK

8.1.2. Configure IP Address in AP Mode

Command Description:

To set IPv4 configuration of AP to create network

Pre-requisites:

AT+WNI and AT+WNIPC commands should be issued

Command Syntax:

```
AT+WNAPIPC=<WHANDLE>,<IPSTARTADDRESS>,<IPRANGEADDRESS>,<LEASETIME>
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Handle, which user got from AT+WNI command
IPSTARTADDRESS	M	Value Range: Format: IP Default value:	Starting IPV4 address to allocate to connected stations
IPRANGEADDRESS	M	Value Range: Format: IP Default value:	Maximum range that AP is going to allocate for stations connected to its network
LEASETIME	M	Value Range: 1-4294967295 Format: UINT Default value:	Lease time for DHCP server

Response:

If AT+WNAIPIC is successful, returns SUCCESS else FAILURE.

Responses	Remarks
The usage of this command is first parameter will be handle which is derived from AT+WNI=1[AP mode]. Parameters are whandle, IPv4 starting address , IPv4 range, lease time. AT+WNI=1<CR><LF> +WNI:0<CR><LF> <CR><LF>OK<CR><LF> AT+WNI=1,1,"192.168.23.2","255.255.255.0","192.168.23.1"<CR><LF> <CR><LF>OK<CR><LF> AT+WNAIPIC=0,"192.168.23.24","192.168.23.55",86400000<CR><LF>	Successful

<CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

The usage of this command is first parameter will be handle which is derived from AT+WNI=1[AP mode]. Parameters are whandle, IPv4 starting address , IPv4 range, lease time.

AT+WNI=1

+WNI:0

OK

AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1"

OK

AT+WNAIPPC=0,"192.168.23.24","192.168.23.55",86400000

OK

8.1.3. NCM Connect

Command Description:

To connect to the configured network at L2 and L3 level

Pre-requisites:

AT+WNI command is a mandate. If user wants to set static IP address then AT+WNIPC command should be issued

Command Syntax:

AT+WNCN=<WHANDLE>,<SSID>,[<PASSPHRASE>,<CHANNEL>]

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Whandle, which user got from AT+WNI=0 command
SSID	M	Value Range: 1-32 Format: STR	SSID to connect

Default value:			
PASSPH RASE	O	Value Range: 0-64 Format: STR Default value: ""	pass phrase to connect to the network
CHANNE L	O	Value Range: 1-14,36,40,44,48,52,56,60,64,100,104,108,112,116,132,136,140,149,153,157,161,165 Format: INT Default value: 0	Channel of the AP to connect

Command Response

If connection is successful '+WNCN:connect status,IP Address,Net Mask,Gate Way' and 'OK'

is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
connect status	not connected, connected, disconnected, IP CONFIGURATION FAILED	0-3
IP Address	IP	
Net Mask	IPV4	
Gate Way	IP	

Response:

If AT+WNCN is successful, returns SUCCESS else FAILURE.

Responses	Remarks
First initialize NCM module by using AT+WNI command. Since connection is done in station mode so 0 is passed.	Successful

If user wants to set the IP address statically, then AT+WNIPC should be issued before calling AT+WNCN command. Here the SSID of the AP is Telit_Guest and the password is Welcome. <pre>AT+WNI=0<CR><LF> +WNI:1<CR><LF> <CR><LF>OK<CR><LF> AT+WNCN=1,"Telit_Guest","Welcome",6<CR><LF> +WNCN:CONNECTED,192.168.3.45,255.255.255.0,192.168.3.1<CR><LF> <CR><LF>OK<CR><LF></pre>	
ERROR	If parameter is not valid

Example:

First initialize NCM module by using AT+WNI command. Since connection is done in station mode so 0 is passed.

If user wants to set the IP address statically, then AT+WNIPC should be issued before calling AT+WNCN command.

Here the SSID of the AP is Telit_Guest and the password is Welcome.

```
AT+WNI=0
+WNI:1
OK
AT+WNCN=1,"Telit_Guest","Welcome",6
+WNCN:CONNECTED,192.168.3.45,255.255.255.0,192.168.3.1
OK
```

8.1.4. Configure AP

Command Description:

To set AP configuration to create a network. This command should be issued before creating a network by using AT+WNCR command or default values will be used.

Pre-requisites:

AT+WNI=1 command should be issued

Command Syntax:

AT+WNAPC=<WHANDLE>,<HIDDENSSID>,<BEACONINTERVAL>,<DTIMPERIOD>

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Whandle, which user got from AT+WNI command
HIDDENSSID	M	Value Range: 0-1 Format: INT Default value:	Hidden SSID for created network\b 0 - Disable\b 1 - Enable\b
BEACONINTERVAL	M	Value Range: 100-1000 Format: INT Default value: 100	Interval between the beacon frames
DTIMPERIOD	M	Value Range: 1-255 Format: INT Default value: 3	Delivery Traffic Indication Map period

Response:

If AT+WNAPC is successful, returns SUCCESS else FAILURE.

Responses	Remarks
Set AP configurations AT+WNI=1<CR><LF> +WNI:0<CR><LF> <CR><LF>OK<CR><LF> AT+WNAPC=0,0,100,3<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

Set AP configurations

AT+WNI=1

+WNI:0

OK

AT+WNAPC=0,0,100,3

OK

8.1.5. Create NCM

Command Description:

To create the network with WHANDLE, SSID, channel, security type, encryption type and pass phrase passed by the user at L2 and L3 level

Pre-requisites:

AT+WNI=1, AT+WNIPC and AT+WNAPIPC commands should be issued

Command Syntax:

```
AT+WNCR=<WHANDLE>,<SSID>,<CHANNEL>,<SECURITYTYPE>,<ENCRYPTIONTYPE>,[<PASSPHRASE>]
```

Parameter Description:

Parameters	Required?	Value	Description
WHANDLE	M	Value Range: 0-1 Format: UINT Default value:	Handle, which user got from AT+WNI=1 command
SSID	M	Value Range: 1-32 Format: STR Default value:	SSID of the creating network
CHANNEL	M	Value Range: 1-14,36,40,44,48,52,56,60,64,100,104,108,112,116,132,136,140,149,153,157,161,165 Format: INT Default value:	Channel of the network where user wants to create
SECURITYT	M	Value Range: 3-4	User has to

YPE		Format: STR Default value:	give the Security type string for secured network . Security types like ["WEP", "WPA", "WPA2"]. For open security user should pass ["NONE"]
ENCRYPTIO NTYPE	M	Value Range: 3-4 Format: STR Default value:	User has to give encryption type. Since it is mandatory parameter, ["WEP", "WPA", "WPA2"] string for secured network . For open security user has to give ["NONE"] string.

PASSPHRA SE ○**Value Range:** 8-64**Format:** STR**Default value:**Passph
rase of
network
to be
created.
Leave it
empty
for
open
security
mode**Command Response**

If creation of network is successful '+WNCR:create status' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
create status	Not created, Created	

Response:

If AT+WNCR is successful, returns SUCCESS else FAILURE.

Responses	Remarks
First initialize the NCM by using AT+WNI=1. Set the Static IP configuration for AP by using AT+WNIPC then set the range of IP addresses so that the connected stations will get IP address in range by using AT+WNAIPIC. After every command returns success, give command AT+WNCR .Here the SSID of the AP is Telit_Guest and the password is 12345678. AT+WNI=1<CR><LF> +WNI=0<CR><LF> <CR><LF>OK<CR><LF> AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1"<CR><LF> <CR><LF>OK<CR><LF> AT+WNAIPIC=0,"192.168.23.24","192.168.23.55",86400000<CR><LF>	Successful

```
<CR><LF>OK<CR><LF>
```

```
AT+WNCR=0,"Telit_Guest",6,"WPA","TKIP","12345678"<CR><LF>
```

```
+WNCR:CREATED<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

ERROR

If
parameter
is not valid

Example:

First initialize the NCM by using AT+WNI=1.

Set the Static IP configuration for AP by using AT+WNIPC then set the range of IP addresses

so that the connected stations will get IP address in range by using AT+WNAPIPC.

After every command returns success, give command AT+WNCR .Here the SSID of the AP is

Telit_Guest and the password is 12345678.

```
AT+WNI=1
```

```
+WNI=0
```

```
OK
```

```
AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1"
```

```
OK
```

```
AT+WNAPIPC=0,"192.168.23.24","192.168.23.55",86400000
```

```
OK
```

```
AT+WNCR=0,"Telit_Guest",6,"WPA","TKIP","12345678"
```

```
+WNCR:CREATED
```

```
OK
```

8.1.6. Initialize NCM

Command Description:

To initialize NCM module

Pre-requisites:

Nothing.

Command Syntax:

```
AT+WNI=<MODE>
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

MODE	M	Value Range: 0-1	Mode to initialize the NCM module. \b
		Format: UINT	0 - STATION MODE \b
		Default value: 0	1 - AP MODE

Command Response

If initialization is successful '+WNI:whandle{0 or 1}' and 'OK' is sent
else 'ERROR' is sent.

Response Name	Response Type	Response Range
whandle	INT	

Response:

If AT+WNI is successful, returns SUCCESS else FAILURE.

Responses	Remarks
User wants the mode as AP , so he/she has given 1 as input parameter. If user wants the mode as STA, he/she should give 0 as input parameter. AT+WNI=1<CR><LF> +WNI:0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

User wants the mode as AP , so he/she has given 1 as input parameter. If user wants the mode as STA, he/she should give 0 as input parameter.

AT+WNI=1

+WNI:0

OK

8.1.7. Interface Statistics in AP Mode

Command Description:

To get statistics of WLAN interface which is in AP mode

Pre-requisites:

AT+WNCR command should be issued

Command Response

If getting statistics is successful '+WNAPST:ucastTxPkts, ucastRxPkts, mcastTxPkts, mcastRxPkts, bcastTxPkts, bcastRxPkts

ucastNonNullTxPkts, ucastNonNullRxPkts, ucastFilteredAcceptedTxPkts, ucastFilteredAcceptedRxPkts, mcastFilteredAcceptedTxPkts, mcastFilteredAcceptedRxPkts,

bcastFilteredAcceptedTxPkts, bcastFilteredAcceptedRxPkts, ucastFilteredRejectedTxPkts, ucastFilteredRejectedRxPkts, mcastFilteredRejectedTxPkts,

mcastFilteredRejectedRxPkts, bcastFilteredRejectedTxPkts, bcastFilteredRejectedRxPkts, nullTxPkts, nullRxPkts, qosNullTxPkts, qosNullRxPkts,

psPollTxPkts, psPollRxPkts, txRetryCnt, beaconMissCnt, beaconsReceivedCnt, beaconResyncSuccessCnt, beaconResyncFailureCnt, currEarlyWakeupAdjInMs,

avgEarlyWakeupAdjInMs, earlyTerminationCnt, uapsdTriggerRxCnt, uapsdTriggerTxCnt, totalActiveTimeInMs, totalPowersaveTimeInMs' and 'OK'

is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
ucastTxPkts	UINT	
ucastRxPkts	UINT	
mcastTxPkts	UINT	
mcastRxPkts	UINT	
bcastTxPkts	UINT	
bcastRxPkts	UINT	
ucastNonNullTxPkts	UINT	
ucastNonNullRxPkts	UINT	
ucastFilteredAcceptedTxPkts	UINT	
ucastFilteredAcceptedRxPkts	UINT	
mcastFilteredAcceptedTxPkts	UINT	

mcastFilteredAcceptedRxPkts	UINT
bcastFilteredAcceptedTxPkts	UINT
bcastFilteredAcceptedRxPkts	UINT
ucastFilteredRejectedTxPkts	UINT
ucastFilteredRejectedRxPkts	UINT
mcastFilteredRejectedTxPkts	UINT
mcastFilteredRejectedRxPkts	UINT
bcastFilteredRejectedTxPkts	UINT
bcastFilteredRejectedRxPkts	UINT
nullTxPkts	UINT
nullRxPkts	UINT
qosNullTxPkts	UINT
qosNullRxPkts	UINT
psPollTxPkts	UINT
psPollRxPkts	UINT
txRetryCnt	UINT
beaconMissCnt	UINT
beaconsReceivedCnt	UINT
beaconResyncSuccessCnt	UINT
beaconResyncFailureCnt	UINT

currEarlyWakeupAdjInMs	UINT
avgEarlyWakeupAdjInMs	UINT
earlyTerminationCnt	UINT
uapsdTriggerRxCnt	UINT
uapsdTriggerTxCnt	UINT
totalActiveTimeInMs	UINT
totalPowersaveTimeInMs	UINT

Command Syntax:

AT+WNAPST

Response:

If AT+WNAPST is successful, returns SUCCESS else FAILURE.

[illegible]

<CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

This command is used to get the all statistics of WLAN interface when it is in AP mode

$$AT + WNI = 1$$
 $+WNI=0$

OK

AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1"

OK

AT+WNAPIPC=0,"192.168.23.24","192.168.23.55",86400000

OK

AT+WNCR=0,"Telit Guest",6,"WPA","TKIP","12345678"

+WNCR:CREATED

OK

AT+WNAPST

+WNAPST:0,0,0,0,0,0,0,-

971,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,971,0,0,0,0,0,0,0,0,0,0,0,610437,0

OK

8.1.8. Get Associated Station Information

Command Description:

To print the information of connected stations with MAC address and IP address when WLAN interface is in AP mode

Pre-requisites:

AT+WNI=1, AT+WNIPC, AT+WNAIPC and AT+WNCR commands should be issued

Command Response

If getting stations connected information is successful '+WNASTAINFO: MAC Address, IP Address' and 'OK' is sent else ERROR is sent.

Response Name	Response Type	Response Range
MAC address	MAC	
IP address	IP	

Command Syntax:

AT+WNASTINFO

Response:

If AT+WNASTAINFO is successful, returns SUCCESS else FAILURE.

Responses	Remarks
First initialize the NCM by using AT+WNI=1. Set the Static IP configuration for AP by using AT+WNIPC then set the range of IP addresses so that the connected stations will get IP address in range by using AT+WNAIPIC. After every command returns success, give command AT+WNCR .Here the SSID of the AP is Telit_Guest and the password is 12345678. At last by giving AT+WNASTAINFO command, connected stations information such as MAC address and IP address will get printed AT+WNI=1<CR><LF> +WNI:0<CR><LF> <CR><LF>OK<CR><LF> AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1"<CR><LF> <CR><LF>OK<CR><LF> AT+WNAIPIC=0,"192.168.23.24","192.168.23.55",86400000<CR><LF> <CR><LF>OK<CR><LF> AT+WNCR=0,"Telit_Guest",6,"WPA","TKIP","12345678"<CR><LF> +WNCR:CREATED<CR><LF> <CR><LF>OK<CR><LF> AT+WNASTAINFO<CR><LF> +WNASTINFO:E0:9D:31:13:BB:38,192.168.3.3<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

First initialize the NCM by using AT+WNI=1.

Set the Static IP configuration for AP by using AT+WNIPC then set the range of IP addresses

so that the connected stations will get IP address in range by using AT+WNAPIPC.

After every command returns success, give command AT+WNCR .Here the SSID of the AP is

Telit_Guest and the password is 12345678. At last by giving AT+WNASTAINFO command, connected stations

information such as MAC address and IP address will get printed

AT+WNI=1

+WNI:0

OK

AT+WNIPC=0,1,"192.168.23.2","255.255.255.0","192.168.23.1"

OK

AT+WNAPIPC=0,"192.168.23.24","192.168.23.55",86400000

OK

AT+WNCR=0,"Telit_Guest",6,"WPA","TKIP","12345678"

+WNCR:CREATED

OK

AT+WNASTAINFO

+WNASTINFO:E0:9D:31:13:BB:38,192.168.3.3

OK

8.1.9. Interface Configuration Information

Command Description:

To get information of different interfaces

Pre-requisites:

AT+WNI command should be issued

Command Response

After connection to the network or creation of network is successful '+WNIFCFG:device_id, mac_addr, state, wstate,

mode, bssid, ssid, channel, security, rssi, IP address, subnet, gateway, dns1, dns2, dns3' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
device_id	UINT	
mac_address	MAC	
state	DOWN, UP	

wstate	NOT CONNECTED, CONNECTED
mode	NONE, STA, AP
bssid	MAC
ssid	STR
channel	INT
security	NONE, WEP, WPA PSK, WPA ENTERPRISE, WPA2 PSK, WPA2 ENTERPRISE
rssi	INT
IP address	IP
SN address	IP
GW address	IP
DNS1 address	IP
DNS2 address	IP
DNS3 address	IP

Command Syntax:

AT+WNIFCFG

Response:

If AT+WNIFCFG is successful, returns SUCCESS else FAILURE.

Responses	Remarks
First initialize the NCM by using AT+WNI=0 or 1. Then AT+WNIFCFG is used to get the information and status of both interfaces.	Successful

```

AT+WNI=0<CR><LF>
+WNI:1<CR><LF>
<CR><LF>OK<CR><LF>
AT+WNCN=1,"telit_guest"<CR><LF>
+WNCN:CONNECTED,192.168.3.45,255.255.255.0,192.168.3.1<CR><LF>
<CR><LF>OK<CR><LF>\b
AT+WNIFCFG<CR><LF>
+WNIFCFG:0,00:00:00:00:00:00,DOWN,NOT
CONNECTED,NONE,00:00:00:00:00:00,"",0,NONE,0,0.0.0.0,0.0.0.0,0.0.0.0,0.0.
0.0,0.0.0.0,0.0.0.0<CR><LF>
+WNIFCFG:1,11:22:33:44:55:66, UP, CONNECTED, STA,00:aa:bb:cc:dd:ee,
telit_guest, 6,
NONE,192.168.3.45,255.255.255.0,192.168.3.1,192.168.3.1,0.0.0.0,0.0.0.0<CR>
<LF>
<CR><LF>OK<CR><LF>

```

ERROR

If
parame
ter is
not
valid

Example:

First initialize the NCM by using AT+WNI=0 or 1. Then AT+WNIFCFG is used to get the information and status of both interfaces.

```

AT+WNI=0
+WNI:1
OK
AT+WNCN=1,"telit_guest"
+WNCN:CONNECTED,192.168.3.45,255.255.255.0,192.168.3.1
OK\b
AT+WNIFCFG
+WNIFCFG:0,00:00:00:00:00:00,DOWN,NOT
CONNECTED,NONE,00:00:00:00:00:00,"",0,NONE,0,0.0.0.0,0.0.0.0,0.0.0.0,0.0.0.0,0.0.0.0,
0.0.0.0
+WNIFCFG:1,11:22:33:44:55:66, UP, CONNECTED, STA,00:aa:bb:cc:dd:ee, telit_guest, 6,
NONE,192.168.3.45,255.255.255.0,192.168.3.1,192.168.3.1,0.0.0.0,0.0.0.0
OK

```

8.1.10. Interface Statistics in Station Mode

Command Description:

To get statistics of WLAN interface which is in AP mode

Pre-requisites:

AT+WNCN command should be issued

Command Response

If getting statistics is successful '+WNSTAST:ucastTxPkts, ucastRxPkts, mcastTxPkts, mcastRxPkts, bcastTxPkts, bcastRxPkts

ucastNonNullTxPkts, ucastNonNullRxPkts, ucastFilteredAcceptedTxPkts, ucastFilteredAcceptedRxPkts, mcastFilteredAcceptedTxPkts, mcastFilteredAcceptedRxPkts,

bcastFilteredAcceptedTxPkts, bcastFilteredAcceptedRxPkts, ucastFilteredRejectedTxPkts, ucastFilteredRejectedRxPkts, mcastFilteredRejectedTxPkts,

mcastFilteredRejectedRxPkts, bcastFilteredRejectedTxPkts, bcastFilteredRejectedRxPkts, nullTxPkts, nullRxPkts, qosNullTxPkts, qosNullRxPkts,

psPollTxPkts, psPollRxPkts, txRetryCnt, beaconMissCnt, beaconsReceivedCnt, beaconResyncSuccessCnt, beaconResyncFailureCnt, currEarlyWakeupAdjInMs,

avgEarlyWakeupAdjInMs, earlyTerminationCnt, uapsdTriggerRxCnt, uapsdTriggerTxCnt, totalActiveTimeInMs, totalPowersaveTimeInMs' and 'OK'

is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
ucastTxPkts	UINT	
ucastRxPkts	UINT	
mcastTxPkts	UINT	
mcastRxPkts	UINT	
bcastTxPkts	UINT	
bcastRxPkts	UINT	
ucastNonNullTxPkts	UINT	
ucastNonNullRxPkts	UINT	
ucastFilteredAcceptedTxPkts	UINT	

ucastFilteredAcceptedRxPkts	UINT
mcastFilteredAcceptedTxPkts	UINT
mcastFilteredAcceptedRxPkts	UINT
bcastFilteredAcceptedTxPkts	UINT
bcastFilteredAcceptedRxPkts	UINT
ucastFilteredRejectedTxPkts	UINT
ucastFilteredRejectedRxPkts	UINT
mcastFilteredRejectedTxPkts	UINT
mcastFilteredRejectedRxPkts	UINT
bcastFilteredRejectedTxPkts	UINT
bcastFilteredRejectedRxPkts	UINT
nullTxPkts	UINT
nullRxPkts	UINT
qosNullTxPkts	UINT
qosNullRxPkts	UINT
psPollTxPkts	UINT
psPollRxPkts	UINT
txRetryCnt	UINT
beaconMissCnt	UINT
beaconsReceivedCnt	UINT

beaconResyncSuccessCnt	UINT
beaconResyncFailureCnt	UINT
currEarlyWakeupAdjInMs	UINT
avgEarlyWakeupAdjInMs	UINT
earlyTerminationCnt	UINT
uapsdTriggerRxCnt	UINT
uapsdTriggerTxCnt	UINT
totalActiveTimeInMs	UINT
totalPowersaveTimeInMs	UINT

Command Syntax:

AT+WNSTAST

Response:

If AT+WNSTAST is successful, returns SUCCESS else FAILURE.

Responses	Remarks
This command is used to get the all statistics of WLAN interface when it is in AP mode AT+WNI=0<CR><LF> +WNI=1<CR><LF> <CR><LF>OK<CR><LF> AT+WNCN=1,"Telit_Guest","12345678"<CR><LF> +WNCN:Connected, 192.168.1.23,192.168.1.1,255.255.255.0<CR><LF> <CR><LF>OK<CR><LF> AT+WNSTAST<CR><LF> +WNSTAST:0,0,5,45,2,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,129,1,0,18,18,0,0,0,77700,0<CR><LF> <CR><LF>OK<CR><LF>	Successful

Response Name	Response Type	Response Range
disconnect status	not disconnected, disconnected, down	0-1

Response:

If AT+WNDK is successful, returns SUCCESS else FAILURE.

Responses	Remarks
If user wants to use this command , he/she should first connect to a network in STATION mode or create a network in AP mode. AT+WNI=0<CR><LF> +WNI:1<CR><LF> <CR><LF>OK<CR><LF> AT+WNCN=1,"Telit"<CR><LF> +WNCN:CONNECTED,192.168.3.5,255.255.255.0,192.168.3.1<CR><LF> <CR><LF>OK<CR><LF> AT+WNDK=1<CR><LF> +WNDK:DISCONNECTED<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

If user wants to use this command , he/she should first connect to a network in STATION mode or create

a network in AP mode.

AT+WNI=0

+WNI:1

OK

AT+WNCN=1,"Telit"

+WNCN:CONNECTED,192.168.3.5,255.255.255.0,192.168.3.1

OK

AT+WNDK=1

+WDC:DISCONNECTED

OK

8.2. Socket

8.2.1. Create Socket

Command Description:

To create socket and add CID entry

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+SC=<Family>,<Type>,[<Protocol>]

Parameter Description:

Parameters	Required?	Value	Description
Family	M	Value Range: 2-5 Format: INT Default value:	specifies communications domain in which a socket is to be created. 2 for AF_INET(IPv4), 3 for AF_INET6(IPv6), 4 for AF_INET_DUAL46(IPv4 and IPv6), 5 for AF_PACKET (packet).
Type	M	Value Range: 1-3 Format: INT Default value:	specifies type of socket to be created. 1 for SOCK_STREAM(eg-TCP), 2 for SOCK_DGRAM(eg-UDP), 3 for SOCK_RAW.
Protocol	O	Value Range: 0-255, 1536-2147483647 Format: INT Default value: 0	specifies a particular protocol to be used with socket. 0 for TCP and UDP.

Command Response

If socket creation is successful '+SC:cid(cid value)' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
cid	UINT	

Response:

If socket create is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+SC=2,2,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+SC=2,2,0

+SC:0

OK

8.2.2. Bind Socket

Command Description:

To bind to a socket

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+SB=<cid>,[<LocalIp>],<port>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by AT+SC command
LocalIp	O	Value Range: Format: IP Default value: 0	Local ip address to bind
port	M	Value Range: 0-65535 Format: UINT	Local port number to bind

Default value:

Response:

If bind is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+SC=2,2,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+SC=2,2,0
+SC:0
OK
AT+SB=0,192.168.120.5,8355
OK
```

8.2.3. Connect Socket

Command Description:

To connect to a server

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+SCO=<cid>,<ServerIp>,<Server_port>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by AT+SC command

ServerIp	M	Value Range:	server IP to connect
		Format: IP	
		Default value:	
Server_port	M	Value Range: 0-65535	server port number to connect
		Format: UINT	
		Default value:	

Response:

If connect is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+SC=2,1,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SCO=0,192.168.140.3,8355<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+SC=2,1,0
+SC:0
OK
AT+SCO=0,192.168.140.3,8355
OK
```

8.2.4. Socket Listen

Command Description:

To listen to a socket

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+SL=<cid>, [<backlog>]
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by the AT+SC command
backlog	O	Value Range: 0-15 Format: INT Default value: 0	Defines maximum length to which queue of pending connections for a handle may grow

Command Response

If Listen is successful module sends 'OK' else

'ERROR' is sent. this may give an asynchronous response as '+SL:cid'.

Response Name	Response Type	Response Range
cid	UINT	

Response:

If listening process is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+SC=2,1,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF> AT+SL=0<CR><LF> <CR><LF>OK<CR><LF> +SL=0<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+SC=2,1,0
```

```
+SC:0
```

```
OK
```

```
AT+SB=0,192.168.120.5,8355
```

```
OK
```

```
AT+SL=0
```

```
OK
```

```
+SL=0
```

8.2.5. Set Socket Option

Command Description:

sets specified socket option.

Pre-requisites:

should bring L2 and L3 layer connection UP.

Command Syntax:

```
AT+SSOPT=<cid>,<optname>,[<optval>]
```

Parameter Description:

Parameters	Required ?	Value	Description
cid	M	Value Range: 0-11 Format: : UINT Default value:	cid of the socket provided by either +SC command or +SA command.
optname	M	Value Range: 0-255 Format: : UINT Default value:	Name of the option. M2M_S2W_SOCKET_BSD_SO_DEBUG1, /* Turn on debugging info recording, Not supported */ M2M_S2W_SOCKET_BSD_SO_ACCEPTCONN2, /* Socket has had listen(), Not supported */ M2M_S2W_SOCKET_BSD_SO_REUSEADDR3, /* Allow local address reuse, always set */ M2M_S2W_SOCKET_BSD_SO_KEEPAIVE4, /* Keep connections alive, not enabled by default */ M2M_S2W_SOCKET_BSD_SO_DONTROUTE5, /* Just use interface addresses, Not supported */ M2M_S2W_SOCKET_BSD_SO_BROADCAST6,

```

/* Permit sending of broadcast msgs, Not supported */
M2M_S2W_SOCKET_BSD_SO_USELOOPBACK7,
/* Permit sending of broadcast msgs, Not supported */
M2M_S2W_SOCKET_BSD_SO_LINGER8, /*
Linger on close if data present, Not supported */
M2M_S2W_SOCKET_BSD_SO_OOBINLINE9, /*
Leave received OOB data in line, Not supported */
M2M_S2W_SOCKET_BSD_SO_DONTLINGER10, /*
Dont Linger, Not supported */
M2M_S2W_SOCKET_BSD_SO_TCPSACK11, /**
Allow TCP SACK (selective acknowledgment). */
M2M_S2W_SOCKET_BSD_SO_WINSIZE12, /**
Set the scaling window option. */
M2M_S2W_SOCKET_BSD_SO_TIMESTAMP13, /**
Set the TCP timestamp option. */
M2M_S2W_SOCKET_BSD_SO_BIGCWIND14, /**
Large initial TCP congestion window. */
M2M_S2W_SOCKET_BSD_SO_HDRINCL15, /**
User access to the IP header for SOCK_RAW. */
M2M_S2W_SOCKET_BSD_SO_NOSLOWSTART16,
/** Suppress slow start on this socket. */
M2M_S2W_SOCKET_BSD_SO_FULLMSS17, /**
Not used. */
M2M_S2W_SOCKET_BSD_SO_SNDBUF18, /*
Send buffer size, supported */
M2M_S2W_SOCKET_BSD_SO_RCVBUF19, /*
Receive buffer size, supported */
M2M_S2W_SOCKET_BSD_SO_SNDLOWAT20,
/* Send low-water mark, Not supported */
M2M_S2W_SOCKET_BSD_SO_RCVLOWAT21,
/* Receive low-water mark, Not supported */
M2M_S2W_SOCKET_BSD_SO_SNDTIMEO22,
/* Send timeout, Not supported */
M2M_S2W_SOCKET_BSD_SO_RCVTIMEO23,
/* Receive timeout, supported */
M2M_S2W_SOCKET_BSD_SO_ERROR24, /*
Get error status and clear */
M2M_S2W_SOCKET_BSD_SO_TYPE25, /*
Get socket type, supported */
M2M_S2W_SOCKET_BSD_SO_HOPCNT26, /**
Not used. */
M2M_S2W_SOCKET_BSD_SO_MAXMSG27, /**
TCP max segment size (MSS). */
M2M_S2W_SOCKET_BSD_SO_RXDATA28, /*

```

```

Receive buffer size,      supported      */
M2M_S2W_SOCKET_BSD_SO_TXDATA29,      /*
Send buffer size,      supported      */
M2M_S2W_SOCKET_BSD_SO_MYADDR30,      /**
Return my IP address. */
M2M_S2W_SOCKET_BSD_SO_NBIO31,  /** Set
socket to Nonblocking mode. */
M2M_S2W_SOCKET_BSD_SO_BIO32,      /** Set
socket to Blocking mode. */
M2M_S2W_SOCKET_BSD_SO_NONBLOCK33,  /**
Set/get the blocking mode via the optval parameter. */
M2M_S2W_SOCKET_BSD_SO_CALLBACK34,  /**
Set/get the TCP zero_copy callback routine. */
M2M_S2W_SOCKET_BSD_SO_UDPCALLBACK35,
/** Set/get the UDP zero_copy callback routine. */

M2M_S2W_SOCKET_BSD_IPPROTO_IP36,      /**
For use with [gs]setsockopt() at IPPROTO_IP 'level'. */
M2M_S2W_SOCKET_BSD_IP_HDRINCL37,      /**
IP header is included with the data. */
M2M_S2W_SOCKET_BSD_IP_ADD_MEMBERSHIP3
8,  /** Add an IPv4 group membership. */
M2M_S2W_SOCKET_BSD_IP_DROP_MEMBERSHIP
39,  /** Drop an IPv4 group membership. */
M2M_S2W_SOCKET_BSD_IPV6_MULTICAST_IF40,
/** Set the egress interface for multicast traffic. */
M2M_S2W_SOCKET_BSD_IPV6_MULTICAST_HOP
S41,  /** Set the number of hops. */
M2M_S2W_SOCKET_BSD_IPV6_MULTICAST_LOOP
42,  /** Enable/disable loopback for multicast. */
M2M_S2W_SOCKET_BSD_IPV6_JOIN_GROUP43,
/** Join an IPv6 MC group. */
M2M_S2W_SOCKET_BSD_IPV6_LEAVE_GROUP44,
/** Leave an IPv6 MC group. */

M2M_S2W_SOCKET_BSD_IP_OPTIONS45,      /**
For use with [gs]setsockopt() at IP_OPTIONS 'level'. */
M2M_S2W_SOCKET_BSD_IP_TOS46,      /**
IPv4 type of service and precedence. */
M2M_S2W_SOCKET_BSD_IP_TTL_OPT47,      /**
IPv4 time to live. */
M2M_S2W_SOCKET_BSD_IPV6_SCOPEID48,
/** IPv6 IF scope ID. */

```

M2M_S2W_SOCKET_BSD_IPV6_UNICAST_HOPS49 , /** IPv6 hop limit. */ M2M_S2W_SOCKET_BSD_IPV6_TCLASS50 /** IPv6 traffic class. */			
optval	O	Value Range: 1-40 Format : STR Default value:	option value to be set.

Response:

Responses	Remarks
AT+SSOPT=0,5,0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+SSOPT=0,5,0

OK

8.2.6. Get Socket Option

Command Description:

gets specified socket option.

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+SGOPT=<cid>,<optname>

Parameter Description:

Parameters	Required ?	Value	Description
cid	M	Value Range: 0-11	cid of the socket provided by either +SC command or +SA command.

		Format : UINT
		Default value:
optname	M	Value Name of the option. Range: 0-255 Format : UINT Default value: M2M_S2W__S2W_SOCKET_BSD_SO_DEBUG1, /* Turn on debugging info recording, Not supported */ M2M_S2W_SOCKET_BSD_SO_ACCEPTCONN2, /* Socket has had listen(), Not supported */ M2M_S2W_SOCKET_BSD_SO_REUSEADDR3, /* Allow local address reuse, always set */ M2M_S2W_SOCKET_BSD_SO_KEEPAIVE4, /* Keep connections alive, not enabled by default */ M2M_S2W_SOCKET_BSD_SO_DONTROUTE5, /* Just use interface addresses, Not supported */ M2M_S2W_SOCKET_BSD_SO_BROADCAST6, /* Permit sending of broadcast msgs, Not supported */ M2M_S2W_SOCKET_BSD_SO_USELOOPBACK7, /* Permit sending of broadcast msgs, Not supported */ M2M_S2W_SOCKET_BSD_SO_LINGER8, /* Linger on close if data present, Not supported */ M2M_S2W_SOCKET_BSD_SO_OOBINLINE9, /* Leave received OOB data in line, Not supported */ M2M_S2W_SOCKET_BSD_SO_DONTLINGER10, /* Dont Linger, Not supported */ M2M_S2W_SOCKET_BSD_SO_TCPSACK11, /** Allow TCP SACK (selective acknowledgment). */ M2M_S2W_SOCKET_BSD_SO_WINSIZE12, /** Set the scaling window option. */ M2M_S2W_SOCKET_BSD_SO_TIMESTAMP13, /** Set the TCP timestamp option. */ M2M_S2W_SOCKET_BSD_SO_BIGCWIND14, /** Large initial TCP congestion window. */ M2M_S2W_SOCKET_BSD_SO_HDRINCL15, /** User access to the IP header for SOCK_RAW. */ M2M_S2W_SOCKET_BSD_SO_NOSLOWSTART16, /** Suppress slow start on this socket. */ M2M_S2W_SOCKET_BSD_SO_FULLMSS17, /** Not used. */ M2M_S2W_SOCKET_BSD_SO_SNDBUF18, /* Send buffer size, supported */ M2M_S2W_SOCKET_BSD_SO_RCVBUF19, /*

```

Receive buffer size,      supported      */
M2M_S2W_SOCKET_BSD_SO_SNDBUFSIZE20,
/* Send low-water mark,    Not supported */
M2M_S2W_SOCKET_BSD_SO_RCVBUFSIZE21,
/* Receive low-water mark,  Not supported */
M2M_S2W_SOCKET_BSD_SO_SNDTIMEO22,
/* Send timeout,           Not supported */
M2M_S2W_SOCKET_BSD_SO_RCVTIMEO23,
/* Receive timeout,        supported     */
M2M_S2W_SOCKET_BSD_SO_ERROR24,          /*
Get error status and clear                */
M2M_S2W_SOCKET_BSD_SO_TYPE25,          /*
Get socket type,                         supported */
M2M_S2W_SOCKET_BSD_SO_HOPCNT26,        /**
Not used. */
M2M_S2W_SOCKET_BSD_SO_MAXMSG27,        /**
TCP max segment size (MSS). */
M2M_S2W_SOCKET_BSD_SO_RXDATA28,        /*
Receive buffer size,      supported      */
M2M_S2W_SOCKET_BSD_SO_TXDATA29,        /*
Send buffer size,         supported      */
M2M_S2W_SOCKET_BSD_SO_MYADDR30,        /**
Return my IP address. */
M2M_S2W_SOCKET_BSD_SO_NBIO31, /** Set
socket to Nonblocking mode. */
M2M_S2W_SOCKET_BSD_SO_BIO32,          /** Set
socket to Blocking mode. */
M2M_S2W_SOCKET_BSD_SO_NONBLOCK33,     /**
Set/get the blocking mode via the optval parameter. */
M2M_S2W_SOCKET_BSD_SO_CALLBACK34,     /**
Set/get the TCP zero_copy callback routine. */
M2M_S2W_SOCKET_BSD_SO_UDPCALLBACK35,  /**
Set/get the UDP zero_copy callback routine. */

M2M_S2W_SOCKET_BSD_IPPROTO_IP36,      /**
For use with [gs]setsockopt() at IPPROTO_IP 'level'. */
M2M_S2W_SOCKET_BSD_IP_HDRINCL37,      /**
IP header is included with the data. */
M2M_S2W_SOCKET_BSD_IP_ADD_MEMBERSHIP38, /**
Add an IPv4 group membership. */
M2M_S2W_SOCKET_BSD_IP_DROP_MEMBERSHIP39, /**
Drop an IPv4 group membership. */
M2M_S2W_SOCKET_BSD_IPV6_MULTICAST_IF40,

```

```

/** Set the egress interface for multicast traffic. */
M2M_S2W_SOCKET_BSD_IPV6_MULTICAST_HOP
S41, /** Set the number of hops. */

M2M_S2W_SOCKET_BSD_IPV6_MULTICAST_LOOP
42, /** Enable/disable loopback for multicast. */

M2M_S2W_SOCKET_BSD_IPV6_JOIN_GROUP43,
/** Join an IPv6 MC group. */

M2M_S2W_SOCKET_BSD_IPV6_LEAVE_GROUP44,
/** Leave an IPv6 MC group. */

M2M_S2W_SOCKET_BSD_IP_OPTIONS45, /**
For use with [gs]etsockopt() at IP_OPTIONS 'level'. */

M2M_S2W_SOCKET_BSD_IP_TOS46, /**
IPv4 type of service and precedence. */

M2M_S2W_SOCKET_BSD_IP_TTL_OPT47, /**
IPv4 time to live. */

M2M_S2W_SOCKET_BSD_IPV6_SCOPEID48,
/** IPv6 IF scope ID. */

M2M_S2W_SOCKET_BSD_IPV6_UNICAST_HOPS49
, /** IPv6 hop limit. */

M2M_S2W_SOCKET_BSD_IPV6_TCLASS50 /**
IPv6 traffic class. */

```

Command Response

If socket option get is successful '+SGOPT:optval' and 'OK' is sent else 'ERROR' is sent.

Response Name	Response Type	Response Range
optval	STR	

Response:

Responses	Remarks
AT+SGOPT=0,19<CR><LF> +SGOPT:1462 <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+SGOPT=0,19
```

```
+SGOPT:1462
```

```
OK
```

8.2.7. Accept Socket

Command Description:

To accept client connection and add CID entry

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+SA=<cid>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by AT+SC command

Command Response

If accept is successful module sends response as

'+SA:Client CID, Client IP, Client port' and 'OK' else 'ERROR' will sent.

Response Name	Response Type	Response Range
Client_Cid	UINT	
Client_IP	IP	
Client_port	UINT	

Response:

If accept is successful, API returns SUCCESS, else reutrns FAILURE.

Responses	Remarks
-----------	---------

AT+SC=2,1,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF> AT+SL=0<CR><LF> <CR><LF>OK<CR><LF> +SL=0<CR><LF> AT+SA=0<CR><LF> +SA:Client CID, Client IP, Client port<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+SC=2,1,0
+SC:0
OK
AT+SB=0,192.168.120.5,8355
OK
AT+SL=0
OK
+SL=0
AT+SA=0
+SA:Client CID, Client IP, Client port
OK

8.2.8. Send Data to Socket

Command Description:

To send data to specified IP address

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+SN=<cid>,[<Dst>,<port>],<data length>,<Data>

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

cid	M	Value Range: 0-11 Format: UINT Default value:	CID provided by either AT+SC command or AT+SA command.
Dst	O	Value Range: Format: IP Default value:	Destination IP address to send data, it is not needed in case of TCP.
port	O	Value Range: 0-65535 Format: UINT Default value:	Destination port number to send data, it is not needed in case of TCP.
data length	M	Value Range: 1-4294967295 Format: INT Default value:	Data to send.
Data	M	Value Range: 1-4294967295 Format: BIN Default value:	Data to send.

Command Response

If Data send is successful module sends 'OK' and no response is sent, otherwise module sends response sends as '+SN:CID,length(total number of bytes sent till then)' and sends 'ERROR'. this is synchronous response.

Response Name	Response Type	Response Range
cid	UINT	
Length	UINT	

Response:

If Data send successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
send the data through the created socket. AT+SC=2,2,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SN=0,192.168.140.3,8355,100,0123456789<CR><LF> <CR><LF>OK<CR><LF> AT+SN=0,192.168.140.3,8355,100,0123456789<CR><LF> <CR><LF>OK<CR><LF> AT+SN=0,192.168.140.3,8355,100,0123456789<CR><LF> +SN=0,20<CR><LF> <CR><LF>ERROR<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

send the data through the created socket.

```
AT+SC=2,2,0
+SC:0
OK
AT+SN=0,192.168.140.3,8355,100,0123456789
OK
AT+SN=0,192.168.140.3,8355,100,0123456789
OK
AT+SN=0,192.168.140.3,8355,100,0123456789
+SN=0,20
ERROR
```

8.2.9. Socket Receive Ready

Command Description:

This command takes care of enabling data reception on a given CID at application level and indicates to the user when data is available on a particular CID.

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+SRR=<cid>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by either AT+SC command or AT+SA command

Command Response

If enabling data reception is successful module sends 'OK' and sends response asynchronously, In case of failure, module sends 'ERROR' and no response is given.

in case of success module sends response asynchronously as
'+SRR:CID,RIP,RP,length'

Response Name	Response Type	Response Range
cid	UINT	
RIP	IP	
RP	UINT	
Length	UINT	

Response:

If Data receive is successful, it returns SUCCESS else FAILURE.

Responses	Remarks
receive the data through the created socket. AT+SC=2,2,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF> AT+SRR=0<CR><LF>	Successful

```
<CR><LF>OK<CR><LF>
```

```
+SRR:0,RIP,RP,length<CR><LF>
```

ERROR

If parameter is not valid

Example:

receive the data through the created socket.

```
AT+SC=2,2,0
```

```
+SC:0
```

```
OK
```

```
AT+SB=0,192.168.120.5,8355
```

```
OK
```

```
AT+SRR=0
```

```
OK
```

```
+SRR:0,RIP,RP,length
```

8.2.10. Receive Socket

Command Description:

To receives data from any CID

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+SR=<cid>,<Length>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	CID value provided by either AT+SC command or AT+SA command
Length	M	Value Range: 1-4294967295 Format: UINT Default value:	number of bytes to receive, indicated in at+SRR command response

Command Response

If data reception is successful module sends

response as '+SR:RIP,RP,TOTAL LENGTH,PRESENT LENGTH,DATA' and 'OK'.

if the data reception is failed or partially received module sends response as

'+SR:RIP,RP,TOTAL LENGTH,PRESENT LENGTH,DATA' and 'ERROR'

here TOTAL LENGTH is requested data length to receive,

PRESENT LENGTH is received data length in current response,

DATA is the received data of size PRESENT LENGTH,

this is synchronous response,

it may have multiple responses in case of bulk data reception.

Response Name	Response Type	Response Range
RIP	IP	
RP	UINT	
data length	INT	
data	BIN	

Response:

If Data receive is successful API returns SUCCESS otherwise returns FAILURE.

Responses	Remarks
receive the data through the created socket. AT+SC=2,2,0<CR><LF> +SC:0<CR><LF> <CR><LF>OK<CR><LF> AT+SB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF> AT+SRR=0<CR><LF> <CR><LF>OK<CR><LF> +SRR:0,RIP,RP,length<CR><LF> AT+SR=2,length<CR><LF> +SR:192.168.120.10,8366,length,data of 100 bytes<CR><LF> <CR><LF>OK<CR><LF>	Successful

ERROR

If parameter is not valid

Example:

receive the data through the created socket.

AT+SC=2,2,0

+SC:0

OK

AT+SB=0,192.168.120.5,8355

OK

AT+SRR=0

OK

+SRR:0,RIP,RP,length

AT+SR=2,length

+SR:192.168.120.10,8366,length,data of 100 bytes

OK

8.2.11. Get Socket Error

Command Description:

gets specified socket errno.

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+SGERR=<cid>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by either +SC command or +SA command.

Command Response

If socket errno get is successful '+SGERR:errno' and 'OK' is sent else

'ERROR' is sent.

Err noError description

(-1) SOCKET ERROR

0 SOCKET NO ERROR

- 1 No buffer space is available.
- 2 Operation timed out.
- 3 Socket is already connected.
- 4 Operation is not supported.
- 5 Software caused a connection abort.
- 6 The socket is marked nonblocking and the requested operation would block.
- 7 Connection was refused.
- 8 Connection was reset by a peer.
- 9 Socket is not connected.
- 9 An invalid descriptor was specified.
- 10 Operation is already in progress.
- 11 Invalid argument was passed.
- 12 Message is too long.
- 13 Local end has been shut down on a connection-oriented socket.
- 14 Destination address is required.
- 15 Cannot send after a socket shutdown.
- 16 Protocol is not available.
- 17 Out of band.
- 18 No memory is available.
- 19 Cannot assign the requested address.
- 20 Address is already in use.
- 21 Address family is not supported by the protocol family.
- 22 Operation is now in progress.
- 23 Lower layer (IP) error.
- 24 Socket operation is on a nonsocket.
- 27 I/O error.
- 28 Too many references.
- 29 Bad address.
- 30 Network is unreachable.

Response Name	Response Type	Response Range
errno	INT	

Response:

Responses	Remarks
AT+SGERR=0<CR><LF> Successful +SGERR:10 <CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

AT+SGERR=0

+SGERR:10

OK

8.2.12. Socket Throughput Test

Command Description:

To test throughput

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+STPTEST=<cid>,<mode>,<iterations>,<pktSize>,[<pktRepeatCnt>
,<logInterval>,<logLocation>,<test Duration>]
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by either AT+SC command or AT+SA command
mode	M	Value Range: 0-6 Format: UINT Default value:	test mode 0 - Txt 1- Rx TCP 2- Both TCP 3- Incremental Packet 4- Rx UDP 5 - Both UDP
iterations	M	Value Range: 1-	number of iterations

		4294967295 Format: UINT Default value:	
pktSize	M	Value Range: 1-1460 Format: UINT Default value:	pktSize to test
pktRepeatCnt	O	Value Range: 1-1000 Format: UINT Default value: 1	packet repeat count to test
logInterval	O	Value Range: 500-3600000 Format: UINT Default value: 5000	log Interval to test.
logLocation	O	Value Range: 1-3 Format: UINT Default value: 1	log Location to test. Forward to Host: 1 Store in RTC: 2 Store in file system: 3
test Duration	O	Value Range: 0-15 Format: UINT Default value: 0	test Duration to test

Command Response

If the throughput test succeeds it prints all statistics.

Response Name	Response Ty
---------------	-------------

Response:

Responses	Remarks
-----------	---------

at+socktptest=0,0,100000,1460<CR><LF>		Successful
Total Time Taken : 51535157 usec<CR><LF>		
Throuhput : 22664208 bps<CR><LF>		
Throuhput : 22133 kbps <CR><LF>		
<CR><LF>OK<CR><LF>		
ERROR		If parameter is not valid

Example:

```
at+socktptest=0,0,100000,1460
```

```
Total Time Taken : 51535157 usec
```

```
Throuhput : 22664208 bps
```

```
Throuhput : 22133 kbps
```

```
OK
```

8.2.13. Close Socket

Command Description:

To close socket and frees CID entry

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+SCL=<cid>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by either AT+SC command or by AT+SA command

Command Response

If socket close is successful module sends '+SCL:cid' and 'OK'
else 'ERROR' will send.

Response Name	Response Type	Response Range
cid	UINT	

Response:

If closing process is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+SCL=0<CR><LF> +SCL=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+SCL=0

+SCL=0

OK

8.3. SSL**8.3.1. Initialize SSL****Command Description:**

To initialize SSL module

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+NSSLINIT=<role>

Parameter Description:

Parameters	Required?	Value	Description
role	M	Value Range: 0-1 Format: UINT Default value: 0	specifies ssl role, 0 - client or 1 - server

Command Response

If sslInit is successful +NSSLINIT:cid is sent else

failure is sent.

Response Name	Response Type	Response Range
cid	UINT	0-15

Response:

If ssl init is successful, it returns cid, else returns FAILURE.

Responses	Remarks
AT+NSSLinit=1<CR><LF> +NSSLINIT:0<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+NSSLinit=1

+NSSLINIT:0

8.3.2. Configure SSL

Command Description:

To configure SSL connection

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+NSSLCFG=<CID>,<CONF_ID>,<CONF_VAL>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: 0	CID given by AT+NSSLINIT command
CONF_ID	M	Value Range: 0-20	the configuration id as described in the list of available configurations

	Format: INT	0 : invalid config param
	Default value: 0	1 : To Set SSL Protocol Version
		2 : To Set SSL Certificate
		3 : To Set SSL CA Certificate
		4 : To Set Cipher Information
		5 : To Enable/Disable Time Validation
		6 : To Enable/Disable SSL Alert
		7 : To Set The Domain
		8 : To Set the Max Fragment Length
		9 : To Set Negotiation
		10 : To Set the SNI
		11 : To Set the ALPN Extension
CONF_VAL M	Value Range: 1-64	specifies string value specific to the configuration provided in CONF_ID.
	Format: STR	
	Default value: ""	CONF_ID :CONF_VAL
		0 : Invalid
		1 : Values for CONF_VAL:
		0 : TLS Version 1.0
		1 : TLS Version 1_1
		2 : TLS Version 1_2
		3 : DTLS Version 1.0
		4 : DTLS Version 1_2
		Note Server will use either TLS or DTLS type internally based on the above config
		2 :ssl certificate Name
		3 : ssl CA Certificate Name
		4 : Cipher Information
		5 : Time Validation
		0 : Disable Time validation
		1 : Enable Time validation
		6 : Enable/Disable SSL Alert
		0 : Disable SSL Alert Indication
		1 : Enable SSL Alert

7: Domain name in String format

8 : Set the Max Fragment Length

9 : Set Negotiation

10 : To Set the SNI

11 : To Set the ALPN Extension

Response:

If ssl config is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NSSLinit = 0<CR><LF> +NSSLINIT:0<CR><LF>OK<CR><LF> AT+NSSLconfig= 0, 2, "clientcert"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+NSSLinit = 0

+NSSLINIT:0OK

AT+NSSLconfig= 0, 2, "clientcert"

OK

8.3.3. SSL Connect

Command Description:

To connect to a ssl server

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+NSSLCO=<cid>,<serverIp>,<server_port>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15	cid given by AT+NSSLINIT command

		Format: UINT Default value: 0	
serverip	M	Value Range: 7-40 Format: IP Default value: 0.0.0.0	server IP to connect
server_port	M	Value Range: 0-65535 Format: UINT Default value: 0	ssl server port number to connect

Response:

If connect is successful, it returns SUCCESS, else reutrns FAILURE.

Responses	Remarks
AT+NSSLINIT=1,2<CR><LF> +NSSLINIT:0<CR><LF> AT+NSSLCFG=0,1,"clientcert"<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLCFG=0,4,"1"<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLCO=0,192.168.140.3,8355<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSSLINIT=1,2
+NSSLINIT:0
AT+NSSLCFG=0,1,"clientcert"
OK
AT+NSSLCFG=0,4,"1"
OK
AT+NSSLCO=0,192.168.140.3,8355
OK
```

8.3.4. Bind SSL

Command Description:

To bind server port to a socket

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+NSSLB=<cid>,[<localIp>],<local_port>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	cid given by AT+NSSLINIT command
localIp	O	Value Range: 7-40 Format: IP Default value: 0.0.0.0	Local IP address to bind
local_port	M	Value Range: 0-65535 Format: UINT Default value: 0	Local port number to bind

Response:

If bind is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NSSLINIT=2,2<CR><LF> +SNSSLINIT:0<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+NSSLINIT=2,2

+SNSSLINIT:0

OK

AT+NSSLB=0,192.168.120.5,8355

OK

8.3.5. SSL Listen

Command Description:

To listen to a SSL socket

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+NSSL=<cid>, [<backlog>]

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	cid given by the AT+NSSLINIT command
backlog	O	Value Range: 0-15 Format: INT Default value: 0	Define's the maximum length to which the queue of pending connections for a handle may grow.

Command Response

If Listen is successful module returns +SL:cid

In case of failure returns failure and no response is given.

Response Name	Response Type	Response Range
cid	UINT	0-15

Response:

If listening process is successful, it returns SUCCESS, else reutrns FAILURE.

Responses	Remarks
<pre>at+nsslinit=2,2<CR><LF> +NSSLINIT:0<CR><LF> <CR><LF>OK<CR><LF> at+nsslb=0,192.168.120.3,9500<CR><LF> <CR><LF>OK<CR><LF> at+nssl=0<CR><LF> <CR><LF>OK<CR><LF> +nssl=0<CR><LF></pre>	Successful
ERROR	If parameter is not valid

Example:

```
at+nsslinit=2,2
+NSSLINIT:0
OK
at+nsslb=0,192.168.120.3,9500
OK
at+nssl=0
OK
+nssl=0
```

8.3.6. Accept SSL

Command Description:

To accept client connection and add it to cid table

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+NSSLA=<cid>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	cid given by the AT+NSSLINIT command

Command Response

If accept is successful +NSSLACCEPT:Client CID, Client IP, Client port is sent else +NSSLACCEPT:FAILED is sent.

Response Name	Response Type	Response Range
Client_Cid	UINT	0-15
Client_IP	IP	7-32
Client_port	UINT	0-65535

Response:

If accept process is successful, it returns Client CID, else returns FAILURE.

Responses	Remarks
AT+NSSLINIT=0,2<CR><LF> +NSSLINIT:0<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLB=0,192.168.120.5,8355<CR><LF> <CR><LF>OK<CR><LF> AT+NSSL=0<CR><LF> <CR><LF>OK<CR><LF> +NSSL=0<CR><LF> AT+NSSLA=0<CR><LF> +NSSLA:Client CID, Client IP, Client port<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSSLINIT=0,2
+NSSLINIT:0
OK
AT+NSSLB=0,192.168.120.5,8355
```

OK

AT+NSSL=0

OK

+NSSL=0

AT+NSSLA=0

+NSSLA:Client CID, Client IP, Client port

OK

8.3.7. Send Data to SSL

Command Description:

To send data to websocket remote device

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+WEBSOCKETSEND=<cid>,<datatype>,<data length>,<Data>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid given by AT+WEBSOCKETNEW command
datatype	M	Value Range: 0-1 Format: UINT Default value:	data type, 1 for text data, 2 for binary data
data length	M	Value Range: 1-4294967295 Format: INT Default value:	Data.
Data	M	Value Range: 1-4294967295 Format: BIN Default value:	Data.

Command Response

If Data send is successful module sends 'OK' and no response is sent,

otherwise module sends response as '+WEBSOCKETSEND:CID,length(total number of bytes sent till then)'

and sends 'ERROR'

this is synchronous response.

Response Name	Response Type	Response Range
cid	UINT	
Length	UINT	

Response:

If data send is successful, it returns number of bytes sent successfully, else returns failure.

Responses	Remarks
AT+WEBSOCKETNEW=2,2,0<CR><LF> +WEBSOCKETNEW:0<CR><LF> AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF> AT+WEBSOCKETSEND=0,0,10,0123456789<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+WEBSOCKETNEW=2,2,0
+WEBSOCKETNEW:0
AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo
OK
AT+WEBSOCKETSEND=0,0,10,0123456789
OK
```

8.3.8. SSL Receive Ready

Command Description:

To Receive data from any client device

Pre-requisites:

before issuing this command, user should bring SSL connection up.

Command Syntax:

AT+NSSLRR=<cid>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	specifies CID given by the AT+NSSLINIT command

Command Response

If Data there is data available , response is

+SRR:CID,RIP,RP,length

In case of failure returns failure and no response is given.

Response Name	Response Type	Response Range
cid	UINT	0-15
RIP	IP	0-1460
RP	UINT	0-65535

Response:

If Data receive is successful, it returns success or failure. The remote device

details are given in the response API. the response API contains the cid, Remote device IP,

Remote device port.

Responses	Remarks
receive the data through the created socket. at+nsslini=1,2<CR><LF> +NSSLINIT:0<CR><LF> <CR><LF>OK<CR><LF> at+nsslco=0,192.168.120.3,9500<CR><LF> <CR><LF>OK<CR><LF>	Successful

```
at+nsslrr=0<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
+nsslrr:0,192.168.120.4,8747<CR><LF>
```

ERROR

If parameter is not valid

Example:

receive the data through the created socket.

```
at+nsslinit=1,2
```

```
+NSSLINIT:0
```

```
OK
```

```
at+nsslco=0,192.168.120.3,9500
```

```
OK
```

```
at+nsslrr=0
```

```
OK
```

```
+nsslrr:0,192.168.120.4,8747
```

8.3.9. Read SSL

Command Description:

To read ssl data from any client device

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+NSSLRD=<cid>,<Length>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	CID is the value given by either +SC command in case of UDP or +SA command in case of TCP.
Length	M	Value Range: 1-1460 Format: UINT Default value: 1460	Length is the value to receive number of bytes.

Command Response

If Data recv is successful module returns +SR:RIP,RP,LENGTH,DATA

where DATA is of size LENGTH. In case of failure the return type returns failure and no response is given.

Response Name	Response Type	Response Range
RIP	IP	7-32
RP	UINT	0-65535
data length	INT	0-1460
data	BIN	0-1460

Response:

If Data receive is successful it returns success otherwise returns failure.

The data information details are given in the response API. the response API contains the cid,

number of bytes of data and the data it received.

Responses	Remarks
receive the data through the created ssl. <pre> AT+NSSLINIT=2,2<CR><LF> +NSSLINIT:0<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLCO=0,192.168.120.3,9500<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLRR=0<CR><LF> <CR><LF>OK<CR><LF> +NSSLRR:0,RIP,RP<CR><LF> AT+NSSLRD=0,length<CR><LF> +NSSLRD:192.168.120.10,8366,length,data of length bytes<CR><LF> <CR><LF>OK<CR><LF> </pre>	Successful
ERROR	If parameter is not

valid

Example:

receive the data through the created ssl.

```
AT+NSSLINIT=2,2
+NSSLINIT:0
OK
AT+NSSLCO=0,192.168.120.3,9500
OK
AT+NSSLRR=0
OK
+NSSLRR:0,RIP,RP
AT+NSSLRD=0,length
+NSSLRD:192.168.120.10,8366,length,data of length bytes
OK
```

8.3.10. Write SSL

Command Description:

This command takes care of sending data to the ssl connection.

Pre-requisites:

before issuing this command,ssl open must be executed for ssl connection.

Command Syntax:

```
AT+NSSLWR=<cid>,[<Dst>,<port>,<data length>,<Data>]
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	cid is the value given by the +NSSLINIT command.
Dst	O	Value Range: 7-40 Format: IP Default value: 0.0.0.0	Destination ip address to send data.
port	O	Value Range: 0-65535 Format: UINT Default value: 0	Destination port number to send data.

data length	O	Value Range: 1-1460 Format: INT Default value: Welcome to Telit	Data to send.
Data	O	Value Range: 1-1460 Format: BIN Default value: Welcome to Telit	Data to send.

Command Response

If Data send is successful module returns success

and no response is sent, if it fails sending, module returns failure and response sends as +NSSLWR:CID,length(length number of bytes sent).

Response Name	Response Type	Response Range
cid	UINT	
Length	UINT	

Response:

If Data send is successful, it returns number of bytes sent successfully, else returns FAILURE.

Responses	Remarks
send the data through the ssl connection. AT+NSSLINIT=1,2<CR><LF> +NSSLINIT:0<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLCFG=0,4,"1"<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLCO=0,192.168.120.3,9500<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLWRITE=0,192.168.120.3,8355,10,0123456789<CR><LF>	Successful

```
+NSSLWRITE=0,10<CR><LF>
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

send the data through the ssl connection.

```
AT+NSSLINIT=1,2
```

```
+NSSLINIT:0
```

```
OK
```

```
AT+NSSLCFG=0,4,"1"
```

```
OK
```

```
AT+NSSLCO=0,192.168.120.3,9500
```

```
OK
```

```
AT+NSSLWRITE=0,192.168.120.3,8355,10,0123456789
```

```
+NSSLWRITE=0,10
```

```
OK
```

8.3.11. Close SSL

Command Description:

To close ssl connection

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+NSSLCL=<cid>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-15 Format: UINT Default value: 0	specifies CID given by the AT+NSSLINIT command

Command Response

reponse handler if closed by peer, response is

```
+NSSLCL:CID,CLOSED
```

Response Name	Response Type	Response Range
cid	UINT	0-15
name	STR	

Response:

If open is successful, it returns SUCCESS, else reutrns FAILURE.

Responses	Remarks
AT+NSSLinit=1,2,<CR><LF> +NSSLINIT:0<CR><LF>OK<CR><LF> AT+NSSLCO=0,192.168.2.1,9600<CR><LF> <CR><LF>OK<CR><LF> AT+NSSLCL=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSSLinit=1,2,
+NSSLINIT:0OK
AT+NSSLCO=0,192.168.2.1,9600
OK
AT+NSSLCL=0
OK
```

8.3.12. SSL Certificate List

Command Description:

To list certificates/CAs list data from nonvolatile memory

Pre-requisites:

user can issue this command at any time independent of any other operations.

Command Syntax:

```
AT+NSSLCERTLIST=<CertType>
```

Parameter Description:

Parameters	Required?	Value	Description
CertType	M	Value Range: 0-3 Format: UINT Default value: 0	specifies the type of the certificate. \b 0 : For CA Certificates \b 1 : For Client/Server Certificates

Command Response

Response Name	Response Type	Response Range
type	STR	
name	STR	

Response:

If cert add is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NSSLCERTLIST=0/1/2<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSSLCERTLIST=0/1/2
OK
```

8.3.13. SSL Certificate Store

Command Description:

To store a certificate/CA list data in nonvolatile memory

Pre-requisites:

user can issue this command at any time independent of any other operations.

Command Syntax:

AT+NSSLCERTSTORE=<CertType>,<sequence>,<Name>,<data length>,<Data>

Parameter Description:

Parameters	Required?	Value	Description
CertType	M	Value Range: 0-3 Format: UINT Default value: 0	specifies the type of the certificate. \b 0 : For CA Certificates \b 1 : For Client/Server Certificates
sequence	M	Value Range: 1-10 Format: UINT Default value: 0	specifies the type of the certificate. \b 0 : For CA Certificates \b 1 : For Client/Server Certificates
Name	M	Value Range: 1-50 Format: STR Default value: 0	specifies the certificate name.
data length	M	Value Range: 1-64000 Format: INT Default value: 0	specifies the certificate data to be store.
Data	M	Value Range: 1-64000 Format: BIN Default value: 0	specifies the certificate data to be store.

Response:

If cert add is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NSSLCertadd=0,"certificate",certificate data<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSSLCertadd=0,"certificate",certificate data
```

```
OK
```

8.3.14. SSL Certificate Deletion

Command Description:

To delete a certificate/CA list data in nonvolatile memory

Pre-requisites:

user can issue this command at any time independent of any other operations.

Command Syntax:

```
AT+NSSLCERTDELETE=<CertType>,<Name>
```

Parameter Description:

Parameters	Required?	Value	Description
CertType	M	Value Range: 0-3 Format: UINT Default value: 0	specifies the type of the certificate. \b 0 : For CA Certificates \b 1 : For Client/Server Certificates
Name	M	Value Range: 1-50 Format: STR Default value: 0	specifies the certificate name.

Response:

If cert add is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NSSLCERTDEL=0,"certificate"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NSSLCERTDEL=0,"certificate"
```

```
OK
```

8.4. HTTPC (Client)

8.4.1. Initialize HTTP Client

Command Description:

To Initiate the HTTP client .

Pre-requisites:

Before issuing HTTP client init, user should have IP configuration, L2 and L3 connection must be up.

Command Syntax:

```
AT+NHTTPCINIT=[<TYPE>,<TIMEOUT>,<MAX_BODY_LEN>,<MAX_HEADER_LEN>]
```

Parameter Description:

Parameters	Required?	Value	Description
TYPE	O	Value Range: 0-2 Format: INT Default value: 0	HTTP type can be 0-HTTP 1-HTTPS 2-HTTP+HTTPs
TIMEOUT	O	Value Range: 0-4294967295 Format: UINT Default value: 0	connection timeout in seconds
MAX_BODY_LEN	O	Value Range: 0-65535 Format: INT Default value: 2048	maximum body length in bytes
MAX_HEADER_LEN	O	Value Range: 0-65535 Format: INT Default value: 2048	maximum header length in bytes

Command Response

If initialization is successful +HTTPCINIT:CID is sent else ERROR is sent.

Response Name	Response Type	Response Range
---------------	---------------	----------------

CID	INT	0-15
------------	------------	-------------

Response:

If initialization is successful, it returns SUCCESS alongwith CID, else returns FAILURE.

Once the allocation is done, registered API is called with the result.

Responses	Remarks
To Initialise an HTTP client . AT+HTTPCINIT=0, 100, 1048, 1048<CR><LF> +NSSLINIT:0<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To Initialise an HTTP client .

AT+HTTPCINIT=0, 100, 1048, 1048

+NSSLINIT:0

8.4.2. Configure HTTP Client

Command Description:

To configure the HTTP client connection parameters.

Pre-requisites:

This command should be called after AT+NHTTPCINIT for a specific CID.

Command Syntax:

AT+NHTTPCCFG=<CID>,<CONF_ID>,<CONF_VAL1>,[<CONF_VAL2>]

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-16 Format: INT Default value: 0	the connection id returned from AT+WNHTTPCINIT

CONF_ID	M	Value Range: 0-10 Format: INT Default value: 0	the configuration id as described in the list of available configurations \b M2M_S2W_HTTPC_CONF_INVLAID = 0, M2M_S2W_HTTPC_CONF_HDR = 1, M2M_S2W_HTTPC_CONF_CGI = 2, M2M_S2W_HTTPC_CONF_RQST_BODY = 3, M2M_S2W_HTTPC_CONF_RSP_HDR = 4,
CONF_VAL1	M	Value Range: 0-65535 Format: STR Default value: ""	First string value specific to the configuration provided in CONF_ID
CONF_VAL2	O	Value Range: 0-65535 Format: STR Default value: ""	Second string value specific to the configuration provided in CONF_ID

Response:

If httpc client config is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NHTTPCINIT=0, 60, 1024, 1024 <CR><LF> +NTTPPCINIT:0<CR><LF> AT+HTTPCCFG=0,4,"1"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NHTTPCINIT=0, 60, 1024, 1024
+NTTPPCINIT:0
```

```
AT+HTTPCCFG=0,4,"1"
```

```
OK
```

8.4.3. HTTP Client Connect

Command Description:

To make http client connection to the given URL and Port

Pre-requisites:

before issuing this command, user should bring L2 and L3 layer connection UP

Command Syntax:

```
AT+NHTTCCO=<CID>,<URL>,<PORT>
```

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-16 Format: INT Default value: 0	the connection id returned from AT+NHTTCCINIT
URL	M	Value Range: 1-64 Format: STR Default value: 0	Server URL or Server IP
PORT	M	Value Range: 1-65535 Format: INT Default value:	server port number

Response:

If connect is successful, it returns OK, else return ERROR.

Responses	Remarks
<pre>AT+NHTTCCINIT=0, 60, 1024, 1024 <CR><LF> +NHTTCCINIT:0<CR><LF> AT+NHTTCCO=0,"http://192.168.2.223",80<CR><LF> <CR><LF>OK<CR><LF></pre>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NHTTPCINIT=0, 60, 1024, 1024
+NHTTPCINIT:0
AT+NHTTPCCO=0,"http://192.168.2.223",80
OK
```

8.4.4. HTTP Client Request

Command Description:

To Send the HTTP client request to the server.

Pre-requisites:

This command should be called after http init and connection is made

Command Syntax:

```
AT+NHTTPCREQ=<CID>,<METHOD>,<FILE_PATH>,[<data length>,<Data>]
```

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: INT Default value: 0	the connection id returned from AT+NHTTPCINIT
METHOD	M	Value Range: 1-4 Format: INT Default value: 0	HTTP method used as part of HTTP request GET = 1, \b POST = 2, \b PUT = 3, \b PATCH = 4 \b
FILE_PATH	M	Value Range: 1-32 Format: STR Default value:	URI path to be used while sending HTTP request
data length	O	Value Range: 1-1460 Format: INT Default value: ""	Data for the given request

Data	O	Value Range: 1-1460 Format: BIN Default value: ""	Data for the given request
-------------	---	--	----------------------------

Command Response

If Data rcv is successful module returns +NHTTPCREQ:STATUS,LENGTH

In case of failure the returns ERROR and no response is given.

Response Name	Response Type	Response Range
STATUS	UINT	0-1024
LENGTH	INT	0-65535

Response:

If http client request is successful it returns response API containing response status and length otherwise returns failure.

Responses	Remarks
AT+NHTTPCINIT=0, 60, 1024, 1024 <CR><LF> +NTTPPCINIT:0<CR><LF> AT+NHTTPCCO=0,"http://192.168.2.223",80<CR><LF> <CR><LF>OK<CR><LF> AT+NHTTPCREQ=0,1,"http/1kb.html"<CR><LF> +NHTTPCREQ=200,564<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NHTTPCINIT=0, 60, 1024, 1024
+NTTPPCINIT:0
AT+NHTTPCCO=0,"http://192.168.2.223",80
OK
```

AT+NHTTPCREQ=0,1,"http/1kb.html"

+NHTTPCREQ=200,564

8.4.5. Read HTTP Client

Command Description:

To Read the response received from server.

Pre-requisites:

This command should be called after httpc request is succesful.

Command Syntax:

AT+NHTTPCRDRSP=<CID>,<LENGTH>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-16 Format: INT Default value: 0	the connection id returned from AT+WNHTTPCINIT
LENGTH	M	Value Range: 1-65535 Format: INT Default value: 0	Length of the Data to Read

Command Response

If Data recv is successful module returns +NHTTPCRDRSP:CID,LENGTH,DATA

where DATA is of size LENGTH. In case of failure returns ERROR and no response is given.

Response Name	Response Type	Response Range
CID	INT	0-16
data length	INT	0-2048
DATA	BIN	0-2048

Response:

If Data read is successful it returns response API otherwise returns failure.

Responses	Remarks
AT+NHTTPCINIT=0, 60, 1024, 1024 <CR><LF> +NHTTPCINIT:0<CR><LF> AT+NHTTPCCO=0,"http://192.168.2.223",80<CR><LF> <CR><LF>OK<CR><LF> AT+NHTTPCREQ=0,1,"http/1kb.html"<CR><LF> +NHTTPCREQ=200,564<CR><LF> AT+NHTTPCRDRSP=0, 1024<CR><LF> +NHTTPCRDRSP:0, 10, HelloWorld<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```

AT+NHTTPCINIT=0, 60, 1024, 1024
+NHTTPCINIT:0
AT+NHTTPCCO=0,"http://192.168.2.223",80
OK
AT+NHTTPCREQ=0,1,"http/1kb.html"
+NHTTPCREQ=200,564
AT+NHTTPCRDRSP=0, 1024
+NHTTPCRDRSP:0, 10, HelloWorld

```

8.4.6. Close HTTP Client

Command Description:

To Close the HTTP client connection.

Pre-requisites:

before issuing this command, user should have httpc client connection.

Command Syntax:

```
AT+NHTTPCCL=<CID>
```

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-16 Format: INT	the connection id returned from AT+WNHTTPCINIT

Default value: 0

Command Response

reponse handler if closed by peer, response is

+NHTTPCCL:CID,CLOSED

Response Name	Response Type	Response Range
CID	UINT	0-15
ERRORCODE	STR	

Response:

If api is successful, it returns SUCCESS, else reutrns FAILURE.

Responses	Remarks
AT+NHTTPCINIT=0, 60, 1024, 1024 <CR><LF> +NTTPPCINIT:0<CR><LF> AT+NHTTPCCO=0,"http://192.168.2.223",80<CR><LF> <CR><LF>OK<CR><LF> AT+NHTTPCREQ=0,1,"http/1kb.html"<CR><LF> +NHTTPCREQ=200,564<CR><LF> AT+NHTTPCRDRSP=0, 1024<CR><LF> +NHTTPCRDRSP:length,data of length bytes<CR><LF> AT+NHTTPCCL=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+NHTTPCINIT=0, 60, 1024, 1024

+NTTPPCINIT:0

AT+NHTTPCCO=0,"http://192.168.2.223",80

OK

AT+NHTTPCREQ=0,1,"http/1kb.html"

+NHTTPCREQ=200,564

AT+NHTTPCRDRSP=0, 1024

+NHTTPCRDRSP:length,data of length bytes

AT+NHTTPCCL=0

OK

8.5. HTTPD (Server)

8.5.1. Initialize HTTP Server

Command Description:

To Initiaise the HTTP Server .

Pre-requisites:

Before issuing HTTP Server init, user should have IP configuration, L2 and L3 connection must be up.

Command Syntax:

AT+NHTTPDINIT=<SERVER_NAME>,<MODE>,<NWK_IF>,<Family>

Parameter Description:

Parameters	Required?	Value	Description
SERVER_NAME	M	Value Range: 0-32 Format: STR Default value: ""	Server Name for http server
MODE	M	Value Range: 0-3 Format: UINT Default value: 0	MODE can be 0 - HTTP, 1- HTTPS or 2- both
NWK_IF	M	Value Range: 0-2 Format: INT Default value: 2	Network interface to bind, \b e.g., 0 for wlan0, 1 for wlan1, 2 for both \b If not specified, the server will bind to every network interface.
Family	M	Value Range: 2-4 Format: INT Default value:	specifies http server to run in IPV4 or IPV6 of both. \b 2 - IPv4, 3 - IPv6, 4 = IPv4 and IPv6

0

Command Response

If initialization is successful +HTTPCINIT:CID is sent else ERROR is sent.

Response Name	Response Type	Response Range
CID	INT	0-15

Response:

If initialization is successful, it returns SUCCESS alongwith CID, else returns FAILURE.

Once the allocation is done, registered API is called with the result.

Responses	Remarks
To Initialise an HTTP server . AT+HTTPDINIT=testServer, 0, 0, 2<CR><LF> +NHTTPDINIT:0<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To Initialise an HTTP server .

AT+HTTPDINIT=testServer, 0, 0, 2

+NHTTPDINIT:0

8.5.2. Configure HTTP Server**Command Description:**

To configure the HTTP server parameters.

Pre-requisites:

This command should be called after AT+NHTTPDINIT with the cid.

Command Syntax:

AT+NHTTDCFG=<CID>,<CONF_ID>,<CONF_VAL1>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: INT Default value: 0	the connection id returned from AT+WNHTTPCINIT
CONF_ID	M	Value Range: 0-10 Format: INT Default value: 0	the configuration id as described in the list of available configurations \b Invalid Value = 0, Port for HTTP = 1, Port for HTTPS = 2, Server Certificate = 3, Receive BufferSize = 4, Transmit BufferSize = 5
CONF_VAL1	M	Value Range: 0-65535 Format: STR Default value: ""	String value specific to the configuration provided in CONF_ID

Response:

If httpd config is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+HTTPDINIT=testServer, 0, 0, 2<CR><LF> +NHTTPDINIT:0<CR><LF> AT+HTTPDCFG=0,1,"10234"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+HTTPDINIT=testServer, 0, 0, 2

+NHTTPDINIT:0

AT+HTTPDCFG=0,1,"10234"

OK

8.5.3. Start/Stop/Close HTTP Server connection

Command Description:

To start/stop/close HTTP server connection.

Pre-requisites:

before issuing this command, user should have http server init.

Command Syntax:

AT+NHTTPDCMD=<CID>,<ACTION>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-16 Format: INT Default value: 0	the connection id returned from AT+WNHTTPDINIT
ACTION	M	Value Range: 0-2 Format: INT Default value: 0	0 - start server, 1 - stop server , 2 - close server

Command Response

Reponse if closed by peer : +NHTTPDCMD:CID,CLOSED

Reponse if Request received from client +NHTTPDCMD:CID,URI_ID,REQUEST,<requestline length> , < length>

Response Name	Response Type	Response Range
CID	UINT	0-15
URI_ID	UINT	0-255
RSP_TYPE	GET,POST_FORMDATA,POST_RAWDATA,CLOSED	0-3

REQUESTLINE_LENGTH	INT	0-16635
BODY_LENGTH	INT	0-16635

Response:

If api is successful, it returns SUCCESS, else reutrns FAILURE.

Responses	Remarks
To Initialise an HTTP server . AT+HTTPDINIT=testServer, 0, 0, 2<CR><LF> +NHTTPDINIT:0<CR><LF> AT+NHTTPDCMD=0,2<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To Initialise an HTTP server .

AT+HTTPDINIT=testServer, 0, 0, 2

+NHTTPDINIT:0

AT+NHTTPDCMD=0,2

OK

8.5.4. Configure HTTP Server URI

Command Description:

To configure the URI for http server.

Pre-requisites:

This command should be called after AT+NHTTPDINIT for a specific CID.

Command Syntax:

AT+NHTTPDCFGURI=<CID>,<URI>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-	the connection id returned from

		16	AT+WNHTTPCINIT
		Format: INT	
		Default value: 0	
URI	M	Value Range: 0-65535	URI value specific
		Format: STR	
		Default value: ""	

Response:

If uri config is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+NHTTPCINIT=0, 60, 1024, 1024 <CR><LF> +NTTTPPCINIT:0<CR><LF> AT+HTTTPCCFG=0,4,"1"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NHTTPCINIT=0, 60, 1024, 1024
+NTTTPPCINIT:0
AT+HTTTPCCFG=0,4,"1"
OK
```

8.5.5. Read HTTP Server

Command Description:

To Read the request received from client.

Pre-requisites:

This command should be called after clinet request is received.

Command Syntax:

```
AT+NHTTTPDRD=<CID>,<URI_ID>,<RD_TYPE>,<LENGTH>
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

CID	M	Value Range: 0-16 Format: INT Default value: 0	the connection id returned from AT+WNHTTPCINIT
URI_ID	M	Value Range: 0-64 Format: INT Default value: 0	the uri id returned in AT+NHTTTPDCMDREP.
RD_TYPE	M	Value Range: 0-10 Format: UINT Default value: 0	type of data to Read 0 : for RequestLine 1 : for Body
LENGTH	M	Value Range: 1-65535 Format: INT Default value: 0	Length of the Data to Read

Command Response

If Data recv is successful module returns +NHTTTPDRD:LENGTH,DATA

where DATA is of size LENGTH. In case of failure returns ERROR and no response is given.

Response Name	Response Type	Response Range
LENGTH	INT	0-65535
data length	INT	0-5000
DATA	BIN	0-5000

Response:

If Data read is successful it returns response API otherwise returns failure.

Responses	Remarks
-----------	---------

AT+NHTTPDINIT="testServer",0,0,2 <CR><LF> +NTPPDINIT:0<CR><LF> AT+NHTTPDCMD=0,0<CR><LF> <CR><LF>OK<CR><LF> +NHTTPCCMD=0,REQUEST,565<CR><LF> AT+NHTTPDRD=0, 565<CR><LF> +NHTTPCRD:length,data of length bytes<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NHTTPDINIT="testServer",0,0,2
+NTPPDINIT:0
AT+NHTTPDCMD=0,0
OK
+NHTTPCCMD=0,REQUEST,565
AT+NHTTPDRD=0, 565
+NHTTPCRD:length,data of length bytes
```

8.5.6. Send Data to HTTP Server

Command Description:

This command takes care of sending data for http request after headers sent.

Pre-requisites:

before issuing this command,httpd server must be received the request and send httpsendHdr.

Command Syntax:

```
AT+NHTTPDSENDDATA=<CID>,<URI_ID>,[<data length>,<Data>]
```

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: 0	cid is the value given by the AT+NHTTPDINIT command.
URI_ID	M	Value Range: 0-15	URI ID to send the data.

		Format: UINT	
		Default value: 0	
data length	O	Value Range: 0-5000	Data to send.
		Format: INT	
		Default value: ""	
Data	O	Value Range: 0-5000	Data to send.
		Format: BIN	
		Default value: ""	

Command Response

If Data send is successful module returns success

and no response is sent, if it fails sending, module returns failure and response sends as +NHTTPDSENDDATA:CID,length(length number of bytes sent).

Response Name	Response Type	Response Range
cid	UINT	
Length	UINT	

Response:

If Data send is successful, it returns number of bytes sent successfully, else returns FAILURE.

Responses	Remarks
send the data through the ssl server connection.	Successful
ERROR	If parameter is not valid

Example:

send the data through the ssl server connection.

8.5.7. Send Response to HTTP Server Request

Command Description:

To Send the HTTP server response to given request.

Pre-requisites:

This command should be used when the client has made an HTTP request

Command Syntax:

```
AT+NHTTPDSENDHDR=<CID>,<URI_ID>,<STATUS_CODE>,[<STATUS_TEXT>],
<CONTENT_LENGTH>,[<CONTENT_TYPE>,<USER_HEADERS>]
```

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: INT Default value: 0	the connection id returned from AT+NHTTPDINIT
URI_ID	M	Value Range: 0-64 Format: INT Default value: 0	the uri id to be used
STATUS_CODE	M	Value Range: 0-1000 Format: INT Default value: 0	HTTP status code e.g 200 for OK, 401 for Bad request
STATUS_TEXT	O	Value Range: 1-32 Format: STR Default value: ""	Optional text associated with the status_code e.g OK or Bad Request
CONTENT_LENGTH	M	Value Range: 0-1000 Format: INT Default value: 0	Size in bytes of the message body

CONTENT_TYPE	O	Value Range: 1-32 Format: STR Default value: ""	Content Type of Msg Body e.g text/html, if NULL, the content type header will not be sent
USER_HEADERS	O	Value Range: 1-255 Format: STR Default value: ""	optional headers to sent if any

Response:

If http server send the headers when request is made from client. if send is successful it returns success otherwise returns failure.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:**8.6. MQTT****8.6.1. MQTT Client Connect****Command Description:**

To make a MQTT connection with MQTT broker

Pre-requisites:

L2 - L3 connections should have been done

Command Syntax:

```
AT+MQTTCONNECT=<HOST>,[<PORT>],<client_id>,[<username>,<password>,<keepalive_in_sec>,<timeout_in_sec>,<use_ssl>,<ca_file>]
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

HOST	M	Value Range: 0-256 Format: STR Default value: ""	Fully Qualified Domain Name of the Server (MQTT Broker) or IP address of the server in String format
PORT	O	Value Range: 1024-65536 Format: INT Default value: 1883	Port number of server
client_id	M	Value Range: 0-256 Format: STR Default value:	unique Client Identifier. It is the user defined string of ASCII characters. Example: QC4020_001122
username	O	Value Range: 0-256 Format: STR Default value:	Username if required by the broker
password	O	Value Range: 0-64 Format: STR Default value: ""	Password, if required by the broker. Valid only if the Username is provided
keepalive_in_sec	O	Value Range: 0-180 Format: UINT Default value: 60	MQTT Keep alive time out that will be conveyed to the broker
timeout_in_sec	O	Value Range: 1-	maximum time limit for connecting to the

		180	server
		Format: UINT	
		Default value: 75	
use_ssl	O	Value Range: 0-1 Format: UINT Default value: 0	Use ssl for secured connection
ca_file	O	Value Range: 0-256 Format: STR Default value: ""	CA certificate file path if ssl connection needs to be done

Command Response

If connection is successful +MCONNECT:connected is sent else

+MCONNECT:not connected is sent

Response Name	Response Type	Response Range
cid	UINT	0-15

Response:

If connection processing is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
Connect to MQTT broker. Here the broker address is api-dev.devicewise.com and the apptoken is F1wS6Rhvuxi432C4 and In Response, will get CID here it is zero. AT+NMQTTCONNECT="api-dev.devicewise.com",1883,"123456789","gainspan","F1wS6Rhvuxi432C4",180,20,0<CR><LF>	Successful

+NMQTTCONNECT:0<CR><LF> <CR><LF>OK<CR><LF>	
ERROR	If paramete r is not valid

Example:

Connect to MQTT broker. Here the broker address is api-dev.devicewise.com and the apptoken is

F1wS6Rhvuxi432C4 and In Response, will get CID here it is zero.

```
AT+NMQTTCONNECT="api-
dev.devicewise.com",1883,"123456789","gainspan","F1wS6Rhvuxi432C4",180,20,
0
+NMQTTCONNECT:0
OK
```

8.6.2. MQTT Client Publish

Command Description:

To PUBLISH MQTT message

Pre-requisites:

L2 - L3 connections should be done and AT+MCONNECT command should be issued

Command Syntax:

```
AT+NMQTTPUBLISH=<CID>,<QOS>,<retainFlag>,<message_id>,<topic>,<data_tlen>,<data>,<data_rlen>
```

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: ""	Connection ID returned at the time of connect command
QOS	M	Value Range: 0-2 Format:	It specifies level of assurance for delivery of application message

		INT Default value: ""	
retainFlag	M	Value Range: 0-1 Format: UINT Default value: 0	If it is set to 1, the Server MUST store the ApplicationMessage and its QoS, so that it can be delivered to future subscribers whose subscriptions match its topic name. If it is set to 0, Server MUST NOT store message and MUST NOT remove or replace any existing retained message
message_id	M	Value Range: 0-65536 Format: UINT Default value: ""	Message ID to be sent along with message being Published
topic	M	Value Range: 0-256 Format: STR Default value: ""	It identifies endpoint to which payload data is published
data_len	M	Value Range: 0-1460 Format: UINT Default value:	Length of data to be published
data	M	Value Range: 1-1460 Format: STR Default value: ""	Data of length as specified in "data_len" parameter.

data_rlen	M	Value Range: 1-1460 Format: UINT Default value:	Length of data to receive
------------------	---	--	---------------------------

Command Response

Response Name	Response Type
---------------	---------------

Response:

If PUBLISH is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
Publish data from MQTT client. Here the CIS is zero,topic is thing/gainspan/property/light,QOS is 1 and data is 123. AT+NMQTTPUBLISH=0,1,0,1234,"thing/gainspan/property/light",3,123<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

Publish data from MQTT client.Here the CIS is zero,topic is thing/gainspan/property/light,QOS is 1 and data is 123.

```
AT+NMQTTPUBLISH=0,1,0,1234,"thing/gainspan/property/light",3,123
OK
```

8.6.3. Receive Request from MQTT Client

Command Description:

To send Receive Request

Pre-requisites:

L2 - L3 connections should be done and AT+MCONNECT command should be issued

Command Syntax:

AT+NMQTTRR=<CID>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: ""	Connection ID returned at the time of connect command

Command Response

whenever a message is received for a subscribed topic,
message will be delivered

Response Name	Response Type	Response Range
CID	UINT	0-15
data_len	UINT	0-1460

Response:

Whenever there is data on the connection, indication in the format

"MQTTRR:<CID>,<data_length>" will be sent to the host.

Responses	Remarks
Enable the Receive request flag.Here the CID is zero. AT+NMQTTRR=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

Enable the Receive request flag.Here the CID is zero.

AT+NMQTTRR=0

OK

8.6.4. MQTT Client Subscribe

Command Description:

To Subscribe to MQTT Topic

Pre-requisites:

L2 - L3 connections should be done and AT+MCONNECT command should be issued

Command Syntax:

AT+NMQTTSUBSCRIBE=<CID>,<QOS>,<message_id>,<topic>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: ""	Connection ID returned at the time of connect command
QOS	M	Value Range: 0-2 Format: INT Default value: ""	It specifies level of assurance for delivery of application message
message_id	M	Value Range: 0-65536 Format: UINT Default value: ""	Message ID to be sent along with message being Published
topic	M	Value Range: 0-256 Format: STR Default value: ""	It identifies endpoint to which payload data is published

Command Response

Response Name	Response Type
---------------	---------------

Response:

If SUBSCRIBE is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
Subscribe topic to receive data on that .Here the CID is zero,topic is thing/gainspan/property/light and QOS is 1. AT+NMQTTSUBSCRIBE=0,1,675,"thing/gainspan/property/light"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

Subscribe topic to receive data on that .Here the CID is zero,topic is thing/gainspan/property/light and QOS is 1.

AT+NMQTTSUBSCRIBE=0,1,675,"thing/gainspan/property/light"

OK

8.6.5. MQTT Client Receive

Command Description:

To Receive data

Pre-requisites:

L2 - L3 connections should be done and AT+MCONNECT, AT+NMQTTRR commands should be issued

Command Syntax:

AT+NMQTTR=<CID>,<Data_Length>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: ""	Connection ID returned at the time of connect command
Data_Length	M	Value Range: 1-1460 Format: UINT Default value: ""	Data length to receive.

Command Response

whenever a message is received for a subscribed topic,
message will be delivered

Response Name	Response Type	Response Range
CID	UINT	0-15
data length	INT	
Data	BIN	

Response:

Data of specified length will be sent over the interface to host.

Responses	Remarks
Receive data from the server.Here the CID is zero and data length is three. AT+NMQTTR=0,3<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

Receive data from the server.Here the CID is zero and data length is three.

AT+NMQTTR=0,3

OK

8.6.6. MQTT Client Disconnect

Command Description:

To Disconnect a connection

Pre-requisites:

L2 - L3 connections should be done and AT+MCONNECT command should be issued

Command Syntax:

AT+NMQTTDISCONNECT=<CID>

Parameter Description:

Parameters	Required?	Value	Description
CID	M	Value Range: 0-15 Format: UINT Default value: ""	Connection ID returned at the time of connect command

Command Response

If disconnection is successful +NMQTTDISCONNECT:cid is sent.

Response Name	Response Type	Response Range
cid	UINT	0-15

Response:

If DISCONNECT is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
Disconnect from MQTT broker.Need to give CID as a param.Here it is zero. AT+NMQTTDISCONNECT=0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

Disconnect from MQTT broker.Need to give CID as a param.Here it is zero.

AT+NMQTTDISCONNECT=0

OK

8.7. [Web Socket](#)**8.7.1. Configure WebSocket****Command Description:**

To configure a websocket

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+WEBSOCKCONF=<cid>,<paramnumber>,[<param1>,<param2>]

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by AT+WEBSOCKNEW command
paramnumber	M	Value Range: 0-8 Format: UINT Default value:	config number 0 for adding http headers 1 for clearing http headers 2 for enable echo mode 3 for disable echo mode
param1	O	Value Range: 1-32 Format: STR Default value:	parameter
param2	O	Value Range: 1-32 Format: STR Default value:	config number

Response:

If config is successful, API returns SUCCESS else returns FAILURE.

Responses	Remarks
AT+WEBSOCKNEW=2,2,0<CR><LF> +WEBSOCKNEW:0<CR><LF> AT+WEBSOCKCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF>	Successful

```
AT+WEBSOCKCONF=0,1<CR><LF>
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

```
AT+WEBSOCKNEW=2,2,0
+WEBSOCKNEW:0
AT+WEBSOCKCONN=0,192.168.16.32,80,/echo
OK
AT+WEBSOCKCONF=0,1
OK
```

8.7.2. Create WebSocket

Command Description:

To create new websocket and add CID entry

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+WEBSOCKNEW=[<origin>,<max_Recv_Chunk_Size>,<handshake_Timeout_Milliseconds>,<ssl_enabled>,<closing_Timeout_Milliseconds>,<subprotocol1>,<subprotocol2>,<subprotocol3>,<subprotocol4>]
```

Parameter Description:

Parameters	Required?	Value	Description
origin	O	Value Range: 0-64 Format: STR Default value:	Client origin.
max_Recv_Chunk_Size	O	Value Range: 1-1460 Format: UINT Default value:	Messages longer than this value will be passed to recv callback in chunks of max_Recv_Chunk_Size.

1024			
handshake_Timeout_Milliseconds	O	Value Range: 1-1000000 Format: UINT Default value: 60000	Handshake timeout in milliseconds.
ssl_enabled	O	Value Range: 0-1 Format: UINT Default value: 0	ssl enabling flag, 0 for disable,1 for enable
closing_Timeout_Milliseconds	O	Value Range: 0-10000 Format: UINT Default value: 0	Closing timeout in milliseconds. If the peer does not respond after we have initiated a close within the timeout, the WebSocket is forcibly shutdown.
subprotocol1	O	Value Range: 0-128 Format: STR Default value:	subprotocol1.
subprotocol2	O	Value Range: 0-128 Format: STR Default value:	subprotocol2.
subprotocol3	O	Value	subprotocol3.

		Range: 0-128 Format: STR Default value:
subprotocol4	O	Value subprotocol4. Range: 0-128 Format: STR Default value:

Command Response

If web socket creation is successful, module sends

'+WEBSOCKNEWC:cid' and 'OK' will be sent, else 'ERROR' will be sent.

Response Name	Response Type	Response Range
cid	UINT	

Response:

If web socket create is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+WEBSOCKNEW=2,2,0<CR><LF> +WEBSOCKNEW:0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+WEBSOCKNEW=2,2,0
+WEBSOCKNEW:0
OK
```

8.7.3. WebSocket Connect

Command Description:

To connect to a websocket server

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+WEBSOCKCONN=<cid>,<ServerIp>,<Server_port>,<Resource_path>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by AT+WEBSOCKNEW command
ServerIp	M	Value Range: Format: IP Default value:	server IP
Server_port	M	Value Range: 0-65535 Format: UINT Default value:	server port
Resource_path	M	Value Range: 0-128 Format: STR Default value:	Resource_path.

Response:

If web socket connect is successful, API returns SUCCESS, else returns FAILURE.

Responses	Remarks
AT+WEBSOCKNEW="192.168.16.132",<CR><LF> +WEBSOCKNEW:0<CR><LF> AT+WEBSOCKCONN=0,192.168.16.32,80,"/" <CR><LF>	Successful

```
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

```
AT+WEBSOCKETNEW="192.168.16.132",
+WEBSOCKETNEW:0
AT+WEBSOCKETCONN=0,192.168.16.32,80,"/"
OK
```

8.7.4. Send WebSocket Data

Command Description:

To send data to websocket remote device

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+WEBSOCKETSEND=<cid>,<datatype>,<data length>,<Data>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid given by AT+WEBSOCKETNEW command
datatype	M	Value Range: 0-1 Format: UINT Default value:	data type, 1 for text data, 2 for binary data
data length	M	Value Range: 1-4294967295 Format: INT Default value:	Data.
Data	M	Value Range: 1-4294967295 Format: BIN Default value:	Data.

Command Response

If Data send is successful module sends 'OK' and no response is sent,
otherwise module sends response as '+WEBSOCKETSEND:CID,length(total number of bytes sent till then)'

and sends 'ERROR'

this is synchronous response.

Response Name	Response Type	Response Range
cid	UINT	
Length	UINT	

Response:

If data send is successful, it returns number of bytes sent successfully, else returns failure.

Responses	Remarks
AT+WEBSOCKETNEW=2,2,0<CR><LF> +WEBSOCKETNEW:0<CR><LF> AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF> AT+WEBSOCKETSEND=0,0,10,0123456789<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+WEBSOCKETNEW=2,2,0
+WEBSOCKETNEW:0
AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo
OK
AT+WEBSOCKETSEND=0,0,10,0123456789
OK
```

8.7.5. WebSocket Ping

Command Description:

To Send a websocket ping

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+WEBSOCKPING=<cid>,<pingdata>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid given by AT+WEBSOCKNEW command
pingdata	M	Value Range: 0-64 Format: STR Default value:	ping payload

Command Response

If ping send is successful module sends 'OK' and pong response is sent asynchronously as '+WEBSOCKPONG:CID,ponglen,pongpayload'
otherwise module sends 'ERROR' and no response is sent.

Response Name	Response Type
---------------	---------------

Response:

If ping send is successful, API returns SUCCESS else returns FAILURE.

Responses	Remarks
AT+WEBSOCKNEW=2,2,0<CR><LF> +WEBSOCKNEW:0<CR><LF> AT+WEBSOCKCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF> AT+WEBSOCKPING=0,"pingpayload"<CR><LF> <CR><LF>OK<CR><LF> +WEBSOCKPONG:0,10,"pongpayload"	Successful

ERROR

If parameter is not valid

Example:

```
AT+WEBSOCKETNEW=2,2,0
+WEBSOCKETNEW:0
AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo
OK
AT+WEBSOCKETPING=0,"pingpayload"
OK
+WEBSOCKETPONG:0,10,"pongpayload"
```

8.7.6. WebSocket Pong

Command Description:

To Send a websocket pong

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+WEBSOCKETPONG=<cid>,<pongpayload>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid given by AT+WEBSOCKETNEW command
pongpayload	M	Value Range: 0-64 Format: STR Default value:	pong payload

Response:

If pong send is successful, API returns SUCCESS else returns FAILURE.

Responses**Remarks**

AT+WEBSOCKNEW=2,2,0<CR><LF> +WEBSOCKNEW:0<CR><LF> AT+WEBSOCKCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF> AT+WEBSOCKPONG=0,"pongpayload"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```

AT+WEBSOCKNEW=2,2,0
+WEBSOCKNEW:0
AT+WEBSOCKCONN=0,192.168.16.32,80,/echo
OK
AT+WEBSOCKPONG=0,"pongpayload"
OK

```

8.7.7. Receive Ready WebSocket Data

Command Description:

This command takes care of enabling data reception on a given CID at application level and indicates to the user when data is available on a particular CID.

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

```
AT+WEBSOCKRECVREADY=<cid>,<RxBufSize>
```

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid given by AT+WEBSOCKNEW command
RxBufSize	M	Value Range: 1-16384 Format: UINT Default value:	receiver buffer size to hold the received data

Command Response

If enabling data reception is successful module sends 'OK' and sends response asynchronously, In case of failure, module sends 'ERROR' and no response is given.

in case of success module sends response asynchronously as
'+WEBSOCKETRECVREADY:CID,RIP,RP,length'

Response Name	Response Type	Response Range
cid	UINT	
RIP	IP	
RP	UINT	
Length	UINT	

Response:

If data send is successful, it returns number of bytes sent successfully, else returns failure.

Responses	Remarks
AT+WEBSOCKETNEW=2,2,0<CR><LF> +WEBSOCKETNEW:0<CR><LF> AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF> AT+WEBSOCKETRECVREADY=0<CR><LF> <CR><LF>OK<CR><LF> +WEBSOCKETRECVREADY:0,192.168.16.129,8082,100<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+WEBSOCKETNEW=2,2,0
+WEBSOCKETNEW:0
AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo
OK
```

AT+WEBSOCKETRCVREADY=0

OK

+WEBSOCKETRCVREADY:0,192.168.16.129,8082,100

8.7.8. Receive WebSocket Data

Command Description:

To receives data from any CID

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+WEBSOCKETRCV=<cid>,<Length>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	CID value given by AT+WEBSOCKETNEW command
Length	M	Value Range: 1-4294967295 Format: UINT Default value:	number of bytes to receive, indicated in at+WEBSOCKETRCVREADY command response

Command Response

If data reception is successful module sends

response as '+WEBSOCKETRCV:RIP,RP,TOTAL LENGTH,PRESNT LENGTH,DATA' and 'OK'.

if the data reception is failed or partially received module sends response as

'+WEBSOCKETRCV:RIP,RP,TOTAL LENGTH,PRESNT LENGTH,DATA' and 'ERROR'

here TOTAL LENGTH is requested data length to receive,

PRESNT LENGTH is received data length in current response,

DATA is the received data of size PRESNT LENGTH,

this is synchronous response,

it may have multiple responses in case of bulk data reception.

Response Name	Response Type	Response Range
---------------	---------------	----------------

RIP	IP	
RP	UINT	0-65535
data length	INT	0-1460
data	BIN	0-1460

Response:

If Data receive is successful it returns success otherwise returns failure.

The data information details are given in the response API. the response API contains the cid,

number of bytes of data and the data it received.

Responses	Remarks
receive the data through the created websocket. AT+WEBSOCKNEW=2,2,0<CR><LF> +WEBSOCKNEW:0<CR><LF> AT+WEBSOCKCONN=0,192.168.16.32,80,/echo <CR><LF>OK<CR><LF> AT+WEBSOCKRCVREADY=0<CR><LF> <CR><LF>OK<CR><LF> +WEBSOCKRCVREADY:0,192.168.16.129,8082,10<CR><LF> AT+WEBSOCKRCV=0,10<CR><LF> +WEBSOCKRCVREADY:192.168.16.129,8082,0123456789<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

receive the data through the created websocket.

AT+WEBSOCKNEW=2,2,0

+WEBSOCKNEW:0

AT+WEBSOCKCONN=0,192.168.16.32,80,/echo

OK

AT+WEBSOCKRCVREADY=0

OK

+WEBSOCKETRCVREADY:0,192.168.16.129,8082,10

AT+WEBSOCKETRCV=0,10

+WEBSOCKETRCVREADY:192.168.16.129,8082,0123456789

8.7.9. Close WebSocket

Command Description:

To close websocket

Pre-requisites:

L2 - L3 connections should be established

Command Syntax:

AT+WEBSOCKETCLOSE=<cid>

Parameter Description:

Parameters	Required?	Value	Description
cid	M	Value Range: 0-11 Format: UINT Default value:	cid provided by AT+WEBSOCKETNEW command

Command Response

If websocket close is successful module sends 'OK',

otherwise module sends 'ERROR' and no response is sent.

for asynchronous websocket close events response is '+WEBSOCKETCLOSE:cid'

Response Name	Response Type	Response Range
cid	UINT	

Response:

If websocket close is successful, API returns SUCCESS else returns FAILURE.

Responses	Remarks
AT+WEBSOCKETNEW=2,2,0<CR><LF> +WEBSOCKETNEW:0<CR><LF> AT+WEBSOCKETCONN=0,192.168.16.32,80,/echo	Successful

```
<CR><LF>OK<CR><LF>
```

```
AT+WEBSOCKCLOSE=0<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

ERROR

If parameter is not valid

Example:

```
AT+WEBSOCKNEW=2,2,0
```

```
+WEBSOCKNEW:0
```

```
AT+WEBSOCKCONN=0,192.168.16.32,80,/echo
```

```
OK
```

```
AT+WEBSOCKCLOSE=0
```

```
OK
```

9. ADVANCED SERVICES

9.1. Network

9.1.1. MDNS

9.1.1.1. MDNS Start

Command Description:

To start the mDNS module

Pre-requisites:

L2 - L3 connections should be done

Response:

If starting mDNS module is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result.

Responses	Remarks
AT+NMDNSSTART<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Command Syntax:

AT+NMDNSSTART

Example:

AT+NMDNSSTART

OK

9.1.1.2. MDNS Host Name Registration

Command Description:

To register hostname in mDNS module

Pre-requisites:

L2 - L3 connections should be done and AT+NMDNSSTART command should be issued

Command Syntax:

AT+NMDNSHNREG=<hostname>

Parameter Description:

Parameters	Required?	Value	Description
hostname	M	Value Range: 1-32 Format: STR	Hostname to be registered

Default value:

Response:

If hostname registration is successful, it returns SUCCESS, else returns FAILURE and registered API is called with the result.

Responses	Remarks
AT+NMDNSSTART<CR><LF> +NMDNSSTART:successful<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSHNREG="Telit_Guest"<CR><LF> +NMDNSHNREG:successful<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NMDNSSTART
+NMDNSSTART:successful
OK
AT+NMDNSHNREG="Telit_Guest"
+NMDNSHNREG:successful
OK
```

9.1.1.3. MDNS Service Registration

Command Description:

To register a service in mDNS module.

Pre-requisites:

L2 - L3 connections should be done and AT+NMDNSSTART command should be issued

Command Syntax:

```
AT+NMDNSSRVREG=<instance_name>,<protocol>,<port>
```

Parameter Description:

Parameters	Required?	Value	Description
instance_name	M	Value Range: 1-32	Instance name of service to be registered

		Format: STR Default value:	
protocol	M	Value Range: 1-32 Format: STR Default value:	Type or Protocol of the service to be registered
port	M	Value Range: 1-9000 Format: UINT Default value:	Port number of the service to be registered

Response:

If service registration is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NMDNSSTART<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSHNREG="Telit_Guest"<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSSRVREG="_QCA4020","_Myprinter._tcp.local",80<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NMDNSSTART
OK
AT+NMDNSHNREG="Telit_Guest"
OK
AT+NMDNSSRVREG="_QCA4020","_Myprinter._tcp.local",80
OK
```

9.1.1.4. MDNS Service De-Registration

Command Description:

To de-registration a service in mDNS module.

Pre-requisites:

L2 - L3 connections should be done. AT+NMDNSSTART and AT+NMDNSSRVREG commands should be issued

Command Syntax:

AT+NMDNSSRVDEREG=<service>

Parameter Description:

Parameters	Required?	Value	Description
service	M	Value Range: 1-64 Format: STR Default value:	Instance name of service to be de-registered

Response:

If service de-registration is successful, it returns SUCCESS, else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NMDNSSTART<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSHNREG="Telit_Guest"<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSSRVREG="_QCA4020","_Myprinter._tcp.local",80<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSSRVDEREG="_QCA4020._MyPrinter._tcp.local"<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+NMDNSSTART

OK

AT+NMDNSHNREG="Telit_Guest"

OK

AT+NMDNSSRVREG="_QCA4020","_Myprinter._tcp.local",80

OK

AT+NMDNSSRVDEREG="_QCA4020._MyPrinter._tcp.local"

OK

9.1.1.5. MDNS Text Update

Command Description:

To update a txt record of a service in mDNS module.

Pre-requisites:

L2 - L3 connections should be done and AT+NMDNSSTART, AT+NMDNSHNREG and AT+NMDNSSRVREG commands should be issued

Command Syntax:

AT+NMDNSUPDATETXT=<service_name>,<txt_record>

Parameter Description:

Parameters	Required?	Value	Description
service_name	M	Value Range: 1-32 Format: STR Default value:	Service name of service of whose txt record should be updated
txt_record	M	Value Range: 1-32 Format: STR Default value:	TXT record of the service to be updated

Response:

If txt updation is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+NMDNSSTART<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSHNREG="Telit_Guest"<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSSRVREG="_QCA4020","_Myprinter._tcp.local",80<CR><LF> <CR><LF>OK<CR><LF> AT+NMDNSUPDATETXT="_QCA4020._MyDevice._tcp.local","updated txt"<CR><LF>	Successful

<CR><LF>OK<CR><LF>	
ERROR	If parameter is not valid

Example:

```
AT+NMDNSSTART
```

```
OK
```

```
AT+NMDNSHNREG="Telit_Guest"
```

```
OK
```

```
AT+NMDNSSRVREG="_QCA4020","_Myprinter._tcp.local",80
```

```
OK
```

```
AT+NMDNSUPDATETXT="_QCA4020._MyDevice._tcp.local","updated txt"
```

```
OK
```

9.1.1.6. MDNS Stop

Command Description:

To stop the mDNS module

Pre-requisites:

L2 - L3 connections should be done and AT+NMDNSSTART command should be issued

Response:

If stopping mDNS module is successful, it returns SUCCESS else returns FAILURE and registered API is called with the result.

Responses	Remarks
AT+NMDNSSTOP<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Command Syntax:

```
AT+NMDNSSTOP
```

Example:

```
AT+NMDNSSTOP
```

```
OK
```

9.1.2. Ping

Command Description:

To test reachability of a host on an Internet Protocol network

Pre-requisites:

L2 - L3 connections should be done

Command Syntax:

```
AT+NPING=<IPAddr>, [<payload>,<count>]
```

Parameter Description:

Parameters	Required?	Value	Description
IPAddr	M	Value Range: 7-32 Format: IP Default value: 0	specifies IP address to ping
payload	O	Value Range: 1-1576 Format: UINT Default value: 64	specifies payload of each packet size to ping
count	O	Value Range: 1-65535 Format: UINT Default value: 1	specifies number of pings

Command Response

If ping is successful +SC:IP,bytes,time in milli seconds, is sent else failure is sent.

Response Name	Response Type	Response Range
dst	IP	7-32
bytes	UINT	1-1576
Time	UINT	1-4294967295
ping status	status success, invalid IP address, Request timed out	0-2

Response:

If ping process is successful, it gives ping result with time in milliseconds,
else returns FAILURE.

Responses	Remarks
AT+NPING=192.168.120.3,1576,3<CR><LF> <CR><LF>OK<CR><LF> +NPING:192.168.120.3,1576,31<CR><LF> +NPING:192.168.120.3,1576,29<CR><LF> +NPING:192.168.120.3,1576,33<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+NPING=192.168.120.3,1576,3
OK
+NPING:192.168.120.3,1576,31
+NPING:192.168.120.3,1576,29
+NPING:192.168.120.3,1576,33
```

9.2. [Application Features](#)

9.2.1. Firmware Upgrade

9.2.1.1. HTTPC Firmware Upgrade

Command Description:

To upgrade firmware using http client that connects to the mentioned http server to fetch ota file and load it as trail firmware on a software reset. Issue AT+FUTRIALIMG to upgrade trail firmware to current firmware or discard it

Pre-requisites:

L2 - L3 connections should be done

Command Syntax:

```
AT+FUHTTPC=<SrvAddr>,[<SrvPort>],<FileUri>,[<sslEnableFlag>,<Timeout>]
```

Parameter Description:

Parameters	Required?	Value	Description
------------	-----------	-------	-------------

SrvAddr	M	Value Range: 1-63 Format: STR Default value: ""	server IP address/Name
SrvPort	O	Value Range: 1-65535 Format: UINT Default value: 80	Server Port.
FileUri	M	Value Range: 1-127 Format: STR Default value: "ota.bin"	URI of firmware file on server
sslEnableFlag	O	Value Range: 0-1 Format: UINT Default value: 0	SSL Enable Flag. 1 - For HTTPS, 0 - For HTTP
Timeout	O	Value Range: 1-1000000 Format: UINT Default value: 100	Http Server Timeout value

Command Response

On success +FUHTTTPC:Success and on failure +FUHTTTPC:Failure is sent

Response Name	Response Type	Response Range
firmwareupgrade status	Success, Failure	0-1

Response:

Registered API is called with the operational result of firware upgradation

Responses	Remarks
AT+FUHTTTPC="192.168.45.22", 80, "ota.bin", 100<CR><LF><CR><LF>OK<CR><LF>	Successful

ERROR

If parameter is not valid

Example:

```
AT+FUHTTPC="192.168.45.22", 80, "\ota.bin", 100
```

```
OK
```

9.2.1.2. HTTPD Firmware Upgrade

Command Description:

To upgrade firmware using http server to which client can post the FWUP file and load it as trail firmware on a software reset. Issue AT+FUTRIALIMG to upgrade trail firmware to current firmware or discard it

Pre-requisites:

L2 - L3 connections should be done

Command Syntax:

```
AT+FUHTTPD=<mode>,<device_id>,<family>,<http_port>,<https_port>
```

Parameter Description:

Parameters	Required?	Value	Description
mode	M	Value Range: 0-2 Format: INT Default value: 0	Mode of the HTTP server. 0 - HTTP, 1 - HTTPS, 2 - Both
device_id	M	Value Range: 0-2 Format: INT Default value: 2	Device Id given my at+WNI command response
family	M	Value Range: 2-4 Format: INT Default value: 0	specifies http server to run in IPV4 or IPV6 of both. \b 2 - IPV4, 3 - IPV6, 4 = IPV4 and IPV6
http_port	M	Value Range: 1-65535 Format: UINT Default value: 80	port number of HTTP server
https_port	M	Value Range: 1-65535 Format: UINT	port number of HTTPS server

Default value:
443

Command Response

On success +FUHTTPD:Success and on failure +FUHTTPD:Failure is sent

Response Name	Response Type	Response Range
status	Success, Failure	0-1

Response:

Registered API is called with the operational result of firmware upgradation

Responses	Remarks
AT+FUHTTPD=0,0,2,80,0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

AT+FUHTTPD=0,0,2,80,0

OK

9.2.1.3. Trial Image

Command Description:

To validate trial firmware either by deleting it or upgrading it to current firmware

Pre-requisites:

L2 - L3 connections should complete and AT+FUHTTTPC command should be executed

Command Syntax:

AT+FUTRIALIMG=[<acceptImgFlag>,<rebootFlag>]

Parameter Description:

Parameters	Required?	Value	Description
acceptImgFlag	O	Value Range: 0-1 Format: UINT Default	Trial Image acceptance Flag. 0 - Rejected, 1 - Accepted

value: 1			
rebootFlag	O	Value Range: 0-1 Format: UINT Default value: 1	Reboot after accepting the trial image. 0 - No Reboot, 1 - Reboot

Response:

On successful trial image handling as per user inputs, it returns SUCCESS else returns FAILURE and registered API is called with the result

Responses	Remarks
AT+FUTRIALIMG=1,1<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

```
AT+FUTRIALIMG=1,1
```

```
OK
```

9.2.2. Provisioning

9.2.2.1. Web Provisioning

Command Description:

To start HTTP provisioning server

Pre-requisites:

L2 - L3 connections should be done

Command Syntax:

```
AT+WNWEBPROV=<START>, [<PORT>, <SSL>, <SERVER_CERT>, <SERVER_KEY>,  
<SERVER_CA>]
```

Parameter Description:

Parameters	Required?	Value	Description
START	M	Value Range: 0-1 Format: INT	Set to 1 for START and 0 to STOP provisioning server

Default value: 0			
PORT	O	Value Range: 0-9999 Format: INT Default value: 80	HTTP port on which server will be running
SSL	O	Value Range: 0-1 Format: INT Default value:	Set this to 1 to enable HTTPS
SERVER_CERT	O	Value Range: "SSL_CERT" Format: STR Default value: "SSL_SERVER"	SSL server certificate name
SERVER_KEY	O	Value Range: "SSL_KEY" Format: STR Default value: "SSL_KEY"	SSL Private Key Name
SERVER_CA	O	Value Range: "SSL_CA" Format: STR Default value: "SSL_CA"	SSL CA certificate name

Command Response

If web server is started successfully +WEBPROV:Success is sent else +WEBPROV:Error is sent.

Response Name	Response Type	Response Range
WEBPROV Send Status	Success, Error	0-1

Response:

If initialization is successful, it returns SUCCESS, else returns FAILURE.

Once the allocation is done, registered API is called with the result.

Responses	Remarks
To allocate an HTTP client handle. AT+HTTPCINIT=0<CR><LF> <CR><LF>OK<CR><LF> AT+HTTPCINIT=0,100,<CR><LF> 0<CR><LF> <CR><LF>OK<CR><LF>	Successful
ERROR	If parameter is not valid

Example:

To allocate an HTTP client handle.

AT+HTTPCINIT=0

OK

AT+HTTPCINIT=0,100,

0

OK

9.2.3. TR50

9.2.3.1. Configure deviceWISE

Command Description:

Configure deviceWISE parameters in following format:

<serverUrl>,<deviceIdSelector>,<appToken>,<security>,<

heartBeat>,<autoReconnect>,<overflowHandling>,<atrUnInstanceId>,<serviceTimeout>

Pre-requisites:**Command Syntax:**

AT+DWCFG=<cmd>

Parameter Description:

Parameters	Required?	Value	Description
cmd	M	Value Range: 0-1,0-1,10-86400,0-3,0-1,0-4,1-120,1-5,128,16	Parameters: <serverUrl> - URL of the M2M Service instance in address:port form. <deviceIdSelector> 0 : 1 (0=IMEI

Format: RAW Default value: ""	1=CCID/ESN), basically 0 if not SIM card or CDMA ID installed <appToken> - The secure application token provided in the Management Portal <security> - Flag indicating if the SSL encryption is enabled. 0 : SSL encryption disabled (default), 1 : SSL encryption enabled <heartBeat> - If no packets are received in the number of seconds specified in the heartbeat field, a heartbeat message will be sent to keep the connection alive. Default: 60, Range: 10 - 86400 <autoReconnect> - Flag indicating if the connection manager should automatically reconnect to the service. 0 : auto-reconnect disabled 1 : auto-reconnect lazy - reconnect on next send and every 3600 seconds. 2 : auto-reconnect moderate (default) - reconnect 120 seconds, then every 3600 seconds after the first day. 3 : auto-reconnect aggressive - reconnect every 120 seconds. <overflowHandling> - Flag indicating if the way to handle overflows in data management. 0 : FIFO (default) 1 : LIFO <atruncInstanceld> - AT instance that will be used by the service to run the AT Command. Default 4, Range 0 : 4 <serviceTimeout> - It defines in seconds the maximum time interval for a service request to the server. Default 5, Range 1 : 120
--	---

Response:

This command sets the parameters related to the deviceWISE

functionality. If the command is executed successfully, "OK" is returned
 else "ERROR" is returned.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

9.2.3.2. deviceWISE Connect/Disconnect

Command Description:

Set command connects/disconnects to the M2M Service

Pre-requisites:

Before issuing publish L2 and L3 connection and TR50 config should have been done and

Command Syntax:

AT+DWCONN=<cmd>

Parameter Description:

Parameters	Required?	Value	Description
cmd	M	Value Range: 0-1 Format: RAW Default value:	Flag to connect/disconnect to the M2M Service 0 : disconnect (default), 1 : connect

Response:

If DWCONN is successful, it returns SUCCESS, else returns FAILURE.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

9.2.3.3. Send deviceWISE Data

Command Description:

Send formatted data to the M2M Service.

Data can be normal request, method, method update
or acknowledgement.

Pre-requisites:

It's possible to use AT+DWSEND only if the connection has been
opened with AT+DWCONN

Command Syntax:

AT+DWSEND=<cmd>

Parameter Description:

Parameters	Required?	Value	Description
cmd	M	Value Range: 0-3 Format: RAW Default value:	Format is <type>,<param_1>[,<param_2>[,'<param_n>]]], Where, <type> - code for the type of message to send. (0 for normal request, 1 for method request, 2 for method update, 3 for method acknowledgement). Type 0: <param_1> - command : the api command to execute. <param_i> - string parameter indicating the i-th parameter Type 1: <param_1> - 'thingKey' : the key of a thing to execute. <param_2> - timeout : time to wait in seconds before returning an error for the request. <param_3> - method : the method key of a thing to execute. <param_4> - is singleton : 0 or 1. 1 if no more than one of these instance can exist. <param_5+> - String parameter indicating the i-th parameter, with i=1,',20. Type 2: <param_1> - id : the identification of the method instance. <param_2> - message : a message represents the current status of the method. Type 3:

<param_1> - id : the identification of the method instance.

<param_2> - status : the integer result status for the execution. 0 is reserved for OK.

<param_3> When status is not 0 - error message associates with the status.

else - param_i should be the name of the element and param_i+1 should be the value of the element.

Response:

if data are successfully sent, then the response is OK.

If data sending fails for some reason, an error code is reported.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

9.2.3.4. Send Raw deviceWISE Data

Command Description:

This command is used to send raw data to the M2M Service.

Content must be valid JSON. The module responds to the command with the prompt <greater_than><space> and waits for the data to send. When dataLen bytes have been sent, operation is automatically completed.

Pre-requisites:

It's possible to use AT+DWSENDNR only if the connection has been opened with AT+DWCONN

Command Syntax:

AT+DWSENDNR=<cmd>

Parameter Description:

Parameters	Required?	Value	Description
cmd	M	Value Range: 1-	dataLen: number of bytes to be sent,

1500 Range: 1 - 1500.
Format: RAW
Default value:

Response:

if data are successfully sent, then the response is OK.

If data sending fails for some reason, an error code is reported.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

9.2.3.5. Receive deviceWISE Data

Command Description:

This command permits the user to read formatted data arriving

from the M2M Service; the module is notified of these data by the URC #DWRING.

If the incoming data are accepted with AT+DWRCV, then the formatted data are received and showed with the following URC:

#DWDATA: <msgid>,<error>,<len>,<param_1>[,<param_2>[,â€¦[,<param_n>]]]

Pre-requisites:

It is possible to use AT+DWRCV only if the connection has been opened with AT+DWCONN

Command Syntax:

AT+DWRCV=<cmd>

Parameter Description:

Parameters	Required?	Value	Description
cmd	M	Value Range: 1-1500 Format: RAW	<msgid> - index of the data message to receive, as indicated in the URC #DWRING. Range: >=1 If the data received are the consequence of a previous data sending (issued by AT+DWSND), then the msgId

Default value:

value is the same of the msgId value reported in the answer of AT+DWSEND.

Response:

If the incoming data are accepted with AT+DWRCV, then the formatted data are

received and showed with the following URC: #DWDATA:
msgId,error,len,param_1[,param_2[,â€¦[,param_n]]].

else error is reported.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

9.2.3.6. Receive Raw deviceWISE Data

Command Description:

This command permits the user to read raw data arriving from the M2M Service; the module is notified of these data by the URC #DWRING.

If the incoming data are accepted with AT+DWRCVR, then the data are received and showed with the following URC:

#DWRDATA: <msgId>,<len>,<data>

Pre-requisites:

It is possible to use AT+DWRCVR only if the connection has been opened with AT+DWCONN

Command Syntax:

AT+DWRCVR=<cmd>

Parameter Description:

Parameters	Required?	Value	Description
cmd	M	Value Range: >=1 Format:	msgId - index of the data message to receive, as indicated in the URC #DWRING. Range: >=1 If the data received are the consequence of a previous data sending (issued by

RAW	AT#DWSENDNR), then the msgId
Default value:	value is the same of the msgId value reported in the answer of AT+DWSENDNR.

Response:

If the incoming data are accepted with AT+DWRVCVR, then the data are received and showed with the following URC: +DWRDATA: <msgId>,<len>,<data>
else error is reported.

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Example:

9.2.3.7. Read Unresolved deviceWISE Data

Command Description:

Command permits the user to obtain information regarding the messages pending from the M2M Service in following format:

#DWLRCV:

```
<msg_number>[,<msgld_1>,<msg_1_len>[,<msgld_2>,<msg_2_len>[,&#39;&#39;<msgld_n>,<msg_n_len>]]]
```

where:

<msg number> - number of messages pending from the M2M Service

Range: ≥ 0

<msgId i> - index of the i-th data message to receive

`<msg_i_len>` - length of the i-th data message to receive

Pre-requisites:

It is possible to use AT+DWLRCV only if the connection has been opened with AT+DWCONN

Response:

Command gives response in the format:

```
msg_number[,msgld_1,msg_1_len[,msgld_2,msg_2_len[,â€¦â€¦,msgld_n,msg_n_len]
```

where:

Service, Range: >=0

msg_number - number of messages pending from the M2M

msgId_i - index of the i-th data message to receive

msg_i_len - length of the i-th data message to receive

Responses	Remarks
	Successful
ERROR	If parameter is not valid

Command Syntax:

AT+DWLRCV

Example:

9.2.3.8. deviceWISE Connection Status

Command Description:

This command returns the status of the connection, including some runtime statistics.

Pre-requisites:

Response:

Command gives response in the format:

#DWSTATUS:

<connected><lastErrorCode>,<latency>,<pktsIn>,<pktsOut>,<bytesIn>,<bytesOut>

<connected> : 3 = waiting to connect, 2 = connected, 1 = trying to connect, 0 = disconnected

<lastErrorCode>: last error code encountered by the client

<latency> : milliseconds measured between last request and reply.

<pktsIn> : number of packets received, tracked by the server

<pktsOut> : number of packets sent.

<bytesIn> : number of bytes received, TCP/IP payload

<bytesOut> : number of bytes sent.

Responses	Remarks
-----------	---------

Successful	
ERROR	If parameter is not valid

Command Syntax:

AT+DWSTATUS

Example:



SUPPORT INQUIRIES

Link to **www.telit.com** and contact our technical support team for any questions related to technical issues.

www.telit.com



Telit Communications S.p.A.
Via Stazione di Prosecco, 5/B
I-34010 Sgonico (Trieste), Italy

Telit IoT Platforms LLC
5300 Broken Sound Blvd, Suite 150
Boca Raton, FL 33487, USA

Telit Wireless Solutions Inc.
3131 RDU Center Drive, Suite 135
Morrisville, NC 27560, USA

Telit Wireless Solutions Co., Ltd.
8th FL., Shinyoung Securities Bld.
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu
Seoul, 150-884, Korea

Telit Wireless Solutions Ltd.
10 Habarzel St.
Tel Aviv 69710, Israel

Telit Wireless Solutions
Tecnologia e Servicos Ltda
Avenida Paulista, 1776, Room 10.C
01310-921 São Paulo, Brazil

Telit reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by Telit at any time. For most recent documents, please visit www.telit.com

Copyright © 2016, Telit