QUECTEL®
Build a Smarter World

MC60 BLE

AT Commands Manual

GSM/GPRS/GNSS Module Series

Rev. MC60_BLE_AT_Commands_Manual_V1.1

Date: 2017-12-21

Status: Released

www.quectel.com

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

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://quectel.com/support/technical.htm
Or email to: 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

About the Document

History

Revision	Date	Author	Description
1.0	2017-05-17	Miles MA	Initial
1.1	2017-12-21	Miles MA	1. Added AT+QGATSETADV 2. Added AT commands of BLE GATT client 3. Added URC +QBTGATCSCAN:, +QBTGATCCON: and +QBTGATCN: 4. Added BLE operation process (Chapter 2) 5. Deleted AT commands AT+QBTFMPSREG and AT+QBTPXPSREG

Contents

Table Index

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 of **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 of **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.

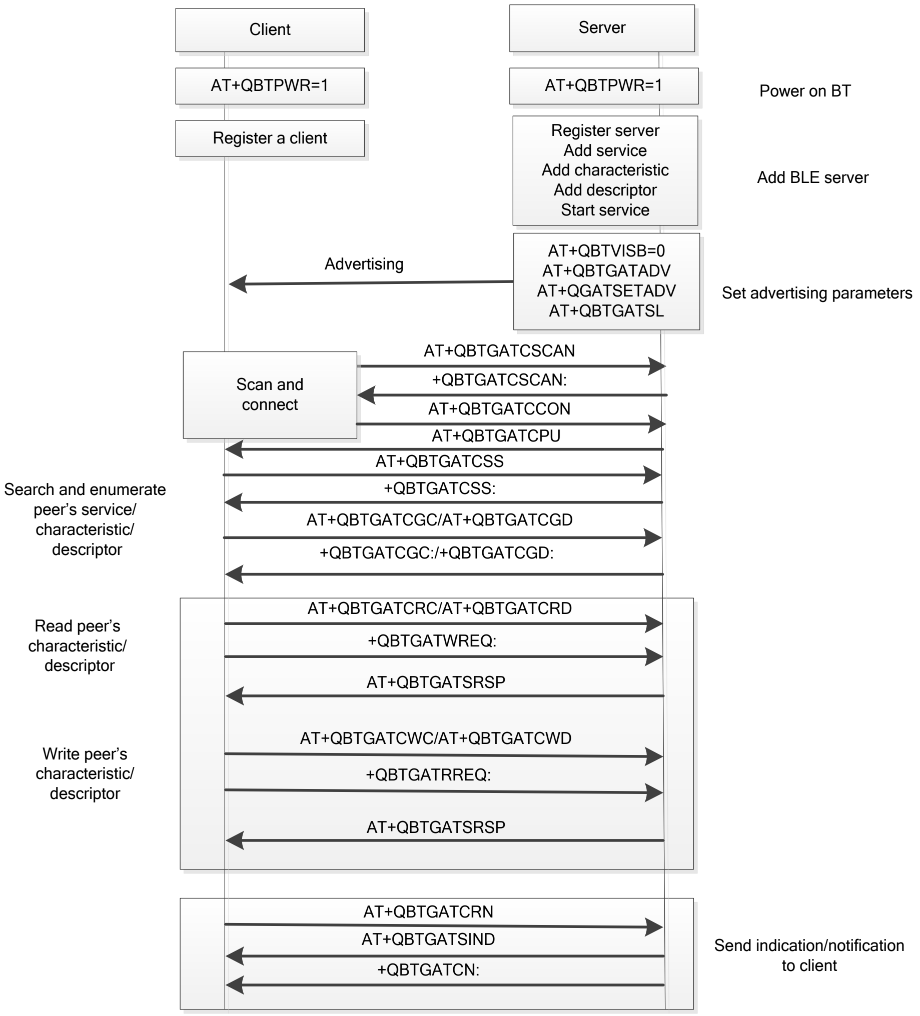

Figure 1: BLE Operation Flow Chat

2.1. BLE Advertising

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

```
AT+QBTPWR=1                          //Power on BT device.
OK
AT+QBTVISB=0                         //Set BT to be invisible, and stop EDR 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
OK
AT+QBTGATSREG=1,"A001"                           //Register a GATT server.
+QBTGATSREG: 1,"A001",0

OK
AT+QBTGATSS=1,"A001","1234",50,1,254             //Add a service.
+QBTGATSS: 1,"A001","1234",1,0,254,256

OK
AT+QBTGATSC=1,"A001",256,"C001",2,58,17          //Add a characteristic.
+QBTGATSC: 1,"A001",256,"C001",2,0,258

OK
AT+QBTGATSD=1,"A001",256,"D001",1,17             //Add a descriptor.
+QBTGATSD: 1,"A001",256,"D001",1,0,261
```

```
OK
AT+QBTGATSST=1,"A001",256,0                  //Start a service.
+QBTGATSST: 1,"A001",0,256

OK

+QBTGATSCON: 1,"A001",0, 0C8ACE3D8734,1           //URC: establish a connection.

+QBTGATRREQ: "A001",1,15,0C8ACE3D8734,258,0, 0 //URC: receive GATT client's Read Request.
AT+QBTGATSRSP="A001",0,1,15,258,"0000"          //Send a response to a client's read operation.
+QBTGATSRSP: 0,"A001",1,258

OK

+QBTGATWREQ:  "A001",1,16,0C8ACE3D8734,258,1234,0,0,0  //URC:  receive  GATT  client's  Write
                                                         Request  that  does  not  need  to
                                                         response.

+QBTGATRREQ: "A001",1,17,0C8ACE3D8734,260,0,0 //URC: receive GATT client's Read Request.
AT+QBTGATSRSP="A001",0,1,17,261,"2222"          //Send a response to a client's read operation.
+QBTGATSRSP: 0,"A001",1,260

OK

+QBTGATWREQ:  "A001",1,18,0C8ACE3D8734,260,AB01,0,0,0  //URC:  receive  GATT  client's  Write
                                                         Request.

AT+QBTGATSIND="A001",1,258,1,"5678"                     //Send an indication or notification to
                                                          a client.
+QBTGATSIND: 0,"A001",1,2

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

+QBTGATSCON: 0,"A001",0,0C8ACE3D8734,1
```

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",0,58
AT+QBTGATCRC="B001",1,"1234",0,1,"C001",0,0      //Read peer’s characteristic of the service.
OK

+QBTGATCRC: 0,"B001",1,"1234",0,1,"C001",0,"0000"
AT+QBTGATCWC="B001",1,1,"1234",0,1,"C001",0,"1234",0      //Write peer’s characteristic.
OK

+QBTGATCWC: 0,"B001",1,"1234",0,1,"C001",0
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,"D001",0
AT+QBTGATCRD="B001",1,"1234",0,1,"C001",0,"D001",0,0      //Read peer’s descriptor of the service
                                                            with UUID.
OK
```

```
+QBTGATCRD: 0,"B001",1,"1234",0,1,"C001",0,"D001",0,"2222"
AT+QBTGATCWD="B001",1,1,"1234",0,1,"C001",0,"D001",0,"AB01",0   //Write peer’s descriptor of
                                                                  the service with UUID.
OK

+QBTGATCWD: 0,"B001",1,"1234",0,1,"C001",0,"D001",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

+QBTGATCN: "B001",1,763CCB96F5DF,"1234",0,1,"C001",0,0,"5678"  //Receive a notification from
                                                                server.
```

3 Description of BLE AT Commands

3.1. General BLE AT Commands

3.1.1. AT+QBTPWR Power on/off BT

AT+QBTPWR Power on/off BT	
Test Command **AT+QBTPWR=?**	Response: **+QBTPWR:** (list of supported **<on_off>**s) **OK**
Read Command **AT+QBTPWR?**	Response: **+QBTPWR: <power status>** **OK**
Write Command **AT+QBTPWR=<on_off>**	Response: **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<on_off>	0	Off
	1	On
<power status>	0	Power off
	1	Power on

Example

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

3.1.2. AT+QBTLEADDR Read Local Address of BLE Device

AT+QBTLEADDR Read Local Address of BLE Device	
Test Command **AT+QBTLEADDR=?**	Response **OK**
Read Command **AT+QBTLEADDR?**	Response **+QBTLEADDR: <ble_addr>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<ble_addr>	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 **AT+QBTLETXPWR=?**	Response **+QBTLETXPWR:** (list of supported **<tx_level>s**) **OK**
Read Command **AT+QBTLETXPWR?**	Response **+QBTLETXPWR: <tx_level>** **OK**
Write Command **AT+QBTLETXPWR=<tx_level>**	Response **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<tx_level>	The level of BLE transmission power. The range is 0-7.

NOTES

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

Example

```
AT+QBTLETXPWR=2
OK
AT+QBTLETXPWR?
+QBTLETXPWR: 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 **AT+QBTNAME=?**	Response **OK** **ERROR**
Read Command **AT+QBTNAME?**	Response **+QBTNAME: <device name>** **OK**
Write Command **AT+QBTNAME=<device name>**	Response **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

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

AT+QBTVISB Set the Current Visibility Mode of BT	
Test Command **AT+QBTVISB=?**	Response **+QBTVISB:** (list of supported **<n>**s)**,**(list of supported **<time>**s) **OK**
Read Command **AT+QBTVISB?**	Response **+QBTVISB: <n>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**
Write Command **AT+QBTVISB=<n>[,<time>]**	Response **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>** If **<n>**=2, response: **OK** **+QBTVISB: 0**

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
AT+QBTVISB=1
OK
AT+QBTVISB=2,60
OK

+QBTVISB: 0                  //The current visibility mode of BT becomes invisible after timeout.
```

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 **AT+QBTGATSREG=?**	Response: **+QBTGATSREG:** (list of supported **<op>s**),**"<gserv_id>"** **OK**
Read Command **AT+QBTGATSREG?**	Response: **+QBTGATSREG: "<gserv_id>",<role>,<ctx_handle>** **…** **OK**
Write Command **AT+QBTGATSREG=<op>,"<gserv_id>"**	Response: **+QBTGATSREG: <op>,"<gserv_id>",<result>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

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 **AT+QBTGATSS=?**	Response: **+QBTGATSS:** (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) **OK**
Read Command **AT+QBTGATSS?**	Response: **+QBTGATSS:** **"<gserv_id>","<service_uuid>",<num_handles>,<service_handle>,<is_primary>,<inst>,<is_started>** **…** **OK**
Write Command Add a service **AT+QBTGATSS=1,"<gserv_id>","<service_uuid>",<num_handles>,<is_primary>,<inst>**	Response: **+QBTGATSS:** **1,"<gserv_id>","<service_uuid>",<is_primary>,<result>,<inst>,<service_handle>**

	OK If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**
Write Command Remove a service **AT+QBTGATSS=0,“<gserv_id>”,<service_handle>**	Response: **+QBTGATSS: 0,“<gserv_id>”,<result>,<service_handle>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<op>	0	Remove
	1	Add
<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 to be even).	
<service_uuid>	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).	
<num_handles>	Number of handles of this service. The range is 1-65535.	
<is_primary>	0	Not primary service
	1	Primary service
<inst>	Instance ID of this service. The range is 0-255.	
<service_handle>	Handle of this service. The range is 0-65535.	
<result>	0	SUCCESS
	1	FAIL
<is_started>	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 **AT+QBTGATSC=?**	Response: **+QBTGATSC: 1,“<gserv_id>”,**(list of supported **<service_handle>**s)**,“<char_uuid>”,**(list of supported **<inst>**s)**,**(list of supported **<prop>**s)**,**(list of supported **<permission>**s) **OK**
Write Command **AT+QBTGATSC=1,“<gserv_id>”,<service_handle>,“<char_uuid>”,<inst>,<prop>,<permission>**	Response: **+QBTGATSC: 1,“<gserv_id>”,<service_handle>,“<char_uuid>”,<inst>,<result>,<char_handle>** **OK** If there is any error relating to ME functionality, response: **+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).
<service_handle>	Handle of this service. The range is 0-65535.
<char_uuid>	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).
<inst>	Instance ID of this characteristic. The range is 0-255.
<prop>	Properties of this characteristic. For more details, please refer to ***Table 2***.
<permission>	Permission of this characteristic. For more details, please refer to ***Table 2***.
<char_handle>	Handle of this characteristic.
<result>	0 SUCCESS 1 FAIL

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**
Test Command **AT+QBTGATSD=?**	Response **+QBTGATSD: 1,“<gserv_id>”,** (list of supported **<service_handle>**s),**“<desc_uuid>”,**(list of supported **<inst>**s),(list of supported **<permission>**s) **OK**
Write Command **AT+QBTGATSD=1,“<gserv_id>”,<service_handle>,“<desc_uuid>”,<inst>,<permission>**	Response **+QBTGATSD: 1,“<gserv_id>”,<service_handle>,“<desc_uuid>”,<inst>,<result>,<desc_handle>** **OK** If there is any error relating to ME functionality, response: **+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).
<desc_uuid>	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).
<inst>	Instance ID of this descriptor. The range is 0-255.
<permission>	Permission of this descriptor. For more details, please refer to ***Table 2***.
<desc_handle>	Handle of this descriptor. The range is 0-65535.
<service_handle>	Handle of this service. The range is 0-65535.
<result>	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 **AT+QBTGATSST=?**	Response: **+QBTGATSST:** (list of supported **<op>**s),**“<gserv_id>”,** (list of supported **<service_handle>**s)[**,**(list of supported **<transport>**s)] **OK**
Write Command Start a service **AT+QBTGATSST=1,“<gserv_id>”,<service_handle>,<transport>**	Response: **+QBTGATSS: 1,“<gserv_id>”,<result>,<service_handle>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**
Write Command Stop a service **AT+QBTGATSST=0,“<gserv_id>”,<service_handle>**	Response: **+QBTGATSST: 0,“<gserv_id>”,<result>,<service_handle>** **OK** If there is any error relating to ME functionality, response: **+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).
<service_handle>	Handle of this service. The range is 0-65535.
<transport>	Transport method to start this service. 0 LE
<result>	0 SUCCESS 1 FAIL

Example

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

OK
```

3.2.6. AT+QBTGATSL Start/Stop BLE Advertising

AT+QBTGATSL Start/Stop BLE Advertising	
Test Command **AT+QBTGATSL=?**	Response: **+QBTGATSL: "<gserv_id>",**(list of supported **<op>**s) **OK**
Write Command **AT+QBTGATSL="<gserv_id>",<op>**	Response: **+QBTGATSL: "<gserv_id>",<result>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<op>	0	Stop
	1	Start
<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

Example

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

OK
```

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

AT+QBTGATSIND Send an Indication or Notification to a Client	
Test Command **AT+QBTGATSIND=?**	Response: **+QBTGATSIND: "<gserv_id>",**(list of supported **<conn_id>**s)**,**(list of supported **<attr_handle>**s)**,** (list of supported **<need_confirm>**s)**,"<ind_value>"**

	OK
Write Command **AT+QBTGATSIND="<gserv_id>",<conn_id>,<attr_handle>,<need_confirm>,"<ind_value>"**	Response: **+QBTGATSIND: <result>,"<gserv_id>",<conn_id>,"<value_handle>"** **OK** If there is any error relating to ME functionality, response: **+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 the '0' at the end of it to even).	
<conn_id>	ID of current connection. The range is 0-255.	
<attr_handle>	Handle of attribute. The range is 0-65535.	
<value_handle>	Handle of this characteristic value. The range is 0-65535.	
<ind_value>	The value that needs to be notified (if the length is odd, then fill '0' at the end of it to be even).	
<need_confirm>	Need client confirmation when sending indication (not when sending notification).	
	0	NO
	1	YES
<result>	0	SUCCESS
	1	FAIL

Example

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

OK
```

3.2.8. 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, response: **+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.
<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
```

3.2.9. 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, response: **+CME ERROR: <errorcode>**

Parameter

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

3.2.10. AT+QGATSETADV Set Advertising Data

AT+ QGATSETADV Set Advertising Data	
Test Command **AT+QGATSETADV=?**	Response **+QGATSETADV: "<gserv_id>",<appearance>**,(list of supported **<string_mode>**s),**"<manufacture_data>","<service_data>","<service_uuid>"** **OK**
Write Command **AT+QGATSETADV="<gserv_id>",<appearance>,<string_mode>,"<manufacture_data>","<service_data>","<service_uuid>"**	Response **+QGATSETADV: <result>** **OK** If there is any error relating to ME functionality, response: **+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 to be even).
<appearance>	The external appearance of the device. The range is 0-65535. For detailed information of this parameter, please visit the website: https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.characteristic.gap.appearance.xml.
<string_mode>	Format type of parameter **<manufacture_data>** and **<service_data>**. 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).
<manufacture_data>	Manufacturer information
<service_data>	Service information
<service_uuid>	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).
<result>	0 SUCCESS 1 FAIL

Example

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

OK
AT+QBTGATSS=1,"ABC2","66FE",50,1,254
+QBTGATSS: 1,"ABCD","2318",1,0,254,256

OK
AT+QGATSETADV="ABC2",64,0,"4F4Bde354192f01e","9612","66FE"
+QGATSETADV: 0

OK
AT+QGATSETADV="ABC2",64,1,"quectel","test","66FE"
+QGATSETADV: 0

OK
```

NOTE

The max length of the advertising data is 31 bytes.

3.2.11. AT+QBTGATCPU Update Connection Parameters

AT+ QBTGATCPU Update Connection Parameters	
Test Command **AT+QBTGATCPU=?**	Response **+QBTGATCPU: <bt_addr>,**(list of supported **<min_interval>**s)**,**(list of supported **<max_interval>**s)**,**(list of supported **<timeout>**s)**,**(list of supported **<latency>**s) **OK**
Write Command **AT+QBTGATCPU=<bt_addr>,<min_interval>,<max_interval>,<timeout>,<latency>**	Response **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

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. Unit: 1.25ms.
<max_interval>	Maximum value of the connection interval. The range is 6-3200. Unit: 1.25ms.
<timeout>	Supervision timeout for the connection. The range is 10-3200. Unit: 10ms.
<latency>	Maximum slave latency allowed for the connection specified as the number of connection events. The range is 0-499.

Example

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

3.2.12. AT+QBTGATSDISC Disconnect Server Actively

AT+ QBTGATSDISC Disconnect Server Actively	
Test Command **AT+QBTGATSDISC=?**	Response **+QBTGATSDISC:** (list of supported **<conn_id>**s) **OK**
Write Command **AT+QBTGATSDISC=<conn_id>**	Response **OK** **+QBTGATSCON:**

	0,“<gserv_id>”,<result>,<bt_addr>,<conn_id> If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

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
```

3.3. BLE GATT Client AT Commands

3.3.1. AT+QBTGATCREG Register/Deregister GATT Client

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

Parameter

<op>	0	Deregister
	1	Register
<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+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 **AT+QBTGATCSCAN=?**	Response **+QBTGATCSCAN:** (list of supported**<op>**s),**“<gclient_id>”** **OK**
Write Command **AT+QBTGATCSCAN=<op>,“<gclient_id>”**	Response **+QBTGATCSCAN: <op>,<result>,“<gclient_id>”** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<op>	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+QBTGATCSCAN=1,"ABCD"
+QBTGATCSCAN: 1,0,"ABCD"

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

+QBTGATCSCAN: "ABCD",FD3B76CE6EF2,53,02011A0B095155454354454C2D4254
AT+QBTGATCSCAN=0,"ABCD"
+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 **AT+QBTGATCCON=?**	Response **+QBTGATCCON:** (list of supported **<op>**s),**"<gclient_id>",<peer_addr>,**(list of supported **<direct>**s) **OK**
Write Command Connect **AT+QBTGATCCON=1,"<gclient_id>",<peer_addr>,<direct>**	Response **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**
Write Command Disconnect **AT+QBTGATCCON=0,"<gclient_id>",<conn_id>**	Response **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

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.

AT+QBTGATCSS Search and Enumerate Peer’s Service	
Test Command **AT+QBTGATCSS=?**	Response **+QBTGATCSS: “<gclient_id>”,<conn_id>** **OK**
Write Command **AT+QBTGATCSS=“<gclient_id>”,<conn_id>**	Response **OK** **+QBTGATCSS: <result>,“<gclient_id>”,<conn_id>,“<service_uuid>”,<service_inst>,<is_primary>** **……** If there is any error relating to ME functionality, response:

	+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.
<service_inst>	Instance ID of the service UUID.
<is_primary>	Service is primary service or not.
<result>	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 **AT+QBTGATCGC=?**	Response **+QBTGATCGC: “<gclient_id>”,<conn_id>,“<service_uuid>”,<service_inst>,<service_is_primary>[,“<char_uuid>”,<char_inst>]** **OK**
Write Command **AT+QBTGATCGC=“<gclient_id>”,<conn_id>,“<service_uuid>”,<service_in**	Response **OK**

st>,<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, response: **+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 primary service or not.
<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 ***Table 2***.

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 customers can only get the first descriptor.

AT+QBTGATCGD Search and Enumerate Peer's Descriptor of the Service with UUID	
Test Command **AT+QBTGATCGD=?**	Response **+QBTGATCGD: "<gclient_id>",<conn_id>,"<service_uuid>",<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>[,"<desc_uuid>",<desc_inst>]** **OK**
Write Command **AT+QBTGATCGD="<gclient_id>",<conn_id>,"<service_uuid>",<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>[,"<desc_uuid>",<desc_inst>]**	Response **OK** **+QBTGATCGD: <result>,"<gclient_id>",<conn_id>,"<service_uuid>",<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>[,"<desc_uuid>",<desc_inst>]** If there is any error relating to ME functionality, response: **+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 primary service or not.

<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.

<desc_uuid> 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, response: **+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 primary service or not.

<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>”,<conn_id>,<write_type>,“<service_uuid>”,<service_inst>,<service_is_primary>,“<char_uuid>”,<char_inst>,“<value>”,<auth_req>**	Response **OK** **+QBTGATCWC: <result>,“<gclient_id>”,<conn_id>,“<service_uuid>”,<service_inst>,<service_is_primary>,“<char_uuid>”,<char_inst>**

	If there is any error relating to ME functionality, response: **+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.

<write_type> 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

<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 primary service or not.

<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+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	
Test Command **AT+QBTGATCRD=?**	Response **+QBTGATCRD: "<gclient_id>",<conn_id>,"<service_uuid>",<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,"<desc_uuid>",<desc_inst>,<auth_req>** **OK**
Write Command **AT+QBTGATCRD="<gclient_id>",<conn_id>,"<service_uuid>",<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,"<desc_uuid>",<desc_inst>,<auth_req>**	Response **OK** **+QBTGATCRD: <result>,"<gclient_id>",<conn_id>,"<service_uuid>",<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,"<desc_uuid>",<desc_inst>,"<value>"** If there is any error relating to ME functionality, response: **+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 primary service or not.
<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.
<desc_uuid>	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.
<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 descriptor.
<result>	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 **AT+QBTGATCWD=?**	Response **+QBTGATCWD: "<gclient_id>",<conn_id>,<write_type>,<service_uuid>,<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,"<desc_uuid>",<desc_inst>,<value>,<auth_req>** **OK**
Write Command **AT+QBTGATCWD="<gclient_id>",<conn_id>,<write_type>,<service_uuid>,<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,"<desc_uuid>",<desc_inst>,<value>,<auth_req>**	Response **OK** **+QBTGATCWD: <result>,"<gclient_id>",<conn_id>,<service_uuid>,<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,"<desc_uuid>",<desc_inst>** If there is any error relating to ME functionality, response: **+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.
<write_type>	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
<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 primary service or not.
<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.
<desc_uuid>	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.
<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+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, response: **+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>** **OK**
Write Command **AT+QBTGATCRN=<op>,“<gclient_id>”,<peer_addr>,<service_uuid>,<service_inst>,<service_is_primary>,“<char_uuid>”,<char_inst>**	Response **+QBTGATCRN: <op>,<result>,“<gclient_id>”,<peer_addr>,<service_uuid>,<service_inst>,<service_is_primary>,“<char_uuid>”,<char_inst>**

	OK If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

Parameter

<op>	0	Deregister
	1	Register
<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)	
<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 primary service or not.	
<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.	
<result>	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+QBTGATGDT Get the Type of Peer Device	
Test Command	Response

AT+QBTGATCGDT=?	**+QBTGATCGDT: "<gclient_id>",<peer_addr>** **OK**
Write Command **AT+QBTGATCGDT="<gclient_id>",<peer_addr>**	Response **+QBTGATCGDT: <result>,"<gclient_id>",<peer_addr>,<dev_type>** **OK** If there is any error relating to ME functionality, response: **+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).
<peer_addr>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<dev_type>	0 Unknown 1 BLE 2 BR/EDR 3 Dual-mode
<result>	0 SUCCESS 1 FAIL

Example

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

OK
```

3.3.1. AT+QBTGATCL Start/Stop Listen Request

AT+QBTGATCL Start/Stop Listen Request	
Test Command **AT+QBTGATCL=?**	Response **+QBTGATCL: "<gclient_id>",<start>** **OK**

Write Command **AT+QBTGATCL=“<gclient_id>”,<start>**	Response **+QBTGATCL: “<gclient_id>”,<result>** **OK** If there is any error relating to ME functionality, response: **+CME ERROR: <errorcode>**

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
```

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>,<n	

eed_rsp>,<is_prepare>,<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 Read Request

Notify the GATT Client’s Read Request	
+QBTGATRREQ: “<gserv_id>”,<conn_id>,<trans_id>,<bt_addr>,<attr_handle>,<is_long>,<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.

<is_long> Notify server the order number of the request (0 means the first request)

<offset> 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
```

4.4. Report Scanning Result

Report Scanning Result	
+QBTGATCSCAN: “<gclient_id>”,<peer_addr>,<RSSI>,<EIR>	

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).

<peer_addr> Address of the discovered BLE device.
(e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)

<RSSI> Signal strength. The range is 0~255, which means the signal strength is -127dbm~127dbm.

<EIR> 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
```

4.5. Notify Client a Connection's Status

Notify Client a Connection's Status	
+QBTGATCCON: <op>,<result>,"<gclient_id>",<peer_addr>,<conn_id>	

Parameter

<op>	0	Disconnect
	1	Connect
<result>	0	SUCCESS
	1	FAIL
<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)	
<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
```

4.6. Notify Client a Notification or Indication Comes

Notify Client a Notification or Indication Comes	
+QBTGATCN: "<gclient_id>",<conn_id>,<peer_addr>,<service_uuid>,<service_inst>,<service_is_primary>,"<char_uuid>",<char_inst>,<is_notify>,<value>	

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.
<peer_addr>	Address of the peer BLE device. (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<service_uuid>	UUID of the service.
<service_inst>	Instance ID of the service UUID.
<service_is_primary>	Service is primary service or not.
<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.
<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
8313	GATT start FMP failed
8314	GATT stop FMP failed
8315	GATT start PXP failed
8316	GATT stop PXP failed