

BC66&BC66-NA LwM2M

Application Note

NB-IoT Module Series

Rev. BC66&BC66-NA_LwM2M_Application_Note_V2.1

Date: 2020-04-03

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.

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: 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

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 WITHOUT PERMISSION ARE FORBIDDEN. 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. 2020. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
1.0	2018-08-28	Randy LI/ Angela SONG	Initial
2.0	2020-03-17	Randy LI	<ol style="list-style-type: none">1. Added introductions of LwM2M in Chapter 1 to Chapter 3.2. Added the following AT commands:<ul style="list-style-type: none">● AT+QLWCFG="rai_enable",<rai_enable>● AT+QLWCFG="auto_ack",<auto_ack_enable>● AT+QLWCFG="recovery_mode",<mode>3. Added examples relating to LwM2M implementation in Chapters 5.6 to 5.8.
2.1	2020-04-03	Taber JIANG	Added the following command: AT+QLWCFG="retransmit_mode",<retransmit_mode>

Contents

About the Document	2
Contents	3
Table Index	5
Figure Index	6
1 Introduction	7
2 General Overview of LwM2M	8
3 LwM2M Implementation	10
3.1. LwM2M Interfaces.....	10
3.1.1. Bootstrap Interface.....	10
3.1.2. Client Registration Interface.....	10
3.1.3. Device Management and Service Enablement Interfaces	11
3.1.4. Information Reporting Interface.....	13
3.2. LwM2M Object	13
3.2.1. Built-in Objects	13
3.2.2. Custom Objects.....	14
3.3. LwM2M Back-off Mechanism.....	14
3.4. LwM2M Auto-Registration Mechanism	15
4 Description of LwM2M AT Commands	16
4.1. AT Command Syntax	16
4.1.1. Definitions.....	16
4.1.2. AT Command Syntax	16
4.2. Description of AT Commands	17
4.2.1. AT+QLWCONFIG Configure Registration Parameters of LwM2M Server.....	17
4.2.2. AT+QLWCFCG Configure Optional Registration Parameters	19
4.2.3. AT+QLWREG Send a Register Request.....	23
4.2.4. AT+QLWUPDATE Send an Update Request.....	24
4.2.5. AT+QLWDEREG Send a Deregister Request	25
4.2.6. AT+QLWADDOBJ Add a LwM2M Object.....	26
4.2.7. AT+QLWDELOBJ Delete a LwM2M Object	27
4.2.8. AT+QLWRDRSP Respond to the Read Request.....	28
4.2.9. AT+QLWRRRSP Respond to the Write Request	30
4.2.10. AT+QLWEXERSP Respond to the Execute Request	30
4.2.11. AT+QLWOBSRSP Respond to the Observe Request	31
4.2.12. AT+QLWNOTIFY Notify the Data to Server	33
4.2.13. AT+QLWRD Read Buffered Data	34
4.2.14. AT+QLWSTATUS Query Current LwM2M Status.....	35
4.2.15. AT+QLWRECOVER Manually Trigger the LwM2M Context Recovery Process	36
4.3. LwM2M Related URCs	37
4.3.1. +QLWURC: "ping" Notify TE of Update Operation Result	38

4.3.2.	+QLWURC: "buffer"	Notify TE that Data is Buffered	38
4.3.3.	+QLWURC: "write"	Inform TE to Respond to the Write Request.....	38
4.3.4.	+QLWURC: "read"	Inform TE to Respond to the Read Request.....	39
4.3.5.	+QLWURC: "execute"	Inform TE to Respond to the Execute Request	40
4.3.6.	+QLWURC: "observe"	Notify TE of an Observe Request.....	40
4.3.7.	+QLWURC: "bs_finished"	Notify TE that the Bootstrap Phase Completed	41
4.3.8.	+QLWURC: "report"	Notify TE that CON Data has been Sent	41
4.3.9.	+QLWURC: "report_ack"	Notify TE that CON Data has been Acked	41
4.3.10.	+QLWURC: "lwstatus"	Notify TE of Current Network Connection Status	41
4.3.11.	+QLWURC: "lifetime_changed"	Notify TE that Lifetime Changed	42
4.3.12.	+QLWURC: "binding_changed"	Notify TE of Binding Node Changed.....	42
4.3.13.	+QLWURC: "min_period_changed"	Notify TE of Minimum Period Changed	43
4.3.14.	+QLWURC: "max_period_changed"	Notify TE of Maximum Period Changed	43
4.3.15.	+QLWURC: "disable"	Inform TE to Disable LwM2M Execution Event	43
4.3.16.	+QLWURC: "factory_reset"	Notify TE of Factory Reset Event	44
4.3.17.	+QLWURC: "bootstrap_request_trigger"	Notify TE of Bootstrap Request Trigger Event	44
4.3.18.	+QLWURC: "recovered"	Notify TE of LwM2M Context Recovery Result	45
5	Examples		46
5.1.	Auto Registration to the LwM2M Server		46
5.2.	Registration Persistence to the LwM2M Server across Reboots		46
5.3.	Manually Trigger Registration to the LwM2M Server		47
5.4.	Custom Object Related Operations		48
5.5.	Manually Trigger LwM2M Context Recovery Process.....		50
5.6.	LwM2M Register to Leshan Server		51
5.7.	LwM2M Updates (Leshan Server)		55
5.7.1.	LwM2M Server Update Operation		55
5.7.2.	LwM2M Client Update Operation.....		56
5.8.	LwM2M Custom Object (Leshan Server)		60
5.8.1.	Observe Operation.....		60
5.8.2.	Notify Operation		61
5.8.3.	Read Operation.....		61
5.8.4.	Execute Operation		61
6	Error Codes		62
7	Appendix A References.....		64

Table Index

Table 1: LwM2M Related URCS	37
Table 2: Summary of LwM2M <err>	62
Table 3: Summary of <status_code>	62
Table 4: References	64
Table 5: Terms and Abbreviations	64

Figure Index

Figure 1: Overall Architecture of LwM2M Enabler	8
Figure 2: Overall Architecture of 3GPP CIoT	9
Figure 3: Client Registration Interface Operation Flow.....	11
Figure 4: Example Data Flow of Device Management	12
Figure 5: Example Data Flow of Service Enablement	12
Figure 6: Example Data Flow of Information Reporting.....	13
Figure 7: Click "Add new client security configuration" (in SECURITY Tab)	52
Figure 8: Create "New security configuration" on Leshan Server	53
Figure 9: Check Created "New security configuration" on Leshan Server	53
Figure 10: Online Client on Leshan Server.....	54
Figure 11: "Read" Operation on Leshan Server.....	55
Figure 12: "Write" Operation on Leshan Server.....	55
Figure 13: "Execute" Operation on Leshan Server	56
Figure 14: Update Example Flow.....	57
Figure 15: Update Registration	58
Figure 16: Update Binding Mode	59
Figure 17: Update Object & Object Instance List.....	60

1 Introduction

OMA Lightweight M2M (LwM2M) is a device management protocol designed for sensor networks and the demands of a machine-to-machine (M2M) environment. The LwM2M protocol, designed for remote management of M2M devices and related service enablement, features a modern architectural design based on REST, defines an extensible resource and data model and builds on an efficient secure data transfer standard called the Constrained Application Protocol (CoAP).

Quectel NB-IoT modules BC66 and BC66-NA support LwM2M V1.0. This document mainly introduces the LwM2M protocol, architecture of OMA LwM2M, and how to use this feature with BC66 and BC66-NA. It also describes the LwM2M AT Command Set supported by the modules.

2 General Overview of LwM2M

OMA specifies a set of LwM2M protocol specifications. Please refer to the following links for more details:

<https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/>

http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180710-A/OMA-TS-LightweightM2M_Core-V1_1-20180710-A.pdf

The OMA LwM2M enabler defines the application layer communication protocol between a LwM2M Server and a LwM2M Client as well as between the LwM2M Bootstrap-Server and the LwM2M Client. The LwM2M Device includes a LwM2M Client component. The enabler includes device management and service enablement for LwM2M Devices. The target LwM2M Devices for this enabler are mainly resource constrained devices. Therefore, this enabler makes use of lightweight and compact protocol mechanisms, as well as an efficient resource data model.

Four interfaces are designed between the three entities, as shown in the architecture below:

- Bootstrap
- Client Registration
- Device Management and Service Enablement
- Information Reporting

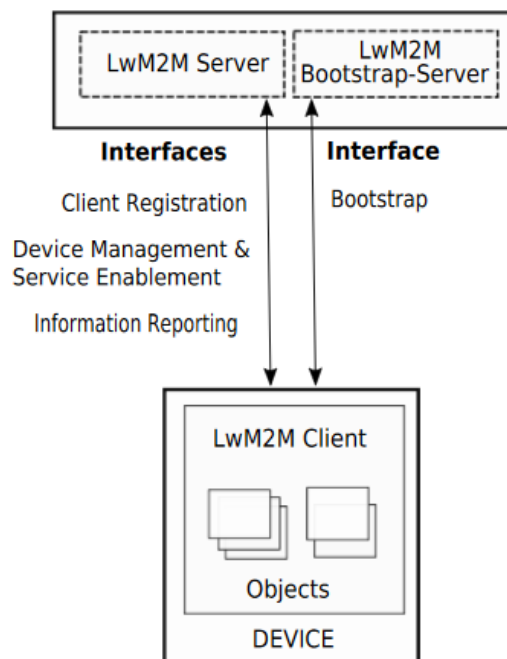


Figure 1: Overall Architecture of LwM2M Enabler

The figure below illustrates the overall architecture of 3GPP CIoT, provided to better understand the LwM2M protocol.

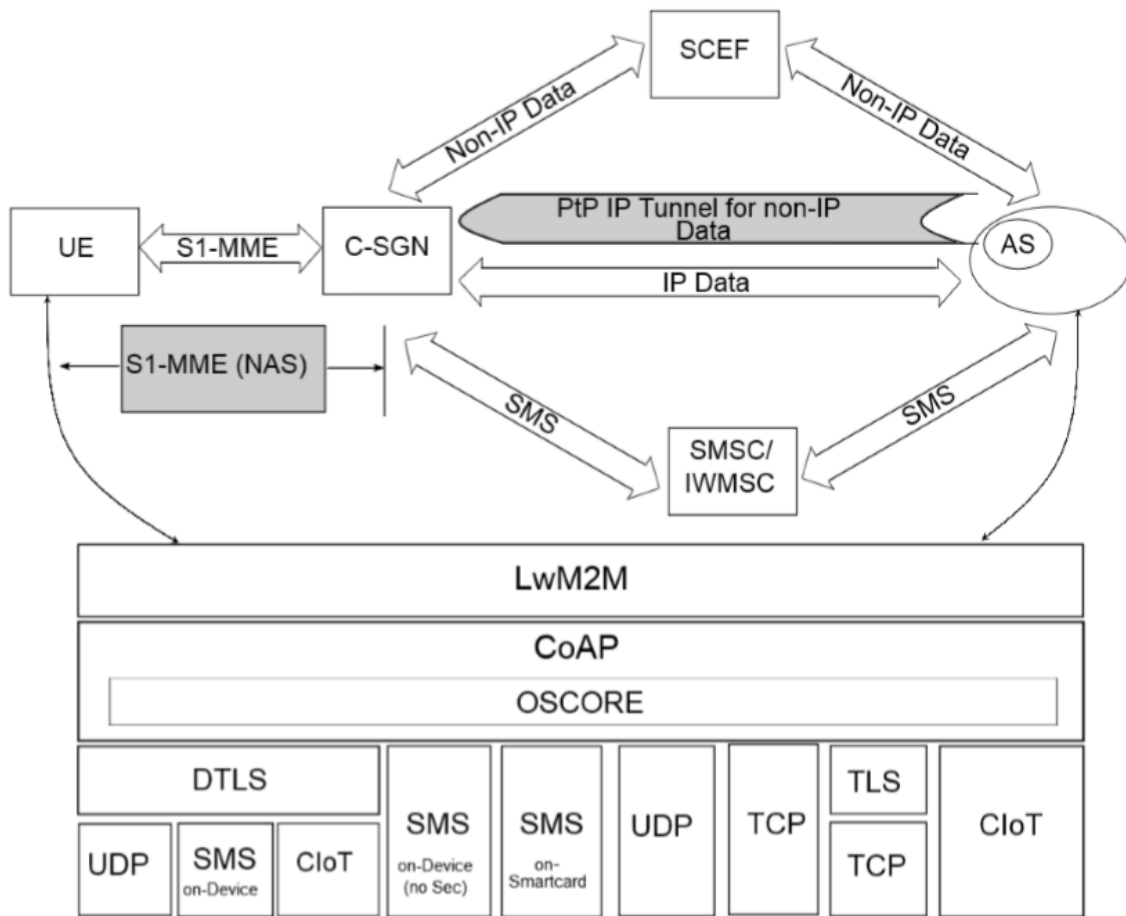


Figure 2: Overall Architecture of 3GPP CIoT

BC66/BC66-NA supports LwM2M Client only. MCU can communicate with LwM2M Server through LwM2M AT commands described in **Chapter 4**.

3 LwM2M Implementation

3.1. LwM2M Interfaces

3.1.1. Bootstrap Interface

The bootstrap interface is used to provision essential information into the LwM2M Client to enable the LwM2M Client to "Register" to one or more LwM2M Servers.

- **Bootstrap Modes Supported by BC66/BC66-NA**

There are four bootstrap modes supported by the LwM2M protocol: Factory Bootstrap, Bootstrap from Smartcard, Client Initiated Bootstrap and Server Initiated Bootstrap. Generally, the LwM2M Client must support at least one bootstrap mode specified in the bootstrap server. Currently, BC66/BC66-NA supports three bootstrap modes: Factory Bootstrap, Client Initiated Bootstrap and Server Initiated Bootstrap.

- **Server and Access Control Configurations**

The bootstrap server or LwM2M Server can be configured with **AT+QLWCONFIG** command, and **AT+QLWREG** command can be sent to register to a dedicated LwM2M Server. During the Client Initiated Bootstrap phase, if disconnection or failure occurs, back-off mechanism will be triggered internally. More details of back-off mechanism are provided in **Chapter 3.3**.

3.1.2. Client Registration Interface

The client registration interface is used by a LwM2M Client to register to one or more LwM2M Servers, maintain each registration to or de-registration from the LwM2M Server.

- **Register**

The LwM2M Client performs a "Register" operation and provides the essential information which is required by the LwM2M Server, such as endpoint name, PSK, lifetime, supported objects/instance etc.

- **Update**

The LwM2M Client performs "Update" operation periodically after registration when lifetime almost expires. If not receiving an update from the LwM2M Client within lifetime, the LwM2M Server will remove the last registration and go to off state.

- **Deregister**

The LwM2M Client performs a "Deregister" operation to deregister from the LwM2M Server.

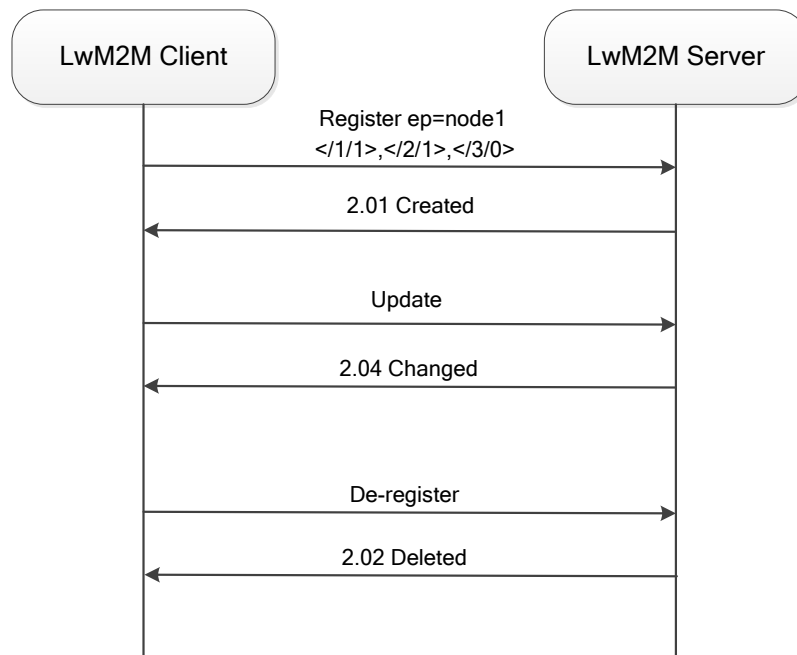


Figure 3: Client Registration Interface Operation Flow

3.1.3. Device Management and Service Enablement Interfaces

Device management and service enablement are very important interfaces in LwM2M protocol.

- Allow LwM2M Server to access object instances and resources available in the LwM2M Client.
- Allow "Create", "Read", "Write", "Delete", "Execute", "Write Attributes", or "Discover" operations from the LwM2M Server.
- Allow the resource operations which are defined in the object definition using the object Template.

The data flow of device management and service enablement are shown as below:

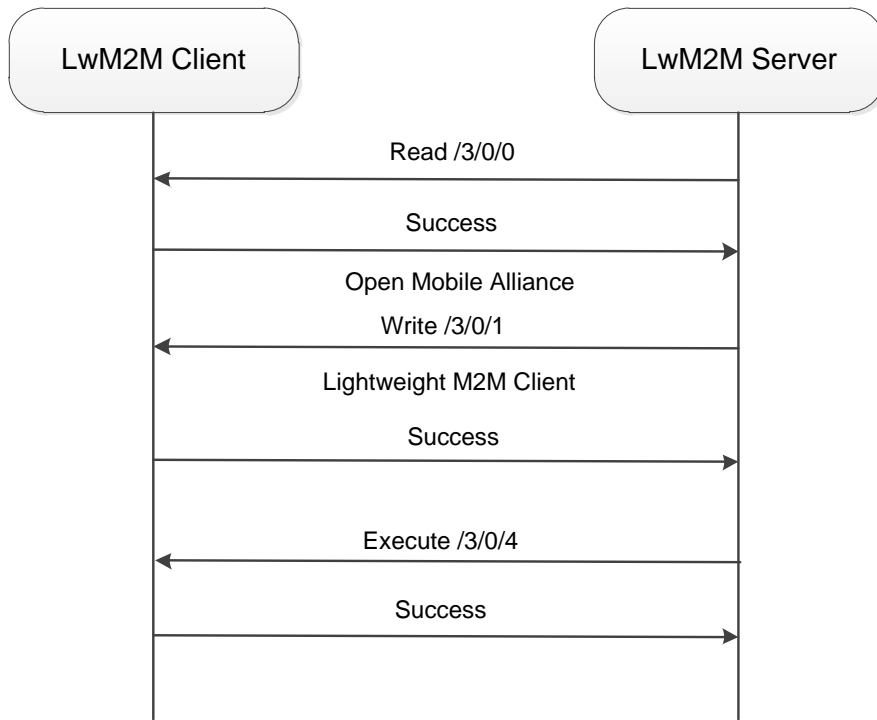


Figure 4: Example Data Flow of Device Management

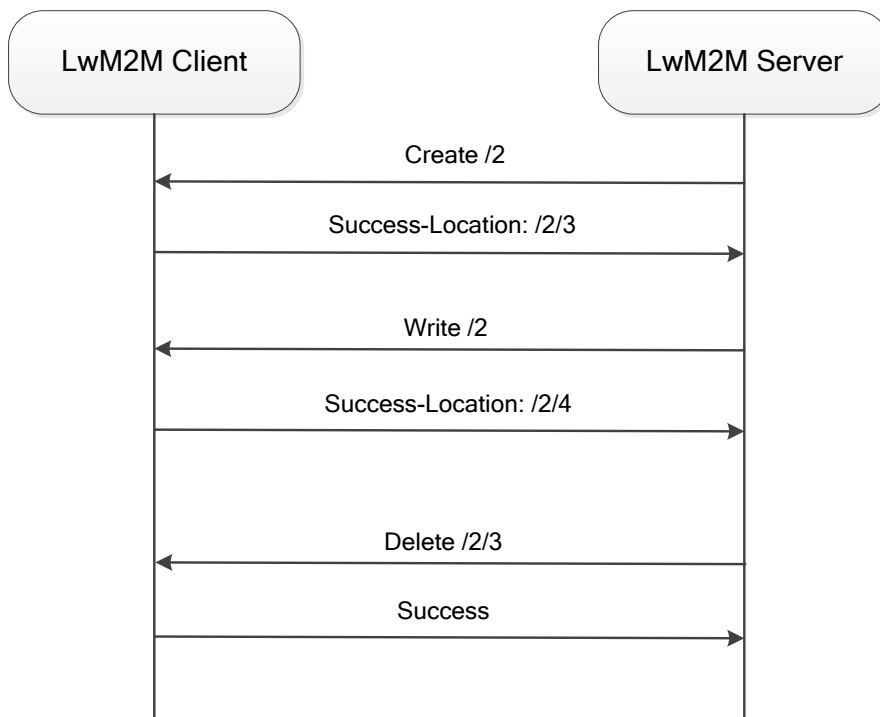


Figure 5: Example Data Flow of Service Enablement

3.1.4. Information Reporting Interface

The information reporting interface is used by a LwM2M Server to observe any change in a Resource on the LwM2M Client, and to receive notifications when new values are available.

- The observation relationship is initiated by sending an "Observe" operation to LwM2M Client for an Object, and Object Instance or a Resource.
- An observation ends when a "Cancel Observation" operation is performed by the LwM2M Server.

BC66/BC66-NA LwM2M Client supports observation and notification of objects, objects instances and resources.

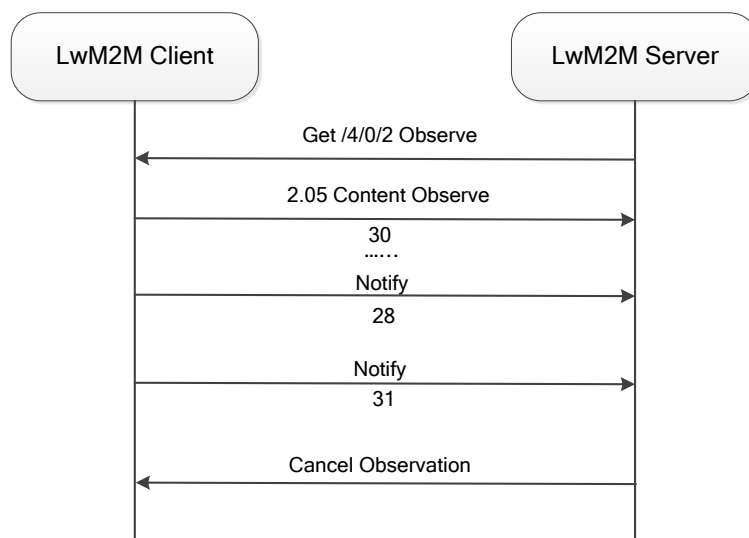


Figure 6: Example Data Flow of Information Reporting

3.2. LwM2M Object

3.2.1. Built-in Objects

There are some built-in objects which are supported by BC66/BC66-NA LwM2M Client by default. (The number in parentheses indicates the object ID.)

- Security object (0)
- Server object (1)
- Access control object (2)
- Device object (3)
- Connectivity monitoring object (4)
- Firmware update object (5)
- Location object (6)
- Connectivity statistics object (7)

- Cellular connectivity object (10)
- APN connection profile object (11)
- Bearer selection object (13)
- Portfolio object (16)
- SCellID object (3353)

3.2.2. Custom Objects

BC66/BC66-NA also supports users to add custom objects through **AT+QLWADDOBJ**. Currently, up to 15 custom objects are supported and each object can add up to 4 instances and each instance can add up to 14 resources.

```
//LwM2M Client has registered to LwM2M Server.
//Add a custom object.
AT+QLWADDOBJ=3303,0,7,5601,5602,5603,5604,5605,5700,5701
OK

+QLWADDOBJ: 0 //0 means the custom object is added successfully.

+QLWURC: "ping",0 //Object list updated successfully
AT+QLWADDOBJ? //Query custom object list
+QLWADDOBJ: 3303,0,7,5601,5602,5603,5604,5605,5700,5701

OK
```

BC66/BC66-NA custom objects only support "Observe", "Read", "Write", and "Execute" operations. Please refer to **Chapter 5.8** for operation examples.

3.3. LwM2M Back-off Mechanism

LwM2M back-off mechanism is used to manage the recovery process of BC66/BC66-NA when errors or timeout happens. The specific processes are described below:

Case 1: If the module sends a registration packet to Bootstrap Server, when timeout or other error occurs, it will retry to register Bootstrap server again. If it still fails, the URC **+QLWREG: 1** will be reported.

Case 2: If the module registers to bootstrap server successfully, but timeout or error occurs when it registers to LwM2M Server, then the module will retry to register LwM2M Server again. If it still fails in LwM2M Server registration, the module will try to register bootstrap server. The URC **+QLWREG: 1** will be reported when it fails in bootstrap server registration.

Case 3: If the module fails in update, it will re-register LwM2M Server. When timeout or error occurs, the

module will retry to register LwM2M Server again. If it still fails in LwM2M Server registration, the module will try to register bootstrap server, and the URC **+QLWREG: 1** will be reported when it fails in bootstrap server registration.

Case 4: When the module wakes up from deep sleep mode, the recovery process will be automatically triggered internally. If recovery fails, the module will re-register LwM2M Server, and when timeout or errors, the module will retry to register LwM2M Server again. If it still fails in LwM2M Server registration, the module will try to register bootstrap server, the URC **+QLWREG: 1** will be reported when it fails in bootstrap server registration.

NOTE

In order to prevent BC66/BC66-NA from entering deep sleep during data interaction, it is recommended that the MCU should actively disable sleep mode of the module before data interaction with **AT+QSCLK=0** and then enable sleep mode with **AT+QSCLK=1** after data interaction is completed. For more details of the command, please refer to the AT commands manual of BC66 and BC66-NA.

3.4. LwM2M Auto-Registration Mechanism

Currently, BC66/BC66-NA includes an LwM2M automatic registration feature for the purpose of meeting T-Mobile certification requirements. It should be noted that LwM2M auto-registration in BC66/BC66-NA is conditional, as illustrated below:

Case 1: If the module uses a T-Mobile USIM card and boots for the first time, the LwM2M auto-registration mechanism will be triggered when the module attaches to the network and obtains the IP address successfully. The module will automatically compute the bootstrap server information for the T-Mobile, for instance, the bootstrap server domain name "bootp.iot.t-mobile.com", the port number 5584 and the corresponding PSK ID, PSK, etc., and the module automatically initiates a registration request to the bootstrap server.

Case 2: If the module uses a non-T-Mobile USIM card, please follow the steps below to enable auto-registration function:

1. Configure information of the server to be connected to with **AT+QLWCONFIG**.
2. Enable automatic registration feature with **AT+QLWCFG="auto_reg",1**, and then reboot the module to make the configuration take effect.

Case 3: If any other server information needs to be configured after LwM2M auto-registration has been initiated internally, it is recommended to set the module into minimum functionality mode with **AT+CFUN=0** first and then configure the server information. After the configuration is completed, please reboot the module or restore the module to full functionality mode with **AT+CFUN=1**.

4 Description of LwM2M AT Commands

4.1. AT Command Syntax

4.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on command line. When an optional parameter is omitted, the new value equals its previous value or its default setting, unless otherwise specified.
- **Underline** Default setting of a parameter.

4.1.2. AT Command Syntax

The **AT** or **at** prefix must be added at the beginning of each command line. Entering **<CR>** will terminate a command line. Commands are usually followed by a response that includes **<CR><LF><response><CR><LF>**. Throughout this document, only the response **<response>** will be presented, **<CR><LF>** are omitted intentionally.

Table 1: Type of AT Commands and Responses

Test Command	AT+<cmd>=?	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	AT+<cmd>?	This command returns the currently set value of the parameter or parameters.
Write Command	AT+<cmd>=<p1> [,<p2>[,<p3>[...]]]	This command sets the user-definable parameter values.
Execution Command	AT+<cmd>	This command reads non-variable parameters affected by internal processes in the module.

4.2. Description of AT Commands

4.2.1. AT+QLWCONFIG Configure Registration Parameters of LwM2M Server

The Write Command configures the registration parameters of LwM2M server to which the LwM2M client will be connected.

AT+QLWCONFIG Configure Registration Parameters of LwM2M Server	
Write Command AT+QLWCONFIG=<BS_enabled>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>]	Response OK If there is any error: +CMEE ERROR: <err>
Read Command AT+QLWCONFIG?	Response When <BS_enabled>=0 and the corresponding server information (e.g. IP address, port number, etc.) are configured, or when <BS_enabled>=1 is configured but no LwM2M server is connected: +QLWCONFIG:<BS_enabled>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>] OK When the setting of <BS_enabled> has been changed, or when bootstrap connection mode is configured and the LwM2M server is also connected: +QLWCONFIG:<BS_enabled>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>] +QLWCONFIG:<BS_enabled>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>] OK
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configurations will be saved to NVRAM automatically.

Parameter

<BS_enabled>	Integer type. Enable/disable bootstrap. 0 Disable Bootstrap 1 Enable Bootstrap
<serverIP>	String type. IP address or domain name of the LwM2M server. The max size is 150 bytes.
<port>	Integer type. Port number of the LwM2M server. The range is 0–65535.
<endpoint_name>	String type. Endpoint name of the device. The maximum size is 150 bytes.
<lifetime>	Integer type. Lifetime of registration. Unit: second. The range is 20–31536000 (365 days). If lifetime ≤ 30 s, then the Actual Update Interval = lifetime / 2 If lifetime ≤ 50 s, then the Actual Update Interval = 15 + (lifetime - 30) × 3/4 If lifetime ≤ 100 s, then the Actual Update Interval = 30 + (lifetime - 50) × 4/5 If lifetime ≤ 300 s, then the Actual Update Interval = 70 + (lifetime - 100) × 9/10 If lifetime > 300 s, then the Actual Update Interval = 250 + (lifetime - 300) × 19/20 When the Actual Update Interval timer times out, the module automatically reports data, updates and resets registration lifetime.
<security_mode>	Integer type. Security mode. 0 Secure mode: PRE_SHARED_KEY 3 No security
<PSK_ID>	String type. PSK identifier. The max size is 150 bytes.
<PSK>	String type. PSK content. It must be Even and Hexadecimal string format. The max size is 256 bytes.
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

NOTES

1. If the module uses a T-Mobile USIM card and boots for the first time, the LwM2M auto-registration mechanism will be triggered when the module attaches to the network and obtains the IP address successfully. The module automatically computes the bootstrap server information for T-Mobile which can be queried with **AT+QLWCONFIG?**. More details are provided in **Chapter 3.4**.
2. LwM2M server information can be configured only when the LwM2M client status is “Not Registered”.

Example

```
AT+QLWCONFIG=1,"bootp.iot.t-mobile.com,5584","urn:imei:867997030052996",900,0,"urn:imei:867997030052996","3083693CF1E6766A2408BFD88278249A2DB498B40A6BF6049300FD47DE325CFC"
OK
AT+QLWCONFIG?
+QLWCONFIG:1,"bootp.iot.t-mobile.com",5584,"urn:imei:867997030052996",900,0,"urn:imei:867997030052996","3083693CF1E6766A2408BFD88278249A2DB498B40A6BF6049300FD47DE325CFC"
```

OK

4.2.2. AT+QLWCFG Configure Optional Registration Parameters

The Write Commands configure optional registration parameters.

AT+QLWCFG Configure Optional Registration Parameters

Read Command

AT+QLWCFG?

Response

+QLWCFG: "auto_reg",<auto_reg_enable>
 +QLWCFG: "per_regstatus_feat",<update_reg_enable>
 +QLWCFG: "rai_enable",<rai_enable>
 +QLWCFG: "retransmit",<ack_timeout>,<retrans_max_times>
 +QLWCFG: "auto_ack",<auto_ack_enable>
 +QLWCFG: "access_mode",<access_mode>
 +QLWCFG: "recovery_mode",<recovery_mode>
 +QLWCFG: "retransmit_mode",<retransmit_mode>

OK

Write Command

Configure whether to enable auto registration

AT+QLWCFG="auto_reg",<auto_reg_enable>]

Response

If <auto_reg_enable> is omitted, query the current configuration:

+QLWCFG: "auto_reg",<auto_reg_enable>

OK

If <auto_reg_enable> is specified, configure whether to enable auto registration:

OK

If there is any error:

+CMEE ERROR: <err>

Write Command

Configure whether to update registration type after module reboot

AT+QLWCFG="per_regstatus_feat",<update_reg_enable>]

Response

If <update_reg_enable> is omitted, query the current configuration:

+QLWCFG: "per_regstatus_feat",<update_reg_enable>

OK

If <update_reg_enable> is specified, configure whether to update registration type after module reboot:

OK

	<p>If there is any error: +CMEE ERROR: <err></p>
<p>Write Command Configure whether to enable automatic update of registration packet with RAI flag 2 AT+QLWCFG="rai_enable",<rai_enable>]</p>	<p>Response If <rai_enable> is omitted, query the current configuration: +QLWCFG: "rai_enable",<rai_enable></p> <p>OK If <rai_enable> is specified, configure whether to enable automatic update of registration packet with RAI flag 2: OK</p> <p>If there is any error: +CMEE ERROR: <err></p>
<p>Write Command Configure response timeout value and maximum number of retransmissions AT+QLWCFG="retransmit",<ack_timeout>,<retrans_max_times>]</p>	<p>Response If <ack_timeout> and <retrans_max_times> are omitted, query the current configuration: +QLWCFG: "retransmit",<ack_timeout>,<retrans_max_times></p> <p>OK</p> <p>If <ack_timeout> and <retrans_max_times> are specified, configure response timeout value and maximum number of retransmissions: OK</p> <p>If there is any error: +CMEE ERROR: <err></p>
<p>Write Command Configure whether to enable automatic reply to observe requests AT+QLWCFG="auto_ack",<auto_ack_enable>]</p>	<p>Response If <auto_ack_enable> is omitted, query the current configuration: +QLWCFG: "auto_ack",<auto_ack_enable></p> <p>OK</p> <p>If <auto_ack_enable> is specified, configure whether to enable automatic reply to observe requests: OK</p> <p>If there is any error: +CMEE ERROR: <err></p>
<p>Write Command Configure the data mode AT+QLWCFG="access_mode",<acce</p>	<p>Response If <access_mode> is omitted, query the current configuration:</p>

<p>ss_mode>]</p>	<p>+QLWCFG: "access_mode",<access_mode></p> <p>OK</p> <p>If <access_mode> is specified, configure the data mode: OK</p> <p>If there is any error: +CMEE ERROR: <err></p>
<p>Write Command Configure the recovery mode. AT+QLWCFG="recovery_mode"[,<recovery_mode>]</p>	<p>Response If <recovery_mode> is omitted, query the current configuration: +QLWCFG: "recovery_mode",<recovery_mode></p> <p>OK</p> <p>If <recovery_mode> is specified, configure the recovery mode: OK</p> <p>If there is any error: +CMEE ERROR: <err></p>
<p>Write Command Configure the retransmission mode. AT+QLWCFG="retransmit_mode"[,<retransmit_mode>]</p>	<p>Response If <retransmit_mode> is omitted, query the current configuration: +QLWCFG: "retransmit_mode",<retransmit_mode></p> <p>OK</p> <p>If <retransmit_mode> is specified, configure the retransmission mode: OK</p> <p>If there is any error: +CMEE ERROR: <err></p>
<p>Maximum Response Time</p>	<p>300 ms</p>
<p>Characteristics</p>	<p>The commands take effect after rebooting. Remain valid after deep-sleep wakeup. The configurations will be saved to NVRAM automatically.</p>

Parameter

<auto_reg_enable>	Integer type. Enable/disable auto registration. 0 Disable auto registration (for more details, please refer to the NOTE) 1 Enable auto registration
<update_reg_enable>	Integer type. Enable/disable registration persistence across reboots. When registration persistence is enabled, the UE will perform an update operation to try to maintain a persistent registration. And the operation result will be notified through URC +QLWURC: "ping" . 0 Registration persistence is enabled across reboots 1 Registration persistence is disabled across reboots
<rai_enable>	Integer type. Enable/disable automatic update registration packet with RAI flag 2. 0 Disable automatic update registration with RAI flag 2. 1 Enable automatic update registration with RAI flag 2.
<access_mode>	Integer type. Data mode. 0 Direct push mode 1 Buffer access mode
<ack_timeout>	Integer type. Response timeout. Unit: second. Timeout = <ack_timeout> × 2 ^(NT-1) "NT" is the number of transmissions. The default is 2 seconds. The maximum number is 20 seconds.
<retrans_max_times>	Integer type. The max number of retransmissions. The default is 4 times. The Maximum number is 8 times.
<auto_ack_enable>	Integer type. Enable/disable automatic reply to observe requests. 0 Disable automatic reply to observe requests. 1 Enable automatic reply to observe requests.
<recovery_mode>	Integer type. 0 Automatically recovers the LwM2M context after awakening from the deep sleep, regardless of lifetime expiration or not. 1 Manually recovers the LwM2M context after awakening from the deep sleep via AT+QLWRECOVER when the lifetime has not time out.
<retransmit_mode>	Integer type. Retransmission mode. 0 Retransmit at an interval of <ack_timeout> × 2 ^(NT-1) 1 Retransmit at an interval of <ack_timeout>
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

NOTES

1. If the module uses a T-Mobile USIM card and boots for the first time, the **<auto_reg_enable>** will be automatically set to 1 when the module attaches to the network and obtains the IP address successfully. More details are provided in **Chapter 3.4**.
2. It is recommended to query the current configurations before configuring parameters, so as to reduce unnecessary repeated configuration.

Example

AT+QLWCFG?

```
+QLWCFG: "auto_reg",0
+QLWCFG: "per_regstatus_feat",0
+QLWCFG: "rai_enable",0
+QLWCFG: "retransmit",2,4
+QLWCFG: "auto_ack",0
+QLWCFG: "access_mode",0
+QLWCFG: "recovery_mode",0
+QLWCFG: "retransmit_mode",0
```

OK

AT+QLWCFG="auto_reg",1

OK

AT+QLWCFG="per_regstatus_feat",1

OK

AT+QLWCFG="rai_enable",1

OK

AT+QLWCFG="retransmit",4,3

OK

AT+QLWCFG="auto_ack",1

OK

AT+QLWCFG="access_mode",1

OK

AT+QLWCFG="recovery_mode",1

OK

AT+QLWCFG="retransmit_mode",1

OK

4.2.3. AT+QLWREG Send a Register Request

The Execution Command sends a register request to LwM2M server.

AT+QLWREG Send a Register Request

Execution Command

AT+QLWREG

Response

OK

+QLWREG: <status_code>

If there is any error:

+CMEE ERROR: <err>

Maximum Response Time

300 ms

Characteristics /

Parameter

<status_code> Integer type. For details of status codes, please refer to **Chapter 6**.
<err> Integer type. For details of error codes, please refer to **Chapter 6**.

Example

AT+QLWREG

OK

+QLWREG: 0 //Registered to LwM2M server successfully

4.2.4. AT+QLWUPDATE Send an Update Request

The Write Command sends an update request to the LwM2M server.

AT+QLWUPDATE Send an Update Request

Write Command

AT+QLWUPDATE=<mode>,<lifetime/binding_mode>

Response

+QLWUPDATE: <messageID>

OK

+QLWUPDATE: <status_code>,<messageID>

If there is any error:

+CMEE ERROR: <err>

Maximum Response Time

300 ms

Characteristics /

Parameter

<mode> Integer type.
 0 Lifetime
 1 Binding mode
<lifetime> Integer type. Lifetime of registration. Unit: second. The range is 20–31536000 (365 days).
 If lifetime ≤ 30 s, then the Actual Update Interval = lifetime / 2
 If lifetime ≤ 50 s, then the Actual Update Interval = 15 + (lifetime - 30) × 3/4
 If lifetime ≤ 100 s, then the Actual Update Interval = 30 + (lifetime - 50) × 4/5

	<p>If lifetime ≤ 300 s, then the Actual Update Interval = 70 + (lifetime - 100) × 9/10</p> <p>If lifetime > 300 s, then the Actual Update Interval = 250 + (lifetime - 300) × 19/20</p> <p>When the Actual Update Interval timer times out, the module automatically reports data, updates and resets registration lifetime.</p>
<binding_mode>	<p>Integer type. The transport binding configured for the LwM2M client.</p> <p>0 UDP mode</p> <p>1 UDP & Queue mode</p>
<messageID>	Integer type. Transport message ID.
<status_code>	Integer type. For details of status codes, please refer to Chapter 6 .
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

Example

```

AT+QLWUPDATE=0,1000 //Update lifetime
+QLWUPDATE: 59797

OK

+QLWUPDATE: 0,59797 //Updated successfully
AT+QLWUPDATE=1,1 //Update binding mode into UDP & Queue mode
+QLWUPDATE: 61459

OK

+QLWUPDATE: 0, 61459 //Updated successfully

```

4.2.5. AT+QLWDEREG Send a Deregister Request

The Execution Command controls the module to launch a deregister request to the LwM2M server.

AT+QLWDEREG Send a Deregister Request

Execution Command	Response
AT+QLWDEREG	<p>OK</p> <p>+QLWDEREG: <status_code></p> <p>If there is any error:</p> <p>+CMEE ERROR: <err></p>
Maximum Response Time	300 ms
Characteristics	/

Parameter

<status_code>	Integer type. For details of status codes, please refer to Chapter 6 .
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

Example

```
AT+QLWDEREG //Send a de-registration request
OK
+QLWDEREG: 0 //De-registered successfully
```

4.2.6. AT+QLWADDOBJ Add a LwM2M Object

The Write Command adds a new LwM2M object.

AT+QLWADDOBJ Add a LwM2M Object

Write Command AT+QLWADDOBJ=<objectID>,<instanceID>,<res_num>,<resourceID>	Response OK +QLWADDOBJ: <err> If there is any error: +CMEE ERROR: <err>
Read Command AT+QLWADDOBJ?	Response +QLWADDOBJ: <objectID>,<instanceID>,<res_num>,<resourceID>[,<resourceID>] OK If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configurations will be saved to NVRAM automatically.

Parameter

<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<res_num>	Integer type. Number of resources.

<resourceID>	Integer type. Resource ID.
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

NOTE

Currently, customers can customize up to 15 objects, each object defines up to 4 instances, and each instance defines up to 14 resources.

Example

//A custom object can be added after successfully connecting to the platform or before the registration is initiated.

```
AT+QLWADDOBJ=3303,0,3,5601,5602,5603 //Add object 3303, and the instance ID is 0. There are 3 resources (5601, 5602, and 5603).
```

OK

```
+QLWADDOBJ: 0
```

```
AT+QLWADDOBJ=3303,1,3,5601,5602,5603
```

OK

```
+QLWADDOBJ: 0
```

```
AT+QLWADDOBJ=3303,2,3,5601,5602,5603
```

OK

```
+QLWADDOBJ: 0
```

```
AT+QLWADDOBJ=3303,3,3,5601,5602,5603
```

OK

```
+QLWADDOBJ: 0
```

```
AT+QLWADDOBJ?
```

```
+QLWADDOBJ: 3303,0,3,5601,5602,5603
```

```
+QLWADDOBJ: 3303,1,3,5601,5602,5603
```

```
+QLWADDOBJ: 3303,2,3,5601,5602,5603
```

```
+QLWADDOBJ: 3303,3,3,5601,5602,5603
```

OK

4.2.7. AT+QLWDELOBJ Delete a LwM2M Object

The Write Command deletes a specified LwM2M object.

AT+QLWDELOBJ Delete a LwM2M Object

Write Command AT+QLWDELOBJ=<objectID>	Response OK +QLWDELOBJ: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configurations will be saved to NVRAM automatically.

Parameter

<objectID>	Integer type. Object ID.
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

Example

```
AT+QLWDELOBJ=17 //Delete object 17
OK
+QLWDELOBJ: 0
```

4.2.8. AT+QLWRDRSP Respond to the Read Request

The Write Command makes the TE respond to the read request from server.

AT+QLWRDRSP Respond to the Read Request

Write Command AT+QLWRDRSP=<messageID>,<result>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>	Response OK +QLWRDRSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	/

Parameter

<messageID>	Integer type. Message ID.																					
<result>	Integer type. The result of read operation and the result codes are as follows: <table border="1"> <thead> <tr> <th>Result code</th> <th>CoAP Response</th> <th>Code Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2.05</td> <td>Content, indicates the correct result.</td> </tr> <tr> <td>11</td> <td>4.00</td> <td>Bad Request</td> </tr> <tr> <td>12</td> <td>4.01</td> <td>Unauthorized</td> </tr> <tr> <td>13</td> <td>4.04</td> <td>Not Found</td> </tr> <tr> <td>14</td> <td>4.05</td> <td>Method Not Allowed</td> </tr> <tr> <td>15</td> <td>4.06</td> <td>Not Acceptable</td> </tr> </tbody> </table>	Result code	CoAP Response	Code Description	1	2.05	Content, indicates the correct result.	11	4.00	Bad Request	12	4.01	Unauthorized	13	4.04	Not Found	14	4.05	Method Not Allowed	15	4.06	Not Acceptable
Result code	CoAP Response	Code Description																				
1	2.05	Content, indicates the correct result.																				
11	4.00	Bad Request																				
12	4.01	Unauthorized																				
13	4.04	Not Found																				
14	4.05	Method Not Allowed																				
15	4.06	Not Acceptable																				
<objectID>	Integer type. Object ID.																					
<instanceID>	Integer type. Instance ID.																					
<resourceID>>	Integer type. Resource ID.																					
<value_type>	Integer type. The value types: <table border="1"> <tbody> <tr><td>1</td><td>String</td></tr> <tr><td>2</td><td>Opaque</td></tr> <tr><td>3</td><td>Integer</td></tr> <tr><td>4</td><td>Float</td></tr> <tr><td>5</td><td>Boolean</td></tr> </tbody> </table>	1	String	2	Opaque	3	Integer	4	Float	5	Boolean											
1	String																					
2	Opaque																					
3	Integer																					
4	Float																					
5	Boolean																					
<len>	Integer type. The value length. When <value_type> is String, <len> is the number of characters. The max size is 1024 bytes. When <value_type> is Opaque, <len> is one-half of the number of characters in the value. The max size is 512 bytes. When <value_type> is Integer, <len> is the number of characters. When <value_type> is Float, <len> is the number of characters. When <value_type> is Boolean, <len> should be 1.																					
<value>	String type. When <value_type> is Opaque, it is in Hexadecimal string format. When <value_type> is Boolean, it is only 0 or 1.																					
<index>	Integer type. The index number of the data. If the data is a combination of several messages, it should be split into several parts. If it is split into N parts, the order number of <index> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the smallest number. If <index> is 0, it means that this is the last message of the data.																					
<err>	Integer type. For details of error codes, please refer to Chapter 6 .																					

Example

```
+QLWURC: "read",62953,9,0,0 //Receive the read request from server
AT+QLWRDRSP=62953,1,9,0,0,1,5,"abcde",0
OK
+QLWRDRSP: 0
```

4.2.9. AT+QLWRRSP Respond to the Write Request

The Write Command makes the TE respond to the write request from server.

AT+QLWRRSP Respond to the Write Request

Write Command AT+QLWRRSP=<messageID>,<result>	Response OK +QLWRRSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	/

Parameter

<messageID>	Integer. Message ID.																		
<result>	Integer. The result of write operation and the result codes are as follows:																		
	<table border="1"> <thead> <tr> <th>Result code</th> <th>CoAP Response</th> <th>Code Description</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>2.04</td> <td>Changed, indicates the correct result.</td> </tr> <tr> <td>11</td> <td>4.00</td> <td>Bad Request</td> </tr> <tr> <td>12</td> <td>4.01</td> <td>Unauthorized</td> </tr> <tr> <td>13</td> <td>4.04</td> <td>Not Found</td> </tr> <tr> <td>14</td> <td>4.05</td> <td>Method Not Allowed</td> </tr> </tbody> </table>	Result code	CoAP Response	Code Description	2	2.04	Changed, indicates the correct result.	11	4.00	Bad Request	12	4.01	Unauthorized	13	4.04	Not Found	14	4.05	Method Not Allowed
Result code	CoAP Response	Code Description																	
2	2.04	Changed, indicates the correct result.																	
11	4.00	Bad Request																	
12	4.01	Unauthorized																	
13	4.04	Not Found																	
14	4.05	Method Not Allowed																	
<err>	Integer type. For details of error codes, please refer to Chapter 6 .																		

Example

```
+QLWURC: "write",36560,9,0,0,2,7,"5155454354454C",0 //Receive the write request from server
AT+QLWRRSP=36560,2
OK
+QLWRRSP: 0
```

4.2.10. AT+QLWEXERSP Respond to the Execute Request

The Write Command makes the TE respond to the execute request from server.

AT+QLWEXERSP Respond to the Execute Request

Write Command AT+QLWEXERSP=<messageID>,<result>	Response OK +QLWEXERSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	/

Parameter

<messageID>	Integer type. Message ID.		
<result>	Integer type. The result of execute operation and the result codes are as follows:		
	Result code	CoAP Response	Code Description
	2	2.04	Changed, indicates the correct result
	11	4.00	Bad Request
	12	4.01	Unauthorized
	13	4.04	Not Found
	14	4.05	Method Not Allowed
<err>	Integer type. For details of error codes, please refer to Chapter 6 .		

Example

```
+QLWURC: "execute",39040,15,0,5 //Receive the execute request from server
AT+QLWEXERSP=39040,2
OK
+QLWEXERSP: 0
```

4.2.11. AT+QLWOBSRSP Respond to the Observe Request

The Write Command makes the TE respond to the observe request from server.

AT+QLWOBSRSP Respond to the Observe Request

Write Command AT+QLWOBSRSP=<messageID>,<result>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<i	Response OK +QLWOBSRSP: <err>
---	---

ndex>	If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	/

Parameter

<messageID>	Integer type. Message ID.																					
<result>	Integer type. The result of observe operation and the result codes are as follows: <table border="1"> <thead> <tr> <th>Result code</th> <th>CoAP Response</th> <th>Code Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2.05</td> <td>Content, indicates the correct result.</td> </tr> <tr> <td>11</td> <td>4.00</td> <td>Bad Request</td> </tr> <tr> <td>12</td> <td>4.01</td> <td>Unauthorized</td> </tr> <tr> <td>13</td> <td>4.04</td> <td>Not Found</td> </tr> <tr> <td>14</td> <td>4.05</td> <td>Method Not Allowed</td> </tr> <tr> <td>15</td> <td>4.06</td> <td>Not Acceptable</td> </tr> </tbody> </table>	Result code	CoAP Response	Code Description	1	2.05	Content, indicates the correct result.	11	4.00	Bad Request	12	4.01	Unauthorized	13	4.04	Not Found	14	4.05	Method Not Allowed	15	4.06	Not Acceptable
Result code	CoAP Response	Code Description																				
1	2.05	Content, indicates the correct result.																				
11	4.00	Bad Request																				
12	4.01	Unauthorized																				
13	4.04	Not Found																				
14	4.05	Method Not Allowed																				
15	4.06	Not Acceptable																				
<objectID>	Integer type. Object ID.																					
<instanceID>	Integer type. Instance ID.																					
<resourceID>>	Integer type. Resource ID.																					
<value_type>	Integer type. The value types. 1 String 2 Opaque 3 Integer 4 Float 5 Boolean																					
<len>	Integer type. The value length. When <value_type> is String, <len> is the number of characters. The max size is 1024 bytes. When <value_type> is Opaque, <len> is one-half of the number of characters in the value. The max size is 512 bytes. When <value_type> is Integer, <len> is the number of characters. When <value_type> is Float, <len> is the number of characters. When <value_type> is Boolean, <len> should be 1.																					
<value>	String type. When <value_type> is Opaque, it is in Hexadecimal string format. When <value_type> is Boolean, it is only 0 or 1.																					
<index>	Integer type. The index number of the data. If the data is a combination of several messages, it should be split into several parts. If it is split into N parts, the order number of <index> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the smallest number. If <index> is 0, it means that this is the last message of the data.																					
<err>	Integer type. For details of error codes, please refer to Chapter 6 .																					

Example

```
+QLWURC: "observe",624,0,9,0,0 //Receive the observe request from server
AT+QLWOBSRSP=624,1,9,0,0,1,5,"abcde",0 //Respond to the observe request
OK
+QLWOBSRSP: 0
```

4.2.12. AT+QLWNOTIFY Notify the Data to Server

The Write Command notifies the data to server.

AT+QLWNOTIFY Notify the Data to Server

Write Command	Response
AT+QLWNOTIFY=<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>[,<ack>[,<rai_flag>]]	OK [+QLWURC: "report",<messageID>] +QLWNOTIFY: <err> [+QLWURC: "report_ack",<status_code>,<messageID>]
Maximum Response Time	300 ms
Characteristics	/

Parameter

<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<resourceID>	Integer type. Resource ID. -1 indicates all resources.
<value_type>	Integer type. The value types. 1 String 2 Opaque 3 Integer 4 Float 5 Boolean
<len>	Integer type. The length of data to be sent. When <value_type> is String, <len> is the number of characters. The max size is 1024 bytes. When <value_type> is Opaque, <len> is one-half of the number of characters in the

	value. The max size is 512 bytes.
	When <value_type> is Integer, <len> is the number of characters.
	When <value_type> is Float, <len> is the number of characters.
	When <value_type> is Boolean, <len> should be 1.
<value>	String type. When <value_type> is Opaque, it is in Hexadecimal string format.
	When <value_type> is Boolean, it is only 0 or 1.
<index>	Integer type. The index number of the data. If the data is combined with several messages, it should be split into several parts. If it is split into N parts, the order number of <index> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the smallest number. If <index> is 0, it means that this is the last message of the data.
<ack>	Integer type. Response type marker.
	0 NON type data
	1 CON type data
<rai_flag>	Integer type. Release assistance information.
	0 No information available (or none of the other options apply)
	1 TE will send only 1 UL packet and no DL packets expected
	2 TE will send only 1 UL packet and only 1 DL packet expected
<status_code>	Integer type. For details of status codes, please refer to Chapter 6 .
<err>	Integer type. For details of error codes, please refer to Chapter 6 .

4.2.13. AT+QLWRD Read Buffered Data

The Execution Command reads cached data in buffer.

AT+QLWRD Read Buffered Data	
Read Command AT+QLWRD?	Response +QLWRD: <caching_nodes_number> OK If there is any error: +CMEE ERROR: <err>
Execution Command AT+QLWRD	Response +QLWRD: <remaining_nodes_number>[,<urc_data>] OK
Maximum Response Time	300 ms
Characteristics	/

Parameter

< caching_nodes_number >	Integer type. Number of caching nodes.
< remaining_nodes_number >	Integer type. Number of remaining nodes.
< urc_data >	String type. Cached URC: If observe: "observe", <messageID>, <flag>, <objectID>, <instanceID>, <resourceID> If read: "read", <messageID>, <objectID>, <instanceID>, <resourceID > If write: "write", <messageID>, <objectID>, <instanceID>, <resourceID>, <value_type>, <len>, <value>, <index> If execute: "execute", < messageID>, <objectID>, <instanceID>, <resourceID>
< err >	Integer type. For details of error codes, please refer to Chapter 6 .

Example

```
AT+QLWRD?
+QLWRD: 4

OK
AT+QLWRD
+QLWRD: 3,"observe",36581,0,9,0,0

OK
```

4.2.14. AT+QLWSTATUS Query Current LwM2M Status

The Read Command queries the current LwM2M status.

AT+QLWSTATUS Query Current LwM2M Status

Read Command AT+QLWSTATUS?	Response +QLWSTATUS: <status> OK If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	/

Parameter

< status >	Integer type. Status of LwM2M. 0 Not Registered
------------	--

- 1 Registering
- 2 Registered
- 3 Deregistering
- 4 Deregistered
- 5 Invalid
- 6 Recovering
- 7 Need Recover

<err> Integer type. For details of error codes, please refer to **Chapter 6**.

4.2.15. AT+QLWRECOVER Manually Trigger the LwM2M Context Recovery Process

The Execution Command is used to manually trigger the LwM2M context recovery process.

AT+QLWRECOVER Manually Trigger the LwM2M Context Recovery Process

Execution Command AT+QLWRECOVER	Response OK +QLWURC: "recovered",<status_code> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	/

Parameter

<status_code> Integer type. For details of status codes, please refer to **Chapter 6**.
<err> Integer type. For details of error codes, please refer to **Chapter 6**.

Example

//The module has already connected to the server, the lifetime has not timeout, and the module is woken up from deep sleep.

```
F1: 0000 0000
V0: 0000 0000 [0001]
00: 0000 0000
U0: 0000 0001 [0000]
T0: 0000 001E
Leaving the BROM
+CPIN: READY
```

AT+QLWSTATUS?

+QLWSTATUS: 7 //The LwM2M context needs to be recovered.

OK

AT+QLWRECOVER

OK

+QLWURC: "recovered",0 //Recovered successfully

4.3. LwM2M Related URCs

This chapter gives LwM2M related URCs and their descriptions.

Table 1: LwM2M Related URCs

Index	Notification Display	Description
[1]	+QLWURC: "ping",<status_code>	Notify TE of the result of periodic update operations.
[2]	+QLWURC: "buffer"	Notify TE that the new data is cached in buffer.
[3]	+QLWURC: "write",<messageID>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>	Inform TE to respond to the write request from server.
[4]	+QLWURC: "read",<messageID>,<objectID>,<instanceID>,<resourceID>	Inform TE to respond to the read request from server.
[5]	+QLWURC: "execute",<messageID>,<objectID>,<instanceID>,<resourceID>	Inform TE to respond to the execute request from server.
[6]	+QLWURC: "observe",<messageID>,<flag>,<objectID>,<instanceID>,<resourceID>	Notify TE that there is an observe request from server.
[7]	+QLWURC: "bs_finished"	Notify TE that the bootstrap phase has completed.
[8]	+QLWURC: "report",<messageID>	Notify TE that CON data has been sent.
[9]	+QLWURC: "report_ack",<status_code>,<messageID>	Notify TE that CON data has been acknowledged.
[10]	+QLWURC: "lwstatus",<status_code>	Notify TE of the current network connection status.
[11]	+QLWURC: "lifetime_changed",<lifetime>	Notify TE that the registration lifetime has been changed.

[12]	+QLWURC: "binding_changed",<binding_mode>	Notify TE that the binding mode has changed.
[13]	+QLWURC: "min_period_changed",<min_period>	Notify TE that the minimum period of an observation has changed.
[14]	+QLWURC: "max_period_changed",<max_period>	Notify TE that the maximum period of an observation has changed.
[15]	+QLWURC: "disable",<timeout>	Inform TE to disable LwM2M execution event.
[16]	+QLWURC: "factory_reset",<status_code>	Notify TE to start the factory reset process.
[17]	+QLWURC: "bootstrap_request_trigger",<status_code>	Notify TE of bootstrap request trigger event.
[18]	+QLWURC: "recovered",<status_code>	Notify TE of the LwM2M context recovery result from deep sleep mode.

4.3.1. **+QLWURC: "ping" Notify TE of Update Operation Result**

This URC is used to notify TE of the result of update operations.

+QLWURC: "ping" Notify TE of Update Operation Result

+QLWURC: "ping",<status_code> Notify TE of the result of update operations.

Parameter

<status_code> Integer type. For details of status codes, please refer to **Chapter 6**.

4.3.2. **+QLWURC: "buffer" Notify TE that Data is Buffered**

This URC is used to notify the TE that the new data is buffered.

+QLWURC: "buffer" Notify TE that Data is Buffered

+QLWURC: "buffer" Notify TE that the new data is cached in buffer.

4.3.3. **+QLWURC: "write" Inform TE to Respond to the Write Request**

The URC is mainly used to inform the TE to respond to the write request from server. The TE should respond to the request with command **AT+QLWRRSP**.

+QLWURC: "write" Inform TE to Respond to the Write Request

+QLWURC: "write",<messageID>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index> Inform TE to respond to the write request from server.

Parameter

<messageID>	Integer type. Message ID.
<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<resourceID>	Integer type. Resource ID.
<value_type>	Integer type. 1 String 2 Opaque 3 Integer 4 Float 5 Boolean
<len>	Integer type. The value length. When <value_type> is String, <len> is the number of characters. The max size is 1024 bytes. When <value_type> is Opaque, <len> is one-half of the number of characters in the value. The max size is 512 bytes. When <value_type> is Integer, <len> is the number of characters. When <value_type> is Float, <len> is the number of characters.
<value>	String type. The value that is received from the server, in hex string format.
<index>	Integer type. The index number of the write request. If the write request is a combination of several messages, it should be split into several parts. If it is split into N parts, the order number of <index> is N-1 to 0 in descending order, and the URC is sorted from the largest one to the smallest. If <index> is 0, it means that this is the last message of the write request. Currently, only <index>=0 is supported.

4.3.4. +QLWURC: "read" Inform TE to Respond to the Read Request

The URC is mainly used to inform the TE to respond to the read request from server. The TE should respond to the request with command **AT+QLWRDRSP**.

+QLWURC: "read" Inform TE to Respond to the Read Request

+QLWURC: "read",<messageID>,<objectID>,<instanceID>,<resourceID> Inform TE to respond to the read request from server.

Parameter

<messageID>	Integer type. Message ID.
<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID. -1 indicates all resources under a specific object.
<resourceID>	Integer type. Resource ID. -1 indicates all resources under a specific instance.

4.3.5. +QLWURC: "execute" Inform TE to Respond to the Execute Request

The URC is mainly used to inform the TE to respond to the execute request from server. The TE should respond to the request with **AT+QLWEXERSP**.

+QLWURC: "execute" Inform TE to Respond to the Execute Request

+QLWURC: "execute",<messageID>,<objectID>,<instanceID>,<resourceID>	Inform TE to respond to the execute request from server.
---	--

Parameter

<messageID>	Integer type. Message ID.
<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<resourceID>	Integer type. Resource ID.

4.3.6. +QLWURC: "observe" Notify TE of an Observe Request

The URC is mainly used to notify the TE that there is an observe request from server.

The TE should respond to the request with **AT+QLWOBSRSP**.

+QLWURC: "observe" Notify TE of an Observe Request

+QLWURC: "observe",<messageID>,<flag>,<objectID>,<instanceID>,<resourceID>	Notify TE that there is an observe request from server.
--	---

Parameter

<messageID>	Integer type. Message ID.
<flag>	Indicates whether or not to observe.
0	Observe
1	Cancel observe

<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID. -1 indicates all resources under all instances.
<resourceID>	Integer type. Resource ID. -1 indicates all resources under a specific instance.

4.3.7. +QLWURC: "bs_finished" Notify TE that the Bootstrap Phase Completed

This URC is used to notify the TE that the bootstrap phase has completed.

+QLWURC: "bs_finished" Notify TE that the Bootstrap Phase Completed

+QLWURC: "bs_finished" Notify TE that the bootstrap phase has completed.

4.3.8. +QLWURC: "report" Notify TE that CON Data has been Sent

This URC is used to notify the TE that CON data has been sent.

+QLWURC: "report" Notify TE that CON Data has been Sent

+QLWURC: "report",<messageID> Notify TE that CON data has been sent.

Parameter

<messageID> Integer type. Message ID.

4.3.9. +QLWURC: "report_ack" Notify TE that CON Data has been Acked

This URC is used to notify the TE that CON data has been acknowledged.

+QLWURC: "report_ack" Notify TE that CON Data has been Acked

+QLWURC: "report_ack",<status_code>,<messageID> Notify TE that CON data has been acknowledged.

Parameter

<status_code> Integer type. For details of status codes, please refer to **Chapter 6**.
<messageID> Integer type. Message ID.

4.3.10. +QLWURC: "lwstatus" Notify TE of Current Network Connection Status

This URC is used to notify the TE of the current network connection status.

+QLWURC: "lwstatus" Notify TE of Current Network Connection Status

+QLWURC: "lwstatus",<result_code> Notify TE of the current network connection status.

Parameter

<result_code> Integer type.
 0 The current network connection is lost.
 1 The current network connection is resumed.

4.3.11. +QLWURC: "lifetime_changed" Notify TE that Lifetime Changed

This URC is used to notify the TE that the registration lifetime has been changed.

+QLWURC: "lifetime_changed" Notify TE that Lifetime changed

+QLWURC: "lifetime_changed",<lifetime> Notify TE that the registration lifetime has been changed.

Parameter

<lifetime> Integer type. Lifetime of registration. Unit: second. The range is 20-31536000.
 If lifetime ≤ 30 s, then the Actual Update Interval = lifetime / 2
 If lifetime ≤ 50 s, then the Actual Update Interval = 15 + (lifetime – 30) × 3/4
 If lifetime ≤ 100 s, then the Actual Update Interval = 30 + (lifetime – 50) × 4/5
 If lifetime ≤ 300 s, then the Actual Update Interval = 70 + (lifetime – 100) × 9/10
 If lifetime > 300 s, then the Actual Update Interval = 250 + (lifetime – 300) × 19/20
 When the Actual Update Interval timer times out, the module automatically reports data, updates and resets registration lifetime.

4.3.12. +QLWURC: "binding_changed" Notify TE of Binding Node Changed

This URC is used to notify the TE that the binding mode has changed.

+QLWURC: "binding_changed" Notify TE of Binding Mode Changed

+QLWURC: "binding_changed",<binding_mode> Notify TE that the binding mode has changed.

Parameter

<binding_mode> String type. Only support "U" or "UQ".

4.3.13. +QLWURC: "min_period_changed" Notify TE of Minimum Period Changed

This URC is used to notify the TE that the minimum period of an observation has changed.

+QLWURC: "min_period_changed" Notify TE of Minimum Period Changed

+QLWURC: "min_period_changed",<min_period> Notify TE that the minimum period of an observation has changed.

Parameter

<min_period> Integer type. Minimum period of an observation. Currently, it is not supported.

4.3.14. +QLWURC: "max_period_changed" Notify TE of Maximum Period Changed

This URC is used to notify the TE that the maximum period of an observation has changed.

+QLWURC: "max_period_changed" Notify TE of Maximum Period Changed

+QLWURC: "max_period_changed",<max_period> Notify TE that the maximum period of an observation has changed.

Parameter

<max_period> Integer type. Maximum period of an observation. Currently, it is not supported.

4.3.15. +QLWURC: "disable" Inform TE to Disable LwM2M Execution Event

This URC is used to inform the TE to disable execution event of Object 1/ Resource 4.

+QLWURC: "disable" URC to Inform TE to Disable LwM2M Execution Event

+QLWURC: "disable",<timeout> If this resource is executed, this LwM2M server object is disabled for a certain period of time defined in <timeout>.

Parameter

<timeout> Integer type. A period to disable the server. After this period, the LwM2M client MUST perform registration process to the server. If this resource is not set, a default timeout value is 86400 seconds (1 day).

4.3.16. +QLWURC: "factory_reset" Notify TE of Factory Reset Event

This URC is used to notify the TE of the factory reset event triggered by Object 3/Resource 5.

+QLWURC: "factory_reset" Notify TE of Factory Reset Event

+QLWURC: "factory_reset",<status_code>	Reboot the LwM2M device to perform factory reset of the LwM2M device to make the LwM2M device go through initial deployment sequence where provisioning and bootstrap sequence is performed.
--	--

Parameter

<status_code>	Integer type. For details of status codes, please refer to Chapter 6 .
---------------	---

NOTES

1. If <access_mode> in **AT+QLWCFG** is set to 0 (direct push mode), the module will perform deregistration process before reporting the URC, and then reset automatically. The module will re-register with the bootstrap server after reset.
2. If <access_mode> in **AT+QLWCFG** is set to 1 (buffer access mode), the module will perform deregistration process first and then report the URC **+QLWURC: "buffer"** (if the buffer is empty) to indicate the incoming URC **+QLWURC: "factory_reset"**. However, the module will not be reset, therefore the MCU should be reset manually to re-register with the bootstrap server.

4.3.17. +QLWURC: "bootstrap_request_trigger" Notify TE of Bootstrap Request

Trigger Event

This URC is used to notify the TE of the bootstrap request trigger event triggered by Object 1/Resource 9.

+QLWURC: "bootstrap_request_trigger" Notify TE of Bootstrap Request Trigger Event

+QLWURC: "bootstrap_request_trigger",<status_code>	When this resource is executed, the LwM2M client MUST initiate a "Client Initiated Bootstrap" procedure in using the LwM2M Bootstrap-Server Account.
--	--

Parameter

<status_code>	Integer type. For details of status codes, please refer to Chapter 6 .
---------------	---

4.3.18. +QLWURC: "recovered" Notify TE of LwM2M Context Recovery Result

This URC is used to notify the TE of the LwM2M context recovery result from deep sleep mode.

+QLWURC: "recovered" Notify TE of LwM2M Context Recovery Result

+QLWURC: "recovered",<status_code> Notify TE of the LwM2M context recovery result from deep sleep mode.

Parameter

<status_code> Integer type. For details of status codes, please refer to **Chapter 6**.

5 Examples

5.1. Auto Registration to the LwM2M Server

//If **AT+QLWCFG="auto_reg",1** and **AT+QLWCONFIG** is configured before network registration or module rebooting, the LwM2M will actively initiate the registration process when the network is registered.

RDY

+CFUN: 1

+CPIN: READY

+IP: 100.119.77.254 //Registered to network successfully and the IP address is reported

+QLWREG: 0 //Registered to LwM2M server successfully

AT+QLWUPDATE=0,1000

+QLWUPDATE: 38177

OK

+QLWUPDATE: 0,38177 //Updated successfully

AT+QLWDEREG

OK

+QLWDEREG: 0 //De-registered successfully

5.2. Registration Persistence to the LwM2M Server across Reboots

//If the module has been successfully registered to network before, and **AT+QLWCFG="auto_reg",1** and **AT+QLWCFG="per_regstatus_feat",0** are configured before reboot, the module will perform an update operation to try to maintain a persistent registration. And the URC **+QLWURC: "ping"** will be reported to notify the module of the update operation result.

RDY

```
+CFUN: 1

+CPIN: READY

+IP: 100.118.64.171

//Just sent update registration packet to server, and update operation performed successfully
+QLWURC: "ping",0
AT+QLWDEREG
OK

+QLWDEREG: 0           //De-registered successfully
```

5.3. Manually Trigger Registration to the LwM2M Server

```
AT+CEREG?
+CEREG: 0,1           //EPS network registered. (+CEREG: 0,5 indicating registered roaming
                       service is also possible.)

OK
AT+CGPADDR=1
+CGPADDR: 1,100.88.40.249

OK

//First please check LwM2M status with AT+QLWSTATUS?, LwM2M information can be configured when
LwM2M status is "Not Registered".
AT+QLWCONFIG=1,"bootp.iot.t-mobile.com",5584,"urn:imei:867997030052996",900,0,"urn:imei:8
67997030052996","3083693CF1E6766A2408BFD88278249A2DB498B40A6BF6049300FD47DE325C
FC"
OK
AT+QLWREG           //Send a registration request
OK

+QLWURC: "lifetime_changed",30

+QLWURC: "min_period_changed",1

+QLWURC: "max_period_changed",3600

+QLWURC: "binding_changed","UQ"
```



```
+QLWURC: "bs_finished"

+QLWREG: 0 //Registered successfully
AT+QLWUPDATE=0,1000
+QLWUPDATE: 59797

OK

+QLWUPDATE: 0,59797 //Updated successfully
AT+QLWDEREG
OK

+QLWDEREG: 0 //De-registered successfully
```

5.4. Custom Object Related Operations

```
AT+CEREG?
+CEREG: 0,1 //EPS network registered. (+CEREG: 0,5 indicating registered roaming
service is also possible.)

OK
AT+CGPADDR=1
+CGPADDR: 1,100.88.40.249

OK

//First please check LwM2M status via AT+QLWSTATUS?, LwM2M information can be configured when
LwM2M status is "Not Registered".
AT+QLWCONFIG=0,220.180.239.212,8085,"Test_dev",30,3 //Configure server relevant parameters.
OK
AT+QLWREG //Send a registration request
OK

+QLWREG: 0 //Registered successfully
AT+QLWADDOBJ=9,0,5,0,1,2,3,4 //Add a custom object
OK

+QLWADDOBJ: 0

+QLWURC: "ping",0 //Updated the object to server successfully
AT+QLWADDOBJ=19,0,4,0,1,2,3
```

OK

+QLWADDOBJ: 0

+QLWURC: "ping",0

AT+QLWADDOBJ=19,1,4,0,1,2,3

OK

+QLWADDOBJ: 0

+QLWURC: "ping",0

AT+QLWADDOBJ? //Query the current custom object

+QLWADDOBJ: 9,0,5,0,1,2,3,4

+QLWADDOBJ: 19,0,4,0,1,2,3

+QLWADDOBJ: 19,1,4,0,1,2,3

OK

//The application server has sent a read request to UE with the intention of reading the resource (9/0/0)

+QLWURC: "read",32191,9,0,0

AT+QLWRDRSP=32191,1,9,0,0,1,7,"quectel",0

OK

+QLWRDRSP: 0

//The server has sent a write request to UE with the intention of writing the resource (9/0/3)

+QLWURC: "write",32193,9,0,3,2,1,"31",0

AT+QLWWRRSP=32193,2

OK

+QLWWRRSP: 0

//The application server has sent an execute request to UE

+QLWURC: "execute",32198,9,0,4

AT+QLWEXERSP=32198,2

OK

+QLWEXERSP: 0

//The application server has observed the resource (9/0/0)

```
+QLWURC: "observe",32201,0,9,0,0

AT+QLWOBSRSP=32201,1,9,0,0,1,7,"quectel",0
OK

+QLWOBSRSP: 0
AT+QLWNOTIFY=9,0,0,1,10,"0123456789",0,1
OK

+QLWURC: "report",34180

+QLWNOTIFY: 0

+QLWURC: "report_ack",0,34180 //Notify the data to server successfully
```

5.5. Manually Trigger LwM2M Context Recovery Process

```
AT+CEREG?
+CEREG: 0,1 //EPS network registered. (+CEREG: 0,5 indicating registered roaming
service is also possible.)

OK
AT+CGPADDR=1
+CGPADDR: 1,100.88.40.249

OK
AT+QLWCFG="recovery_mode",1 //Set recovery mode to manually trigger.
OK

//First please check LwM2M status via AT+QLWSTATUS?, LwM2M information can be configured when
LwM2M status is "Not Registered".
AT+QLWCONFIG=1,"bootp.iot.t-mobile.com",5584,"urn:imei:867997030052996",900,0,"urn:imei:8
67997030052996","3083693CF1E6766A2408BFD88278249A2DB498B40A6BF6049300FD47DE325C
FC"
OK
AT+QLWREG //Send a registration request.
OK

+QLWURC: "lifetime_changed",30

+QLWURC: "min_period_changed",1
```

```
+QLWURC: "max_period_changed",3600

+QLWURC: "binding_changed","UQ"

+QLWURC: "bs_finished"

+QLWREG: 0 //Registered successfully.
//After waiting for some time, the module enters deep sleep.
//Wake up the module and <recovery_mode> was set to manual trigger mode.
F1: 0000 0000
V0: 0000 0000 [0001]
00: 0000 0000
U0: 0000 0001 [0000]
T0: 0000 001E
Leaving the BROM
+CPIN: READY
AT+QLWSTATUS?
+QLWSTATUS: 7 //LwM2M context needs to be recovered.

OK
AT+QLWRECOVER
OK

+QLWURC: "recovered",0 //Recovered successfully and then the module can interact with server
normally.
```

5.6. LwM2M Register to Leshan Server

LwM2M Client features can be tested on Leshan server. Please follow the steps below to register to Leshan server:

- In non-security mode:

```
AT+QLWCONFIG=0,"leshan.eclipse.org",5683,"urn:imei:866971030000717",900,3
OK
```

- In security mode with PSK:

```
AT+QLWCONFIG=0,"leshan.eclipse.org",5684,"urn:imei:867997030056338",900,0,"urn:imei:86799
7030056338","30313233343536373839"
OK
```

NOTE

If security mode is used, client security configuration should be added into the web interface of the server, according to steps illustrated below.

The Leshan public key (SubjectPublicKeyInfo der encoded)

Elliptic Curve parameters : secp256r1 [NIST P-256, X9.62 prime256v1] (1.2.840.10045.3.1.7)
 Public x coord : fcc28728c123b155be410fc1c0651da374fc6ebe7f96606e90d927d188894a73
 Public y coord : d2ffaa73957d76984633fc1cc54d0b763ca0559a9dff9706e9f4557dacc3f52a

Hex :
 3059301306072A8648CE3D020106082A8648CE3D03010703420004FCC28728C123B1
 55BE410FC1C0651DA374FC6EBE7F96606E90D927D188894A73D2FFAA73957D769846
 33FC1CC54D0B763CA0559A9DFF9706E9F4557DACC3F52A

Base64 :
 MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE/MKHKMEjsVM
 +QQ/BwGUdo3T8br5/1mBukNkn0Y1jSnPS/6pz1X12mEYz/B
 zFTQt2PKBVmp3/1wbp9FV9rMP1Kg==

Add new client security configuration

Client Endpoint	Security Mode	Security Information
866971030000710	Pre-Shared Key	Identity : 866971030000710 Key : 77c755bd8cbb400abf2f0c420eed2cb0
866971030000711	Pre-Shared Key	Identity : 866971030000711 Key : 77c755bd8cbb400abf2f0c420eed2cb1
866971030578730	Pre-Shared Key	Identity : bc26654321 Key : 1234567891

Figure 7: Click "Add new client security configuration" (in SECURITY Tab)

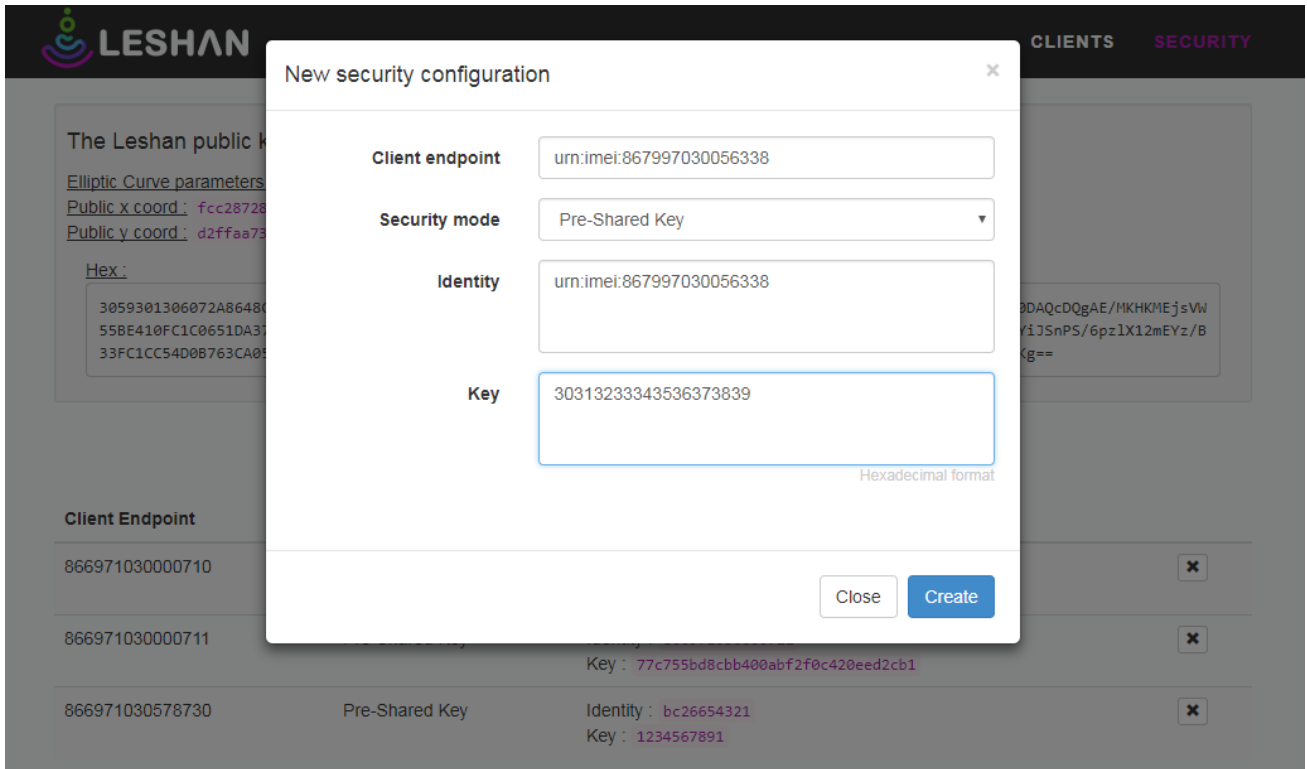


Figure 8: Create "New security configuration" on Leshan Server

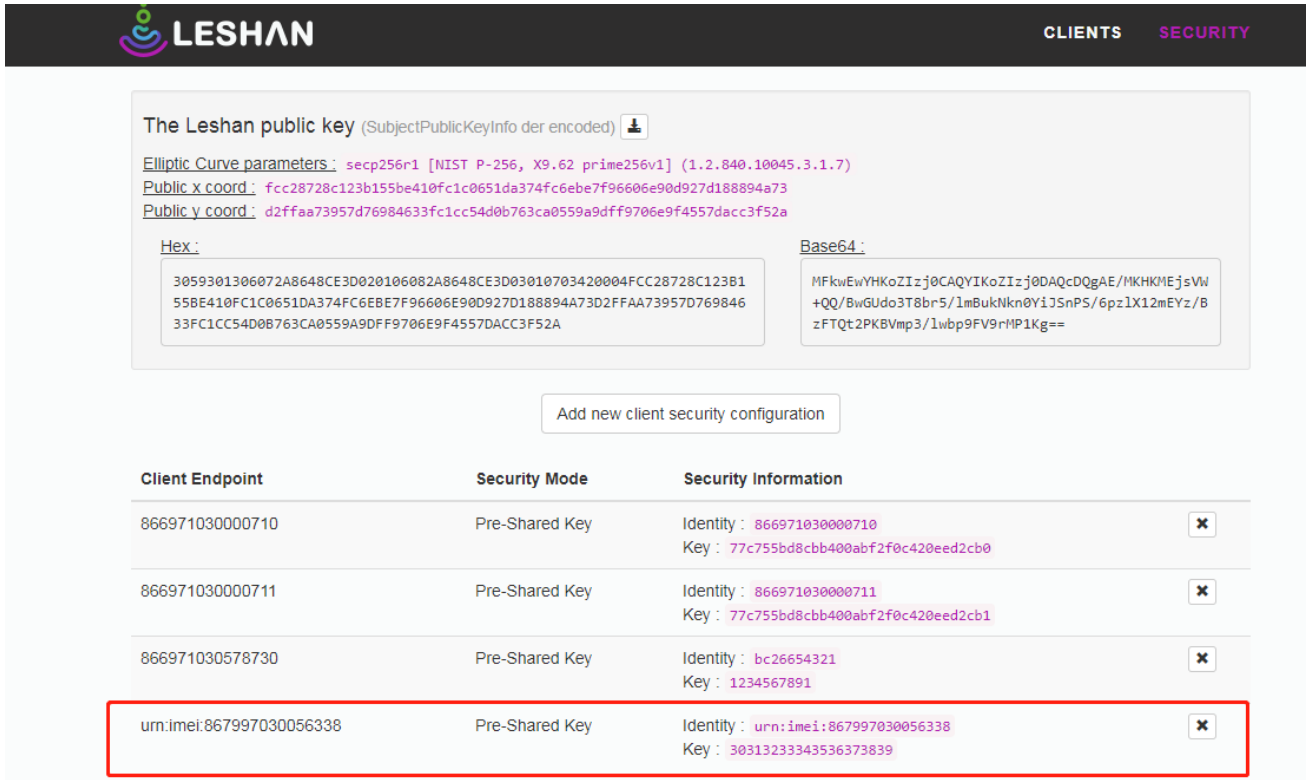


Figure 9: Check Created "New security configuration" on Leshan Server

```

AT+CEREG?
+CEREG: 0,1           //EPS network registered. (+CEREG: 0,5 indicating registered roaming
                       service is also possible.)

OK
AT+CGPADDR?
+CGPADDR: 1,100.97.37.69

OK
AT+QLWCONFIG=0,"leshan.eclipse.org",5684,"urn:imei:867997030056338",900,0,"urn:imei:86799
7030056338","30313233343536373839"
OK
AT+QLWREG
OK

+QLWREG: 0           //Registered successfully
AT+QLWSTATUS?       //Query the current LwM2M status
+QLWSTATUS: 2       //LwM2M status: registered

OK
    
```

After successful connection to Leshan server, the server will show various operation permissions for each resource on the webpage. For instance, the "Lifetime" value can be read through clicking "Read" button on the webpage, and any resource value can be updated through clicking "Write" button and then inputting a new value on the webpage.

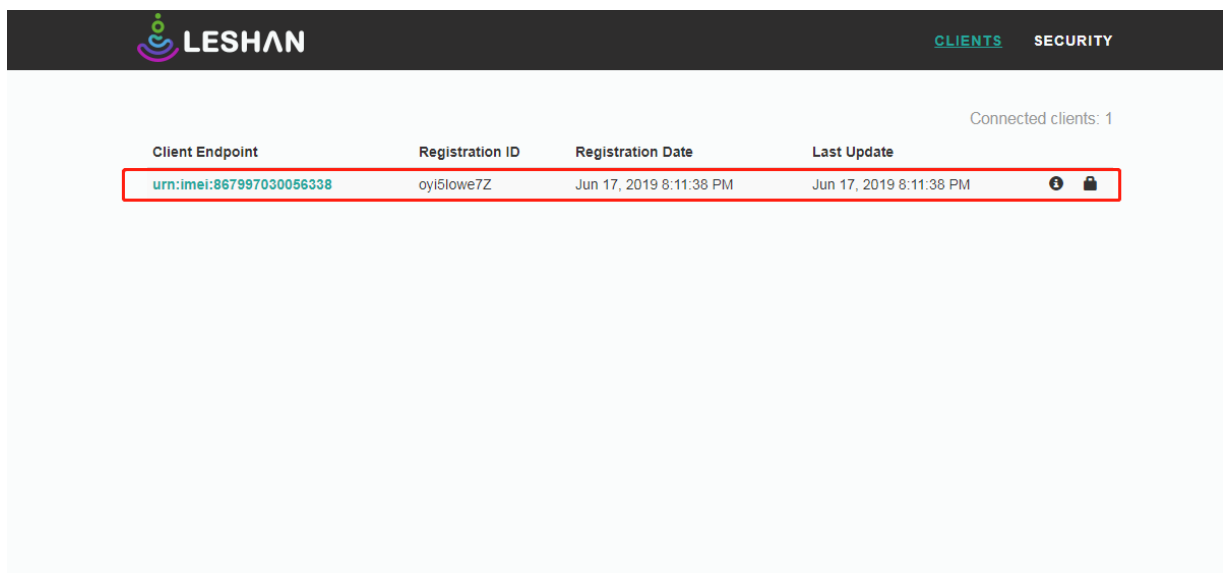


Figure 10: Online Client on Leshan Server

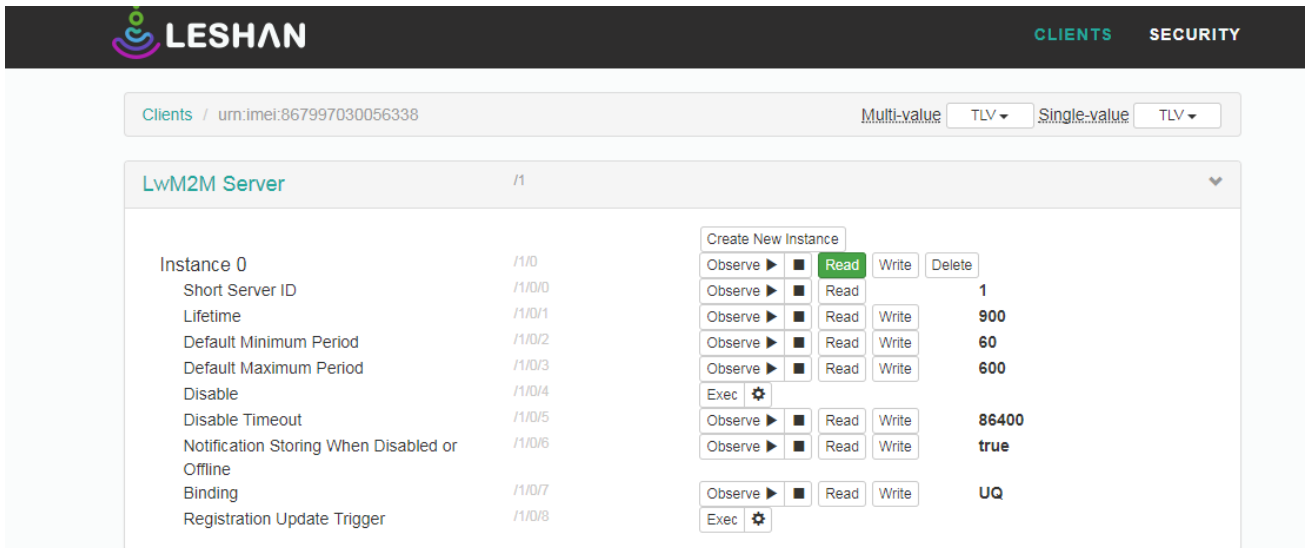


Figure 11: "Read" Operation on Leshan Server

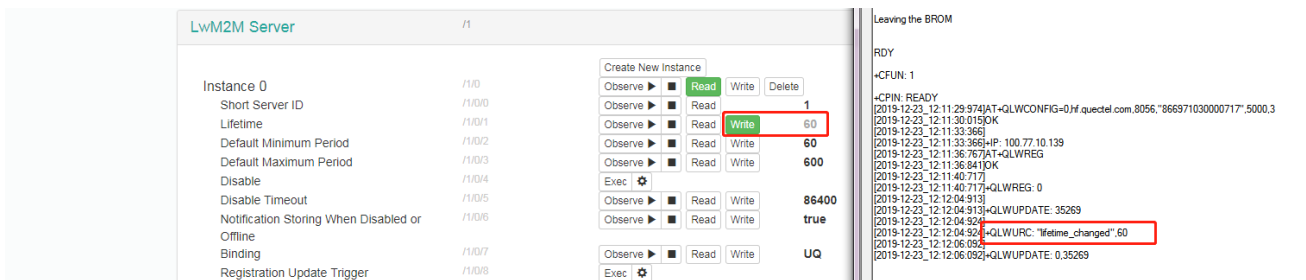


Figure 12: "Write" Operation on Leshan Server

5.7. LwM2M Updates (Leshan Server)

Periodically or based on certain events within the LwM2M Client or initiated by the LwM2M Server, the LwM2M Client updates its registration information with a LwM2M Server by sending an "Update" operation to the LwM2M Server.

5.7.1. LwM2M Server Update Operation

The "Update" operation can be initiated by the LwM2M Server via an "Execute" operation on the "Registration Update Trigger" Resource of the LwM2M Server Object. The LwM2M Client can perform an "Update" operation to refresh the lifetime of its registration to the LwM2M Server.

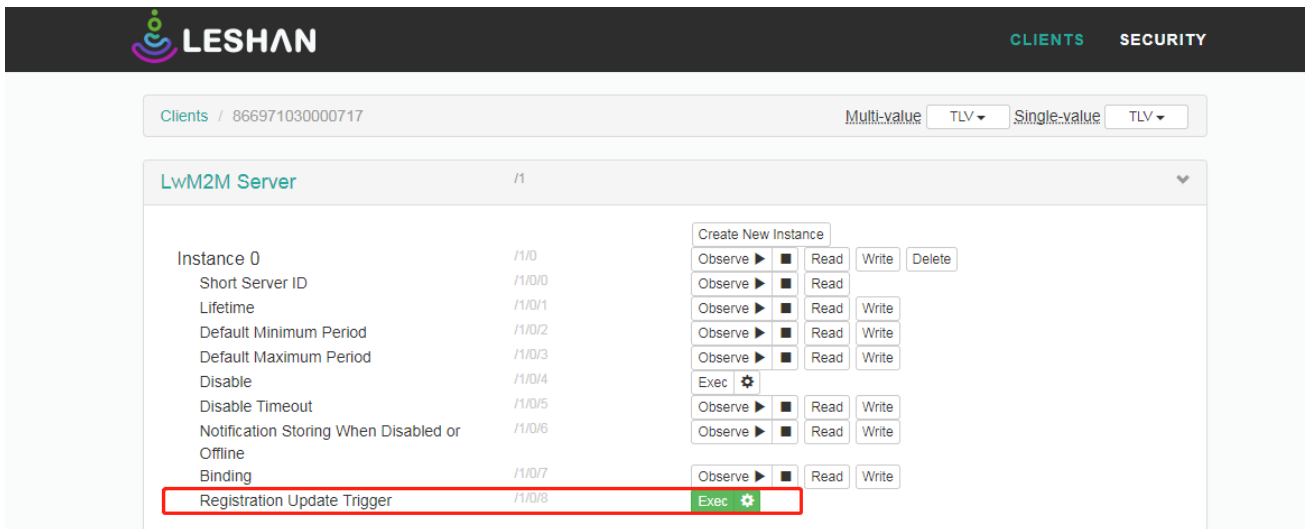


Figure 13: "Execute" Operation on Leshan Server

+QLWURC: "ping",0 // "Update" operation initiated by the LwM2M Server successfully

5.7.2. LwM2M Client Update Operation

When any of the parameters listed in the table below changes, the LwM2M Client must send an "Update" operation to the LwM2M Server. The "Update" operation must contain only the following parameters which have changed compared to the last registration parameters sent to the LwM2M Server.

Table 2: Update Parameters

Parameter	Required
Lifetime	No
Binding Mode	No
SMS Number	No
Objects and Object Instances	No

NOTE

SMS number updating is not supported currently.

Three common LwM2M Client update operations are provided in the subsequent chapters.

5.7.2.1. Extend the Lifetime of a Registration

In this case the LwM2M Client sends an "Update" operation with no parameters or with lifetime.

The figure below shows an example exchange where the LwM2M Client sends an "Update" operation that only refreshes the registration, i.e. the message does not contain any parameter. With the second "Update" the Client changes the lifetime field to 6000 (seconds) and hence the **<lifetime>** parameter is included in the message.

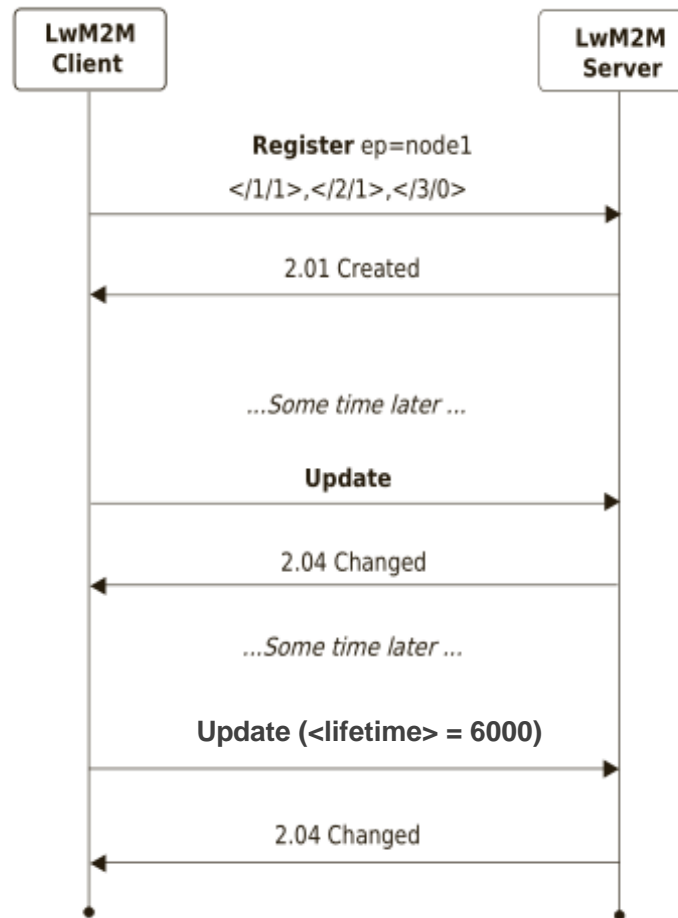


Figure 14: Update Example Flow

//The LwM2M Client has registered to Leshan Server successfully, and the default update period is 30 seconds. The module will update periodically (once every 30 seconds), and it can change the lifetime by **AT+QLWUPDATE**.

AT+CEREG?

+CEREG: 0,1

//EPS network registered. (+CEREG: 0,5 indicating registered roaming service is also possible.)

OK

AT+CGPADDR?

+CGPADDR: 1,100.97.37.69

OK

AT+QLWCONFIG=0,"leshan.eclipse.org",5683,"urn:imei:866971030000717",30,3

OK

AT+QLWREG

OK

+QLWREG: 0 //Registered successfully

+QLWURC: "ping",0 //Lifetime will timeout, and the module will trigger registration auto-update

AT+QLWUPDATE=0,200

+QLWUPDATE: 35270

OK

+QLWUPDATE: 0,35270 //Updated successfully

No.	Time	Source	Destination	Protocol	Length	Info
454	2019/10/11 11:32:44	36.113.226.200	172.172.169.176	CoAP	305	CON, MID:35268, POST, TKN:c6 89 0a 00, /rd/lm2m=1.08ep-urn:imei:86697103000071788-UQ81t-30
455	2019/10/11 11:32:44	172.172.169.176	36.113.226.200	CoAP	64	ACK, MID:35268, 2.01 Created, TKN:c6 89 0a 00, /rd
2540	2019/10/11 11:33:00	36.113.226.200	172.172.169.176	CoAP	64	CON, MID:35269, POST, TKN:c6 89 1d 00, /rd/fff7f185ps3 Update Registraion Without Parameters
2543	2019/10/11 11:33:00	172.172.169.176	36.113.226.200	CoAP	50	ACK, MID:35269, 2.04 Changed, TKN:c6 89 1d 00, /rd/fff7f185ps3
2084	2019/10/11 11:33:02	36.113.226.200	172.172.169.176	CoAP	73	CON, MID:35270, POST, TKN:c6 89 1d 00, /rd/fff7f185ps3?lt=200 Update Registraion With Lifetime
2085	2019/10/11 11:33:02	172.172.169.176	36.113.226.200	CoAP	50	ACK, MID:35270, 2.04 Changed, TKN:c6 89 1d 00, /rd/fff7f185ps3

Figure 15: Update Registration

5.7.2.2. Update the Binding Mode of a Registration

//The LwM2M Client has registered to Leshan Server successfully.

AT+QLWUPDATE=1,0 //Update binding mode to UDP mode

+QLWUPDATE: 38296

OK

+QLWUPDATE: 0,38296 //Updated successfully

No.	Time	Source	Destination	Protocol	Length	Info
275	2019/10/11 11:36:19	36.113.245.216	172.172.169.176	CoAP	987	CON, MID:38295, POST, TXN:97 95 0a 00, //rd/lm2m=1.0&epmurn:imei:866971030000717&cid=3800
274	2019/10/11 11:36:19	172.172.169.176	36.113.245.216	CoAP	64	ACK, MID:38295, 2.01.Created, TXN:92 95 0a 00, //rd
832	2019/10/11 11:36:23	36.113.245.216	172.172.169.176	CoAP	68	CON, MID:38296, POST, TXN:98 95 0e 00, //rd/Uup43KizRb=U
833	2019/10/11 11:36:23	172.172.169.176	36.113.245.216	CoAP	58	ACK, MID:38296, 2.04.Changed, TXN:98 95 0e 00, //rd/Uup43KizR Update Binding Mode

```

0000 48 4d 7e ca 4b 2c 08 08 39 45 22 5b 08 00 45 00  H9-K, f 9ESP-E
0010 00 36 00 03 00 00 f4 11 56 0d 24 71 f5 08 ac ac  6.....V.Sq...
0020 a9 b0 ca 91 1f 78 00 22 c8 ee 44 02 95 98 98 95  ....x."D.....
0030 0e 00 62 72 04 0e 55 41 75 70 34 4a 4b 57 7a 52  ...rd UK up43KizR
0040 81 62 9d 95  CBU
  
```

Figure 16: Update Binding Mode

5.7.2.3. Add/Remove Objects and Object Instances

In this case the LwM2M Client sends an "Update" with a body listing the complete list of objects and object instances.

As illustrated below, the LwM2M Client starts with an initial registration with lifetime of 3000 seconds. Later, **AT+QLWADDOBJ** is executed to add objects 3303/0 and 3303/1. With the "Update" operation both of the LwM2M Client and the LwM2M Server include the new list of Objects and Object Instances.

```

AT+CEREG?
+CEREG: 0,1 //EPS network registered. (+CEREG: 0,5 indicating registered roaming service is also possible.)

OK
AT+CGPADDR?
+CGPADDR: 1,100.97.37.69

OK
AT+QLWCONFIG=0,"leshan.eclipse.org",5683,"urn:imei:866971030000717",3000,3
OK
AT+QLWREG
OK
+QLWREG: 0 //Registered successfully
  
```

AT+QLWADDOBJ=3303,0,3,5601,5602,5603

OK

+QLWADDOBJ: 0

+QLWURC: "ping",0 //Object list updated successfully

No.	Time	Source	Destination	Protocol	Length	Info
1860	2019/169 14:18:54	36.113.238.130	172.172.169.176	CoAP	296	CON, MID:32242, POST, TKN:f2 7d 0b 00, /rd/lwm2m=1.0&epurnn:imei:866971030000717&b=U&t=3000
1861	2019/169 14:18:54	172.172.169.176	36.113.238.130	CoAP	64	ACK, MID:32242, 2.01 Created, TKN:f2 7d 0b 00, /rd
2539	2019/169 14:18:59	36.113.238.130	172.172.169.176	CoAP	264	CON, MID:32243, POST, TKN:f3 7d 10 00, /rd/KsyNlgCF4k
2540	2019/169 14:18:59	172.172.169.176	36.113.238.130	CoAP	50	ACK, MID:32243, 2.04 Changed, TKN:f3 7d 10 00, /rd/KsyNlgCF4k

Update Object & Object Instance List

Figure 17: Update Object & Object Instance List

5.8. LwM2M Custom Object (Leshan Server)

Currently, BC66/BC66-NA supports instance or resource level operations only for custom objects.

5.8.1. Observe Operation

//The LwM2M Client has registered to Leshan Server successfully, and custom object 3303 is observed by the server.

+QLWURC: "observe",16987,0,3303,0,5601 //The Server initiates an observation request for resource level (3303/0/5601) changes

AT+QLWOBSRSP=16987,1,3303,0,5601,4,4,88.88,0

OK

+QLWOBSRSP: 0 //Observed successfully

```
+QLWURC: "observe",16988,0,3303,0,-1 //The Server initiates an observation request for instance
                                         level (3303/0) changes
AT+QLWOBSRSP=16988,1,3303,0,5601,4,4,88.88,0
OK
+QLWOBSRSP: 0 //Observed successfully
```

5.8.2. Notify Operation

```
//The LwM2M Client has registered to Leshan Server successfully and object 3303 has been observed
successfully by the server.
AT+QLWNOTIFY=3303,0,5601,4,4,88.88,0,1 //Send CON data
OK
+QLWURC: "report",17111
+QLWNOTIFY: 0
+QLWURC: "report_ack",0,17111 //Successfully responded to the notify request
```

5.8.3. Read Operation

```
//The LwM2M Client has registered to Leshan Server successfully and a read request related to
3303/0/5601 has been issued from the LwM2M Server.
+QLWURC: "read",17123,3303,0,5601
AT+QLWRDRSP=17123,1,3303,0,5601,4,4,66.66,0
OK
+QLWRDRSP: 0 //Successfully responded to the read request
```

5.8.4. Execute Operation

```
//The LwM2M Client has registered to Leshan Server successfully and an execute request related to
3303/0/5605 has been issued from the LwM2M Server.
+QLWURC: "execute",27452,3303,0,5605
AT+QLWEXERSP=27452,2
OK
+QLWEXERSP: 0 //Successfully responded to the execute request
```

6 Error Codes

Table 2: Summary of LwM2M <err>

<err>	Description
0	Operation successful
1	Other error
2	Parameter number error
3	Parameter value error
4	Not register error
7	Disable error
13	Data length odd error
15	Not ready receive error
32	Keep connecting error
33	Already registered error
34	Create lwm2m error

Table 3: Summary of <status_code>

<status_code>	Description
0	Success
1	Timeout
2	Packet not sent
3	Recovered failed

4	Update failed
9	Reset
10	Bad request
13	Forbidden
14	Not found
22	Precondition failed

7 Appendix A References

Table 4: References

SN	References
[1]	3GPP 27.007 AT Command Set for User Equipment
[2]	3GPP 27.005 Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)
[3]	http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html
[4]	http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180710-A/
[5]	https://www.omaspecworks.org/
[6]	OMA-ERELD-LightweightM2M-V1_1-20180710-A
[7]	OMA-RD-LightweightM2M-V1_1-20180710-A
[8]	OMA-TS-LightweightM2M_Core-V1_1-20180710-A
[9]	OMA-TS-LightweightM2M_Transport-V1_1-20180710-A
[10]	https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/

Table 5: Terms and Abbreviations

Abbreviation	Description
3GPP	3 rd Generation Partnership Project
CoAP	Constrained Application Protocol
IMEI	International Mobile Equipment Identity
NB-IoT	Narrowband Internet of Thing
UDP	User Datagram Protocol

UE	User Equipment
URC	Unsolicited Result Code
CON	Confirmable
NON	Non-Confirmable
IoT	Internet of Things
LwM2M	Lightweight Machine to Machine
OMA	Open Mobile Alliance
REST	Representational State Transfer
PSK	Pre-shared Key
