

# **MC60 BLE**

# **AT Commands Manual**

**GSM/GPRS/GNSS Module Series**

Rev. MC60\_BLE\_AT\_Commands\_Manual\_V1.2

Date: 2018-09-30

Status: Released



**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

7<sup>th</sup> Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

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

**Or our local office. For more information, please visit:**

<http://www.quectel.com/support/sales.htm>

**For technical support, or to report documentation errors, please visit:**

<http://www.quectel.com/support/technical.htm>

Or email to: [support@quectel.com](mailto:support@quectel.com)

## **GENERAL NOTES**

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

## **COPYRIGHT**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2018. All rights reserved.***

# About the Document

## History

Revision	Date	Author	Description
1.0	2017-05-17	Miles MA	Initial
1.1	2017-12-21	Miles MA	<ol style="list-style-type: none"> <li>Added AT+QGATSETADV</li> <li>Added AT commands of BLE GATT client</li> <li>Added URC +QBTGATCSCAN:, +QBTGATCCON: and +QBTGATCN:</li> <li>Added BLE operation process (Chapter 2)</li> <li>Deleted AT commands AT+QBTFMPSREG and AT+QBTPXPSREG</li> </ol>
1.2	2018-09-30	Miles MA	<ol style="list-style-type: none"> <li>Added command AT+QGATADVDATA, AT+QGATSCANRSP, AT+QBTGATSENLE and AT+QGATCFGSCAN. (Chapter 3.2.10, 3.2.11, 3.2.15 and 3.3.15)</li> <li>Added URC +QBTGATEWREQ: (Chapter 4.3)</li> <li>Updated the description of command AT+QGATSETADV, AT+QBTGATCCON and AT+QBTGATCSS (Chapter 3.2.9, 3.3.3 and 3.3.4)</li> <li>Updated the examples in Chapter 2.2 and 3.1.5</li> <li>Updated codes of &lt;errorcode&gt; and their meanings in Table 3</li> </ol>

## Contents

About the Document .....	2
Contents .....	3
Table Index .....	5
<b>1 Introduction .....</b>	<b>6</b>
<b>2 BLE Operation Process .....</b>	<b>7</b>
2.1. BLE Advertising .....	8
2.2. Simplified Process of BLE Communication .....	8
2.2.1. BLE Process When Module is as BLE Server .....	8
2.2.2. BLE Process When Module is as BLE Client .....	10
<b>3 Description of BLE AT Commands .....</b>	<b>12</b>
3.1. General BLE AT Commands .....	12
3.1.1. AT+QBTPWR Power ON/OFF BT Device .....	12
3.1.2. AT+QBTLEADDR Read Local Address of BLE Device .....	13
3.1.3. AT+QBTLETXPWR Set the Level of BLE Transmission Power .....	13
3.1.4. AT+QBTNAME Read/Write the Local Name of BT Device .....	14
3.1.5. AT+QBTVISB Set the Current Visibility Mode of BT .....	15
3.2. BLE GATT Server AT Commands .....	16
3.2.1. AT+QBTGATSREG Register/Deregister a GATT Server .....	16
3.2.2. AT+QBTGATSS Add/Remove a Service .....	17
3.2.3. AT+QBTGATSC Add a Characteristic to an Existing Service .....	18
3.2.4. AT+QBTGATSD Add a Descriptor to an Existing Service .....	20
3.2.5. AT+QBTGATSST Start/Stop a Service .....	21
3.2.6. AT+QBTGATSIND Send an Indication or Notification to a Client .....	22
3.2.7. AT+QBTGATSRSP Send a Response to a Client's Read or Write Operation .....	23
3.2.8. AT+QBTGATADV Set Advertising Parameters .....	24
3.2.9. AT+QGATSETADV Set Advertising Data .....	24
3.2.10. AT+QGATADVDATA Set Advertising Data .....	26
3.2.11. AT+QGATSCANRSP Set Scan Response Data .....	27
3.2.12. AT+QBTGATCPU Update Connection Parameters .....	28
3.2.13. AT+QBTGATSDISC Disconnect Server Actively .....	29
3.2.14. AT+QBTGATSL Start/Stop BLE Advertising .....	29
3.2.15. AT+QBTGATSENLE Start/Stop BLE Advertising .....	30
3.3. BLE GATT Client AT Commands .....	31
3.3.1. AT+QBTGATCREG Register/Deregister GATT Client .....	31
3.3.2. AT+QBTGATCSCAN Start/Stop Scanning LE Device .....	32
3.3.3. AT+QBTGATCCON Connect/Disconnect GATT Client to Remote LE Device .....	33
3.3.4. AT+QBTGATCSS Search and Enumerate Peer's Service .....	34
3.3.5. AT+QBTGATCGC Search and Enumerate Peer's Characteristic of the Service .....	35
3.3.6. AT+QBTGATCGD Search and Enumerate Peer's Descriptor of the Service with UUID .....	37

3.3.7.	AT+QBTGATCRC	Read Peer's Characteristic of the Service with UUID .....	38
3.3.8.	AT+QBTGATCWC	Write Peer's Characteristic of the Service with UUID.....	39
3.3.9.	AT+QBTGATCRD	Read Peer's Descriptor of the Service with UUID.....	41
3.3.10.	AT+QBTGATCWD	Write Peer's Descriptor of the Service with UUID .....	42
3.3.11.	AT+QBTGATCEW	Execute Write Command .....	44
3.3.12.	AT+QBTGATCRN	Register/Deregister Notification Request.....	44
3.3.13.	AT+QBTGATCGDT	Get the Type of Peer Device .....	46
3.3.14.	AT+QBTGATCL	Start/Stop Listen Request .....	47
3.3.15.	AT+QGATCFGSCAN	Set Scan Interval and Scan Window .....	47
<b>4</b>	<b>Description of URCs .....</b>		<b>49</b>
4.1.	Notify the GATT Connection or Disconnection.....		49
4.2.	Notify the GATT Client's Write Request .....		49
4.3.	Notify the GATT Client's Write Request Execution.....		50
4.4.	Notify the GATT Client's Read Request .....		52
4.5.	Report Scanning Result .....		52
4.6.	Notify Client a Connection's Status .....		53
4.7.	Notify Client a Notification or Indication Comes .....		54
<b>5</b>	<b>Appendix A References .....</b>		<b>56</b>

## Table Index

TABLE 1: TERMS AND ABBREVIATIONS .....	56
TABLE 2: FORMAT MAP OF PROPERTIES AND PERMISSION .....	56
TABLE 3: DIFFERENT CODING SCHEMES OF +CME ERROR: <ERRORCODE> .....	57

# 1 Introduction

This document presents the BLE AT Commands Set for Quectel MC60 module. It is only applicable to the MC60 module with ordering code **MC60ECA-04-BLE**.

## 2 BLE Operation Process

A set of AT commands are provided to support basic BLE operation, including scanning, advertising, connecting and so on. Quectel MC60 module with ordering code **MC60ECA-04-BLE** supports GATT profile which can be used to communicate between module and other BLE devices. The module can act as a peripheral (server) role or central (client) role.

The following figure shows the detailed operation of BLE communication by AT commands.

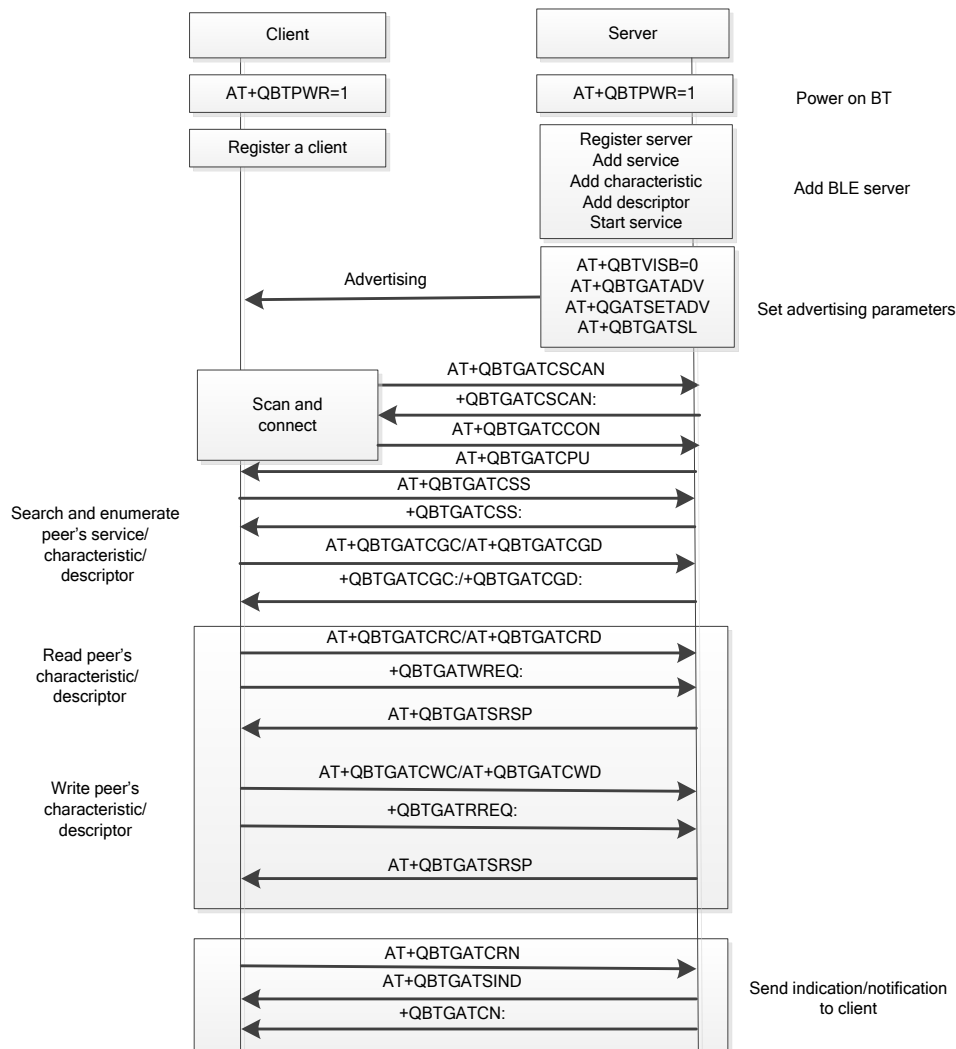


Figure 1: BLE Operation Flow Chat



## 2.1. BLE Advertising

If the Bluetooth function is turned on, both the classic BT and BLE will be broadcasted. The separate process of BLE broadcasting is as below.

```
AT+QBTPWR=1                //Power on BT device.
OK
AT+QBTVISB=0                //Set BT to be invisible, and stop classic BT and BLE advertising.
OK
AT+QBTVISB?
+QBTVISB: 0

OK
AT+QBTGATSREG=1,"ABC2"      //Register a GATT server.
+QBTGATSREG: 1,"ABC2",0

OK
AT+QBTGATSL="ABC2",1        //Start advertising.
+QBTGATSL: "ABC2",0

OK
```

## 2.2. Simplified Process of BLE Communication

### 2.2.1. BLE Process When Module is as BLE Server

```
AT+QBTPWR=1                //Power on BT device.
OK
AT+QBTLEADDR?
+QBTLEADDR: DFF596CB3C76

OK
AT+QBTGATSREG=1,"A001"      //Register a GATT server.
+QBTGATSREG: 1,"A001",0

OK
AT+QBTGATSS=1,"A001","1234",15,1,254 //Add a service.
+QBTGATSS: 1,"A001","1234",1,0,254,256

OK
AT+QBTGATSC=1,"A001",256,"C001",2,16,17 //Add a characteristic.
+QBTGATSC: 1,"A001",256,"C001",2,0,258
```

OK

**AT+QBTGATSD=1,"A001",256,"0229",1,17** //Add a descriptor.

+QBTGATSD: 1,"A001",256,"0229",1,0,259

OK

//If property "notify" or "indicate" of characteristic is added, please add a descriptor with UUID "0229" after it.

//If some Android versions are unable to register "notify", please try to add "indicate".

**AT+QBTGATSC=1,"A001",256,"C002",2,6,17** //Add a characteristic.

+QBTGATSC: 1,"A001",256,"C001",2,0,261

OK

**AT+QBTGATSST=1,"A001",256,0** //Start a service.

+QBTGATSST: 1,"A001",0,256

OK

+QBTGATSCON: 1,"A001",0,61625C58B558,1 //URC: establish a connection.

+QBTGATWREQ: "A001",1,14,61625C58B558,259,0100,0,0,0

+QBTGATRREQ: "A001",1,15,61625C58B558,261,0,0 //URC: receive GATT client's Read Request.

**AT+QBTGATSRSP="A001",0,1,15,261,"0000"** //Send a response to a client's read operation.

+QBTGATSRSP: 0,"A001",1,261

OK

+QBTGATWREQ: "A001",1,15,61625C58B558,261,2222,0,0,0 //URC: receive GATT client's Write Request that does not need to response.

+QBTGATRREQ: "A001",1,16,61625C58B558,259,0,0 //URC: receive GATT client's Read Request.

**AT+QBTGATSRSP="A001",0,1,16,259,"0100"** //Send a response to a client's read operation.

+QBTGATSRSP: 0,"A001",1,259

OK

**AT+QBTGATSIND="A001",1,258,1,"3333"** //Send an indication or notification to a client.

+QBTGATSIND: 0,"A001",1,2

OK

**AT+QBTGATSDISC=1** //Disconnect server.

OK

```
+QBTGATSCON: 0,"A001",0,61625C58B558,1
```

## 2.2.2. BLE Process When Module is as BLE Client

```
AT+QBTPWR=1           //Power on BT device.
OK
AT+QBTGATCREG=1,"B001" //Register a GATT client.
+QBTGATCREG: 1,0,"B001"

OK
AT+QBTGATCSCAN=1,"B001" //Start scanning LE device.
+QBTGATCSCAN: 1,0,"B001"

OK

+QBTGATCSCAN: "B001",763CCB96F5DF,77,02011A05094C593032
AT+QBTGATCSCAN=0,"B001" //Stop scanning LE device.
+QBTGATCSCAN: 0,0,"B001"

OK
AT+QBTGATCCON=1,"B001",763CCB96F5DF,1 //Connect GATT client.
OK

+QBTGATCCON: 1,0,"B001",763CCB96F5DF,1 //Establish a connection.
AT+QBTGATCSS="B001",1 //Enumerate peer's service.
OK

+QBTGATCSS: 0,"B001",1,"0018",0,1

+QBTGATCSS: 0,"B001",1,"0118",0,1

+QBTGATCSS: 0,"B001",1,"1234",0,1
AT+QBTGATCGC="B001",1,"1234",0,1 //Enumerate peer's characteristic of the service.
OK

+QBTGATCGC: 0,"B001",1,"1234",0,1,"C001",0,16
AT+QBTGATCGC="B001",1,"1234",0,1,"C001",0
OK

+QBTGATCGC: 0,"B001",1,"1234",0,1,"C002",0,6
AT+QBTGATCGD="B001",1,"1234",0,1,"C001",0 //Search and enumerate peer's descriptor of the
service.
OK
```

```
+QBTGATCGD: 0,"B001",1,"1234",0,1,"C001",0,"0229",0
AT+QBTGATCRN=1,"B001",763CCB96F5DF,"1234",0,1,"C001",0 //Register notification request.
+QBTGATCRN: 1,0,"B001",763CCB96F5DF,"1234",0,1,"C001",0

OK
AT+QBTGATCWD="B001",1,1,"1234",0,1,"C001",0,"0229",0,"0100",0 //Write peer's descriptor of the
OK                                         service with UUID

+QBTGATCWD: 0,"B001",1,"1234",0,1,"C001",0,"0229",0
AT+QBTGATCRC="B001",1,"1234",0,1,"C002",0,0 //Read peer's characteristic of the service.
OK

+QBTGATCRC: 0,"B001",1,"1234",0,1,"C002",0,"0000"
AT+QBTGATCWC="B001",1,1,"1234",0,1,"C002",0,"2222",0 //Write peer's characteristic.
OK

+QBTGATCWC: 0,"B001",1,"1234",0,1,"C002",0
AT+QBTGATCRD="B001",1,"1234",0,1,"C001",0,"0229",0,0
OK

+QBTGATCRD: 0,"B001",1,"1234",0,1,"C001",0,"0229",0,"0100"
+QBTGATCN: "B001",1,763CCB96F5DF,"1234",0,1,"C001",0,0,"3333" //Receive a notification from
                                                                    server.

AT+QBTGATCCON?
+QBTGATCCON: 1, 763CCB96F5DF

OK
+QBTGATCCON: 0,0,"B001",763CCB96F5DF,1
```

# 3 Description of BLE AT Commands

The BT device in this chapter includes classic BT and BLE devices.

## 3.1. General BLE AT Commands

### 3.1.1. AT+QBTPWR Power ON/OFF BT Device

The command is used to power on/off BT device.

#### AT+QBTPWR Power ON/OFF BT Device

Test Command <b>AT+QBTPWR=?</b>	Response: <b>+QBTPWR:</b> (list of supported <b>&lt;on_off&gt;</b> s)  <b>OK</b>
Read Command <b>AT+QBTPWR?</b>	Response: <b>+QBTPWR:</b> <b>&lt;power_status&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTPWR=&lt;on_off&gt;</b>	Response: <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;on_off&gt;</b>	<u>0</u>	Off
	1	On
<b>&lt;power_status&gt;</b>	<u>0</u>	Power off
	1	Power on

#### Example

```
AT+QBTPWR=1      //Power on BT device.
OK
```

### 3.1.2. AT+QBTLEADDR Read Local Address of BLE Device

#### AT+QBTLEADDR Read Local Address of BLE Device

Test Command <b>AT+QBTLEADDR=?</b>	Response <b>OK</b>
Read Command <b>AT+QBTLEADDR?</b>	Response <b>+QBTLEADDR: &lt;ble_addr&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;ble_addr&gt;</b>	Local address of BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02C3)
-------------------------	------------------------------------------------------------------------------------------------------

#### Example

```
AT+QBTLEADDR?
+QBTLEADDR: A662616202C3
OK
```

### 3.1.3. AT+QBTLETXPWR Set the Level of BLE Transmission Power

#### AT+ QBTLETXPWR Set the Level of BLE Transmission Power

Test Command <b>AT+QBTLETXPWR=?</b>	Response <b>+QBTLETXPWR: (list of supported &lt;tx_level&gt;s)</b>  <b>OK</b>
Read Command <b>AT+QBTLETXPWR?</b>	Response <b>+QBTLETXPWR: &lt;tx_level&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTLETXPWR=&lt;tx_level&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

**<tx\_level>** The level of BLE transmission power. The range is 0-7.

## NOTES

1. This command takes effect after BT function is restarted.
2. The maximum level of BLE transmission power is 7.

## Example

```
AT+QBTLEXPWR=2
OK
AT+QBTLEXPWR?
+QBTLEXPWR: 2
OK
```

### 3.1.4. AT+QBTNAME Read/Write the Local Name of BT Device

AT+QBTNAME Read/Write the Local Name of BT Device	
Test Command <b>AT+QBTNAME=?</b>	Response <b>OK</b>
Read Command <b>AT+QBTNAME?</b>	Response <b>+QBTNAME: &lt;device_name&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTNAME=&lt;device_name&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

**<device\_name>** String type. BT device local name.

**NOTE**

The max length of the BT device local name is 54 bytes.

**Example**

```
AT+QBTNAME?
+QBTNAME: "MYBTDEVICE"
OK
```

### 3.1.5. AT+QBTVISB Set the Current Visibility Mode of BT

The classic BT and BLE will stop advertising when the BT settings are invisible.

#### AT+QBTVISB Set the Current Visibility Mode of BT

Test Command <b>AT+QBTVISB=?</b>	Response <b>+QBTVISB:</b> (list of supported <n>s),(list of supported <time>s)  OK
Read Command <b>AT+QBTVISB?</b>	Response <b>+QBTVISB:</b> <n>  OK  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command <b>AT+QBTVISB=&lt;n&gt;[,&lt;time&gt;]</b>	Response OK  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>  If <n>=2, response: OK  <b>+QBTVISB: 0</b>



## Parameter

<n>	0	Invisible
	1	Visible forever
	2	Visible temporarily in the time period that Bluetooth can be found by other devices
<time>	Integer type. Time period that Bluetooth can be found by other devices. The range is 1-255. Unit: second.	

## Example

```
AT+QBTVISB=0           //Set the current visibility mode of BT to be invisible.
OK
```

## 3.2. BLE GATT Server AT Commands

### 3.2.1. AT+QBTGATSREG Register/Deregister a GATT Server

AT+QBTGATSREG Register/Deregister a GATT Server	
Test Command <b>AT+QBTGATSREG=?</b>	Response: <b>+QBTGATSREG:</b> (list of supported <op>s), "<gserv_id>"  <b>OK</b>
Read Command <b>AT+QBTGATSREG?</b>	Response: <b>+QBTGATSREG:</b> "<gserv_id>",<role>,<ctx_handle> ...  <b>OK</b>
Write Command <b>AT+QBTGATSREG=&lt;op&gt;,"&lt;gserv_id&gt;"</b>	Response: <b>+QBTGATSREG:</b> <op>,"<gserv_id>",<result>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<op>	0	Deregister
	1	Register
<gserv_id>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character	

	of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<result>	0	SUCCESS
	1	FAIL
<role>	1	Server
<ctx_handle>	Handle of the registered GATT server.	

### Example

```
AT+QBTGATSREG=1,"ABC2"           //Register a GATT server.
+QBTGATSREG: 1,"ABC2",0
OK
```

### 3.2.2. AT+QBTGATSS Add/Remove a Service

#### AT+QBTGATSS Add/Remove a Service

Test Command <b>AT+QBTGATSS=?</b>	Response: <b>+QBTGATSS:</b> (list of supported <op>s), "<gserv_id>", "<service_uuid>", (list of supported <num_handles>s), (list of supported <is_primary>s), (list of supported <inst>s)  <b>OK</b>
Read Command <b>AT+QBTGATSS?</b>	Response: <b>+QBTGATSS:</b> "<gserv_id>", "<service_uuid>", <num_handles>,<service_handle>,<is_primary>,<inst>,<is_started> ...  <b>OK</b>
Write Command Add a service <b>AT+QBTGATSS=1,"&lt;gserv_id&gt;","&lt;service_uuid&gt;",&lt;num_handles&gt;,&lt;is_primary&gt;,&lt;inst&gt;</b>	Response: <b>+QBTGATSS:</b> 1,"<gserv_id>",<service_uuid>,<is_primary>,<result>,<inst>,<service_handle>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command Remove a service <b>AT+QBTGATSS=0,"&lt;gserv_id&gt;",&lt;service_handle&gt;</b>	Response: <b>+QBTGATSS:</b> 0,"<gserv_id>",<result>,<service_handle>  <b>OK</b>

If there is any error relating to ME functionality:

**+CME ERROR: <errorcode>**

## Parameter

<b>&lt;op&gt;</b>	0	Remove
	1	Add
<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end to be even).	
<b>&lt;service_uuid&gt;</b>	UUID of this service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).	
<b>&lt;num_handles&gt;</b>	Number of handles of this service. The range is 1-65535.	
<b>&lt;is_primary&gt;</b>	Service is the primary one or not.	
	0	Not primary service
	1	Primary service
<b>&lt;inst&gt;</b>	Instance ID of this service. The range is 0-255.	
<b>&lt;service_handle&gt;</b>	Handle of this service. The range is 0-65535.	
<b>&lt;result&gt;</b>	0	SUCCESS
	1	FAIL
<b>&lt;is_started&gt;</b>	0	Service is in stop state
	1	Service is in start state

## Example

```
AT+QBTGATSS=1,"ABC2","3418",15,1,254 //Add a service.
+QBTGATSS: 1,"ABC2","3418",1,0,254,256
```

OK

```
AT+QBTGATSS=0,"ABC2",256 //Remove a service.
+QBTGATSS: 0,"ABC2",0,256
```

OK

### 3.2.3. AT+QBTGATSC Add a Characteristic to an Existing Service

#### AT+QBTGATSC Add a Characteristic to an Existing Service

Test Command	Response:
AT+QBTGATSC=?	+QBTGATSC: 1,"<gserv_id>",(list of supported

	<p>&lt;service_handle&gt;s),“&lt;char_uuid&gt;”,(list of supported &lt;inst&gt;s),(list of supported &lt;prop&gt;s),(list of supported &lt;permission&gt;s)</p> <p>OK</p>
<p>Write Command</p> <p><b>AT+QBTGATSC=1,“&lt;gserv_id&gt;”,&lt;service_handle&gt;,”&lt;char_uuid&gt;”,&lt;inst&gt;,&lt;prop&gt;,&lt;permission&gt;</b></p>	<p>Response:</p> <p><b>+QBTGATSC: 1,“&lt;gserv_id&gt;”,&lt;service_handle&gt;,”&lt;char_uuid&gt;”,&lt;inst&gt;,&lt;result&gt;,&lt;char_handle&gt;</b></p> <p>OK</p> <p>If there is any error relating to ME functionality:</p> <p><b>+CME ERROR: &lt;errorcode&gt;</b></p>

## Parameter

<gserv_id>	<p>User ID (or the name) of GATT server.</p> <p>A hex value string (string should be included in quotation marks). Each character of it should be in set {‘0’~‘9’,‘a’~‘f’,‘A’~‘F’}. The max length is 32 bytes (if the length is odd, then fill ‘0’ at the end of it to be even).</p>
<service_handle>	Handle of this service. The range is 0-65535.
<char_uuid>	<p>UUID of this characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill ‘0’ at the end of it to be even).</p>
<inst>	Instance ID of this characteristic. The range is 0-255.
<prop>	Properties of this characteristic. For more details, please refer to <b>Table 2</b> .
<permission>	Permission of this characteristic. For more details, please refer to <b>Table 2</b> .
<char_handle>	Handle of this characteristic.
<result>	<p>0 SUCCESS</p> <p>1 FAIL</p>

## Example

```

AT+QBTGATSC=1,“ABC2”,256,“332a”,2,58,17           //Add a characteristic.
+QBTGATSC: 1,“ABC2”,256,“332A”,2,0,260

OK

```

### 3.2.4. AT+QBTGATSD Add a Descriptor to an Existing Service

AT+QBTGATSD Add a Descriptor to an Existing Service	
<b>Test Command</b> <b>AT+QBTGATSD=?</b>	<b>Response</b> <b>+QBTGATSD: 1,"&lt;gserv_id&gt;",(list of supported &lt;service_handle&gt;s),"&lt;desc_uuid&gt;",(list of supported &lt;inst&gt;s),(list of supported &lt;permission&gt;s)</b>  <b>OK</b>
<b>Write Command</b> <b>AT+QBTGATSD=1,"&lt;gserv_id&gt;",&lt;service_handle&gt;,"&lt;desc_uuid&gt;",&lt;inst&gt;,&lt;permission&gt;</b>	<b>Response</b> <b>+QBTGATSD: 1,"&lt;gserv_id&gt;",&lt;service_handle&gt;,"&lt;desc_uuid&gt;",&lt;inst&gt;,&lt;result&gt;,&lt;desc_handle&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;desc_uuid&gt;</b>	UUID of this descriptor. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;inst&gt;</b>	Instance ID of this descriptor. The range is 0-255.
<b>&lt;permission&gt;</b>	Permission of this descriptor. For more details, please refer to <b>Table 2</b> .
<b>&lt;desc_handle&gt;</b>	Handle of this descriptor. The range is 0-65535.
<b>&lt;service_handle&gt;</b>	Handle of this service. The range is 0-65535.
<b>&lt;result&gt;</b>	0      SUCCESS 1      FAIL

#### Example

```

AT+QBTGATSD=1,"ABC2",256,"1329",1,17           //Add a descriptor.
+QBTGATSD: 1,"ABC2",256,"1329",1,0,262
OK
  
```

### 3.2.5. AT+QBTGATSST Start/Stop a Service

AT+QBTGATSST Start/Stop a Service	
Test Command <b>AT+QBTGATSST=?</b>	Response: <b>+QBTGATSST:</b> (list of supported <op>s), "<gserv_id>", (list of supported <service_handle>s)[, (list of supported <transport>s)]  <b>OK</b>
Write Command Start a service <b>AT+QBTGATSST=1,"&lt;gserv_id&gt;",&lt;service_handle&gt;,&lt;transport&gt;</b>	Response: <b>+QBTGATSST: 1,"&lt;gserv_id&gt;",&lt;result&gt;,&lt;service_handle&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command Stop a service <b>AT+QBTGATSST=0,"&lt;gserv_id&gt;",&lt;service_handle&gt;</b>	Response: <b>+QBTGATSST: 0,"&lt;gserv_id&gt;",&lt;result&gt;,&lt;service_handle&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<gserv_id>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<service_handle>	Handle of this service. The range is 0-65535.
<transport>	Transport method to start this service.
<result>	0 LE 0 SUCCESS 1 FAIL

#### Example

```
AT+QBTGATSST=1,"ABC2",256,0 //Start a service.
+QBTGATSST: 1,"ABC2",0,256
```

OK

### 3.2.6. AT+QBTGATSIND Send an Indication or Notification to a Client

AT+QBTGATSIND Send an Indication or Notification to a Client	
<b>Test Command</b> <b>AT+QBTGATSIND=?</b>	<b>Response:</b> <b>+QBTGATSIND: "&lt;gserv_id&gt;",(list of supported &lt;conn_id&gt;s),(list of supported &lt;attr_handle&gt;s), (list of supported &lt;need_confirm&gt;s), "&lt;ind_value&gt;"</b>  <b>OK</b>
<b>Write Command</b> <b>AT+QBTGATSIND="&lt;gserv_id&gt;",&lt;conn_id&gt;,&lt;attr_handle&gt;,&lt;need_confirm&gt;,&lt;ind_value&gt;"</b>	<b>Response:</b> <b>+QBTGATSIND: &lt;result&gt;,&lt;gserv_id&gt;,&lt;conn_id&gt;,&lt;value_handle&gt;"</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill the '0' at the end of it to even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;attr_handle&gt;</b>	Handle of attribute. The range is 0-65535.
<b>&lt;value_handle&gt;</b>	Handle of this characteristic value. The range is 0-65535.
<b>&lt;ind_value&gt;</b>	The value that needs to be notified (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;need_confirm&gt;</b>	Need client confirmation when sending indication (not when sending notification). 0 NO 1 YES
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

#### Example

```
AT+QBTGATSIND="ABC2",1,258,1,"74ab" //Send an indication to a client.
+QBTGATSIND: 0,"ABC2",1,2
```

OK

### 3.2.7. AT+QBTGATSRSP Send a Response to a Client's Read or Write Operation

#### AT+QBTGATSRSP Send a Response to a Client's Read or Write Operation

Test Command

AT+QBTGATSRSP=?

Response:

**+QBTGATSRSP:** "<gserv\_id>",(list of supported <response>s),(list of supported <conn\_id>s),(list of supported <trans\_id>s),(list of supported <attr\_handle>s),"<rsp\_value>"

OK

Write Command

AT+QBTGATSRSP="<gserv\_id>",<response>,<conn\_id>,<trans\_id>,<attr\_handle>,"<rsp\_value>"

Response:

**+QBTGATSRSP:** <result>,"<gserv\_id>",<conn\_id>,<attr\_handle>

OK

If there is any error relating to ME functionality:

**+CME ERROR:** <errorcode>

#### Parameter

<gserv_id>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<response>	Response result for client request. The range is 0-255. 0: Read request successfully Other values: Read request failed
<conn_id>	ID of current connection. The range is 0-255.
<trans_id>	ID of current transaction. The range is 0-65535.
<attr_handle>	Handle of attribute. The range is 0-65535.
<rsp_value>	The character value that needs to be responded (if the value length is odd, then fill '0' at the end of it to be even).
<result>	0 SUCCESS 1 FAIL

#### Example

**+QBTGATRREQ:** "ABC2",1,452,107602B38034,258,0,0

//Client's read operation.

**AT+QBTGATSRSP="ABC2",0,1,452,258,"adb2"**

//Send response to a client.



```
+QBTGATSRSP: 0,"ABC2",1,258
```

```
OK
```

### 3.2.8. AT+QBTGATADV Set Advertising Parameters

#### AT+QBTGATADV Set Advertising Parameters

Test Command

**AT+QBTGATADV=?**

Response

**+QBTGATADV:** (list of supported <min\_interval>s),(list of supported <max\_interval>s)

**OK**

Write Command

**AT+QBTGATADV=<min\_interval>,<max\_interval>**

Response

**OK**

If there is any error relating to ME functionality:

**+CME ERROR: <errorcode>**

#### Parameter

<min_interval>	Minimum advertising interval for undirected and low duty cycle directed advertising. The range is 32-16384. Default: 256. Unit: 0.625ms.
<max_interval>	Maximum advertising interval for undirected and low duty cycle directed advertising. The range is 32-16384. Default: 512. Unit: 0.625ms.

#### Example

```
AT+QBTGATADV=800,2000
```

```
OK
```

#### NOTE

The actual value of advertising interval used is usually the median of maximum and minimum values.

### 3.2.9. AT+QGATSETADV Set Advertising Data

The command adds the specified AD structure into the advertising data. Without any other special setting, the device name will be automatically included in the advertising information after executing this command.

## AT+QGATSETADV Set Advertising Data

Test Command <b>AT+QGATSETADV=?</b>	Response <b>+QGATSETADV:</b> "<gserv_id>",<appearance>,(list of supported <string_mode>s),"<manufacture_data>",<service_data>",<service_uuid>"  <b>OK</b>
Write Command <b>AT+QGATSETADV</b> <b>= "&lt;gserv_id&gt;",&lt;appearance&gt;,&lt;string_mode&gt;,&lt;manufacture_data&gt;",&lt;service_data&gt;",&lt;service_uuid&gt;"</b>	Response <b>+QGATSETADV: &lt;result&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

### Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end to be even).
<b>&lt;appearance&gt;</b>	The external appearance of the device. The range is 0-65535. For detailed information of this parameter, please visit the website: <a href="https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.characteristic.gap.appearance.xml">https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.characteristic.gap.appearance.xml</a> .
<b>&lt;string_mode&gt;</b>	Format type of parameter <b>&lt;manufacture_data&gt;</b> and <b>&lt;service_data&gt;</b> . 0 Hex string. A hex value string (string should be included in quotation marks). 1 String. A string (string should be included in quotation marks).
<b>&lt;manufacture_data&gt;</b>	Manufacturer information
<b>&lt;service_data&gt;</b>	Service information
<b>&lt;service_uuid&gt;</b>	UUID of this service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

### NOTE

The max length of the advertising data is 31 bytes.

## Example

```
AT+QBTGATSREG=1,"ABC2"
+QBTGATSREG: 1,"ABC2",0
OK
AT+QGATSETADV="ABC2",0,0,"400103","9612","66FE"
+QGATSETADV: 0

OK
AT+QGATSETADV="ABC2",0,1,"ble","test","66FE"
+QGATSETADV: 0

OK
```

### 3.2.10. AT+QGATADVDATA Set Advertising Data

The function of this command is similar to that of the command **AT+QGATSETADV**, but this command can be used to set any AD structure that customers want to add.

AT+QGATADVDATA Set Advertising Data	
Test Command <b>AT+QGATADVDATA=?</b>	Response <b>+QGATADVDATA: "&lt;gserv_id&gt;","&lt;adv_data&gt;"</b>  <b>OK</b>
Write Command <b>AT+QGATADVDATA="&lt;gserv_id&gt;","&lt;adv_data&gt;"</b>	Response <b>OK</b>  If there is any error related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300ms
Reference	

## Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end to be even).
<b>&lt;adv_data&gt;</b>	Raw advertising data string. Hex format.

## NOTES

1. **<adv\_data>** needs to conform to *Bluetooth 4.0 Specification*.
2. The max length of the advertising data is 31 bytes.

### Example

```
AT+QBTPWR=1
OK
AT+QBTGATSREG=1,"ABC2"
+QBTGATSREG: 1,"ABC2",0

OK
AT+QGATADVDATA="ABC2","020106070931323334353606FF4400323169"
OK
```

### 3.2.11. AT+QGATSCANRSP Set Scan Response Data

The command adds the required additional data such as the device name or 128-bit service UUID into scan response data.

#### AT+QGATSCANRSP Set Scan Response Data

Test Command <b>AT+QGATSCANRSP=?</b>	Response <b>+QGATSCANRSP: "&lt;gserv_id&gt;","&lt;rsp_data&gt;"</b>  <b>OK</b>
Write Command <b>AT+QGATSCANRSP="&lt;gserv_id&gt;","&lt;rsp_data&gt;"</b>	Response <b>OK</b>  If there is any error related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300ms
Reference	

### Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end to be even).
<b>&lt;rsp_data&gt;</b>	Raw scan response data string. A hex value string (string should conform to <i>Bluetooth 4.0 Specification</i> ). The max length is 31 bytes.

## Example

```
AT+QGATSCANRSP="ABC2","11069ECADC240EE5A9E093F3A3B50100406E"
OK
```

### 3.2.12. AT+QBTGATCPU Update Connection Parameters

#### AT+QBTGATCPU Update Connection Parameters

Test Command <b>AT+QBTGATCPU=?</b>	Response <b>+QBTGATCPU:</b> <bt_addr>,(list of supported <min_interval>s),(list of supported <max_interval>s),(list of supported <timeout>s),(list of supported <latency>s)  <b>OK</b>
Write Command <b>AT+QBTGATCPU=&lt;bt_addr&gt;,&lt;min_interval&gt;,&lt;max_interval&gt;,&lt;timeout&gt;,&lt;latency&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<bt_addr>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<min_interval>	Minimum value of the connection interval. The range is 6-3200. Default: 10. Unit: 1.25ms.
<max_interval>	Maximum value of the connection interval. The range is 6-3200. Default: 15. Unit: 1.25ms.
<timeout>	Supervision timeout for the connection. The range is 10-3200. Default: 500. Unit: 10ms.
<latency>	Maximum slave latency allowed for the connection specified as the number of connection events. The range is 0-499. Default: 0.

## Example

```
AT+QBTGATCPU=E62CA017A503,288,304,600,4
OK
```

### 3.2.13. AT+QBTGATSDISC Disconnect Server Actively

#### AT+QBTGATSDISC Disconnect Server Actively

Test Command <b>AT+QBTGATSDISC=?</b>	Response <b>+QBTGATSDISC:</b> (list of supported <conn_id>s)  <b>OK</b>
Write Command <b>AT+QBTGATSDISC=&lt;conn_id&gt;</b>	Response <b>OK</b>  <b>+QBTGATSCON:</b> 0,"<gserv_id>",<result>,<bt_addr>,<conn_id>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<conn_id>	ID of current connection. The range is 0-255.
<gserv_id>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<result>	0 SUCCESS 1 FAIL
<bt_addr>	Address of the peer BLE device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)

#### Example

```
AT+QBTGATSDISC=1
OK
```

### 3.2.14. AT+QBTGATSL Start/Stop BLE Advertising

The command is only valid when **AT+QBTVISB=0**. And it is applied to scenarios where BLE is broadcasted and classic BT is not broadcasted.

#### AT+QBTGATSL Start/Stop BLE Advertising

Test Command <b>AT+QBTGATSL=?</b>	Response: <b>+QBTGATSL:</b> "<gserv_id>",(list of supported <op>s)
--------------------------------------	-----------------------------------------------------------------------

	<b>OK</b>
Write Command <b>AT+QBTGATSL="&lt;gserv_id&gt;",&lt;op&gt;</b>	Response: <b>+QBTGATSL: "&lt;gserv_id&gt;",&lt;result&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<b>&lt;op&gt;</b>	0 Stop 1 Start
<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

## Example

```
AT+QBTGATSL="ABC2",1
+QBTGATSL: "ABC2",0
OK
```

### 3.2.15. AT+QBTGATSENLE Start/Stop BLE Advertising

The command can control BLE advertising independently because command **AT+QBTGATSL** is only valid when **AT+QBTVISB=0**. It is applied to scenarios where BLE is not broadcasted and classic BT is broadcasted.

<b>AT+QBTGATSENLE Start/Stop BLE Advertising</b>	
Test Command <b>AT+QBTGATSENLE=?</b>	Response <b>+QBTGATSENLE:</b> (list of supported <op>s)  <b>OK</b>
Write Command <b>AT+QBTGATSENLE=&lt;op&gt;</b>	Response <b>OK</b>  If error is related to ME functionality:

	<b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300ms
Reference	

## Parameter

<b>&lt;op&gt;</b>	0	Stop.
	1	Start.

## Example

```
AT+QBTGATSENLE=1
OK
```

## 3.3. BLE GATT Client AT Commands

### 3.3.1. AT+QBTGATCREG Register/Deregister GATT Client

<b>AT+QBTGATCREG Register/Deregister GATT Client</b>	
Test Command <b>AT+QBTGATCREG=?</b>	Response <b>+QBTGATCREG:</b> (list of supported <op>s), "<gclient_id>"  <b>OK</b>
Write Command <b>AT+QBTGATCREG=&lt;op&gt;,"&lt;gclient_id&gt;"</b>	Response <b>+QBTGATCREG:</b> <op>,<result>,"<gclient_id>"  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<b>&lt;op&gt;</b>	0	Deregister
	1	Register
<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length	



	is odd, then fill '0' at the end of it to be even).	
<b>&lt;result&gt;</b>	0	SUCCESS
	1	FAIL

### Example

```
AT+QBTGATCREG=1,"ABCD"
+QBTGATCREG: 1,0,"ABCD"

OK
```

### 3.3.2. AT+QBTGATCSCAN Start/Stop Scanning LE Device

AT+QBTGATCSCAN Start/Stop Scanning LE Device	
Test Command <b>AT+QBTGATCSCAN=?</b>	Response <b>+QBTGATCSCAN:</b> (list of supported<op>s), "<gclient_id>"  <b>OK</b>
Write Command <b>AT+QBTGATCSCAN=&lt;op&gt;,"&lt;gclient_id&gt;"</b>	Response <b>+QBTGATCSCAN:</b> <op>,<result>,"<gclient_id>"  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

### Parameter

<b>&lt;op&gt;</b>	0 Stop 1 Start
<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

### Example

```
AT+QBTGATCSCAN=1,"ABCD" //Start scanning LE device.
+QBTGATCSCAN: 1,0,"ABCD"
```

```

OK
+QBTGATCSCAN: "ABCD",76DCAB964FEF,40,02011A0B095155454354454C2D4254

+QBTGATCSCAN: "ABCD",FD3B76CE6EF2,53,02011A0B095155454354454C2D4254
AT+QBTGATCSCAN=0,"ABCD"           //Stop scanning LE device.
+QBTGATCSCAN: 0,0,"ABCD"

OK

```

#### NOTE

Execute the command to start scanning. When a device is scanned, the device information is reported to the serial port. If the device scanned is the same device scanned before, even if its advertising information has changed, the device information is still reported to the serial port only once.

### 3.3.3. AT+QBTGATCCON Connect/Disconnect GATT Client to Remote LE Device

AT+QBTGATCCON Connect/Disconnect GATT Client to Remote LE Device	
Test Command <b>AT+QBTGATCCON=?</b>	Response <b>+QBTGATCCON:</b> (list of supported <op>s), "<gclient_id>", <peer_addr>, (list of supported <direct>s)  <b>OK</b>
Read command <b>AT+QBTGATCCON?</b>	Response <b>+QBTGATCCON:</b> <conn_id>, <peer_addr> .....  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command Connect <b>AT+QBTGATCCON=1,"&lt;gclient_id&gt;",&lt;peer_addr&gt;,&lt;direct&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command Disconnect <b>AT+QBTGATCCON=0,"&lt;gclient_id&gt;",&lt;conn_id&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<op>	0	Disconnect
	1	Connect
<gclient_id>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<peer_addr>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)	
<direct>	0	Non-direct connection
	1	Direct connection
<conn_id>	ID of current connection. The range is 0-255.	

## Example

```
AT+QBTGATCCON=1,"ABCD",2711C92E38C6,1
OK
```

```
+QBTGATCCON: 1,0,"ABCD",2711C92E38C6,1
AT+QBTGATCCON=0,"ABCD",1
OK
```

```
+QBTGATCCON: 0,255,"ABCD",2711C92E38C6,1
```

### 3.3.4. AT+QBTGATCSS Search and Enumerate Peer's Service

The command is used to search and enumerate all the GATT services on the connected device. If the connection parameters in command **AT+QBTGATCPU** are long, the return time will be long correspondingly.

#### AT+QBTGATCSS Search and Enumerate Peer's Service

Test Command <b>AT+QBTGATCSS=?</b>	Response <b>+QBTGATCSS: "&lt;gclient_id&gt;",&lt;conn_id&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATCSS="&lt;gclient_id&gt;",&lt;conn_id&gt;</b>	Response <b>OK</b>  <b>+QBTGATCSS: &lt;result&gt;,&lt;gclient_id&gt;,&lt;conn_id&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;is_primary&gt;</b>  .....

If there is any error relating to ME functionality:

**+CME ERROR: <errorcode>**

## Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;service_uuid&gt;</b>	UUID of the service.
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;is_primary&gt;</b>	Service is the primary one or not. 0 Not primary service 1 Primary service
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

## Example

```
AT+QBTGATCSS="ABCD",1
OK

+QBTGATCSS: 0,"ABCD",1,"0018",0,1

+QBTGATCSS: 0,"ABCD",1,"0118",0,1

+QBTGATCSS: 0,"ABCD",1,"3418",0,1
```

### 3.3.5. AT+QBTGATCGC Search and Enumerate Peer's Characteristic of the Service

The command is used to search and enumerate all the peer's characteristics of a given service. If the characteristics info is set to NULL, then customers can only get the first characteristic.

#### AT+QBTGATCGC Search and Enumerate Peer's Characteristic of the Service

Test Command <b>AT+QBTGATCGC=?</b>	Response <b>+QBTGATCGC: "&lt;gclient_id&gt;",&lt;conn_id&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;[,&lt;char_uuid&gt;,&lt;char_inst&gt;]</b>  <b>OK</b>
Write Command <b>AT+QBTGATCGC="&lt;gclient_id&gt;",&lt;co</b>	Response <b>OK</b>

nn\_id>,"<service\_uuid>",<service\_inst>,<service\_is\_primary>[,"<char\_uuid>",<char\_inst>]

+QBTGATCGC: <result>,"<gclient\_id>",<conn\_id>,"<service\_uuid>",<service\_inst>,<service\_is\_primary>[,"<char\_uuid>",<char\_inst>,<prop>]

If there is any error relating to ME functionality:

+CME ERROR: <errorcode>

## Parameter

<gclient_id>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<result>	0 SUCCESS 1 FAIL
<conn_id>	ID of current connection. The range is 0-255.
<service_uuid>	UUID of the service.
<service_inst>	Instance ID of this service. The range is 0-255.
<service_is_primary>	Service is the primary one or not. 0 Not primary service 1 Primary service
<char_uuid>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<char_inst>	Instance ID of this characteristic. The range is 0-255.
<prop>	The properties of the characteristic. For more details, please refer to <b>Table 2</b> .

## Example

```
AT+QBTGATCGC="ABCD",1,"3418",0,1 //Enumerate peer's first characteristic of the service.
OK

+QBTGATCGC: 0,"ABCD",1,"3418",0,1,"C001",0,58
AT+QBTGATCGC="ABCD",1,"3418",0,1,"C001",0 //Enumerate peer's second characteristic of the
service.
OK

+QBTGATCGC: 0,"ABCD",1,"3418",0,1,"C002",0,58
```

### 3.3.6. AT+QBTGATCGD Search and Enumerate Peer's Descriptor of the Service with UUID

The command is used to search and enumerate all the descriptors for a given characteristic. If the descriptors info is set to NULL, then only the first descriptor will be got.

AT+QBTGATCGD Search and Enumerate Peer's Descriptor of the Service with UUID	
Test Command <b>AT+QBTGATCGD=?</b>	Response <b>+QBTGATCGD: "&lt;gclient_id&gt;",&lt;conn_id&gt;,"&lt;service_uuid&gt;",&lt;service_inst&gt;,&lt;service_is_primary&gt;,"&lt;char_uuid&gt;",&lt;char_inst&gt;["&lt;desc_uuid&gt;",&lt;desc_inst&gt;"]</b>  <b>OK</b>
Write Command <b>AT+QBTGATCGD="&lt;gclient_id&gt;",&lt;conn_id&gt;,"&lt;service_uuid&gt;",&lt;service_inst&gt;,&lt;service_is_primary&gt;,"&lt;char_uuid&gt;",&lt;char_inst&gt;["&lt;desc_uuid&gt;",&lt;desc_inst&gt;"]</b>	Response <b>OK</b>  <b>+QBTGATCGD: &lt;result&gt;,&lt;gclient_id&gt;,&lt;conn_id&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,&lt;char_uuid&gt;,&lt;char_inst&gt;["&lt;desc_uuid&gt;",&lt;desc_inst&gt;"]</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;service_uuid&gt;</b>	UUID of the service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;service_is_primary&gt;</b>	Service is the primary one or not. 0 Not primary service 1 Primary service
<b>&lt;char_uuid&gt;</b>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;char_inst&gt;</b>	Instance ID of this characteristic. The range is 0-255.
<b>&lt;desc_uuid&gt;</b>	UUID of the descriptor. A string with hex value (string should be included in

	quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<desc_inst>	Instance ID of this descriptor. The range is 0-255.
<result>	0 SUCCESS 1 FAIL

### Example

```
AT+QBTGATCGD="ABCD",1,"3418",0,1,"332A",0
OK
```

```
+QBTGATCGD: 0,"ABCD",1,"3418",0,1,"332A",0,"1329",0
```

### 3.3.7. AT+QBTGATCRC Read Peer's Characteristic of the Service with UUID

AT+QBTGATCRC Read Peer's Characteristic of the Service with UUID	
Test Command AT+QBTGATCRC=?	Response +QBTGATCRC: "<gclient_id>",<conn_id>",<service_uuid>",<service_inst>,<service_is_primary>",<char_uuid>",<char_inst>,<auth_req>  OK
Write Command AT+QBTGATCRC="<gclient_id>",<conn_id>",<service_uuid>",<service_inst>,<service_is_primary>",<char_uuid>",<char_inst>,<auth_req>	Response OK  +QBTGATCRC: <result>,<gclient_id>,<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>,<value>  If there is any error relating to ME functionality: +CME ERROR: <errorcode>

### Parameter

<gclient_id>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<conn_id>	ID of current connection. The range is 0-255.
<service_uuid>	UUID of the service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).

<service_inst>	Instance ID of the service UUID.
<service_is_primary>	Service is the primary one or not. 0 Not primary service 1 Primary service
<char_uuid>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<char_inst>	Instance ID of this characteristic. The range is 0-255.
<auth_req>	Authentication requirement for reading characteristic. 0 auth_req_NONE 1 auth_req_NO_MITM 2 auth_req_MITM 3 auth_req_SIGNED_NO_MITM 4 auth_req_MITM<value>
<value>	The value of the characteristic
<result>	0 SUCCESS 1 FAIL

### Example

```
AT+QBTGATCRC="ABCD",1,"3418",0,1,"332A",0,0 //Wait for server response.
OK
+QBTGATCRC: 0,"ABCD",1,"3418",0,1,"332A",0,"ADB2"
```

### 3.3.8. AT+QBTGATCWC Write Peer's Characteristic of the Service with UUID

AT+QBTGATCWC Write Peer's Characteristic of the Service with UUID	
Test Command AT+QBTGATCWC=?	Response +QBTGATCWC: "<gclient_id>",<conn_id>,<write_type>, "<service_uuid>",<service_inst>,<service_is_primary>, "<char_uuid>",<char_inst>",<value>",<auth_req>  OK
Write Command AT+QBTGATCWC="<gclient_id>",<c onn_id>,<write_type>",<service_uuid >",<service_inst>,<service_is_primar y>",<char_uuid>",<char_inst>",<valu e>",<auth_req>	Response OK  +QBTGATCWC: <result>,<gclient_id>,<conn_id>,<se rvice_uuid>,<service_inst>,<service_is_primary>,<cha r_uuid>,<char_inst>  If there is any error relating to ME functionality:



**+CME ERROR: <errorcode>**

## Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;write_type&gt;</b>	Type of sending a write request to peer device. 1 Not need the response of peer device to confirm the receiving of write request 2 Need the response of peer device to confirm the receiving of write request 3 Prepare to write the request
<b>&lt;service_uuid&gt;</b>	UUID of the service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;service_is_primary&gt;</b>	Service is the primary one or not. 0 Not primary service 1 Primary service
<b>&lt;char_uuid&gt;</b>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;char_inst&gt;</b>	Instance ID of this characteristic. The range is 0-255.
<b>&lt;auth_req&gt;</b>	Authentication requirement for reading characteristic. 0 auth_req_NONE 1 auth_req_NO_MITM 2 auth_req_MITM 3 auth_req_SIGNED_NO_MITM 4 auth_req_MITM<value>
<b>&lt;value&gt;</b>	The value of the characteristic
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

## Example

```
AT+QBTGATCWC="ABCD",1,1,"3418",0,1,"332A",0,"3435",0
OK

+QBTGATCWC: 0,"ABCD",1,"3418",0,1,"332A",0
```

### 3.3.9. AT+QBTGATCRD Read Peer's Descriptor of the Service with UUID

AT+QBTGATCRD Read Peer's Descriptor of the Service with UUID	
<b>Test Command</b> <b>AT+QBTGATCRD=?</b>	<b>Response</b> <b>+QBTGATCRD: "&lt;gclient_id&gt;",&lt;conn_id&gt;,"&lt;service_uuid&gt;",&lt;service_inst&gt;,&lt;service_is_primary&gt;,"&lt;char_uuid&gt;",&lt;char_inst&gt;,"&lt;desc_uuid&gt;",&lt;desc_inst&gt;,&lt;auth_req&gt;</b>  <b>OK</b>
<b>Write Command</b> <b>AT+QBTGATCRD="&lt;gclient_id&gt;",&lt;conn_id&gt;,"&lt;service_uuid&gt;",&lt;service_inst&gt;,&lt;service_is_primary&gt;,"&lt;char_uuid&gt;",&lt;char_inst&gt;,"&lt;desc_uuid&gt;",&lt;desc_inst&gt;,&lt;auth_req&gt;</b>	<b>Response</b> <b>OK</b>  <b>+QBTGATCRD: &lt;result&gt;,&lt;gclient_id&gt;,&lt;conn_id&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,&lt;char_uuid&gt;,&lt;char_inst&gt;,&lt;desc_uuid&gt;,&lt;desc_inst&gt;,&lt;value&gt;</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;service_uuid&gt;</b>	UUID of the service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;service_is_primary&gt;</b>	Service is the primary one or not. 0 Not primary service 1 Primary service
<b>&lt;char_uuid&gt;</b>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;char_inst&gt;</b>	Instance ID of this characteristic. The range is 0-255.
<b>&lt;desc_uuid&gt;</b>	UUID of the descriptor. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;desc_inst&gt;</b>	Instance ID of this descriptor. The range is 0-255.

<b>&lt;auth_req&gt;</b>	Authentication requirement for reading characteristic.
0	auth_req_NONE
1	auth_req_NO_MITM
2	auth_req_MITM
3	auth_req_SIGNED_NO_MITM
4	auth_req_MITM<value>
<b>&lt;value&gt;</b>	The value of the descriptor.
<b>&lt;result&gt;</b>	0 SUCCESS
1	FAIL

### Example

```
AT+QBTGATCRD="ABCD",1,"3418",0,1,"332A",0,"1329",0,0 //Wait for server's response.
OK

+QBTGATCRD: 0,"ABCD",1,"3418",0,1,"332A",0,"1329",0,1234
```

### 3.3.10. AT+QBTGATCWD Write Peer's Descriptor of the Service with UUID

#### AT+QBTGATCWD Write Peer's Descriptor of the Service with UUID

Test Command <b>AT+QBTGATCWD=?</b>	Response <b>+QBTGATCWD: "&lt;gclient_id&gt;",&lt;conn_id&gt;,&lt;write_type&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,&lt;char_uuid&gt;,&lt;char_inst&gt;,&lt;desc_uuid&gt;,&lt;desc_inst&gt;,&lt;value&gt;,&lt;auth_req&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATCWD="&lt;gclient_id&gt;",&lt;conn_id&gt;,&lt;write_type&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,&lt;char_uuid&gt;,&lt;char_inst&gt;,&lt;desc_uuid&gt;,&lt;desc_inst&gt;,&lt;value&gt;,&lt;auth_req&gt;</b>	Response <b>OK</b>  <b>+QBTGATCWD: &lt;result&gt;,&lt;gclient_id&gt;,&lt;conn_id&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,&lt;char_uuid&gt;,&lt;char_inst&gt;,&lt;desc_uuid&gt;,&lt;desc_inst&gt;</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

### Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the
---------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;write_type&gt;</b>	Type of sending a write request to peer device. 1 Not need the response of peer device to confirm the receiving of write request 2 Need the response of peer device to confirm the receiving of write request 3 Prepare to write the request
<b>&lt;service_uuid&gt;</b>	UUID of the service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;service_is_primary&gt;</b>	Service is the primary one or not. 0 Not primary service 1 Primary service
<b>&lt;char_uuid&gt;</b>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;char_inst&gt;</b>	Instance ID of this characteristic. The range is 0-255.
<b>&lt;desc_uuid&gt;</b>	UUID of the descriptor. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;desc_inst&gt;</b>	Instance ID of this descriptor. The range is 0-255.
<b>&lt;auth_req&gt;</b>	Authentication requirement for reading characteristic. 0 auth_req_NONE 1 auth_req_NO_MITM 2 auth_req_MITM 3 auth_req_SIGNED_NO_MITM 4 auth_req_MITM<value>
<b>&lt;value&gt;</b>	The value of the characteristic.
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

## Example

```
AT+QBTGATCWD="ABCD",1,1,"3418",0,1,"332A",0,"1329",0,"4F4B",0
OK

+QBTGATCWD: 0,"ABCD",1,"3418",0,1,"332A",0,"1329",0
```

### 3.3.11. AT+QBTGATCEW Execute Write Command

AT+QBTGATCEW Execute Write Command	
Test Command AT+QBTGATCEW=?	Response +QBTGATCEW: "<gclient_id>",<conn_id>,<execute>  OK
Write Command AT+QBTGATCEW="<gclient_id>",<conn_id>,<execute>	Response +QBTGATCEW: <result>,<gclient_id>,<conn_id>  OK  If there is any error relating to ME functionality: +CME ERROR: <errorcode>

#### Parameter

<gclient_id>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<conn_id>	ID of current connection. The range is 0-255.
<execute>	0 Cancel execution 1 Execute
<result>	0 SUCCESS 1 FAIL

#### Example

```
AT+QBTGATCEW="ABCD",1,1
+QBTGATCEW: 0,"ABCD",1
OK
```

### 3.3.12. AT+QBTGATCRN Register/Deregister Notification Request

AT+QBTGATCRN Register/Deregister Notification Request	
Test Command AT+QBTGATCRN=?	Response +QBTGATCRN: <op>,<gclient_id>,<peer_addr>,<service_uuid>,<service_inst>,<service_is_primary>,<char_uuid>,<char_inst>

	<b>OK</b>
Write Command <b>AT+QBTGATCRN=&lt;op&gt;,"&lt;gclient_id&gt; &lt;peer_addr&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,"&lt;char_uuid&gt;",&lt;char_inst&gt;</b>	Response <b>+QBTGATCRN: &lt;op&gt;,&lt;result&gt;,"&lt;gclient_id&gt;",&lt;peer_addr&gt;,&lt;service_uuid&gt;,&lt;service_inst&gt;,&lt;service_is_primary&gt;,"&lt;char_uuid&gt;",&lt;char_inst&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<b>&lt;op&gt;</b>	0      Deregister 1      Register
<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;peer_addr&gt;</b>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<b>&lt;service_uuid&gt;</b>	UUID of the service. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;service_is_primary&gt;</b>	Service is the primary one or not. 0      Not primary service 1      Primary service
<b>&lt;char_uuid&gt;</b>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;char_inst&gt;</b>	Instance ID of this characteristic. The range is 0-255.
<b>&lt;result&gt;</b>	0      SUCCESS 1      FAIL

## Example

```
AT+QBTGATCRN=1,"ABCD",2711C92E38C6,"3418",0,1,"332A",0
+QBTGATCRN: 1,0,"ABCD",2711C92E38C6,"3418",0,1,"332A",0
```

**OK** //Server sends indication/notification to client.

```
+QBTGATCN: "ABCD",1,2711C92E38C6,"3418",0,1,"332A",0,0,"74AB"
```

### 3.3.13. AT+QBTGATCGDT Get the Type of Peer Device

The command is used to get the type of the peer device, such as BLE, BR/EDR or dual-mode.

AT+QBTGATCGDT Get the Type of Peer Device	
Test Command <b>AT+QBTGATCGDT=?</b>	Response <b>+QBTGATCGDT: "&lt;gclient_id&gt;",&lt;peer_addr&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATCGDT="&lt;gclient_id&gt;",&lt;peer_addr&gt;</b>	Response <b>+QBTGATCGDT: &lt;result&gt;,"&lt;gclient_id&gt;",&lt;peer_addr&gt;,&lt;dev_type&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;peer_addr&gt;</b>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<b>&lt;dev_type&gt;</b>	0 Unknown 1 BLE 2 BR/EDR 3 Dual-mode
<b>&lt;result&gt;</b>	0 SUCCESS 1 FAIL

#### Example

```
AT+QBTGATCGDT="ABCD",2711C92E38C6
+QBTGATCGDT: 0,"ABCD",2711C92E38C6,1
OK
```

### 3.3.14. AT+QBTGATCL Start/Stop Listen Request

AT+QBTGATCL Start/Stop Listen Request	
Test Command <b>AT+QBTGATCL=?</b>	Response <b>+QBTGATCL: "&lt;gclient_id&gt;",&lt;start&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATCL="&lt;gclient_id&gt;",&lt;start&gt;</b> <b>&gt;</b>	Response <b>+QBTGATCL: "&lt;gclient_id&gt;",&lt;result&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<start>	0	Stop
	1	Start
<gclient_id>	User ID (or the name) of GATT client.	
	A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<result>	0	SUCCESS
	1	FAIL

#### Example

```
AT+QBTGATCL="ABCD",1
+QBTGATCL: "ABCD",0
OK
```

### 3.3.15. AT+QGATCFGSCAN Set Scan Interval and Scan Window

The command sets the scan interval and scan window.

AT+QGATCFGSCAN Set Scan Interval and Scan Window	
Test Command <b>AT+QGATCFGSCAN=?</b>	Response <b>+QGATCFGSCAN:</b> (list of supported <scan_interval>s),(list of supported <scan_window>s)



	OK
Read Command <b>AT+QGATCFGSCAN?</b>	Response <b>+QGATCFGSCAN: &lt;scan_interval&gt;,&lt;scan_window&gt;</b>
	OK
Write Command <b>AT+QGATCFGSCAN=&lt;scan_interval&gt;,&lt;scan_window&gt;</b>	Response <b>OK</b>  If there is any error related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300ms
Reference	

### Parameter

<b>&lt;scan_interval&gt;</b>	The interval time of each scan. The range is 4-16384. Default: 320. Unit: 0.625ms.
<b>&lt;scan_window&gt;</b>	The duration of each scan. The range is 4-16384. Default: 160. Unit: 0.625ms.

### NOTE

Scan interval is the time difference between the start times of two consecutive scan windows, including the time of the scanning and the scanning break.

### Example

```

AT+QGATCFGSCAN?
+QGATCFGSCAN: 320,160

OK
AT+QGATCFGSCAN=?
+QGATCFGSCAN: <scan_interval>,<scan_window>

OK
AT+QGATCFGSCAN=320,200
OK
  
```

## 4 Description of URCs

### 4.1. Notify the GATT Connection or Disconnection

#### Notify the GATT Connection or Disconnection

```
+QBTGATSCON: <op>,"<gserv_id>",<result>,<bt_addr>,<conn_id>
```

#### Parameter

<op>	0	Disconnect
	1	Connect
<gserv_id>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<bt_addr>	Address of the peer BLE device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)	
<result>	0	SUCCESS
	1	FAIL
<conn_id>	ID of current connection. The range is 0-255.	

#### Example

```
+QBTGATSCON: 1,"ABC2",0,CB2CD7923F46,1
```

### 4.2. Notify the GATT Client's Write Request

#### Notify the GATT Client's Write Request

```
+QBTGATWREQ: "<gserv_id>",<conn_id>,<trans_id>,<bt_addr>,<attr_handle>,<value>,<need_rsp>,<is_prep>
```

are>,<offset>

## Parameter

<gserv_id>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<conn_id>	ID of current connection. The range is 0-255.
<trans_id>	ID of current transaction. The range is 0-255.
<bt_addr>	Address of the peer BLE device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<attr_handle>	Handle of attribute. The range is: 0-65535.
<value>	The value that needs to be written. Hex format.
<need_rsp>	Whether the client needs server's response. 1 YES 0 NO
<is_prepare>	Whether the server executes request immediately. 0 NO 1 YES
<offset>	Offset of the request. The range is 0-65535.

## Example

```
+QBTGATWREQ: "ABC2",1,19,CB2CD7923F46,259,37383336,1,0,0
AT+QBTGATSRSP="ABC2",0,1,19,259,"adb2"
+QBTGATSRSP: 0,"ABC2",1,259
OK
```

## 4.3. Notify the GATT Client's Write Request Execution

### Notify the GATT Client's Write Request Execution

```
+QBTGATEWREQ: "<gserv_id>",<conn_id>,<trans_id>,<bt_addr>,<cancel>
```

## Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;trans_id&gt;</b>	ID of current transaction. The range is 0-255.
<b>&lt;bt_addr&gt;</b>	Address of the peer BLE device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<b>&lt;cancel&gt;</b>	0 Cancel execution 1 Execution

## Example

```
+QBTGATWREQ:
"ABC2",1,36,205CB121455D,258,313233343536373839303132333435363738,1,1,0
AT+QBTGATSRSP="ABC2",0,1,36,258,"00"
+QBTGATSRSP: 0,"ABC2",1,258

OK

+QBTGATWREQ: "ABC2",1,37,205CB121455D,258,393031323334353637383930313233343536,1,1,
18
AT+QBTGATSRSP="ABC2",0,1,37,258,"00"
+QBTGATSRSP: 0,"ABC2",1,258

OK

+QBTGATWREQ: "ABC2",1,38,205CB121455D,258,3738393031323334353637383930,1,1,36
AT+QBTGATSRSP="ABC2",0,1,38,258,"00"
+QBTGATSRSP: 0,"ABC2",1,258

OK

+QBTGATEWREQ: "ABC2",1,39,205CB121455D,0
AT+QBTGATSRSP="ABC2",0,1,39,258,"00"
+QBTGATSRSP: 0,"ABC2",1,258

OK
```

## 4.4. Notify the GATT Client's Read Request

### Notify the GATT Client's Read Request

```
+QBTGATRREQ: "<gserv_id>",<conn_id>,<trans_id>,<bt_addr>,<attr_handle>,<is_long>,<offset>
```

#### Parameter

<b>&lt;gserv_id&gt;</b>	User ID (or the name) of GATT server. A hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;trans_id&gt;</b>	ID of current transaction. The range is 0-255.
<b>&lt;bt_addr&gt;</b>	Address of the peer BLE device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<b>&lt;attr_handle&gt;</b>	Handle of attribute. The range is 0-65535.
<b>&lt;is_long&gt;</b>	Notify server the order number of the request (0 means the first request)
<b>&lt;offset&gt;</b>	Offset of the request. The range is 0-65535.

#### Example

```
+QBTGATRREQ: "ABC2",1,18,CB2CD7923F46,259,0,0
AT+QBTGATSRSP="ABC2",0,1,18,259,"adb2"
+QBTGATSRSP: 0,"ABC2",1,259
```

OK

## 4.5. Report Scanning Result

### Report Scanning Result

```
+QBTGATCSCAN: "<gclient_id>",<peer_addr>,<RSSI>,<EIR>
```

## Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;peer_addr&gt;</b>	Address of the discovered BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<b>&lt;RSSI&gt;</b>	Signal strength. The range is 0-255, which means the signal strength is -127dbm~127dbm.
<b>&lt;EIR&gt;</b>	Raw EIR data string. Hex format.

## Example

```
AT+QBTGATCSCAN=1,"ABCD"
+QBTGATCSCAN: 1,0,"ABCD"
```

OK

```
+QBTGATCSCAN: "ABCD",76DCAB964FEF,40,02011A0B095155454354454C2D4254
```

```
+QBTGATCSCAN: "ABCD",FD3B76CE6EF2,53,02011A0B095155454354454C2D4254
```

## 4.6. Notify Client a Connection's Status

### Notify Client a Connection's Status

```
+QBTGATCCON: <op>,<result>,"<gclient_id>",<peer_addr>,<conn_id>
```

## Parameter

<b>&lt;op&gt;</b>	0	Disconnect
	1	Connect
<b>&lt;result&gt;</b>	0	SUCCESS
	1	FAIL
<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<b>&lt;peer_addr&gt;</b>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)	

**<conn\_id>** ID of current connection. The range is 0-255.

### Example

```
AT+QBTGATCCON=1,"ABCD",2711C92E38C6,1
OK

+QBTGATCCON: 1,0,"ABCD",2711C92E38C6,1
AT+QBTGATCCON=0,"ABCD",1
OK

+QBTGATCCON: 0,255,"ABCD",2711C92E38C6,1
```

## 4.7. Notify Client a Notification or Indication Comes

### Notify Client a Notification or Indication Comes

```
+QBTGATCN: "<gclient_id>",<conn_
id>,<peer_addr>,<service_uuid>,<ser
vice_inst>,<service_is_primary>,<c
har_uuid>",<char_inst>,<is_notify>,<
value>
```

### Parameter

<b>&lt;gclient_id&gt;</b>	User ID (or the name) of GATT client. A hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;conn_id&gt;</b>	ID of current connection. The range is 0-255.
<b>&lt;peer_addr&gt;</b>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<b>&lt;service_uuid&gt;</b>	UUID of the service.
<b>&lt;service_inst&gt;</b>	Instance ID of the service UUID.
<b>&lt;service_is_primary&gt;</b>	Service is the primary one or not. 0 Not primary service 1 Primary service
<b>&lt;char_uuid&gt;</b>	UUID of the characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<b>&lt;char_inst&gt;</b>	Instance ID of this characteristic. The range is 0-255.

---

<is_notify>	0	Indication
	1	Notification
<value>	The value of notification or indication.	

---

### Example

```
AT+QBTGATCRN=1,"ABCD",2711C92E38C6,"3418",0,1,"332A",0
+QBTGATCRN: 1,0,"ABCD",2711C92E38C6,"3418",0,1,"332A",0

OK                                     //Server sends indication.

+QBTGATCN: "ABCD",1,2711C92E38C6,"3418",0,1,"332A",0,0,"74AB"
```



# 5 Appendix A References

**Table 1: Terms and Abbreviations**

Abbreviation	Description
BLE	Bluetooth Low Energy
BR	Basic Rate
BT	Bluetooth
DTE	Data Terminal Equipment
EDR	Enhanced Data Rate
FMP	Find Me Profile
GATT	Generic Attribute Profile
GSM	Global System for Mobile Communication
ID	Identification
ME	Mobile Equipment
PXP	Proximity Profile
UE	User Equipment
UUID	Universally Unique Identifier

**Table 2: Format Map of Properties and Permission**

Properties	Format Map
Default	0
Broadcast	1

Read	2
Write without response	4
Write	8
Notify	16
Indicate	32
Signed write	64
Extended properties	128
Permission	Format Map
Read	1
Read with encrypted protection	2
Read with MITM protection	4
Write	16
Write with encrypted protection	32
Write with MITM protection	64
Signed write	128
Signed write with MITM protection	256

**Table 3: Different Coding Schemes of +CME ERROR: <errorcode>**

Code of <errorcode>	Meaning
4	Operation not supported
8300	GATT server register failed
8301	GATT server deregister failed
8302	GATT add service failed
8303	GATT remove service failed
8304	GATT add include service failed

---

8305	GATT add characteristic failed
8306	GATT add descriptor failed
8307	GATT service start failed
8308	GATT service stop failed
8309	GATT indication to client failed
8310	GATT send response to client failed
8311	GATT start listening failed
8312	GATT stop listening failed
8320	GATT BLE adv param too long
8321	GATT client register failed
8322	GATT client deregister failed
8323	GATT client connect failed
8324	GATT client disconnect failed

---