

Enabler Test Specification for Lightweight M2M

Approved Version 1.0.2 – 15 Aug 2018

Open Mobile Alliance
OMA-ETS-LightweightM2M-V1_0_2-20180815-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2018 Open Mobile Alliance All Rights Reserved.

Used with the permission of the Open Mobile Alliance under the terms set forth above.

Contents

1.	SCOPE	6
2.	REFERENCES	7
2.1	NORMATIVE REFERENCES	7
2.2	INFORMATIVE REFERENCES	7
3.	TERMINOLOGY AND CONVENTIONS	8
3.1	CONVENTIONS	8
3.2	DEFINITIONS	8
3.3	ABBREVIATIONS	8
4.	INTRODUCTION	9
5.	LIGHTWEIGHTM2M CONFORMANCE TEST CASES	10
6.	LIGHTWEIGHTM2M INTEROPERABILITY TEST CASES	11
6.1	LWM2M CORE SPECIFICATION TEST CASES [0-999]	11
6.1.1	Bootstrap Interface: [0-99]	11
6.1.1.1	<i>LightweightM2M-1.0-int-0 – Client Initiated Bootstrap</i>	11
6.1.1.2	<i>LightweightM2M-1.0-int-1 – Client Initiated Bootstrap Full (PSK)</i>	12
6.1.1.3	<i>LightweightM2M-1.0-int-2 – Client Initiated Bootstrap Full (Cert)</i>	13
6.1.1.4	<i>LightweightM2M-1.0-int-3 – Simple Bootstrap from Smartcard</i>	15
6.1.2	Registration Interface [100-199]	16
6.1.2.1	<i>LightweightM2M-1.0-int-101 – Initial Registration</i>	16
6.1.2.2	<i>LightweightM2M-1.0-int-102 – Registration Update</i>	17
6.1.2.3	<i>LightweightM2M-1.0-int-103 – Deregistration</i>	17
6.1.2.4	<i>LightweightM2M-1.0-int-104 – Registration Update Trigger</i>	18
6.1.3	Device management & Service Enablement Interface [200-299]	19
6.1.3.1	<i>LightweightM2M-1.0-int-201 – Querying basic information in Plain Text format</i>	19
6.1.3.2	<i>LightweightM2M-1.0-int-202 – Querying basic information in Opaque format</i>	19
6.1.3.3	<i>LightweightM2M-1.0-int-203 – Querying basic information in TLV format</i>	20
6.1.3.4	<i>LightweightM2M-1.0-int-204 – Querying basic information in JSON format</i>	20
6.1.3.5	<i>LightweightM2M-1.0-int-205 – Setting basic information in Plain Text format</i>	21
6.1.3.6	<i>LightweightM2M-1.0-int-210 – Setting basic information in Opaque format</i>	21
6.1.3.7	<i>LightweightM2M-1.0-int-215 – Setting basic information in TLV format</i>	21
6.1.3.8	<i>LightweightM2M-1.0-int-220 – Setting basic information in JSON format</i>	22
6.1.3.9	<i>LightweightM2M-1.0-int-241 – Executable Resource: Rebooting the device</i>	24
6.1.3.10	<i>LightweightM2M-1.0-int-260 – Discover Command</i>	24
6.1.3.11	<i>LightweightM2M-1.0-int-270 – Create Object Instance</i>	25
6.1.3.12	<i>LightweightM2M-1.0-int-271 – Create Multiple Resource Instance</i>	25
6.1.3.13	<i>LightweightM2M-1.0-int-290 – Delete Object Instance</i>	25
6.1.4	Information Reporting Interface [300-399]	26
6.1.4.1	<i>LightweightM2M-1.0-int-301 – Observation and Notification of parameter values</i>	26
6.1.4.2	<i>LightweightM2M-1.0-int-302 – Cancel Observations using Reset Operation</i>	26
6.1.4.3	<i>LightweightM2M-1.0-int-303 – Cancel observations using Observe with Cancel parameter</i>	27
6.1.5	Security [400-499]	29
6.1.5.1	<i>LightweightM2M-1.0-int-401 – UDP Channel Security – Pre-shared Key Mode</i>	29
6.1.5.2	<i>LightweightM2M-1.0-int-402 – UDP Channel Security – Certificate Mode</i>	29
6.1.5.3	<i>LightweightM2M-1.0-int-403 – UDP Channel Security – Certificate Mode (Server Identify Verification Failure)</i>	30
6.1.6	Core Specific Objects Test cases [500-999]	30
6.1.6.1	<i>Security Object (ID 0) [500-549]</i>	30
6.1.6.2	<i>Server Object (ID 1) [550-599]</i>	30
6.1.6.2.1	<i>LightweightM2M-1.0-int-551 – Access Check to the Resources</i>	31
6.1.6.2.2	<i>LightweightM2M-1.0-int-555 – Check of the Disable (De-Registration) capability</i>	31
6.1.6.2.3	<i>LightweightM2M-1.0-int-556 – Check of the Update Registration capability</i>	31
6.1.6.2.4	<i>LightweightM2M-1.0-int-560 – Delayed Report Notification</i>	31
6.1.6.2.5	<i>LightweightM2M-1.0-int-565 – Create Object Instance</i>	31
6.1.6.2.6	<i>LightweightM2M-1.0-int-566 – Delete Object Instance</i>	31
6.1.6.3	<i>Access Control Object (ID 2) [600-649]</i>	31
6.1.6.4	<i>LwM2M Device Object (ID 3) [650-699]</i>	31
6.1.6.4.1	<i>LightweightM2M-1.0-int-651 – Check Access to the Resources</i>	32
6.1.6.4.2	<i>LightweightM2M-1.0-int-652 – Querying the firmware version from the client</i>	33

6.1.6.4.3	LightweightM2M-1.0-int-655 – Check of the “Reboot” capability	33
6.1.6.4.4	LightweightM2M-1.0-int-656 – Check of the “Factory Reset” capability	33
6.1.6.4.5	LightweightM2M-1.0-int-657 – Check of the “Error Code” functionality	33
6.1.6.4.6	LightweightM2M-1.0-int-660 – Basic Observation and notification of Device Object Resources	33
6.1.6.4.7	LightweightM2M-1.0-int-661 – Extended Observation and notification of Device Object Resources	33
6.1.6.4.8	LightweightM2M-1.0-int-670 – Create Multiple Resource Instances	33
6.1.6.4.9	LightweightM2M-1.0-int-680 – Create Object Instance	33
6.1.6.4.10	LightweightM2M-1.0-int-685 – Delete Object Instance	34
6.1.6.5	<i>LwM2M Connectivity Monitoring (ID:4) [700-749]</i>	34
6.1.6.5.1	LightweightM2M-1.0-int-701 – Querying the readable resources of object	34
6.1.6.5.2	LightweightM2M-1.0-int-705 – Setting the writable resources	35
6.1.6.5.3	LightweightM2M-1.0-int-710 – Basic Observation and notification on Connectivity Monitoring Object Resources	36
6.1.6.5.4	LightweightM2M-1.0-int-711 – Extended Observation and notification of Connectivity Monitoring Object Resources	36
6.1.6.5.5	LightweightM2M-1.0-int-720 – Create Multiple Resource Instances	36
6.1.6.5.6	LightweightM2M-1.0-int-730 – Create Object Instance	36
6.1.6.5.7	LightweightM2M-1.0-int-735 – Delete Object Instance	36
6.1.6.6	<i>Firmware Update Object (ID 5) [750-799]</i>	37
6.1.6.6.1	LightweightM2M-1.0-int-751 – Querying the readable resources	37
6.1.6.6.2	LightweightM2M-1.0-int-755 – Setting the writable Resource Package	37
6.1.6.6.3	LightweightM2M-1.0-int-756 – Setting the writable Resource Package URI	38
6.1.6.6.4	LightweightM2M-1.0-int-760 – Basic Observation and notification on Firmware Update Object Resources	39
6.1.6.6.5	LightweightM2M-1.0-int-770 – Successful Firmware update (via COAP)	40
6.1.6.6.6	LightweightM2M-1.0-int-771 – Successful Firmware update (via alternative mechanism)	41
6.1.6.6.7	LightweightM2M-1.0-int-772 – Error Case 1: firmware installation without downloaded package	42
6.1.6.6.8	LightweightM2M-1.0-int-773 – Error Case 2: shortage of storage memory	43
6.1.6.6.9	LightweightM2M-1.0-int-774 – Error Case 3: out of memory	44
6.1.6.6.10	LightweightM2M-1.0-int-775 – Error Case 4: Connection lost during download (package URI)	45
6.1.6.6.11	LightweightM2M-1.0-int-776 – Error Case 5: Package Integrity check failure	46
6.1.6.6.12	LightweightM2M-1.0-int-777 – Error Case 6: unsupported package type	47
6.1.6.6.13	LightweightM2M-1.0-int-778 – Error Case 7: invalid URI (package URI)	48
6.1.6.6.14	LightweightM2M-1.0-int-779 – Error Case 8: Unsuccessful Firmware Update	49
6.1.6.6.15	LightweightM2M-1.0-int-780 – Error Case 9: Unsupported protocol	50
6.1.6.7	<i>LwM2M Location Object (ID:6) [800-849]</i>	51
6.1.6.7.1	LightweightM2M-1.0-int-801 – Querying the readable resources of object	51
6.1.6.7.2	LightweightM2M-1.0-int-805 – Setting the writable resources	51
6.1.6.7.3	LightweightM2M-1.0-int-810 – Basic Observation and notification on Location Object Instance	51
6.1.6.7.4	LightweightM2M-1.0-int-811 – Extended Observation and notification of Location Object Instance	52
6.1.6.7.5	LightweightM2M-1.0-int-820 – Create Multiple Resource Instances	52
6.1.6.7.6	LightweightM2M-1.0-int-830 – Create Object Instance	52
6.1.6.7.7	LightweightM2M-1.0-int-835 – Delete Object Instance	52
6.1.6.8	<i>Connectivity Statistics (ID 7) [900-949]</i>	53
6.1.6.8.1	LightweightM2M-1.0-int-901 – Querying a Data Collection from Connectivity Object Instance	53
6.1.6.8.2	LightweightM2M-1.0-int-905 – Setting the writable resources	53
6.1.6.8.3	LightweightM2M-1.0-int-910 – Basic Observation and notification on Connectivity Monitoring Object Instance	54
6.1.6.8.4	LightweightM2M-1.0-int-911 – Extended Observation and notification of Connectivity Statistics Object Instance	55
6.1.6.8.5	LightweightM2M-1.0-int-920 – Create Multiple Resource Instances	55
6.1.6.8.6	LightweightM2M-1.0-int-930 – Create Object Instance	55
6.1.6.8.7	LightweightM2M-1.0-int-935 – Delete Object Instance	55
6.2	MULTI-SERVERS CONTEXT [950-999]	56
6.2.1	LightweightM2M-1.0-int-950 – Multi-Servers Registration	56
6.2.2	LightweightM2M-1.0-int-951 – Multi-Servers & Attributes	56
6.3	LWM2M ADDITIONAL OBJECTS TEST CASES [1000-1999]	57
6.3.1	Lock and Wipe Object (ID 8) [1000-1099]	57
6.3.2	Software Management Object (ID 9) [1100-1199]	57
6.3.3	Connectivity Management Objects (ID 10, 11, 12, 13) [1200-1499]	58
6.3.3.1	<i>Cellular Network Connectivity ID:10 [1200-1249]</i>	58
6.3.3.1.1	LightweightM2M-1.0-int-1200 – Querying the readable resources of Object ID:10	58
6.3.3.1.2	LightweightM2M-1.0-int-1201 – Querying the readable resources of Object ID:10 in version 1.1	58
6.3.3.1.3	LightweightM2M-1.0-int-1202 – Setting the Power Saving Mode Resources of Object ID:10 (version 1.1 only)	59
6.3.3.1.4	LightweightM2M-1.0-int-1203 – Observation and notification on Object ID:10 related to Power Saving Mode Resources (version 1.1 only)	60
6.3.3.1.5	LightweightM2M-1.0-int-1204 – Observation and notification on Object ID: 10 related to timers	61

6.3.3.2	APN connection profile ID:11 [1250-1299]	61
6.3.3.2.1	LightweightM2M-CONMGMT-1.0-int-1250 – APN configuration.....	61
6.3.3.2.2	WLAN Connectivity ID:12 [1300-1349].....	62
6.3.3.2.3	Bearer Selection ID:13 [1350-1399].....	62
6.3.4	Device Capability Management Object (ID 15) [1500-1599].....	63
6.3.5	Portfolio Object (ID 16) [1600-1699].....	63
6.3.5.1	LightweightM2M-1.0-int-1600 – Querying the readable resources of object	63
6.3.5.2	LightweightM2M-1.0-int-1605 – Setting the writable resources.....	64
6.3.5.3	LightweightM2M-1.0-int-1610 – Basic Observation and notification of a Portfolio Object Instance	64
6.3.5.4	LightweightM2M-1.0-int-1611 – Extended Observation and notification of a Portfolio Object Instance	64
6.3.5.5	LightweightM2M-1.0-int-1620 – Create Multiple Resource Instances	64
6.3.5.6	LightweightM2M-1.0-int-1630 – Create Portfolio Object Instance.....	64
6.3.5.7	LightweightM2M-1.0-int-1635 – Delete Portfolio Object Instance	65
6.3.6	BinaryAppDataContainer Object (ID 19) [1900-1999]	66
6.3.6.1	LightweightM2M-1.0-int-1900 – Observation and notification on Object ID: 19	66
6.3.6.2	LightweightM2M-1.0-int-1901 – Setting the writable Resource Data	67
6.3.7	Event Log Object (ID 20) [2000-2099]	67
6.3.7.1	LightweightM2M-1.0-int-2000 – Control the Log Function	67
6.3.7.2	LightweightM2M-1.0-int-2001 – Querying the readable resources.....	68
APPENDIX A.	CHANGE HISTORY (INFORMATIVE).....	69
A.1	APPROVED VERSION HISTORY	69
APPENDIX B.	ADDITIONAL INFORMATION	70
B.1	EXAMPLE OF TEST CONFIGURATION AND SETUP	70
APPENDIX C.	LWM2M CONFIGURATIONS	71
C.1	BASIC CONFIGURATION 1	71
C.2	BASIC CONFIGURATION 2	72
C.3	CONFIGURATION 3	73
C.4	CONFIGURATION 4	74
C.5	CONFIGURATION 5	75
C.6	CONFIGURATION 6	76
C.7	CONFIGURATION 7	77
C.8	CONFIGURATION 8	78
C.9	CONFIGURATION 9	79
C.10	BASIC CONNECTIVITY MANAGEMENT CONFIGURATION 10	80
C.11	BASIC CONNECTIVITY MANAGEMENT CONFIGURATION 11 SUPPORTING OBJECT ID:10 VERSION 1.1	81
C.12	CONFIGURATION 12	82
C.13	BOOTSTRAP SERVER CONTACT CONFIGURATION.....	83
C.14	MULTI-SERVERS CONTEXT INITIAL CONFIGURATION.....	84
C.15	BINARY APPDATA CONTAINER CONFIGURATION	86
C.16	EVENT LOG CONFIGURATION	87
C.17	BASIC CONFIGURATION 17	88
C.18	BASIC CONFIGURATION 18	89
APPENDIX D.	CORE TEST COVERAGE.....	90
APPENDIX E.	ENABLER VALIDATION PLAN.....	95
E.1	ENTRY CRITERIA FOR TESTFEST	95
E.2	ENABLER VALIDATION TEST CASES	95

Tables

Table 1: Listing of Tests for Entry Criteria for TestFest	95
Table 2: Enabler Validation Test Cases.....	95

1. Scope

This document describes in detail available test cases for LightweightM2M as specified in OMA-TS-LightweightM2M-V1_0.

The test cases are split in two categories, conformance and interoperability test cases.

The conformance test cases are aimed to verify the adherence to normative requirements described in the technical specifications.

The interoperability test cases are aimed to verify that implementations of the specifications work satisfactory.

If either conformance or interoperability tests do not exist at the creation of the test specification this part should be marked not available.

2. References

2.1 Normative References

- [3GPP-TS_23.003] 3GPP TS 23.003 “Numbering, addressing and identification”
- [CoAP] Shelby, Z., Hartke, K., Bormann, C., and B. Frank, “Constrained Application Protocol (CoAP)”, draft-ietf-core-coap-18, Jun 2013.
- [ETSI TS 102.221] “Smart Cards; UICC-Terminal interface; Physical and logical characteristics”, (ETSI TS 102 221 release 11), [URL:http://www.etsi.org/](http://www.etsi.org/)
- [GlobalPlatform SCP 02] GlobalPlatform v2.2.1 - January 2011 - Appendix E: Secure Channel Protocol 02 (SCP 02)
- [IOPPROC] “OMA Interoperability Policy and Process”, Version 1.13, Open Mobile Alliance™, OMA-IOP-Process-V1_13, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [LwM2M-AD] “Lightweight Machine to Machine Architecture”, Open Mobile Alliance™, OMA-AD-LightweightM2M-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [LwM2M-TS] “Lightweight Machine to Machine Technical Specification”, Open Mobile Alliance™, OMA-TS-LightweightM2M-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [OBSERVE] Hartke, K. “Observing Resources in CoAP”, draft-ietf-core-observe-11 (work in progress), Oct 2013.
- [PKCS#15] PKCS #15 v1.1: Cryptographic Token Information Syntax Standard”, RSA Laboratories, June 6, 2000. [URL:ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-15/pkcs-15v1_1.pdf](ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-15/pkcs-15v1_1.pdf)
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, [URL:http://www.ietf.org/rfc/rfc2119.txt](http://www.ietf.org/rfc/rfc2119.txt)
- [RFC2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997, [URL:http://www.ietf.org/rfc/rfc2234.txt](http://www.ietf.org/rfc/rfc2234.txt)
- [RFC4122] “A Universally Unique Identifier (UUID) URN Namespace”, P. Leach, et al. July 2005, [URL:http://www.ietf.org/rfc/rfc4122.txt](http://www.ietf.org/rfc/rfc4122.txt)
- [RFC5246] The Transport Layer Security (TLS) Protocol Version 1.2
- [RFC5289] TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)
- [RFC5487] Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode
- [RFC6347] Rescorla, E. and N. Modadugu, “Datagram Transport Layer Security Version 1.2”, [RFC6347](https://tools.ietf.org/html/rfc6347), January 2012.
- [RFC6655] McGrew, D. and D. Bailey, “AES-CCM Cipher Suites for TLS”, RFC6655, July 2012.
- [RFC6690] Shelby, Z. “Constrained RESTful Environments (CoRE) Link Format”, RFC6690, Aug 2012.

2.2 Informative References

- [DMREPPRO] “OMA Device Management Representation Protocol, Version 1.3”. Open Mobile Alliance™. OMA-TS-DM_RepPro-V1_3. [URL:http://www.openmobilealliance.org](http://www.openmobilealliance.org/)
- [OMADICT] “Dictionary for OMA Specifications”, Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope”, are normative, unless they are explicitly indicated to be informative.

The following numbering scheme is used:

xxx-y.z-con-number where:

xxx	Name of enabler, e.g. MMS or Browsing
y.z	Version of enabler release, e.g. 1.2 or 1.2.1
'con'	Indicating this test is a conformance test case
number	Leap number for the test case

Or

xxx-y.z-int-number where:

xxx	Name of enabler, e.g. MMS or Browsing
y.z	Version of enabler release, e.g. 1.2 or 1.2.1
'int'	Indicating this test is a interoperability test case
number	Leap number for the test case

3.2 Definitions

Application	1) The customer's realisation of a service through M2M - e.g. satnav, smart metering 2) Individual software components that run on top of the device's operating system. Access Point Name
Device	The hardware that is realising a function for the customer e.g. a smart meter.
Module	A modular component of a device e.g. the radio module housing the SIM

3.3 Abbreviations

API	Application Programming Interface
APN	Access Point Name
CoAP	Constrained Application Protocol
CON	Conformance
DM	Device Management
GDSP	Global Data Service Platform
GUI	Graphical User Interface
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IOP	Interoperability
LwM2M	Lightweight Machine to Machine (refers to this OMA enabler)
M2M	Machine to Machine
MSISDN	Mobile Station International Subscriber Directory Number
OMA	Open Mobile Alliance
OpCo	Operating Company
OS	Operating System
SIM	Subscriber Interface Module
UI	User Interface

4. Introduction

The purpose of this document is to provide test cases for LightweightM2M Enabler Release V1.0.

The implementation of some features is optional for the Clients and/or the Servers in the LightweightM2M Enabler. The tests associated with these optional features are marked as "(Includes Optional Features)" in the test specification.

The following items on an overall level are needed to adequately test the LwM2M enabler:

- A LwM2M Server
- A LwM2M client e.g. embedded in a M2M device or module connected via UDP and SMS with the LwM2M Server

The LwM2M enabler tests are carried out using the LwM2M protocol and objects, and using the underlying protocols such as [CoAP].

The four data formats used in that document, namely Plain Text, Opaque, TLV and JSON data formats are respectively associated to the text/plain, application/octet-stream, application/vnd.oma.lwm2m+tlv, application/vnd.oma.lwm2m+json Media Types referred by LwM2M TS 1.0 and registered in IANA.

5. LightweightM2M Conformance Test Cases

None.

6. LightweightM2M Interoperability Test Cases

6.1 LwM2M Core Specification Test cases [0-999]

6.1.1 Bootstrap Interface: [0-99]

6.1.1.1 LightweightM2M-1.0-int-0 – Client Initiated Bootstrap

Test Case Id	LightweightM2M-1.0-int- 0
Test Object	Client and Server
Test Case Description	Test the Client capability to connect the Bootstrap Server according to the Client Initiated Bootstrap Mode
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.13 as defined in Annex C ○ The Client is supporting the Client Initiated Bootstrap Mode ○ The Client doesn't contain any Instance of LwM2M Server Object ID:1 (minimum condition to trigger a Bootstrap request message when the device is on)
Test Procedure	<p>The Client requests the Bootstrap Server to setup a configuration enabling the Client to contact a LwM2M Server.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Without Instance of Server Object, the Client performs a BOOTSTRAP-REQUEST (CoAP POST /bs?ep{Endpoint Client Name}) in using the Resource values of the Instance 1 of Security Object ID:0 to contact the Bootstrap Server 2. The Bootstrap Server updates the Configuration in the Client (C13+C1) in performing two BOOTSTRAP-WRITE operations (CoAP PUT /0, CoAP PUT /1) in using the set of values defined in 0-SetOfValue 3. The Bootstrap Server performs a BOOTSTRAP-DISCOVER operation (CoAP GET Accept:40 /) to verify the Client setup 4. The Bootstrap Server performs a BOOTSTRAP-FINISH operation (CoAP POST /bs) to end-up that BS phase
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Bootstrap Server received a Success Message (“2.04” Changed) related to the BOOTSTRAP-REQUEST of the Client B. In test step 2., Client received WRITE operation(s) to update the initial configuration with the values contains in 0-SetOfValue C. In test step 2., Server received Success (“2.04” Changed) message(s) from Server related to WRITE operation(s) D. In test step 3., the Bootstrap Server received the Success message (“2.05” Content) along with the payload related to the BOOTSTRAP-DISCOVER request and containing : <ul style="list-style-type: none"> lwm2m="1.0",</0/0>;ssid=1,</0/1>, </1/0>;ssid=1,</3/0> E. In test step 4., the Bootstrap Server received the Success message (“2.04” Changed) : no inconsistency detected

0-SetOfValues definition

lwm2m+json version

```

{“bn”：“/”,
  “e”:[
    {“n”：“0/0/0”,“sv”:<LwM2M Server URI>},
    {“n”：“0/0/1”,“bv”:false},
    {“n”：“0/0/2”,“v”：0},
    {“n”：“0/0/3”,“sv”:<PSK Identity>}, // opaque in Base64
    {“n”：“0/0/4”,“sv”:<n/a> },
    {“n”：“0/0/5”,“sv”:<Secret Key>}, // opaque in Base64
    {“n”：“0/0/10”,“v”：1},
    {“n”：“1/0/0”,“v”：1},
    {“n”：“1/0/1”,“v”：86400},
    {“n”：“1/0/6”,“bv”:false},
    {“n”：“1/0/7”,“sv”：“U”},
  ]
}

```

6.1.1.2 LightweightM2M-1.0-int-1 – Client Initiated Bootstrap Full (PSK)

Test Case Id	LightweightM2M-1.0-int- 1
Test Object	Client and Server
Test Case Description	Test the Client capability to connect the Bootstrap Server according to the Client Initiated Bootstrap Mode followed by registering to the LwM2M Server afterwards. In this test case the Bootstrap Server provisions the LwM2M Client for PSK-based authentication.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.14 as defined in Annex C ○ The Client is supporting the Client Initiated Bootstrap Mode ○ The Client doesn't contain any Instance of LwM2M Server Object ID:1 (minimum condition to trigger a Bootstrap request message when the device is on) ○ The Client is capable of using PSK-based DTLS credentials
Test Procedure	<p>The Client requests the Bootstrap Server to setup a configuration enabling the Client to contact a LwM2M Server.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Without Instance of Server Object, the Client performs a BOOTSTRAP-REQUEST (CoAP POST /bs?ep{Endpoint Client Name}) in using the Resource values of the Instance 1 of Security Object ID:0 to contact the Bootstrap Server 2. The Bootstrap Server uploads the Configuration C.1 in the Client in performing two BOOTSTRAP-WRITE (CoAP PUT /0, CoAP PUT /1) in using the set of values defined in 0-SetOfValue_1 and 0-SetOfValue_2 (Object Device ID:3,is

	<p>automatically created and filled-up by the Client)</p> <ol style="list-style-type: none"> 3. The Bootstrap Server performs a BOOTSTRAP-DISCOVER operation (CoAP GET / Accept:40 /) to verify the Client setup 4. The Bootstrap Server performs a BOOTSTRAP-FINISH operation (CoAP POST /bs) to end-up that BS phase 5. The Lwm2m Client connects to the provisioned Lwm2m server using the PSK-based credentials, as described in LightweightM2M-1.0-int-401
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Bootstrap Server received a Success Message (“2.04” Changed) related to the BOOTSTRAP-REQUEST of the Client B. In test step 2., Client received WRITE operation(s) to setup the C.1 configuration (0-SetOfValue) C. In test step 2., Server received Success (“2.04” Changed) message(s) from Server related to WRITE operation(s) D. In test step 3., the Bootstrap Server received the Success message (“2.05” Content) along with the payload related to the BOOTSTRAP-DISCOVER request and containing : <pre style="margin-left: 40px;">lwm2m="1.0",</0/0>;ssid=1,</0/1>, </1/0>;ssid=1,</3/0></pre> E. In test step 4., the Bootstrap Server received the Success message (“2.04” Changed) : no inconsistency detected F. A modular component of a device e.g. the radio module housing the SIM

0-SetOfValues definition

lwm2m+json version

```
{“bn”：“/”,
  “e”:[
    {“n”：“0/0/0”,“sv”: <Lwm2m Server URI>},
    {“n”：“0/0/1”,“bv”:false},
    {“n”：“0/0/2”,“v”:0},
    {“n”：“0/0/3”,“sv”:<PSK Identity>}, // opaque in Base64
    {“n”：“0/0/4”,“sv”: <n/a> },
    {“n”：“0/0/5”,“sv”: <Secret Key>}, // opaque in Base64
    {“n”：“0/0/10”,“v”:1},
    {“n”：“1/0/0”,“v”:1},
    {“n”：“1/0/1”,“v”:86400},
    {“n”：“1/0/6”,“bv”:false},
    {“n”：“1/0/7”,“sv”：“U”},
  ]
}
```

6.1.1.3 LightweightM2M-1.0-int-2 – Client Initiated Bootstrap Full (Cert)

Test Case Id	LightweightM2M-1.0-int- 2
Test Object	Client and Server

Test Case Description	Test the Client capability to connect the Bootstrap Server according to the Client Initiated Bootstrap Mode followed by registering to the LwM2M Server afterwards. In this test case the Bootstrap Server provisions the LwM2M Client for certificate-based authentication.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.14 as defined in Annex C ○ The Client is supporting the Client Initiated Bootstrap Mode ○ The Client doesn't contain any Instance of LwM2M Server Object ID:1 (minimum condition to trigger a Bootstrap request message when the device is on) ○ The Client is capable of using certificate-based DTLS credentials
Test Procedure	<p>The Client requests the Bootstrap Server to setup a configuration enabling the Client to contact a LwM2M Server.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Without Instance of Server Object, the Client performs a BOOTSTRAP-REQUEST (CoAP POST /bs?ep{Endpoint Client Name}) in using the Resource values of the Instance 1 of Security Object ID:0 to contact the Bootstrap Server 2. The Bootstrap Server uploads the Configuration C.1 in the Client in performing two BOOTSTRAP-WRITE (CoAP PUT /0, CoAP PUT /1) in using the set of values defined in 0-SetOfValue_1 and 0-SetOfValue_2 (Object Device ID:3,is automatically created and filled-up by the Client) 3. The Bootstrap Server performs a BOOTSTRAP-DISCOVER operation (CoAP GET Accept:40 /) to verify the Client setup 4. The Bootstrap Server performs a BOOTSTRAP-FINISH operation (CoAP POST /bs) to end-up that BS phase 5. The LwM2M Client connects to the provisioned LwM2M server using the certificate-based credentials, as described in LightweightM2M-1.0-int-402
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Bootstrap Server received a Success Message ("2.04" Changed) related to the BOOTSTRAP-REQUEST of the Client B. In test step 2., Client received WRITE operation(s) to setup the C.1 configuration (0-SetOfValue) C. In test step 2., Server received Success ("2.04" Changed) message(s) from Server related to WRITE operation(s) D. In test step 3., the Bootstrap Server received the Success message ("2.05" Content) along with the payload related to the BOOTSTRAP-DISCOVER request and containing: <ul style="list-style-type: none"> lwm2m="1.0",</0/0>;ssid=1,</0/1>, </1/0>;ssid=1,</3/0> E. In test step 4., the Bootstrap Server received the Success message ("2.04" Changed) : no inconsistency detected F. The test case in LightweightM2M-1.0-int-402 executes successfully.

0-SetOfValues definition

lwm2m+json version

```

{“bn”：“/”,
  “e”:[
    {“n”：“0/0/0”,“sv”：coaps://server.example.com },
    {“n”：“0/0/1”,“bv”：false},
    {“n”：“0/0/2”,“v”：2},
    {“n”：“0/0/3”,“sv”：<Client Certificate >}, // Contains the client.crt
    {“n”：“0/0/4”,“sv”：<Server Certificate> }, // Contains the server.crt
    {“n”：“0/0/5”,“sv”：<Client Private Key>}, // Contains the client.key
    {“n”：“0/0/10”,“v”：1},
    {“n”：“1/0/0”,“v”：1},
    {“n”：“1/0/1”,“v”：86400},
    {“n”：“1/0/6”,“bv”：false},
    {“n”：“1/0/7”,“sv”：“U”},
  ]
}

```

6.1.1.4 LightweightM2M-1.0-int-3 – Simple Bootstrap from Smartcard

Test Case Id	LightweightM2M-1.0-int-3
Test Object	Client and Server
Test Case Description	Test the Client capability to connect the expected LwM2M Server according to the Bootstrap Information uploaded from the Smartcard (Bootstrap from Smartcard)
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the “Bootstrap from Smartcard” mode as defined in LwM2M 1.0.2: no Smartcard with Bootstrap Information is inserted ○ The Client is configured with the minimum Configuration C.1 as defined in Annex C with LwM2M Object Server URI “X” & Lifetime=86400s ○ The Client is connected to the LwM2M Server X ○ A Smartcard is prepared containing a Bootstrap Information file as specified in Annex G and H of the TS LwM2M 1.0.2; this Bootstrap Information contains a Server Account different from the C.1 one (LwM2M Server Object URI “Y” with “Y” different of “X” and Lifetime of 43200s - ID :”/1/0/1”)

Test Procedure	<p>The Device is powered-off and the prepared Smartcard (pre-conditions) is inserted in the Device. After power-on, the Bootstrap procedure from Smartcard occurs and the Client registers to LwM2M Server “Y”.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The LwM2M Client is connected to the LwM2M Server “X” 2. The LwM2M Server retrieves (READ /1/0/1) the Server Object lifetime of the LwM2M Client 3. The Device is powered-off, the prepared Smartcard is inserted in the Device, then the Device is powered-on again. 4. The Client registers to the LwM2M Server “Y” as specified in the Smartcard Bootstrap Information file. 5. The LwM2M Server “Y” verifies (READ /1/0/1) that the Server Object lifetime contains the expected value (as specified in the Smartcard Bootstrap Information file)
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., Initial Registration for Server “X” has passed (pass criteria A&B of Test Case int-101) B. In test step 2, the READ operation on the Resource “Lifetime” of the Server Object (READ /1/0/1) of the LwM2M Client provides the value given by the C.1 configuration (86400) along with the success message (“2.05” Content) C. In test step 4, Initial Registration for Server “Y” has passed (pass criteria A&B of Test Case int-101 but in using Configuration set by the Bootstrap from Smartcard) D. In test step 5, the READ operation on the Resource “Lifetime” of the Server Object (READ /1/0/1) of the LwM2M Client provides the value given by the new configuration (43200) along with the success message (“2.05” Content)

6.1.2 Registration Interface [100-199]

6.1.2.1 LightweightM2M-1.0-int-101 – Initial Registration

Test Case Id	LightweightM2M-1.0-int-101
Test Object	Client and Server
Test Case Description	Test that the Client registers with the Server.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.1 as defined in Annex C. ○ Device is turned on. ○ The bootstrap procedure has been completed or the required bootstrap information is available to the Client.
Test Procedure	<p>The Client automatically registers at the Server, once this information is available.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Registration message (CoAP POST) is sent from Client to Server.

Pass-Criteria	<p>A. In test step 1. , the Server receives the REGISTER command along with the following information:</p> <ul style="list-style-type: none"> • Endpoint Client Name • registration lifetime • LwM2M version • binding mode (optional) • SMS number (optional) • Objects and Object Instances (mandatory and optional objects / object instances) ; possibly with Version of Objects <p>B. In test step 1. , Client received “Success” message from Server (2.01 Created) related to the REGISTER command.</p>
----------------------	--

6.1.2.2 LightweightM2M-1.0-int-102 – Registration Update

Test Case Id	LightweightM2M-1.0-int-102
Test Object	Client and Server
Test Case Description	Test that the client updates the registration information on the Server.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.1 as defined in Annex C ○ Client is registered with the Server ○ The Server will be prepared to change the Client lifetime registration to 20 sec to set the conditions for a Client UPDATE operation. ○ On option, the Client will be prepared to update its registration before the registration time expires.
Test Procedure	<p>The Client is registered with the Server and updates its registration according to the change of its registration information (Lifetime) (UPDATE Operation)</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server set the lifetime resource of the Server Object Instance to 20 sec (CoAP PUT /1/0/1) 2. UPDATE (Registration) message (CoAP POST) is sent from Client to Server with Lifetime parameter set to 20 sec 3. On option, before the registration expires (20 sec) the Client send a new UPDATE message without parameter to the Server
Pass-Criteria	<p>A. In test step 1., the Server received a Success Message (2.04 Changed) related to its setting request</p> <p>B. In test step 2., Server has received UPDATE operation with lifetime parameter = 20 sec</p> <p>C. In test step 2., Client has received “Success” (2.04) message from Server related to the UPDATE command</p> <p>D. In test step 3., either the Server received an UPDATE operation with no parameter or a de_registratoin occurs in the Server after the 20s</p>

6.1.2.3 LightweightM2M-1.0-int-103 – Deregistration

Test Case Id	LightweightM2M-1.0-int-103
Test Object	Client and Server

Test Case Description	Test that the client is able to deregister at the Server.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.3 as defined in Annex C ○ Client is registered with the Server
Test Procedure	<p>Client will no longer be available, thus, it should de-register</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server sends EXECUTE command on the Disable Resource of the Server Object (ID:1) Instance. 2. Deregistration message (CoAP DELETE) is sent from Client to Server.
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Server receives the “Success” message (2.04 Changed) from the Client B. In test step 2., the Client receives “Success” message (2.02 Deleted) from the Server C. Client is removed from the servers registration database

6.1.2.4 LightweightM2M-1.0-int-104 – Registration Update Trigger

Test Case Id	LightweightM2M-1.0-int-104
Test Object	Client and Server
Test Case Description	Test that the Client updates its registration with the Server when triggered with the Registration Update Trigger (see LwM2M Server object)
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.1 as defined in Annex C ○ Device is turned on ○ The bootstrap procedure has been completed or the required bootstrap information is available to the client ○ The Client is registered with the Server with a Lifetime of 20 sec (LightweightM2M-1.0-int-102)
Test Procedure	<p>Before Client Registration lifetime expired (lifetime is 20 sec) the Server sends Registration Update Trigger and the Client updates its registration via the UPDATE Operation</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Before Client registration expires on the Server (for test purposes a short registration lifetime is chosen 20 sec) a Registration Update Trigger message CoAP POST /1/0/8 is sent from Server to Client 2. UPDATE (Registration) message (CoAP POST) is sent from Client to Server

Pass-Criteria	<ul style="list-style-type: none"> A. In test step 1., Client received a Registration Update Trigger B. In test step 1., Server received a “Success” message (2.04 Changed) related to the EXECUTE command (Registration Update Trigger). C. In test step 2., Server received UPDATE operation without parameter D. In test step 2., Client has received “Success” message (2.04 Changed) from Server related to its UPDATE message E. After test step 2., the Client is still registered in the Server while the initial Registration lifetime expired
----------------------	--

6.1.3 Device management & Service Enablement Interface [200-299]

6.1.3.1 LightweightM2M-1.0-int-201 – Querying basic information in Plain Text format

Test Case Id	LightweightM2M-1.0-int-201
Test Object	Client and Server
Test Case Description	<p>Querying the following data on the client (Device Object: ID 3) using Plain Text data format</p> <ul style="list-style-type: none"> • Manufacturer Name • Model number • Serial number
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.1 as defined in Annex C ○ Client is registered with the LwM2M Server ○ Client is supporting Plain Text data format (0)
Test Procedure	<p>READ operations from Server on the targeted Resources have been received by the client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Successive READ (CoAP GET) operations are performed on the targeted Resources of the Device Object (ID:3) Instance, with the CoAP Accept option set to Plain Text data format (0)
Pass-Criteria	<p>A. In test step 1, Server has received all the expected “Success” messages (2.05 Content) along with the requested information with the expected values and according to the Plain Text data format:</p> <ul style="list-style-type: none"> • Manufacturer Name (ID:0) • Model number (ID:1) • Serial number (ID:2)

6.1.3.2 LightweightM2M-1.0-int-202 – Querying basic information in Opaque format

<Test Cases to fill-up>

6.1.3.3 LightweightM2M-1.0-int-203 – Querying basic information in TLV format

Test Case Id	LightweightM2M-1.0-int-203
Test Object	Client and Server
Test Case Description	Querying the Resources Values of Device Object ID:3 on the Client Resources Values of Device Object ID:3
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The client supports the Configuration C.1 as defined in Annex C ○ Client is registered with the LwM2M Server
Test Procedure	<p>A READ operation from Server on the Device Object Instance has been received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) operation on Device Object Instance (/3/0), with the CoAP Accept option set to TLV data format (11542) 2. Bits 7-6=00= Object Instance in which case the Value contains one or more Resource TLVs 3. Bits 7-6=11= Resource with Value
Pass-Criteria	<p>A. In test step 1. Server has received the “Success” message (2.05 Content), along with the requested information in the TLV data format (11542) and containing the expected values concerning:</p> <ul style="list-style-type: none"> • Manufacturer Name (ID:0) • Model number (ID:1) • Serial number (ID:2) • Firmware Version (ID:3) • Error Code (ID:11) • Supported Binding and Modes (ID:16) (“U”)

6.1.3.4 LightweightM2M-1.0-int-204 – Querying basic information in JSON format

Test Case Id	LightweightM2M-1.0-int-204
Test Object	Client and Server
Test Case Description	Querying the Resources Values of Device Object ID:3 on the Client using JSON data format
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.1 as defined in Annex C ○ Client is supporting JSON data format (11543) ○ Client is registered with the LwM2M Server
Test Procedure	<p>A READ operation from Server on the Device Object Instance has been received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) operation on the Device Object Instance (/3/0), with the CoAP Accept option set to JSON data format (11543)

Pass-Criteria	<p>A. In test step 1. Server has received the success message (2.05 Content) along with the requested information in JSON data format (11543) and containing the expected values concerning:</p> <ul style="list-style-type: none"> • Manufacturer Name (ID:0) • Model number (ID:1) • Serial number (ID:2) • Firmware Version (ID:3) • Error Code (ID:11) • Supported Binding and Modes (ID:16) (“U”)
----------------------	--

6.1.3.5 LightweightM2M-1.0-int-205 – Setting basic information in Plain Text format

Test Case Id	LightweightM2M-1.0-int-205
Test Object	Client and Server
Test Case Description	Setting writable Resources of Server Object (ID:1) Instance 0 using Plain Text data format (0)
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.3 as defined in Annex C ○ Client is supporting Plain Text data format (0) ○ Client is registered with the Server
Test Procedure	<p>Successive WRITE operations from Server on the Resources of the Server Object (ID:1) Instance have been received by the client. This test has to set the following resources with specific values:</p> <ul style="list-style-type: none"> • Default minimum period (ID:2) : 0101 sec • Default maximum period (ID:3) : 1010 sec • Disable timeout (ID:5) : 2000 sec <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Successive WRITE (CoAP PUT) operations are performed on the Resources of the Instance 0 of the Server Object in using the predefined values above. The Plain Text data format (0) is used 2. The Server verifies (READs/CoAP GET) the result of the WRITE operations by querying the Instance 0 of the Server Object in using the TLV data format. 3. The steps 1., 2. of the test are re-played with the initial values of the Configuration 3
Pass-Criteria	<p>A. In test step 1. the Server receives a “Sucesss” message (2.04 Changed) for each WRITE operation</p> <p>B. In test step 2, Server received the “Success” message (2.05 Content), along with the requested information in the TLV data format and containing the expected values defined for this test</p> <p>C. In test step 3, the final stage confirms that the initial Configuration C.3 has been restored in the Client.</p>

6.1.3.6 LightweightM2M-1.0-int-210 – Setting basic information in Opaque format

<Test Cases to fill-up>

6.1.3.7 LightweightM2M-1.0-int-215 – Setting basic information in TLV format

Test Case Id	LightweightM2M-1.0-int-215
---------------------	----------------------------

Test Object	Client and Server
Test Case Description	Setting writable Resources of Server Object (ID:1) Instance 0 in using the TLV data format (11542) and restoring these Resources to their initial values
Preconditions	The Client supports the Configuration C.3 as defined in Annex C The Client is registered with the Server The current values of the Server Object (ID:1) Instance 0, are saved on the Server
Test Procedure	A WRITE operation from Server on the Server Object (ID:1) Instance 0 has been received by the client using TLV data format (11542). The set of values to write is given in the sample below (215-SetOfValues) Normal flow: <ol style="list-style-type: none"> 1. A 1st WRITE (CoAP POST) operation on the Server Object (ID:1) Instance 0 is performed in using the set of values contains in the 215-SetOfValues sample below.The data format TLV is used. (11542) 2. The Server READs the result of the WRITE operation by querying the Server Object (ID:1) Instance 0. 3. A 2nd WRITE (CoAP PUT) operation on the Server Object (ID:1) Instance 0 is performed in using the Initial values preserved on the Server in conformance to the test pre-conditions. 4. The Server READs the result of the previous WRITE operation by querying the Server Object (ID:1) Instance 0.
Pass-Criteria	<ol style="list-style-type: none"> A. The Server receives the success messages for the steps 1.(2.04), 2. (2.05), 3. (2.04) and 4. (2.05) of the test B. In test step 4. , the received values are consistent with the pre-defined values of the 215-SetOfValues sample C. In test step 6, the received values are consistent with the values present in the initial Configuration C.3

215-SetOfValues definition	
lwm2m+tlv version	
C1 02 65	; /1/0/2 101sec
C2 03 03 F2	; /1/0/3 1010 sec
C2 05 07 D0	; /1/0/5 2000 sec
C1 06 01	; /1/0/6 true
C2 07 55 51	; /1/0/7 UQ

6.1.3.8 LightweightM2M-1.0-int-220 – Setting basic information in JSON format

Test Case Id	LightweightM2M-1.0-int-220
Test Object	Client and Server
Test Case Description	Setting writable Resources of Server Object (ID:1) Instance 0 in using the JSON data format (11543) and restoring these Resources to their initial values

Preconditions	<p>The Client supports the Configuration C.3 as defined in Annex C</p> <p>The Client is registered with the Server</p> <p>The current values of the Server Object (ID:1) Instance 0, are saved on the Server</p>
Test Procedure	<p>A WRITE operation from Server on the Server Object (ID:1) Instance 0 has been received by the client using JSON data format (11543). The set of values to write is given in the sample below (220-SetOfValues)</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A 1st WRITE (CoAP POST) operation on the Server Object (ID:1) Instance 0 is performed in using the set of values contains in the 220-SetOfValues sample below.The data format JSON is used. (11543) 2. The Server READs the result of the WRITE operation by querying the the previously targeted Resources of the Server Object (ID:1) Instance 0. 3. A 2nd WRITE (CoAP PUT) operation on the Server Object (ID:1) Instance 0 is performed in using the Initial values which have been preserved (pre-conditions). 4. The Server READs the result of the previous WRITE operation by querying the Server Object (ID:1) Instance 0.
Pass-Criteria	<ol style="list-style-type: none"> A. The Server receives the correct status codes for the steps 1. (2.04), 2.(2.05), 3. (2.04) and 4. (2.05) of the test. B. In test step 2. , the received values are consistent with the 220-SetOfValues sample C. In test step 4, the received values are consistent with the values present in the initial Configuration 3

220-SetOfValues definition

lwm2m+json version

```
{
  "bn": "/",
  "e": [
    {
      "n": "1/0/2",
      "v": "0101"
    },
    {
      "n": "1/0/3",
      "v": "1010"
    },
    {
      "n": "1/0/5",
      "v": "2000"
    },
    {
      "n": "1/0/6",
      "bv": "true"
    },
    {
      "n": "1/0/7",
      "sv": "UQ"
    }
  ]
}
```

6.1.3.9 LightweightM2M-1.0-int-241 – Executable Resource: Rebooting the device

Test Case Id	LightweightM2M-1.0-int-241
Test Object	Client and Server
Test Case Description	Test that the Device can be remotely reboot via the Device Object (ID:3)
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The client supports the Configuration C.1 as defined in Annex C ○ Device is switched on ○ Client is registered with the Server. (Client might be in a state that requires a reboot)
Test Procedure	<p>An EXECUTE operation from Server on the resource Reboot of the Device Object (ID:3) has been received by the client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Server performs Execute (CoAP POST) operation on the Reboot Resource of Device Object (ID:3) Instance 2. Client registers again with the Server as in the test LightweightM2M-1.0-int-101
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the success message (2.04 Changed) related to the EXECUTE operation on the Client B. In test step2., Device reboots and the Client registers successfully with the Server again (see LightweightM2M-1.0-int-101)

6.1.3.10 LightweightM2M-1.0-int-260 – Discover Command

Test Case Id	LightweightM2M-1.0-int-260
Test Object	Client and Server
Test Case Description	Discovering the implemented Resources of the Device Object as well as reporting the various Attributes attached to Object / Object Instance and Resources
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.4 as defined in Annex C ○ The Client is registered with the Server ○ No <NOTIFICATION> Attribute is initially attached to the Object Device
Test Procedure	<p>Various DISCOVER operations are performed by the Server on the Object Device. Several WRITE-ATTRIBUTES commands are interleaved to modify the content of the DISCOVER command replies</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The initial DISCOVER (CoAP GET Accept:40) operation is performed on Object Device ID:3 2. A WRITE-ATTRIBUTES operation set the pmin=10 & pmax=200 <NOTIFICATION> Attributes at the Object level 3. DISCOVER command is performed at Object Instance level of Object ID:3 4. The WRITE-ATTRIBUTES operation set the lt=1, gt=6 et st=1 <NOTIFICATION> Attributes of the Resource ID:7 (Power Source Voltage) 5. Same DISCOVER command as in test step 3. is performed 6. DISCOVER operation is performed on Resource ID:7 of the Object Device Instance

Pass-Criteria	<p>A. In test step 1, the Success Message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and containing :</p> <pre></3>,</3/0>,</3/0/1>, </3/0/2>, </3/0/3>, </3/0/4>,</3/0/6> </3/0/7>,</3/0/8>,</3/0/9>,</3/0/11>,</3/0/16></pre> <p>B. In test step 2, a Success message is received by the Server (“2.04” Changed) related to the WRITE-ATTRIBUTES operation</p> <p>C. In test step 3, the Success message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and containing :</p> <pre></3/0>,</3/0/1>, </3/0/2>, </3/0/3>, </3/0/4>, </3/0/6>;dim=2, </3/0/7>;dim=2, </3/0/8>;dim=2, </3/0/9>,</3/0/11>,</3/0/16></pre> <p>D. In test step 4, a “Success” message is received by the Server (“2.04” Changed) related to the WRITE-ATTRIBUTES operation</p> <p>E. In test step 5, the Success message (“2.05” Content) is received by the Server along with the same payload received in step 3, except for the Resource 7 :</p> <pre></3/0>,</3/0/1>, </3/0/2>, </3/0/3>, </3/0/4>, </3/0/6>;dim=2, </3/0/7>;dim=2, ;lt=1;gt=6;st=1, </3/0/8>;dim=2,</3/0/9>, </3/0/11>,</3/0/16></pre> <p>F. In test step 6, the Success message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and just containing :</p> <pre></3/0/7>;pmin=10;pmax=200;dim=2 ;lt=1;gt=6 ;st=1</pre>
----------------------	---

6.1.3.11 LightweightM2M-1.0-int-270 – Create Object Instance

As this functionality is not simply available through usage of Core Objects, the result of such Test Case is delegated to the result of any Test Case belonging to the following list:

- LightweightM2M-1.0-int-1630 – Create Portfolio Object Instance

6.1.3.12 LightweightM2M-1.0-int-271 – Create Multiple Resource Instance

<Test Cases to fill-up>

6.1.3.13 LightweightM2M-1.0-int-290 – Delete Object Instance

As this functionality is not simply available through usage of Core Objects, the result of such Test Case is delegated to the result of any Test Case belonging to the following list:

- LightweightM2M-1.0-int-1635 – Delete Portfolio Object Instance

6.1.4 Information Reporting Interface [300-399]

6.1.4.1 LightweightM2M-1.0-int-301 – Observation and Notification of parameter values

Test Case Id	LightweightM2M-1.0-int-301
Test Object	Client and Server
Test Case Description	Sending the observation policy to the device.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.4 as defined in Annex C ○ Device is switched on and operational. ○ Client is registered with the LwM2M Server
Test Procedure	<p>The Server establishes an Observation relationship with the Client to acquire conditional notifications about Resources from the Device Object Instance:</p> <ul style="list-style-type: none"> • Power Source Voltage (ID:7) • Power Source Current (ID:8) <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Server Determines which Power Sources are on the Device (READ access to Resource ID:6 of Device Object ID:3) 2. Server communicates with the Device Object ID:3 via WRITE-ATTRIBUTE (CoAP PUT) operations, to set the pmin & pmax Attributes of each targeted Resources (e.g. /3/0/7?pmin=5&pmax=15 and /3/0/8?pmin10&pmax=20) 3. Server sends OBSERVE (CoAP GET operation with Observe Option set to 0) messages for the targeted Resources (ID:7 & ID:8) to activate reporting 4. Client reports requested information with a NOTIFY message (CoAP responses)
Pass-Criteria	A. The Server regularly receives the requested information and displays “Power Source Voltage” and “Power Source Current” values to the user if possible.

6.1.4.2 LightweightM2M-1.0-int-302 – Cancel Observations using Reset Operation

Test Case Id	LightweightM2M-1.0-int-302
Test Object	Client and Server
Test Case Description	Cancel the Observation relationship by sending “Cancel Observation” operation.
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The client supports the Configuration C.4 as defined in Annex C ○ Device is switched on and operational. ○ Client is registered at the LwM2M Server ○ Server established a successful Observation relationship with the Client as defined in Test Case Id: LightweightM2M-1.0-int-301

Test Procedure	<p>The Server removes a pre-established Observation relationship by sending “Cancel Observation”. The Client removes conditional notifications about:</p> <ul style="list-style-type: none"> • Power Source Voltage • Power Source Current <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Client reports requested information with NOTIFY messages (CoAP responses) on Resource ID:7 and ID:8 2. On receiving a NOTIFY message from Resource ID:7, the Server sends a Cancel Observation (CoAP RESET message) to remove the Observation relationship with that Resource . 3. On receiving a NOTIFY message from Resource ID:8, the Server sends a Cancel Observation (CoAP RESET message) to remove the Observation relationship with that Resource . 4. Client stops reporting requested information and removes associated entries from the list of observers.
Pass-Criteria	<ol style="list-style-type: none"> A. The Server receives regular NOTIFICATION from Device Object ID:3 on Resources Power Source Voltage (ID:7) and Power Source Current (ID:8) B. The Server stops receiving information first on “Power Source Voltage” then on “Power Source Current” after having sent the Cancel Observation (Reset Operation) on that Resources. Associated entries from the list of observers are removed.

6.1.4.3 LightweightM2M-1.0-int-303 – Cancel observations using Observe with Cancel parameter

Test Case Id	LightweightM2M-1.0-int-303
Test Object	Client and Server
Test Case Description	Cancel the Observation relationship by sending an OBSERVE operation with the CANCEL parameter
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The client supports the Configuration C4 as defined in Annex C ○ Device is switched on and operational. ○ Client is registered at the LwM2M Server ○ Server established a successful Observation relationship with the Client as defined in Test Case Id: LightweightM2M-1.0-int-301

Test Procedure	<p>The Server removes a pre-established Observation relationship by sending “Observe” with Cancel Parameter. The Client removes conditional notifications about:</p> <ul style="list-style-type: none"> • Power Source Voltage • Power Source Current <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Client reports requested information with a NOTIFY message (CoAP responses) 2. After receiving few notifications, Server sends OBSERVE operation with the CANCEL parameter (CoAP GET message with Observe option set to 1) to cancel the Observation relationship with the Instance of the Device Object (/3/0) 3. Client stops reporting requested information on both Resources and removes associated entries from the list of observers.
Pass-Criteria	<ol style="list-style-type: none"> A. The Server receives regular NOTIFICATION from Device Object ID:3 on Resources Power Source Voltage (ID:7) and Power Source Current (ID:8) B. The Server stops receiving information on “Power Source Voltage” and “Power Source Current” after having sent the Cancel Observation (with Cancel parameter) on the Device Object Instance (/3/0).Associated entries from the list of observers are removed.

6.1.5 Security [400-499]

6.1.5.1 LightweightM2M-1.0-int-401 – UDP Channel Security – Pre-shared Key Mode

Test Case Id	LightweightM2M-1.0-int-401
Test Object	Client and Server
Test Case Description	Establishing a successful DTLS session using UDP pre-shared key mode
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The client supports the Configuration C.1 as defined in Annex C ○ The bootstrap procedure has been completed or the required bootstrap information is available to the client. ○ The bootstrap information includes the Security Mode resource of the object LwM2M Security set to 0: Pre-Shared Key mode
Test Procedure	<p>Device is switched on.</p> <p>A DTLS session is established between client and Server and the Client automatically registers with that Server</p> <p>Using this DTLS session, a simple data exchange is performed between the Server and the Client, to verify the encrypted connection is properly established.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DTLS Session is established 2. Registration message (CoAP POST) is sent from LwM2M Client to LwM2M Server. 3. Client receives Success message (2.01 Created) from the Server. 4. READ (CoAP GET) on the Instance of the Device Object is performed using the default TLV data format (cf Test LwM2M-1.0-int-203) 5. Server receives success message (2.05 Content) and the requested values (encrypted)
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 2 & 3, Registration command of the Client on the Server is performed successfully over the DTLS session B. In test step 4 & 5 the READ command work successfully over the DTLS session.

6.1.5.2 LightweightM2M-1.0-int-402 – UDP Channel Security – Certificate Mode

Test Case Id	LightweightM2M-1.0-int-402
Test Object	Client and Server
Test Case Description	Establishing a successful DTLS session using certificate mode
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The client supports the Configuration C.17 as defined in Annex C ○ The bootstrap procedure has been completed or the required bootstrap information is available to the client. ○ DNS is configured in such a way that the server responds to server.example.com.

Test Procedure	<p>Device is switched on.</p> <p>A DTLS session is established between client and Server and the Client automatically registers with that Server</p> <p>Using this DTLS session, a simple data exchange is performed between the Server and the Client, to verify the encrypted connection is properly established.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DTLS Session is established 2. Registration message (CoAP POST) is sent from LwM2M Client to LwM2M Server. 3. Client receives Success message (2.01 Created) from the Server. 4. READ (CoAP GET) on the Instance of the Device Object is performed using the default TLV data format (cf Test LwM2M-1.0-int-203) 5. Server receives success message (2.05 Content) and the requested values (encrypted)
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 2 & 3, Registration command of the Client on the Server is performed successfully over the DTLS session B. In test step 4 & 5 the READ command work successfully over the DTLS session.

6.1.5.3 LightweightM2M-1.0-int-403 – UDP Channel Security – Certificate Mode (Server Identify Verification Failure)

Test Case Id	LightweightM2M-1.0-int-403
Test Object	Client and Server
Test Case Description	Failure to establish a DTLS session using certificate mode because of Server Identity Verification Failure
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> o The client supports the Configuration C.18 as defined in Annex C o The bootstrap procedure has been completed or the required bootstrap information is available to the client. o DNS is configured in such a way that the server responds to server-fail.example.com.
Test Procedure	<p>Device is switched on.</p> <p>A DTLS handshake is started between the DTLS Client and the DTLS Server. The DTLS handshake fails because the content of the certificate (server.example.com) does not match the configured LwM2M Server URI (server-fail.example.com).</p>
Pass-Criteria	The DTLS Handshake fails

6.1.6 Core Specific Objects Test cases [500-999]

6.1.6.1 Security Object (ID 0) [500-549]

6.1.6.2 Server Object (ID 1) [550-599]

6.1.6.2.1 LightweightM2M-1.0-int-551 – Access Check to the Resources

LightweightM2M-1.0-int-215 Test Case (Setting basic information in TLV format) is covering the same functional scope than this LightweightM2M-1.0-int-551 Test Case. The Test Status of both Test Cases must be considered as identical.

6.1.6.2.2 LightweightM2M-1.0-int-555 – Check of the Disable (De-Registration) capability

<Test Case to fill-up>

Note : LightweightM2M-1.0-int-103 Test Case (De-Registration) is partially addressing the scope of this LightweightM2M-1.0-int-555 Test Case.

6.1.6.2.3 LightweightM2M-1.0-int-556 – Check of the Update Registration capability

LightweightM2M-1.0-int-104 Test Case (Registration Update Trigger) is covering the same functional scope than this LightweightM2M-1.0-int-556 Test Case. The Test Status of both Test Cases must be considered as identical.

6.1.6.2.4 LightweightM2M-1.0-int-560 – Delayed Report Notification

<Test Cases to fill-up>

6.1.6.2.5 LightweightM2M-1.0-int-565 – Create Object Instance

<Test Cases to fill-up>

6.1.6.2.6 LightweightM2M-1.0-int-566 – Delete Object Instance

<Test Cases to fill-up>

6.1.6.3 Access Control Object (ID 2) [600-649]

6.1.6.4 LwM2M Device Object (ID 3) [650-699]

6.1.6.4.1 LightweightM2M-1.0-int-651 – Check Access to the Resources

Test Case Id	LightweightM2M-1.0-int-651
Test Object	Client and Server
Test Case Description	Setting writable Resources of Device Object (ID:3) Instance in using the TLV data format (11542), checking the modifications and querying the readable Resources. Then restoring the Resources to their initial values with verification
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.6 as defined in Annex C ○ The Client is registered with the Server ○ The Initial values of the Device Object (ID:3) Instance, are saved on the Server
Test Procedure	<p>A WRITE operation from Server on the Device Object (ID:3) Instance has been received by the client using TLV data format (11542). The set of values to write is given in the sample below (651-SetOfValues)</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A 1st WRITE (CoAP POST) operation on the Device Object (ID:3) Instance is performed in using the set of values contains in the 651-SetOfValues sample below. The data format TLV is used. (11542) 2. The Server READs the result of the WRITE operation by querying the Device Object (ID:3) Instance. 3. A 2nd WRITE (CoAP PUT) operation on the Device Object (ID:3) Instance is performed in using the Initial values which have been preserved (pre-conditions). 4. The Server READs the result of the previous WRITE operation by querying the Server Object (ID:3) Instance.
Pass-Criteria	<ol style="list-style-type: none"> A. The Server receives the correct status codes for the steps 1. (2.04), 2. (2.05), 3. (2.04), & 4 (2.05) of the test. B. In test step 3., the received values are as expected (readable resources) and consistent with the 651-SetOfValues sample. C. In test step 7, the received values are consistent with the values present in the initial Configuration C.3

651-SetOfValues definition

lwm2m+json version

```
{“bn”：“/”,
  “e”:[
    {“n”：“3/0/13”,“v”：“1367491215”},
    {“n”：“3/0/14”,“sv”：“+02:00”},
    {“n”：“3/0/15”,“sv”：“Europe/Paris”}
  ]
}
```

lwm2m+tlv version

```
C4 0D 51 82 42 8F
C6 0E 2B 30 32 3A 30 30
C8 0F 0C 45 75 72 6F 70 65 2F 50 61 72 69 73
```

6.1.6.4.2 LightweightM2M-1.0-int-652 – Querying the firmware version from the client

Test Case Id	LightweightM2M-1.0-int-652
Test Object	Client and Server
Test Case Description	Querying the version of the firmware present in the LwM2M Client
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.6 as defined in Annex C ○ The Client is registered with the Server
Test Procedure	<p>A READ operation from Server on the Firmware Version Resource (ID:3) has been received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) operation is performed on Device Object Resource “Firmware Version”
IPass-Criteria	<ol style="list-style-type: none"> A. In test step 2, the Server received the requested information (Firmware Version) in the data format preferred by the Client B. In test step 2 the Server receives the success message (2.05 Content)

6.1.6.4.3 LightweightM2M-1.0-int-655 – Check of the “Reboot” capability

LightweightM2M-1.0-int-241 Test Case (Executable Resource: Rebooting the Device) is covering the same functional scope than this LightweightM2M-1.0-int-655 Test Case. The Test Status of both Test Cases must be considered as identical.

6.1.6.4.4 LightweightM2M-1.0-int-656 – Check of the “Factory Reset” capability

Note: this test should be performed only if the client is able to carry out one of the Bootstrap methods defined in [LwM2M TS 1.0] section 5.2.3

<Test Cases to fill-up>

6.1.6.4.5 LightweightM2M-1.0-int-657 – Check of the “Error Code” functionality

<Test Cases to fill-up>

6.1.6.4.6 LightweightM2M-1.0-int-660 – Basic Observation and notification of Device Object Resources

LightweightM2M-1.0-int-301 Test Case (Observation and Notification of parameter values) is covering the same functional scope than this LightweightM2M-1.0-int-660 Test Case. The Test Status of both Test Cases must be considered as identical.

6.1.6.4.7 LightweightM2M-1.0-int-661 – Extended Observation and notification of Device Object Resources

<Test Cases to fill-up>

6.1.6.4.8 LightweightM2M-1.0-int-670 – Create Multiple Resource Instances

<Test Cases to fill-up>

6.1.6.4.9 LightweightM2M-1.0-int-680 – Create Object Instance

Note: This operation is not allowed on this Object since the Device Object is a Mandatory and Single-instance Object.

Test Case Id	LightweightM2M-1.0-int-680
Test Object	Client and Server
Test Case Description	Verifying that Creating an Instance of the Device Object is not allowed
Tool	n/a

Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.1 as defined in Annex C ○ The Client is registered with the Server
Test Procedure	<p>A CREATE operation targeting the Device Object ID:3 is performed by the Server and received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. CREATE (CoAP POST) operation is performed on Device Object
Pass-Criteria	A. In test step 1, the Server receives the failure message (4.05 Method Not Allowed)

6.1.6.4.10 LightweightM2M-1.0-int-685 – Delete Object Instance

Note: This operation is not allowed on this Object since the Device Object is a Mandatory and Single-instance Object fully maintained by the lwm2m Client. It must be always present when the Client is registered.

Test Case Id	LightweightM2M-1.0-int-685
Test Object	Client and Server
Test Case Description	Verifying that Deleting the Instance of the Device Object is not allowed
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.1 as defined in Annex C ○ The Client is registered with the Server
Test Procedure	<p>A DELETE operation targeting the Device Object ID:3 is performed by the Server and received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DELETE (CoAP DELETE) operation is performed on the Device Object Instance (/3/0)
Pass-Criteria	A. In test step 1 the Server receives the failure message (4.05 Method Not Allowed)

6.1.6.5 LwM2M Connectivity Monitoring (ID:4) [700-749]

6.1.6.5.1 LightweightM2M-1.0-int-701 – Querying the readable resources of object

Test Case Id	LightweightM2M-1.0-int-701
Test Object	Client and Server
Test Case Description	Querying the readable Resources of the Connectivity Monitoring Object Instance and verifying the received values are as expected and coherent.
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.5 as defined in Annex C ○ Client is registered with the Server
Test Procedure	<p>A READ operation from Server on the Connectivity Monitoring Object Instance has been received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) operation is performed by the Server on the Connectivity Monitoring Object Instance of the Client
Pass-Criteria	A. In test step 1., the Client received a READ (Coap GET) command from the Server on the Connectivity Monitoring Object Instance

	<ul style="list-style-type: none">B. In test step 1., the Server receives the status code “2.05” for READ message successC. In test step 1., along with the success message, the mandatory Resources (ID:0, 1,2, 4) and the optional ones, are received by the Server with expected values in compliance with LwM2M technical specification TS 1.0.
--	--

6.1.6.5.2 LightweightM2M-1.0-int-705 – Setting the writable resources

There is no writable resources for this object.

6.1.6.5.3 LightweightM2M-1.0-int-710 – Basic Observation and notification on Connectivity Monitoring Object Resources

Test Case Id	LightweightM2M-1.0-int-710
Test Object	Client and Server
Test Case Description	Sending the observation policy to the Client for resources of Connectivity Monitoring (ID:4)
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.5 as defined in Annex C ○ Device is switched on and operational ○ Client is registered with the LwM2M server
Test Procedure	<p>The Server establishes an Observation relationship with the Client to acquire condition notifications about the Connectivity Monitoring Object Instance.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server communicates to the Client the pmin=2 and pmax=10 periods, threshold values with a WRITE ATTRIBUTE (CoAP PUT) operation at the Connectivity Monitoring Object Instance level 2. The Server sends OBSERVE (CoAP Observe Option) message to activate reporting on the Monitoring Object Instance 3. The Client reports requested information with a NOTIFY message (CoAP response)
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Client has received a WRITE ATTRIBUTE Command (CoAP PUT) targeting the Connectivity Monitoring Object Instance with the proper pmin=2, and pmax=10 parameters B. In test step 1, the Server received the success message (2.04 Changed) in response to the WRITE ATTRIBUTE command C. In test step 2. the Client received the OBSERVE operation targeting the Connectivity Monitoring Object Instance D. In test step 2., in response to its OBSERVE request, the Server receives the success message (Content 2.05) along with the Connectivity Object Instance initial values E. In test step 3., based on pmin/pmax periods parameters received in test step 1., the Client reports Information to the Server with a NOTIFY message containing the Connectivity Object Instance updated values. F. In test step 3., the values receives by the Server along with the success message (Content 2.05) must be as expected

6.1.6.5.4 LightweightM2M-1.0-int-711 – Extended Observation and notification of Connectivity Monitoring Object Resources

<Test Cases to fill-up>

6.1.6.5.5 LightweightM2M-1.0-int-720 – Create Multiple Resource Instances

<Test Cases to fill-up>

6.1.6.5.6 LightweightM2M-1.0-int-730 – Create Object Instance

<Test Cases to fill-up>

6.1.6.5.7 LightweightM2M-1.0-int-735 – Delete Object Instance

<Test Cases to fill-up>

6.1.6.6 Firmware Update Object (ID 5) [750-799]

6.1.6.6.1 LightweightM2M-1.0-int-751 – Querying the readable resources

Test Case Id	LightweightM2M-1.0-int-751
Test Object	Client and Server
Test Case Description	Querying Information on the Firmware Update Object Instance for determining the current states and supported characteristics of the Client for such capability.
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ The Client is registered with the Server ○ The Fw Update Object Instance is in its initial state (State=0)
Test Procedure	<p>A READ operation from the Server on the Firmware Update Object Instance requesting TLV format has been received by the client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) operation is performed on the Firmware Update Object Instance
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the status code “2.05” for READ operation success B. In test step 1, the returned values regarding State (ID:3) and Update Result (ID:5) prove the Client FW update Capability is in initial state (State=Idle & Update Result= Initial Value). C. In test step 1, the returned values regarding Firmware Update Protocol Support (ID:8) & Firmware Update Delivery Method (ID:9) allow to determine the supported characteristics of the Client FW Update Capability.

6.1.6.6.2 LightweightM2M-1.0-int-755 – Setting the writable Resource Package

Test Case Id	LightweightM2M-1.0-int-755
Test Object	Client and Server
Test Case Description	Setting the writable Resource Package (ID:0) of Firmware Update Object Instance and verifying the triggered actions produce the expected results
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Appendix C ○ The Client is registered with the Server ○ According to Test LightweightM2M-1.0-int-751, the FW Update Object Instance supports the PUSH Delivery Method
Test Procedure	<p>A WRITE operation from Server to set to NULL value (‘\0’) the Resource Package (ID:0) has been received by the client. The test verifies the FW Update Object Instance is reset to its Initial state. Then a WRITE operation is performed by the Server to load a valid image in the Client in setting the Package Resource.</p> <p>The various Resources of this FW Update Objects must be set as expected by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A WRITE (CoAP PUT) operation with a NULL value (‘\0’) is performed by the Server on the Package Resource (ID:0) of the FW Update Object Instance 2. The Server READs (CoAP GET) the FW Object Instance to get

	<p>the values of the State (ID:3) and Update Result (ID:5) Resources</p> <ol style="list-style-type: none"> 3. A WRITE (CoAP PUT) operation with a valid image is performed by the Server on the Package Resource (ID:0) of the FW Update Object Instance 4. The Server READs (CoAP GET) the FW Object Instance to get the values of the State (ID:3) and Update Result (ID:5) Resources
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the success message “2.04” associated with the WRITE operation B. In test step 2, the Server receives the success message “2.05” along with the value of State and Update Result Resources values. C. In test step 2, the queried State and Update Result Resources values are both 0 (Idle / Initial value): FW Update Object Instance is in the Initial state. D. In test step 3, the Server receives the success message “2.04” associated with the WRITE request for loading the firmware image. E. In test step 4, the Server receives the success message “2.05” along with the State and Update Result Resources values. F. In test step 4, the queried value of State resource is 2 (Downloaded) and the value of Update Result value is still 0 (Initial Value)

6.1.6.6.3 LightweightM2M-1.0-int-756 – Setting the writable Resource Package URI

Test Case Id	LightweightM2M-1.0-int-756
Test Object	Client and Server
Test Case Description	Setting the writable Resource Package URI (ID:1) of Firmware Update Object Instance and verifying the triggered actions produce the expected results
Preconditions	<ul style="list-style-type: none"> o The Client supports the Configuration C.8 as defined in Appendix C o The Client is registered with the Server o According to Test LightweightM2M-1.0-int-751, the FW Update Object Instance supports the PULL Delivery Method
Test Procedure	<p>A WRITE operation from Server to set the empty string to the Resource Package URI (ID:0) has been received by the client. The test verifies the FW Update Object Instance is reset to its Initial state. Then a WRITE operation is performed by the Server to load a valid URI in the Client in setting the Package URI Resource.</p> <p>The various Resources of this FW Update Objects must be set as expected by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A WRITE (CoAP PUT) operation with an empty string value is performed by the Server on the Package Resource (ID:0) of the FW Update Object Instance 2. The Server READs (CoAP GET) the FW Object Instance to get the values of the State (ID:3) and Update Result (ID:5) Resources 3. A WRITE (CoAP PUT) operation with a valid image is performed by the Server on the Package Resource (ID:0) of the FW Update Object Instance 4. The Server READs (CoAP GET) the FW Object Instance to get the values of the State (ID:3) and Update Result (ID:5) Resources

Pass-Criteria	<p>A. In test step 1, the Server receives the success message “2.04” associated with the WRITE operation</p> <p>B. In test step 2, the Server receives the success message “2.05” along with the value of State and Update Result Resources values.</p> <p>C. In test step 2, the queried State and Update Result Resources values are both 0 (Idle / Initial value): FW Update Object Instance is in the Initial state.</p> <p>D. In test step 3, the Server receives the success message “2.04” associated with the WRITE request for the loaded image.</p> <p>E. In test step 4, the Server receives the success message “2.05” along with the State and Update Result Resources values.</p> <p>F. In test step 4, the queried value of State resource is 2 (Downloaded) and the value of Update Result value is still 0 (Initial Value)</p>
----------------------	---

6.1.6.6.4 LightweightM2M-1.0-int-760 –Basic Observation and notification on Firmware Update Object Resources

Test Case Id	LightweightM2M-1.0-int-760
Test Object	Client and Server
Test Case Description	Sending the observation policy to the Client for State Resource of Firmware Update Object (ID:5)
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Appendix C ○ Device is switched on and operational ○ Client is registered at the LwM2M server ○ According to Test LightweightM2M-1.0-int-751, the FW Update Object Instance supports the PUSH Delivery Method ○ Test LightweightM2M-1.0-int-755 already passed (an firmware image can be successfully loaded) ○ The Client is in Initial State regarding the FW Update functionality (State=Idle)
Test Procedure	<p>The Server establishes an Observation relationship with the Client to acquire notifications about State Resource of the Fw Update Object Instance</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server communicates to the Client pmin=2 and pmax=10 periods with a WRITE-ATTRIBUTE (CoAP PUT) operation at the FW Update Object Instance level. 2. The Server Sends OBSERVE (CoAP Observe Option) message to activate reporting on the State Resource (/5/0/3) of the FW Update Object Instance. 3. The Server delivers the firmware to the Client through a WRITE (CoAP PUT) operation on the Package Resource (/5/0/0) 4. The Client reports requested information with a NOTIFY message (CoAP response)
Pass-Criteria	<p>A. In test step 1, the Server receives the success message “2.04” associated with the WRITE-ATTRIBUTE operation.</p> <p>B. In test step 2, the Server receives the success message “2.05” associated with the OBSERVE operation, along with the value of State =Idle</p> <p>C. In test step 3, the Server receives the success message “2.04” associated with the WRITE operation delivering the firmware image.</p> <p>D. In test step 4, the State Resource value returned by the Client in NOTIFY message is set to “Downloaded”</p>

6.1.6.6.5 LightweightM2M-1.0-int-770 – Successful Firmware update (via COAP)

Test Case Id	LightweightM2M-1.0-int-770
Test Object	Client and Server
Test Case Description	Perform a device firmware update remotely triggered by the LWM2M server in using Package Resource (CoAP mechanisms)
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ According to Test LightweightM2M-1.0-int-751, the FW Update Object Instance supports the PULL Delivery Method ○ The Client already passed the LightweightM2M-1.0-int-755 test ○ Device is switched on and operational ○ The Client is registered with the LWM2M server ○ Firmware Update is available on the Server ○ State is “0” (Idle) or “1” (Downloaded)
Test Procedure	<p>After setting the Client in the Initial stage for supporting Firmware Update process, the Server delivers the Package to the Client before to trigger the installation of the downloaded firmware. Finally, the Server checks the result of the full process.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Step 1 – Package Delivery <ol style="list-style-type: none"> a. The Server places the Client in the initial state of the FW Update process : A WRITE (CoAP PUT) operation with a NULL value (‘0’) is performed by the Server on the Package Resource (ID:0) of the FW Update Object Instance b. The Server delivers the firmware to the Client through a WRITE (CoAP PUT) operation on the Package Resource (/5/0/0) c. Polling (READ command) or Notification on Update Result and State Resources is performed, up to the time State Resource takes the ‘Downloaded’ value (2) 2. Step 2 – Firmware Update <ol style="list-style-type: none"> a. When the download is completed (State Resource value is ‘2’ Downloaded) , the Server initiates a firmware update by triggering EXECUTE command on Update Resource (CoAP POST /5/0/2) b. Polling (READ command) or Notification on Update Result and State Resources is performed, up to the time State Resource is turned back to Idle value (0) or Update Result Resource contains an other value than the Initial one (0) 3. Step 3 – Process verification <ol style="list-style-type: none"> a. The Server READs Update Result (“/5/0/5”) and State (“/5/0/3”) Resources to know the result of the firmware update procedure. a. The Server READs the Resource “Firmware Update” from the Object Device Instance (“/3/0/3”)
Pass-Criteria	<ol style="list-style-type: none"> A. Step 1 – Package Delivery <ol style="list-style-type: none"> a. In the test step 1.a, the Server receives the status code “2.04” for the WRITE success setting the Client in the FW update initial state. b. In the test step 1.b, The Server receives success message with either a “2.31” status code (Continue) or a final “2.04” status code.

	<p>c. In the test step 1.c State Resource can take the value “1” (Downloading) during this sub-step and will take the value “2” at the end (Downloaded)</p> <p>d. Update Result is “0” (Initial Value) during the whole step</p> <p>B. Step 2 – Firmware Update</p> <p>a. In test step 2.a, the Server receives a success message “2.04” (Changed) in response to the EXECUTE command</p> <p>b. In test step 2.b, the Server receives success message(s) “2.05” Contents along with a State Resource value of “3” (Updating) or “0” (Idle) and an Update Ressource value of “0” (Initial Value) or “1” (Firmware updated successfully)</p> <p>C. Step 3 – Process verification</p> <p>a. In test step 3.a, the Server receives success message(s) “2.05” Content” along with a State Resource value of “0” (Idle) and an Update Ressource value of “1” (Firmware updated successfully)</p> <p>b. In test step 3.b, the Server receives success message “2.05” Content” along with the expected value of the Resource Firmware Version from the Object Device Instance</p>
--	---

6.1.6.6 LightweightM2M-1.0-int-771 – Successful Firmware update (via alternative mechanism)

Test Case Id	LightweightM2M-1.0-int-771
Test Object	Client and Server
Test Case Description	Perform a device firmware update remotely triggered by the LwM2M server in using Package URI Resource
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ According to Test LightweightM2M-1.0-int-751, the FW Update Object Instance supports the PUSH Delivery Method ○ The Client passed the LightweightM2M-1.0-int-755 test ○ Device is switched on and operational ○ The Client is registered with the LwM2M server ○ Firmware Update is available on the Server ○ State is “0” (Idle) or “1” (Downloaded)
Test Procedure	<p>After setting the Client in the Initial stage for supporting Firmware Update process, the Server delivers the Package URI to the Client, waits for the firmware to be downloaded and then triggers the installation of the downloaded firmware. Finally, the Server checks the result of the full process.</p> <p>Normal flow:</p> <p>1. Step 1 – Package Delivery</p> <p>a. The Server places the Client in the initial state of the FW Update process : A WRITE (CoAP PUT) operation with an empty string value is performed by the Server on the Package URI Resource (ID:1) of the FW Update Object Instance</p> <p>b. The Server delivers the Package URI to the Client through a WRITE (CoAP PUT) operation on the Package URI Resource (/5/0/1)</p> <p>c. The Client downloads the firmware from the provided URI via an alternative mechanism (not CoAP)</p> <p>d. Polling (successive READ commands) or Notification on</p>

	<p>Update Result and State Resources is performed, up to the time State Resource takes the 'Downloaded' value (2)</p> <p>2. Step 2 – Firmware Update</p> <p>a. When the download is completed (State Resource value is '2' Downloaded) , the Server initiates a firmware update by triggering EXECUTE command on Update Resource (CoAP POST /5/0/2)</p> <p>b. Polling (READ command) or Notification on Update Result and State Resources is performed, up to the time State Resource is turned back to Idle value (0) or Update Result Resource contains an other value than the Initial one (0)</p> <p>3. Step 3 – Process verification</p> <p>a. The Server READs Update Result (“/5/0/5”) and State (“/5/0/3”) Resources to know the result of the firmware update procedure.</p> <p>b. The Server READs the Resource “Firmware Update” from the Object Device Instance (“/3/0/3”)</p>
Pass-Criteria	<p>A. Step 1 – Package Delivery</p> <p>a. In the test step 1.a, the Server receives the status code “2.04” for the WRITE success setting the Client in the FW update initial state.</p> <p>b. In the test step 1.b, the Server receives the status code “2.04” for the WRITE success setting the Package URI Client in the FW update Object Instance</p> <p>c. In the test step 1.c, The Server receives success message with either a “2.31” status code (Continue) or a final “2.04” status code.</p> <p>d. In the test step 1.d State Resource can take the value “1” (Downloading) during this sub-step and will take the value “2” at the end (Downloaded)</p> <p>e. Update Result is “0” (Initial Value) during the whole test step 1</p> <p>B. Step 2 – Firmware Update</p> <p>a. In test step 2.a, the Server receives a success message “2.04” (Changed) in response to the EXECUTE command</p> <p>b. In test step 2.b, the Server receives success message(s) “2.05” Contents along with a State Resource value of “3” (Updating) or “0” (Idle) and an Update Ressource value of “0” (Initial Value) or “1” (Firmware updated successfully)</p> <p>C. Step 3 – Process verification</p> <p>a. In test step 3.a, the Server receives success message(s) “2.05” Content” along with a State Resource value of “0” (Idle) and an Update Ressource value of “1” (Firmware updated successfully)</p> <p>b. In test step 3.b, the Server receives success message “2.05” Content” along with the expected value of the Resource Firmware Version from the Object Device Instance</p>

6.1.6.6.7 LightweightM2M-1.0-int-772 – Error Case 1: firmware installation without downloaded package

Test Case Id	LightweightM2M-1.0-int-772
Test Object	Client and Server

Test Case Description	Try to perform a Device firmware installation when there is no downloaded firmware package
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ Device is switched on and operational ○ Client is registered with the LwM2M server ○ Firmware Update is available on the Server ○ The State Resource (/5/0/3) of the FW Object Instance is different from “2” (Downloaded)
Test Procedure	<p>The Server initiates a firmware installation on the Client while this Client has no downloaded package in it.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server send a READ operation (CoAP GET /5/0) to the Client on the FW Update Object Instance to obtain the values of the State and Update Resources. 2. the Client receives an EXECUTE operation on the Update Resource (CoAP POST /5/0/2) of the FW Update Object Instance 3. The Server send a READ operation again (CoAP GET /5/0/3) to the Client on the FW Update Object Instance to obtain the State and the Update Resource values
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives a success message (“2.05” Content) associated to its READ command along with a State Resource value which is not “2” (Downloaded) and a valid (0..9) Update Resource value B. In test step 2, the Server receives the status code “4.05” for method not allowed associated to its EXECUTE command C. In test step 3, the Server receives a success message (“2.05” Content) associated to its READ command along with a State Resource value and an Update Result Resource value, identical to the ones retrieved in Pass-Criteria A. The firmware has not been installed.

6.1.6.6.8 LightweightM2M-1.0-int-773 – Error Case 2: shortage of storage memory

Test Case Id	LightweightM2M-1.0-int-773
Test Object	Client and Server
Test Case Description	Try to perform a Device firmware update remotely triggered by the LWM2M Server with a firmware package exceeding the Client storage memory capacity.
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ Prepare a package with a size exceeding the Client storage memory capacity ○ Device is switched on and operational ○ Client is registered with the LwM2M server ○ Firmware Update is available on the Server ○ The State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)
Test Procedure	<p>A Package delivery initiated by a Server in using either PULL or PUSH delivery methods with a firmware package exceeding the Client storage memory capacity, is received by the Client</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server verifies through a READ (CoAP GET) command on

	<p>/5/0/3 (State) the FW Update Object Instance of the Client is in Idle State</p> <ol style="list-style-type: none"> 2. The Server delivers the firmware package to the Client either through a WRITE (CoAP PUT) operation in the Package Resource (/5/0/0) or through a WRITE operation of an URI in the Package URI Resource. 3. The firmware downloading process is running The Server sends repeated READs or OBSERVE on State and Update Result Resources (CoAP GET /5/0) of the FW Update Object Instance to determine when the download is completed or if an error occurred. Before the end of download, the device runs out of storage and cannot finish the download 4. When the Package delivery is stopped the server READs Update Result to know the result of the firmware update procedure.
<p>Pass-Criteria</p>	<ol style="list-style-type: none"> A. In test step 1, the Server receives the status code “2.05 “ (Content) for the READ success command, along with the State Resource value of “0” (Idle) B. In test step 2., the Server receives the status code “2.04” (Changed) for the WRITE command setting either the Package URI Resource or setting the Package Resource, according to the chosen firmware delivery method. C. In test step 3., the State Resource retrieved with a value of “1” from successive Server READs or Client NOTIFY messages, indicates the download stage of the Package Delivery is engaged D. In test step 3., the Update Result Resource (/5/0/5) retrieved from successive Server READs or Client NOTIFY messages will take the value “2” indicating an error occurred during the downloading process related to shortage of storage memory The State Resource value never reaches the Downloaded value (“3”) E. In test step 4., the success READ message(s) (status code “2.05” Content) on State Resource with value “0” (Idle) and Update Result Resource with value “2” indicates the firmware Package Delivery aborted due to shortage of storage memory.

6.1.6.6.9 LightweightM2M-1.0-int-774 – Error Case 3: out of memory

<p>Test Case Id</p>	<p>LightweightM2M-1.0-int-774</p>
<p>Test Object</p>	<p>Client and Server</p>
<p>Test Case Description</p>	<p>Try to perform a Device firmware update remotely triggered by the LWM2M Server while the Client runs out of RAM during download</p>
<p>Preconditions</p>	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ Prepare a firmware package to make the Client running out of RAM when downloading it. ○ Device is switched on and operational ○ Client is registered at the Lwm2M server ○ Firmware Update is available on the Server ○ The State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)
<p>Test Procedure</p>	<p>A Package delivery initiated by a Server using either PULL or PUSH delivery methods is received by the Client. This firmware package will make the Client to run out of RAM during its download, Normal flow:</p>

	<ol style="list-style-type: none"> 1. The Server verifies through a READ (CoAP GET) command on /5/0/3 (State) the FW Update Object Instance of the Client is in Idle State 2. The Server delivers the firmware package to the Client either through a WRITE (CoAP PUT) operation in the Package Resource (/5/0/0) or through a WRITE operation of an URI in the Package URI Resource. 3. The firmware download process is running The Server sends repeated READs or OBSERVE on State and Update Result Resources (CoAP GET /5/0) of the FW Update Object Instance to determine when the download is completed or if an error occurred. Before the end of download, the Client runs out of RAM and cannot finish the download 4. When the Package delivery is stopped the server READs Update Result to know the result of the firmware update procedure.
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the status code “2.05 “ (Content) for the READ success command, along with the State Resource value of “0” (Idle) B. In test step 2., the Server receives the status code “2.04” (Changed) for the WRITE command setting either the Package URI Resource or setting the Package Resource, according to the chosen firmware delivery method. C. In test step 3., the State Resource retrieved with a value of “1” from successive Server READs or Client NOTIFY messages, indicates the download stage of the Package Delivery is engaged D. In test step 3., the Update Result Resource (/5/0/5) retrieved from successive Server READs or Client NOTIFY messages will take the value “2” indicating an error occurred during the download process related to shortage of RAM The State Resource value never reaches the Downloaded value (“3”) E. In test step 4., the success READ message(s) (status code “2.05” Content) on State Resource with value “0” (Idle) and Update Result Resource with value “2” indicates the firmware Package Delivery aborted due to shortage of RAM.

6.1.6.6.10 LightweightM2M-1.0-int-775 – Error Case 4: Connection lost during download (package URI)

Test Case Id	LightweightM2M-1.0-int-775
Test Object	Client and Server
Test Case Description	Try to perform a Device firmware update remotely triggered by the LWM2M server while connection is lost during download
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ Prepare a method for loosing the connection when firmware package download is engaged. ○ Device is switched on and operational ○ Client is registered at the LwM2M server ○ Firmware Update is available on the Server ○ The State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)

Test Procedure	<p>A Package delivery initiated by a Server in using either the PULL delivery methods is received by the Client</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server verifies through a READ (CoAP GET) command on /5/0/3 (State) the FW Update Object Instance of the Client is in Idle State 2. The Server delivers the firmware package to the Client through a WRITE operation of an URI in the Package URI Resource. 3. The Server sends repeated READs or OBSERVE on State and Update Result Resources (CoAP GET /5/0) of the FW Update Object Instance to determine when the download is completed or if an error occurred. Before the end of download, the connection is intentionally lost and the download cannot be finished. 4. When the Package delivery is stopped the Server READs Update Result to know the result of the firmware update procedure.
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Server receives the status code “2.05 “ (Content) for the READ success command, along with the State Resource value of “0” (Idle) B. In test step 2., the Server receives the status code “2.04” (Changed) for the WRITE command setting the Package URI Resource according to the PULL firmware delivery method. C. In test step 3., the State Resource value set to “1” retrieved from successive Server READs or Client NOTIFY messages, indicates the Package Delivery process is engaged in a Download stage D. In test step 3., the Update Result Resource (/5/0/5) retrieved from successive Server READs or Client NOTIFY messages will take the value “4” indicating an error occurred during the downloading process related to connection lost E. In test step 4., the success READ message(s) (status code “2.05” Content) on State Resource with value “0” (Idle) and Update Result Resource with value “4” indicates the firmware Package Delivery aborted due to connection lost dur the Package delivery.

6.1.6.6.11 LightweightM2M-1.0-int-776 – Error Case 5: Package Integrity check failure

Test Case Id	LightweightM2M-1.0-int-776
Test Object	Client and Server
Test Case Description	Try to perform a device firmware update remotely triggered by the LWM2M Server while the downloaded package does not pass the integrity check
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ Prepare a firmware package with a wrong Data Integrity Check in it. ○ Device is switched on and operational ○ Client is registered at the Lwm2M server ○ Firmware Update is available on the Server ○ The State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)
Test Procedure	<p>A Package delivery initiated by a Server in using either PULL or PUSH delivery methods is received by the Client. This firmware package has been generated with a wrong Data Integrity Check,</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server verifies through a READ (CoAP GET) command on

	<p>/5/0/3 (State) the FW Update Object Instance of the Client is in Idle State</p> <ol style="list-style-type: none"> 2. The Server delivers the firmware package to the Client either through a WRITE (CoAP PUT) operation in the Package Resource (/5/0/0) or through a WRITE operation of an URI in the Package URI Resource. 3. The Server sends repeated READs or OBSERVE on State and Update Result Resources (CoAP GET /5/0) of the FW Update Object Instance to determine when the download is completed or if an error occurred. The firmware package Integrity Check failure stopped the download process. 4. When the Package delivery is stopped the server READs Update Result to know the result of the firmware update procedure.
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the status code “2.05 “ (Content) for the READ success command, along with the State Resource value of “0” (Idle) B. In test step 2., the Server receives the status code “2.04” (Changed) for the WRITE command setting either the Package URI Resource or setting the Package Resource, according to the chosen firmware delivery method. C. In test step 3., the State Resource value set to “1” retrieved from successive Server READs or Client NOTIFY messages, indicates the Package Delivery process is maintained in Downloading stage D. In test step 3., the Update Result Resource (/5/0/5) retrieved from successive Server READs or Client NOTIFY messages will take the value “5” indicating an error occurred during the downloading process related to the failure of the firmware package integrity check E. In test step 4., the success READ message(s) (status code “2.05” Content) on State Resource with value “0” (Idle) and Update Result Resource with value “5” indicates the firmware Package Delivery aborted due to a Firmware Package Integrity failure.

6.1.6.6.12 LightweightM2M-1.0-int-777 – Error Case 6: unsupported package type

Test Case Id	LightweightM2M-1.0-int-777
Test Object	Client and Server
Test Case Description	Try to perform a device firmware update remotely triggered by the LWM2M Server with an unsupported package type
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.8 as defined in Annex C ○ Prepare a firmware package containing a package type not supported by the Client ○ Device is switched on and operational ○ Client is registered at the Lwm2M server ○ Firmware Update is available on the Server ○ The State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)
Test Procedure	<p>A Package delivery initiated by a Server in using either PULL or PUSH delivery methods is received by the Client. The package type is not supported by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server verifies through a READ (CoAP GET) command on /5/0/3 (State) the FW Update Object Instance of the Client is in Idle State

	<ol style="list-style-type: none"> 2. The Server delivers the firmware package to the Client either through a WRITE (CoAP PUT) operation in the Package Resource (/5/0/0) or through a WRITE operation of an URI in the Package URI Resource. 3. The Server sends repeated READs or OBSERVE on State and Update Result Resources (CoAP GET /5/0) of the FW Update Object Instance to determine when the download is completed or if an error occurred. The Download cannot be finished since the firmware package type is not supported by the Client 4. When the Package delivery is stopped the server READs Update Result to know the result of the firmware update procedure.
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the status code “2.05” (Content) for the READ success command, along with the State Resource value of “0” (Idle) B. In test step 2., the Server receives the status code “2.04” (Changed) for the WRITE command setting either the Package URI Resource or setting the Package Resource, according to the chosen firmware delivery method. C. In test step 3., the State Resource value set to “1” retrieved from successive Server READs or Client NOTIFY messages, indicates the Package Delivery process is in Downloading stage D. In test step 3., the Update Result Resource (/5/0/5) retrieved from successive Server READs or Client NOTIFY messages will take the value “6” indicating an error occurred during the downloading process related to the firmware package type not supported by the Client. E. In test step 4., the success READ message(s) (status code “2.05” Content) on State Resource with value “1” (Downloading) and Update Result Resource with value “6” indicates the firmware Package Delivery aborted due to a firmware package type not supported by the Client.

6.1.6.6.13 LightweightM2M-1.0-int-778 – Error Case 7: invalid URI (package URI)

Test Case Id	LightweightM2M-1.0-int-778
Test Object	Client and Server
Test Case Description	Try to perform a device firmware update remotely triggered by the LWM2M server with an invalid URI
Preconditions	<ul style="list-style-type: none"> o The Client supports the Configuration C.8 as defined in Annex C o According to Test LightweightM2M-1.0-int-751, the FW Update Object Instance supports the PULL Update Delivery Method o Prepare an invalid URI from which the Client will try to fetch a firmware package. o Device is switched on and operational o Client is registered with the LwM2M server o Firmware Update is available on the ServerThe State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)
Test Procedure	<p>A Package delivery initiated by a Server in using the PULL delivery methods is received by the Client. The URI provided is a fake one.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server verifies through a READ (CoAP GET) command on /5/0/3 (State) the FW Update Object Instance of the Client is in Idle State

	<ol style="list-style-type: none"> 2. The Server initiates a firmware package delivery to the Client through a WRITE operation of an invalid URI in the Package URI Resource. 3. The Server sends repeated READs or OBSERVE on State and Update Result Resources (CoAP GET /5/0) of the FW Update Object Instance to determine when the download is completed or if an error occurred. The download process is stopped by the Client due to the usage of a bad URI. 4. When the Package delivery is stopped the server READs Update Result to know the result of the firmware update procedure.
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server receives the status code “2.05 “ (Content) for the READ success command, along with the State Resource value of “0” (Idle) B. In test step 2., the Server receives the status code “2.04” (Changed) for the WRITE command setting the Package URI Resource C. In test step 3., the State Resource value set to “1” retrieved from successive Server READs or Client NOTIFY messages, indicates the Package Delivery process is maintained in Downloading stage D. In test step 3., the Update Result Resource (/5/0/5) retrieved from successive Server READs or Client NOTIFY messages will take the value “7” indicating an error occurred during the downloading process related to the usage of a bad URI E. In test step 4., the success READ message(s) (status code “2.05” Content) on State Resource with value “0” (Idle) and Update Result Resource with value “7” indicates the firmware Package Delivery aborted due to the connection to an Invalid URI for the firmware package delivery.

6.1.6.6.14 LightweightM2M-1.0-int-779 – Error Case 8: Unsuccessful Firmware Update

Test Case Id	LightweightM2M-1.0-int-779
Test Object	Client and Server
Test Case Description	Perform a device firmware update remotely triggered by the LWM2M server in using Package Resource (CoAP mechanisms) or but with installation failure
Preconditions	<ul style="list-style-type: none"> o The Client supports the Configuration C.8 as defined in Annex C o Prepare a firmware package which can be downloaded but cannot be successfully installed. o Device is switched on and operational o Client is registered with the Lwm2M Server o Firmware Update is available on the Server o The State Resource (/5/0/3) of the FW Object Instance is set to “0” (Idle)
Test Procedure	<p>A Package delivery initiated by a Server in using either PULL or PUSH delivery methods is received by the Client. But its installation will fail.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Step 1 – Package Delivery <ol style="list-style-type: none"> a. The Server verifies through a READ (CoAP GET) command on /5/0/3 (State) the FW Update Object Instance of the Client is in Idle State b. The Server retrieves (CoAP GET) the initial value of the Firmware Version Resource from the Object Device Instance for verification in the Pass Criteria (C) c. The Server delivers the firmware package to the Client either through a WRITE (CoAP PUT) operation in the Package Resource

	<p>(/5/0/0) or through a WRITE operation of an URI in the Package URI Resource.</p> <p>d. Polling (successive READ commands) or Notification on Update Result and State Resources is performed, up to the time State Resource takes the ‘Downloaded’ value (2)</p> <p>2. Step 2 – Installation Failure</p> <p>a. When the download is completed (State Resource value is ‘2’ Downloaded) , the Server initiates a firmware update by triggering EXECUTE command on Update Resource (CoAP POST /5/0/2)</p> <p>b. Polling (READ command) or Notification on Update Result and State Resources is performed, up to the time State Resource is turned back to 2 (Downloaded) or the Update Result Resource contains the value “8” (Firmware update failed)</p> <p>3. Step 3 – Process Verification</p> <p>a. The server READs Update Result & State Resources to know the result of the firmware update procedure.</p> <p>b. The Server READs the Firmware Version Resource from the Object Device Instance</p>
Pass-Criteria	<p>A. Package Delivery</p> <p>a. In test step 1.a, the Server receives the status code “2.05 “ (Content) for the READ success command, along with the State Resource value of “0” (Idle)</p> <p>b. In test step 1.b, the Server receives the status code “2.05 “ (Content) for the READ success command, along with the initial value of the Firmware version Resource available from the Object Device Instance.</p> <p>c. In test step 1.c, the Server receives the status code “2.04” (Changed) for the WRITE command setting either the Package URI Resource or setting the Package Resource, according to the chosen firmware delivery method.</p> <p>d. In at this end of test step 1.d, the State Resource take the value “2” (Downloaded)</p> <p>B. Installation failure</p> <p>a. In test step 2.a, the Server receives a success message “2.04” (Changed) in response to the EXECUTE command</p> <p>b. In test step 2.b, the Server receives success message(s) “2.05” Contents along with a State Resource value of “3” (Updating) or “2” (Downloaded) and an Update Ressource value of “0” (Initial Value) and “8” at the end (Firmware updated failure)</p> <p>C. Process Verification</p> <p>a. In test step 3.a, the Server receives success message(s) “2.05” Content” along with a State Resource value of “2” (Downloaded) and an Update Ressource value of “8” (Firmware updated failed)</p> <p>b. In test step 3.b the Server receives success message(s) “2.05” Content” along with a Firmware Version Resource value form the Object Device Instance which has not changed compared to the one retrieved in Pass Criteria A.b</p>

6.1.6.6.15 LightweightM2M-1.0-int-780 – Error Case 9: Unsupported protocol

6.1.6.7 LwM2M Location Object (ID:6) [800-849]**6.1.6.7.1 LightweightM2M-1.0-int-801 – Querying the readable resources of object**

Test Case Id	LightweightM2M-1.0-int-801
Test Object	Client and Server
Test Case Description	Querying the readable resources of Object Location ID:6 Instance and verifying the received values are as expected and coherent.
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.7 as defined in Annex C ○ Client is registered with the Server
Test Procedure	<p>A READ operation from Server on the Location Object Instance has been received by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) is performed by the Server on the Location Object Instance of the Client
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Client receives a READ (CoAP GET) command from the Server on the Location Object Instance B. In test step 1., the Server receives the status code “2.05” for READ message success C. In test step 1., along with the success message, the mandatory Resources (Latitude, Longitude, Timestamp) and optional ones, are received by the Server with expected values in compliance with LwM2M technical specification 1.0

6.1.6.7.2 LightweightM2M-1.0-int-805 – Setting the writable resources

There are no writable resources for this object.

6.1.6.7.3 LightweightM2M-1.0-int-810 – Basic Observation and notification on Location Object Instance

Test Case Id	LightweightM2M-1.0-int-810
Test Object	Client and Server
Test Case Description	Sending the observation policy to the device for resources of Object Location ID:6
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.7 as defined in Annex C ○ Device is switched on and operational ○ Client is registered with the LWM2M server
Test Procedure	<p>The Server establishes an Observation relationship with the Client to acquire condition notifications about the Location Object Instance observable resources.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server communicates to the Client pmin=2 pmax=10 period threshold values with a WRITE ATTRIBUTE (CoAP PUT) operation at the Location Object Instance level 2. The Server sends OBSERVE (CoAP Observe Option) message to activate reporting on the Location Object Instance. 3. The Client reports requested information with a NOTIFY message (CoAP response)

Pass-Criteria	<ul style="list-style-type: none"> A. In test step 1., the Server received a WRITE ATTRIBUTE command (CoAP PUT) targeting the Location Object Instance with the proper pmin=2 and pmin=10 parameters. B. In test step 1., the Server received the success message (2.04 Changed) in response to the WRITE ATTRIBUTE command C. In test step 2., the Client received the OBSERVE operation targeting the Location Object Instance D. In test step 2., in response to its OBSERVE operation, the Server receives the success message (Content 2.05) along with the initial values of the Location Object Instance E. In test step 3., based on pmin/pmax periods parameters received in test step 1., the Client reports information to the Server with NOTIFY messages containing the Location Object Instance updated values. F. In test step 3., the values received by the Server along with the success message (Content 2.05) must be as expected : at less the Mandatory Timestamp Resource must have admissible values according to the pmin and pmax parameters.
----------------------	---

6.1.6.7.4 LightweightM2M-1.0-int-811 – Extended Observation and notification of Location Object Instance

<Test Cases to fill-up>

6.1.6.7.5 LightweightM2M-1.0-int-820 – Create Multiple Resource Instances

<Test Cases to fill-up>

6.1.6.7.6 LightweightM2M-1.0-int-830 – Create Object Instance

<Test Cases to fill-up>

6.1.6.7.7 LightweightM2M-1.0-int-835 – Delete Object Instance

<Test Cases to fill-up>

6.1.6.8 Connectivity Statistics (ID 7) [900-949]

6.1.6.8.1 LightweightM2M-1.0-int-901 – Querying a Data Collection from Connectivity Object Instance

Test Case Id	LightweightM2M-1.0-int-901
Test Object	Client and Server
Test Case Description	Starting and stopping after few seconds a data collection, then querying the readable resources of Connectivity Statistics Object ID:7 Instance and verifying the received values are as expected and coherent
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.9 as defined in Annex C ○ Client is registered with the Server
Test Procedure	<p>A READ operation from Server on the Connectivity Statistics Object Instance has been received by the Client after the data collection has been triggered then stopped after few seconds.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. EXECUTE operation (CoAP POST) is performed on the Start Resource of the Connectivity Statistics Object Instance 2. After few seconds, an EXECUTE operation (CoAP POST) is performed on the Stop Resource of the Connectivity Statistics Object Instance 3. A READ (CoAP GET) operation is performed by the Server on the Connectivity Object Instance of the Client
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Client receives a Start command for the Connectivity Statistics Object Instance (CoAP POST) B. In test step 1., the Server receives the success message (“2.4” Changed) in response of its EXECUTE command on the Client C. In test step 2., the Client receives a Stop command for the Connectivity Statistics Object Instance (CoAP POST) D. In test step 2., the Server receives the success message (“2.4” Changed) in response of its EXECUTE command on the Client E. In test step 3., the Client receives a READ operation on the Connectivity Object Instance F. In test step 3., the Server receives the status code “2.05” for the READ message success G. In test step 3., along with the success message, the Server receives the Connectivity Statistics Object Instance values according to the Client preferred data format. H. In test step 3, the received Connectivity Statistics Object Instance contains expected values regarding the data collection represented by the Resources 0 to 5 of the Connectivity Statistics Object Instance

6.1.6.8.2 LightweightM2M-1.0-int-905 – Setting the writable resources

Test Case Id	LightweightM2M-1.0-int-905
Test Object	Client and Server
Test Case Description	Query the readable Resources of the Connectivity Statistics Object ID:7 Instance, before setting the Collection Period to another value than the initial value one. Then query the readable Resources again and verify the Collection Period contains the expected value.

Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.9 as defined in Annex C ○ Client is registered with the Server
Test Procedure	<p>A WRITE operation is received by the Client for setting the Collection Period Resource of the Connectivity Statistics Object Instance. This WRITE operation is framed by 2 READ operations on the same Object Instance.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A READ (CoAP GET) operation is performed by the Server on the Connectivity Object Instance of the Client 2. A WRITE (CoAP PUT) operation is performed on the Client targeting the Collection Period Resource (ID:8) of the Connectivity Object Instance with a value different of the one collected in step 2. 3. A new READ (CoAP GET) operation is performed by the Server on the Connectivity Object Instance of the Client
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Client receives a READ operation on the Connectivity Object Instance B. In test step 1., the Server receives the status code “2.05” for the READ success message along with the Connectivity Statistics Object Instance value according to the Client preferred data format. C. In step 2, the Client receives a WRITE command (CoAP PUT) targeting the Collection Period of the Connectivity Statistics Object Instance with a value different of the one collected in pass B. D. In test step 2, the Server received receives a success message (2.04 Changed) related to its WRITE operation E. In test step 3., the Client receives a new READ operation on the Connectivity Object Instance F. In step 3, the Server receives success message (2.05 Content) and the requested value in the LwM2M Client preferred data format G. The value of the Collection Period Resource value collected in Pass F, correspond to the value which was set in step 2

6.1.6.8.3 LightweightM2M-1.0-int-910 – Basic Observation and notification on Connectivity Monitoring Object Instance

Test Case Id	LightweightM2M-1.0-int-910
Test Object	Client and Server
Test Case Description	Sending the observation policy to the device for resources of Connectivity Object ID:7 Instance
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.9 as defined in Annex C ○ Device is switched on and operational ○ Client is registered with the LwM2M server

Test Procedure	<p>The Server establishes an Observation relationship with the Client to acquire condition notifications about the Connectivity Statistics Instance.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server communicates to the Client pmin=2 & pmax=10 periods with a WRITE ATTRIBUTE (CoAP PUT) operation at the Connectivity Statistics Instance level. 2. The Server set (CoAP PUT) the Collection Period Resource to 0 in the Connectivity Statistics Object Instance. 3. The Server sends OBSERVE (CoAP Observe Option) message to activate reporting on the Connectivity Statistics Instance. 4. The Server starts the data collection on the Connectivity Statistics Object Instance by triggering the Start Resource of this Object Instance (CoAP POST) 5. The Client reports requested information with NOTIFY messages (CoAP response) 6. Several NOTIFY messages are received by the Server related to the Connectivity Statistics Object Instance 7. The Server stops the data collection on the Connectivity Statistics Object Instance by triggering the Stop Resource of this Object Instance (CoAP POST) 8. The Server stops the OBSERVATION process
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Server received a WRITE ATTRIBUTE command (CoAP PUT) targeting the Connectivity Statistics Object Instance with the proper pmin=2 and pmax=10 parameters. B. In test step 1., the Server received the success message (2.04 Changed) in response to the WRITE ATTRIBUTE command C. In test step 2 the Client received the OBSERVE operation targeting the Connectivity Statistics Object Instance D. In test step2, in response to its OBSERVE operation, the Server receives the success message (Content 2.05) along with the initial values of the Connectivity Statistics Object Instance E. In test step 3, based on pmin/pmax periods parameters received in test step 1., the Client reports information to the Server with NOTIFY messages containing the updated values of the Connectivity Statistics. F. In test step 3., the values returned by the Server along with the success message (Content 2.05) must be as expected.

6.1.6.8.4 LightweightM2M-1.0-int-911 – Extended Observation and notification of Connectivity Statistics Object Instance

<Test Cases to fill-up>

6.1.6.8.5 LightweightM2M-1.0-int-920 – Create Multiple Resource Instances

<Test Cases to fill-up>

6.1.6.8.6 LightweightM2M-1.0-int-930 – Create Object Instance

<Test Cases to fill-up>

6.1.6.8.7 LightweightM2M-1.0-int-935 – Delete Object Instance

<Test Cases to fill-up>

6.2 Multi-Servers Context [950-999]

6.2.1 LightweightM2M-1.0-int-950 – Multi-Servers Registration

Test Case Id	LightweightM2M-1.0-int-950
Test Object	Client and Server
Test Case Description	Test the Client capability to register 2 and connect 2 different Servers
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.14 as defined in Annex and containing 2 (Registered-)Server accounts ○ Device is turned-on ○ The Bootstrap Information is available in the Client with 2 Server-Accounts to which the Client is likely to register, and support of the Access Control Object)
Test Procedure	<p>The Client automatically registers with 2 Servers in using the 2 Server Accounts available in the bootstrapped configuration</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. Registration message (CoAP POST) is sent from the Client to Server #1 2. Registration message (CoAP POST) is sent from the Client to Server #2
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Server receives the REGISTER command along with the information related to the Server Account #1 (see LwM2M-1.0-int-101 Test Case for more) B. In test step 1., the Client receives the “Success” message from Server #1 (2.01 Created) C. In test step 2., the Server receives the REGISTER command along with the information related to the Server Account #2 (see LwM2M-1.0-int-101 Test Case for more) D. In test step 2., the Client receives the “Success” message from Server #2 (2.01 Created)

6.2.2 LightweightM2M-1.0-int-951 – Multi-Servers & Attributes

Test Case Id	LightweightM2M-1.0-int-951
Test Object	Client and Server
Test Case Description	Test the Client capability to identify and to report information according to a specific Server context
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.14 as defined in Annex and containing 2 (Registered-)Server accounts as well as simple Access Control Structure ○ Device is turned-on ○ The Bootstrap Information is available in the Client with 2 Server-Accounts to which the Client is likely to register ○ The Client pass the test case LightweightM2M-1.0-int-950 and is connected to Server#1 et Server#2

Test Procedure	<p>Server #1 & Server#2 set different Observation period values for the Device Object Instance; A check is performed to verify the effective setting</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. The Server #1 communicates to the Client pmin=2 and pmax=10 periods with a WRITE-ATTRIBUTE (CoAP PUT) operation at the Device Object Instance level. 2. The Server #2 communicates to the Client pmin=15 and pmax=50 periods with a WRITE-ATTRIBUTE (CoAP PUT) operation at the Device Object Instance level. 3. The Server #1 sends a DISCOVER command to the Client 4. The Server #2 send a DISCOVER command to the Client
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., the Server#1 receives the “Success” message from the Client (2.04 Changed) B. In test step 2., the Server#2 receives the “Success” message from the Client (2.04 Changed) C. In test step 3., the Server#1 receives the “Success” message from Client (2.05 Content) related to its DISCOVER command along with the payload containing the following information regarding the Device Object Instance : </3/0>;pmin=2;pmax=10,</3/0/0>,</3/0/1>,</3/0/2>,</3/0/3>,</3/0/11>,</3/0/16> D. In test step 4., the Server#2 receives the “Success” message from Client (2.05 Content) related to the DISCOVER command along with the payload containing the following information regarding the Device Object Instance : </3/0>;pmin=15;pmax=50,</3/0/0>,</3/0/1>,</3/0/2>,</3/0/3>,</3/0/11>,</3/0/16>

6.3 LwM2M Additional Objects Test cases [1000-1999]

6.3.1 Lock and Wipe Object (ID 8) [1000-1099]

<Test Cases to fill-up>

6.3.2 Software Management Object (ID 9) [1100-1199]

<Test Cases to fill-up>

6.3.3 Connectivity Management Objects (ID 10, 11, 12, 13) [1200-1499]

6.3.3.1 Cellular Network Connectivity ID:10 [1200-1249]

6.3.3.1.1 LightweightM2M-1.0-int-1200 – Querying the readable resources of Object ID:10

Test Case Id	LightweightM2M-1.0-int-1200
Test Object	Client and Server
Test Case Description	Querying the readable resources of Cellular network connectivity object ID:10 and verifying the received values are as expected and coherent regarding the Configuration C.10
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.10 as defined in Annex C ○ The Client is registered with the Server
Test Procedure	<p>A READ operation from Server on the Cellular Network Connectivity Object Instance requesting TLV format is sent to the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET /10/0) operation on the Instance of the Object ID:10 is received by the Client and its answer is sent back to the Server
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1 :the Server receives the success message (2.05 Content) associated to its READ request B. In step 1: the values returned by the Client in TLV format are consistent with the Configuration C 10 used for that test (PSM timer, Active Timer, eDRX parameters for WB-S1 mode, eDRX parameters for NB-S1 mode and Serving PLMN rate Control...).

6.3.3.1.2 LightweightM2M-1.0-int-1201 – Querying the readable resources of Object ID:10 in version 1.1

Test Case Id	LightweightM2M-1.0-int-1201
Test Object	Client and Server
Test Case Description	Querying the readable resources of Cellular network connectivity object ID:10 version 1.1 and verifying the received values are as expected and coherent regarding the Configuration C.11
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.11 as defined in Annex C ○ The Client is registered with the Server in indicating the Object ID:10 is in version 1.1
Test Procedure	<p>A DISCOVER operation is requested by the Server to verify the Object ID:10 is in version 1.1.</p> <p>A READ operation from Server on the Cellular Network Connectivity Object Instance requesting TLV format is sent to the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DISCOVER (CoAP GET Accept:40) operation is performed by the Server on Object ID:10 2. READ (CoAP GET /10/0) operation on the Instance of the Object ID:10 is received by the Client and its answer is sent back to the Server

Pass-Criteria	<p>A. In test step 1, the Server – along with the success message 2.05 – received the information related to Object ID:10 including </10>;ver="1.1", </10/0>, </10/0/4>, </10/0/5>, </10/0/6>, </10/0/8>, </10/0/9>, </10/0/11>, </10/0/13>, </10/0/14></p> <p>B. In test step 2 :the Server receives the success message (2.05 Content) associated to its READ request on Object Instance /10/0</p> <p>C. In test step 2: the values returned by the Client in TLV format are consistent with the Configuration C 10. used for that test (Activated Profile Names, PSM timer, Active Timer, eDRX parameters for WB-S1 mode, eDRX parameters for NB-S1 mode, Serving PLMN rate Control, Power Saving Modes, Active Power Saving Modes...).</p>
----------------------	---

6.3.3.1.3 LightweightM2M-1.0-int-1202 – Setting the Power Saving Mode Resources of Object ID:10 (version 1.1 only)

Test Case Id	LightweightM2M-1.0-int-1202
Test Object	Client and Server
Test Case Description	Setting the writable Resources of Cellular network connectivity Object ID:10 in version 1.1
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.11 as defined in Annex C ○ The Client is registered with the Server in indicating the Object ID:10 is in version 1.1 ○ The Power Saving Modes of Device is not 0
Test Procedure	<p>A DISCOVER operation is requested by the Server to verify the Object ID:10 is in version 1.1.</p> <p>A WRITE operation from Server on the Cellular Network Connectivity Object Instance requesting TLV format is sent to the Client to set the power saving mode. This test has to be run for the following resources:</p> <p style="padding-left: 40px;">a) Active Power Saving Modes</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DISCOVER (CoAP GET Accept:40) operation is performed by the Server on Object ID:10 2. A READ (CoAP GET /10/0/13) operation on the Instance 0 of the Object ID:10 is performed by the Server 3. A WRITE operation is performed by the Server on the “Active Power Saving Modes” Resource (CoAP PUT/POST 10/0/14) of the Object ID:10 Instance 0, in using the set of values contains in the 1210-SetOfValues sample below.The data format TLV is used. (11542) 4. A READ (CoAP GET /10/0/14) operation is performed by the Server on the Instance 0 of the Object ID:10
Pass-Criteria	<p>A. In test step 1, the Server – along with the success message 2.05 – received the information related to Object ID:10 including </10>;ver="1.1", </10/0>,</10/0/6>,</10/0/11>,</10/0/13>,</10/0/14></p> <p>B. In test step 2 : the Server receives the success message (2.05 Content) and the requested value of the “Power Saving Modes” Resource (/10/0/13). This value must mach the value present in the C.11 configuration.</p> <p>C. In step 3, the Server receives the success message (2.04 Changed) related to the WRITE command</p> <p>D. In test step 4 : the Server receives the success message (2.05 Content) associated to its READ request and the received value for the “Active Power Saving Modes” Resource“ must match the value contains in the</p>

1210-SetOfValues sample.
<p>1210-SetOfValues definition</p> <p>lwm2m+json version</p> <pre>{“bn”:“/”, “e”:[{“n”:“10/0/14”,“v”:2},] }</pre> <p>lwm2m+tlv version</p> <p>C10 0D 02 ; /10/0/14 2</p>

6.3.3.1.4 LightweightM2M-1.0-int-1203 – Observation and notification on Object ID:10 related to Power Saving Mode Resources (version 1.1 only)

Test Case Id	LightweightM2M-1.0-int-1203
Test Object	Client and Server
Test Case Description	Sending the observation policy to the device for resources of Cellular network connectivity Object ID:10 in version 1.1
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.11 as defined in Annex C ○ The Client is registered with the Server in indicating the Object ID:10 is in version 1.1
Test Procedure	<p>A DISCOVER operation is requested by the Server to verify the Object ID:10 is in version 1.1.</p> <p>The Server establishes an Observation relationship with the Client to acquire conditional notifications about Resources from the Connectivity Management Object Instance:</p> <ul style="list-style-type: none"> • Power Saving Modes(ID:13) • Active Power Saving Modes (ID:14) <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DISCOVER (CoAP GET Accept:40) operation is performed by the Server on Object ID:10 2. Server communicates with the Object ID:10 in version 1.1 via WRITE-ATTRIBUTE (CoAP PUT) operations, to set the pmin & pmax Attributes of each targeted Resources (e.g. /10/0/13?pmin=5&pmax=15 and /10/0/14?pmin=10&pmax=20) 3. Server sends OBSERVE (CoAP GET operation with Observe Option set to 0) messages for the targeted Resources (ID:13 & ID:14) to activate reporting 4. Client reports requested information with a NOTIFY message (CoAP responses)
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Server – along with the success message 2.05 – received the information related to Object ID:10 including </10>;ver="1.1", </10/0>, </10/0/6>, </10/0/11>, </10/0/13>, </10/0/14> B. In test step 2, the Server receives success messages (“2.04” Changed) related to the WRITE-ATTRIBUTES operations C. In test step 3, the Server receives success messages (“2.05” Content)

	<p>along with the initial values of Resource ID:13 and ID:14</p> <p>D. In test step 4, the Server regularly receives consistent information on the targeted Resources ID:13 & ID:14 of the Object ID:10 Instance 0.</p>
--	---

6.3.3.1.5 LightweightM2M-1.0-int-1204 – Observation and notification on Object ID: 10 related to timers

Test Case Id	LightweightM2M-1.0-int-1204
Test Object	Client and Server
Test Case Description	Sending the observation policy to the device for resources of Cellular network connectivity Object ID:10
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.10 as defined in Annex C ○ The Client is registered with the Server in indicating the Object ID:10
Test Procedure	<p>A DISCOVER operation is requested by the Server to verify the Object ID: 10. The Server establishes an Observation relationship with the Client to acquire conditional notifications about Resources from the Connectivity Management Object Instance:</p> <ul style="list-style-type: none"> • PSM Timer (ID:4) • Active Timer (ID:5) • eDRX parameters for WB-S1 mode(ID:8) • eDRX parameters for WB-S1 mode(ID:9) <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DISCOVER (CoAP GET Accept:40) operation is performed by the Server on Object ID:10 2. Server communicates with the Object ID:10 via WRITE-ATTRIBUTE (CoAP PUT) operations, to set the pmin & pmax Attributes of each targeted Resources (e.g. /10/0/4?pmin=5&pmax=15 and /10/0/5?pmin=10&pmax=20 and /10/0/8?pmin=5&pmax=15 and /10/0/9?pmin=10&pmax=20) 3. Server sends OBSERVE (CoAP GET operation with Observe Option set to 0) messages for the targeted Resources (ID:4 & ID:5 & ID:9 & ID:10) to activate reporting 4. Client reports requested information with a NOTIFY message (CoAP responses)
Pass-Criteria	<p>A. In test step 1, the Server – along with the success message 2.05 – received the information related to Object ID:10 including </10>;ver="1.0", </10/0>, </10/0/4>, </10/0/5>, </10/0/6>, </10/0/8>, </10/0/9>, B. In test step 2, the Server receives success messages ("2.04" Changed) related to the WRITE-ATTRIBUTES operations C. In test step 3, the Server receives success messages ("2.05" Content) along with the initial values of Resource ID:4 ,ID:5, ID:8 and ID:9 D. In test step 4, the Server regularly receives consistent information on the targeted Resources ID:4 , ID:5, ID:8 and ID:9 of the Object ID:10 Instance 0.</p>

6.3.3.2 APN connection profile ID:11 [1250-1299]

6.3.3.2.1 LightweightM2M-CONMGMT-1.0-int-1250 – APN configuration

Test Case Id	LightweightM2M-CONMGMT-1.0-int-1201
Test Object	Client and Server
Test Case Description	Creating and enabling a new APN profile

Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.10 as defined in Annex C ○ Test 104 on registration Update Trigger successfully passed ○ Device is switched on and operational ○ Client is registered at the LwM2M Server ○ Cellular connectivity is established with the parameters given in the APN Connection Profile Object Instance (/11/0).
Test Procedure	<p>A new APN Object is created, then activated. The list of the Active APN Connection Profiles is requested by the Server to the Client</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. CREATE (COAP POST) operation is performed by the Server targeting APN Connection Profile Object (ID:11) to create a 2nd instance of the APN connection profile Object with a new APN which is not active yet. 2. The Server triggers a Registration Update Trigger for forcing an UPDATE registration from the Client 3. UPDATE (registration) message (COAP POST) is sent from Client to Server including information about the supported Objects and Object Instances and namely the new Instance of the APN Connection Profile Object 4. Server activates the new APN Connection Profile in changing Enable status to True by WRITE operation (COAP PUT /11/1/3) 5. Server reads the list of active APN Connection Profiles by performing a TLV READ targeting Resource ID:11 of Object 10 (/10/0/11)
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1., The Server receives a Success message (“2.01” Created) associated to the Instantiation of the APN connection profile Object. B. In test step 2., the Client receives a Registration Update Trigger request on Resource ID:7 of the Device Object Instance (ID:3) C. In test step 2., the Server receives the Success message (“2.04” Changed) associated to the Update Registration Request D. In test step 3, the Server receives an UPDATE (registration) operation from the Client along with the updated list of Object/Object Instances in that Client. This list contains the new Instance of the APN connection profile Object (/11/1) E. In test step 4, the Server receives the Success message associated with the Server WRITE operation for the new APN activation. F. In test step 5., the Server receives a Success message (“2.05” Content) along with the list of the active APN containing the new Created APN (11:1) <p>NOTE: In case the device only supports one active APN profile this test is passed when the new APN profile is activated.</p>

6.3.3.2.2 WLAN Connectivity ID:12 [1300-1349]

<Test cases to fill-up>

6.3.3.2.3 Bearer Selection ID:13 [1350-1399]

6.3.3.2.3.1. LightweightM2M-CONMGMT-1.0-int-1350 – Bearer Selection

Test Case Id	LightweightM2M-CONMGMT-1.0-int-1350
Test Object	Client and Server
Test Case Description	Controlling bearers using Bearer Selection Object
Tool	n/a
Test code	n/a
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the minimum Configuration C.10 as defined in Annex C ○ Device is switched on and operational ○ Client is registered at the LwM2M server ○ The Client supports a Cellular Network Connectivity Object Instance and one instance of an APN Connection Profile Object. Cellular connectivity is established with the parameters given in the APN Connection Profile Object Instance. Also, the Client supports the WLAN Connectivity Object but WLAN radio is not enabled.
Test Procedure	<p>Normal flow:</p> <ol style="list-style-type: none"> 1. CREATE (COAP POST) operation is performed by the server targeting /13 to create the instance of the Bearer selection object with Preferred Communication Bearer (13/0/0) as WLAN preferred. 2. Server receives success message (2.01 Created) 3. Client shall turn on the WLAN radio and use it for connectivity with the Server. Client shall send Update to the Server indicating the update in registration as the Client's IP address (and port) has changed. 4. Server checks the status of the WLAN connectivity object by performing a READ on /12/0. The interface shall be enabled and running. The Server shall verify the values of Enable and Status resources for the same. 5. Server performs WRITE operation on Preferred Communication Bearer (13/0/0) resource and updates its value to 3GPP PS Preferred. 6. Client shall turn on the Cellular network connectivity (if not already enabled) and use it for connectivity with the Server. Client shall send Update to the Server indicating the update in registration as the Client's IP address (and port) has changed. 7. Server checks the status of the Cellular Network Connectivity object by performing a READ on /10. The interface shall be enabled and running.
Pass-Criteria	A. Bearer Selection Object is allowing the Server to control Client interface for communication

6.3.4 Device Capability Management Object (ID 15) [1500-1599]

<Test Cases too fill-up>

6.3.5 Portfolio Object (ID 16) [1600-1699]**6.3.5.1 LightweightM2M-1.0-int-1600 – Querying the readable resources of object**

<Test Cases to fill-up>

6.3.5.2 LightweightM2M-1.0-int-1605 – Setting the writable resources

<Test Cases to fill-up>

6.3.5.3 LightweightM2M-1.0-int-1610 – Basic Observation and notification of a Portfolio Object Instance

<Test Cases to fill-up>

6.3.5.4 LightweightM2M-1.0-int-1611 – Extended Observation and notification of a Portfolio Object Instance

<Test Cases to fill-up>

6.3.5.5 LightweightM2M-1.0-int-1620 – Create Multiple Resource Instances

<Test Cases to fill-up>

6.3.5.6 LightweightM2M-1.0-int-1630 – Create Portfolio Object Instance

Test Case Id	LightweightM2M-1.0-int-1630
Test Object	Client and Server
Test Case Description	Creating a new Instance of the Portfolio Object
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.12 as defined in Annex C ○ The Client is registered with the Server
Test Procedure	<p>A DISCOVER operations is performed by the Server on the Portfolio Object to retrieve its initial characteristics. A Portfolio CREATE operation is performed to add a new Instance in the LwM2M Client. Additional DISCOVER and READ check the results of the various operations</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. An initial DISCOVER (CoAP GET Accept:40) operation is performed on Instance 0 of Portfolio Object ID:16 2. A CREATE operation is performed with the Portfolio Object as target and the 1630-SetOfValues as payload in TLV format 3. A new DISCOVER operation is performed on Instance 1 of Portfolio Object ID:16 4. A READ request is performed by the Server on the Portfolio Object (CoAP GET /16).
Pass-Criteria	<ol style="list-style-type: none"> A. In test step 1, the Success Message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and containing : </16/0/0/>;dim=4 B. In test step 2, a Success message is received by the Server (“2.01” Created) related to the CREATE operation along with the new created Instance ID (should be /16/1) C. In test step 3, the Success message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and containing : </16/1/0/>;dim=2 D. In test step 4., the Success message (“2.05” Content) is received by the Server along with a TLV payload related to the READ Request and containing 2 Instances of the Portfolio Object : <ul style="list-style-type: none"> • Instance 0 with Identity Resource (ID:0) composed of 4 Instances as defined in Configuration C.12 • Instance 1 with Identity Resource (ID:0) composed of 2

Instances as defined in 1630-SetOfValues below.

1630-SetOfValues definition

lwm2m+tlv version

```

08 01 2E
  80 00 2B
    48 00 11 486F737420446576696365204944202332
    48 01 14 486F737420446576696365204D6F64656C202332
    
```

lwm2m+json version

```

{"bn": "/16/1/0",
 "e": [
   {"n": "0", "sv": "Host Device ID #2"},
   {"n": "1", "sv": "Host Device Model #2"}
 ]
}
    
```

6.3.5.7 LightweightM2M-1.0-int-1635 – Delete Portfolio Object Instance

Test Case Id	LightweightM2M-1.0-int-1635
Test Object	Client and Server
Test Case Description	Deleting all Instances of the Portfolio Object
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.12 as defined in Annex C ○ The Client is registered with the Server
Test Procedure	<p>A DISCOVER operations is performed by the Server on the Portfolio Object to retrieve its initial characteristics. Portfolio DELETE operations are performed for removing all the Portfolio Object Instances implemented in the LwM2M Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. An initial DISCOVER (CoAP GET Accept:40) operation is performed on Instance 0 of Portfolio Object ID:16 2. DELETE operations are performed for each Instance of the Portfolio Object 3. A new DISCOVER operation is performed on Portfolio Object ID:16

Pass-Criteria	<p>A. In test step 1, the Success Message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and containing at least: </16/0>,</16/0/0/>;dim=4</p> <p>B. In test step 2, successive Success messages are received by the Server (“2.04” Deleted) related to each DELETE operation targeting a Portfolio Object Instance</p> <p>C. In test step 3, the Success message (“2.05” Content) is received by the Server along with the payload related to the DISCOVER request and containing: </16> (Object ID16 present but without any Instance implemented in the Client)</p>
----------------------	---

6.3.6 BinaryAppDataContainer Object (ID 19) [1900-1999]

6.3.6.1 LightweightM2M-1.0-int-1900 – Observation and notification on Object ID: 19

Test Case Id	LightweightM2M-1.0-int-1900
Test Object	Client and Server
Test Case Description	Sending the observation policy to the device for resources of BinaryAppDataCont Object ID:19
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.15 as defined in Annex C ○ The Client is registered with the Server in indicating the Object ID:19 ○ Instance 0 is used to send application data from LwM2M client to LwM2M server. ○ Instance 1 is used to send application data from LwM2M server to LwM2M client.
Test Procedure	<p>A DISCOVER operation is requested by the Server to verify the Object ID:19</p> <p>The Server establishes an Observation relationship with the Client to acquire conditional notifications about Resources from the Binary AppData Container Instance 0:</p> <ul style="list-style-type: none"> • Data (ID: 0) <p>Normal flow:</p> <ol style="list-style-type: none"> 1. DISCOVER (CoAP GET Accept:40) operation is performed by the Server on Object ID:19 2. Server communicates with the Object ID:19 via WRITE-ATTRIBUTE (CoAP PUT) operations, to set the pmin & pmax Attributes of the targeted Resource (e.g. /19/0/0?pmin=300&pmax=3600) 3. Server sends OBSERVE (CoAP GET operation with Observe Option set to 0) messages for the targeted Resources(ID:0) to activate reporting 4. Client reports requested information with a NOTIFY message (CoAP responses)
Pass-Criteria	<p>A. In test step 1, the Server – along with the success message 2.05 – received the information related to Object ID:19 including </19>;ver="1.0", </19/0>,</19/0/0></p> <p>B. In test step 2, the Server receives success messages (“2.04” Changed) related to the WRITE-ATTRIBUTES operations</p> <p>C. In test step 3, the Server receives success messages (“2.05” Content)</p>

	<p>along with the initial values of Resource ID: 0</p> <p>D. In test step 4, the Server regularly receives consistent information on the targeted Resources ID: 0 Instance 0.</p>
--	---

6.3.6.2 LightweightM2M-1.0-int-1901 – Setting the writable Resource Data

Test Case Id	LightweightM2M-1.0-int-1901
Test Object	Client and Server
Test Case Description	Setting the writable Resource Data (ID:0) of Binary AppData Container Instance 1 and verifying the triggered actions produce the expected results
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.15 as defined in Appendix C ○ The Client is registered with the Server in indicating the Object ID:19 ○ Instance 0 is used to send application data from LwM2M client to LwM2M server. ○ Instance 1 is used to send application data from LwM2M server to LwM2M client.
Test Procedure	<p>A WRITE operation is performed by the Server to update a valid value in the Client in setting the Data Resource.</p> <p>The various Resources of this Binary AppData Container Objects must be set as expected by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A WRITE (CoAP PUT) operation with a valid value is performed by the Server on the Data Resource (/19/1/0) of the Binary AppData Container Object Instance 1 2. The Server READs (CoAP GET) the Binary AppData Container Object Instance 1 to get the values of the Data (/19/1/0)
Pass-Criteria	<p>A. In test step 1, the Server receives the success message “2.04” associated with the WRITE request.</p> <p>B. In test step 2, the Server receives the success message “2.05” along with the Data Resources values.</p> <p>C. In test step2, the queried value of Data resource is the same with that in Step1.</p>

6.3.7 Event Log Object (ID 20) [2000-2099]

6.3.7.1 LightweightM2M-1.0-int-2000 – Control the Log Function

Test Case Id	LightweightM2M-1.0-int-2000
Test Object	Client and Server
Test Case Description	Setting the writable Resource LogStart (ID:4011) of Event Log and verifying the triggered actions produce the expected results
Preconditions	<ul style="list-style-type: none"> ○ The Client supports the Configuration C.16 as defined in Appendix C ○ The Client is registered with the Server in indicating the Object ID:20
Test Procedure	<p>A WRITE operation is performed by the Server to update a valid value in the Client in setting the LogStart Resource.</p> <p>The various Resources of this Event Log Objects must be set as expected by the Client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. A WRITE (CoAP PUT) operation with a valid value is performed by

	<p>the Server on the LogStart Resource (/20/0/4011) of the Event Log Object</p> <p>2. The Server READs (CoAP GET) the Event Log Object to get the values of the LogStart (/20/0/4011)</p>
Pass-Criteria	<p>A. In test step 1, the Server receives the success message “2.04” associated with the WRITE request.</p> <p>B. In test step 2, the Server receives the success message “2.05” along with the LogStart Resources values.</p> <p>C. In test step2, the queried value of LogStart resource is the same with that in Step1.</p>

6.3.7.2 LightweightM2M-1.0-int-2001 – Querying the readable resources

Test Case Id	LightweightM2M-1.0-int-2001
Test Object	Client and Server
Test Case Description	Querying Information on the Firmware Update Object Instance for determining the current states and supported characteristics of the Client for such capability.
Preconditions	<ul style="list-style-type: none"> o The Client supports the Configuration C.16 as defined in Annex C o The Client is registered with the Server in indicating the Object ID:20
Test Procedure	<p>A READ operation from the Server on the EventLog Object Instance requesting TLV format has been received by the client.</p> <p>Normal flow:</p> <ol style="list-style-type: none"> 1. READ (CoAP GET) operation is performed on the EventLog Object Instance
Pass-Criteria	<p>A. In test step 1 :the Server receives the success message (2.05 Content) associated to its READ request</p> <p>B. In step 1: the values returned by the Client in TLV format are consistent with the Configuration C 16 used for that test (LogStatus (ID:4013), LogClass (ID:4010), and LogData(4014))</p>

Appendix A. Change History

(Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-ETS-LightweightM2M-V1_0_1-20170926-A	26 Sep 2017	Status changed to Approved by TP TP Ref # OMA-TP-2017-0039-INP_LightweightM2M-V1_0_ETS_for_Final_Approval
OMA-ETS-LightweightM2M-V1_0_2-20180815-A	15 Aug 2018	Status changed to Approved by DM Doc Ref # OMA-DM&SE-2018-0088-INP_LightweightM2M_V1_0_ETS_for_Final_Approval

Appendix B. Additional Information

B.1 Example of Test Configuration and Setup

The following hardware components were part of a test setup.

- M2M device equipped with LwM2M client
- Computer that runs the browser interface to the LwM2M Server component
- Server that runs the LwM2M Server software
- USIM provisioned for use on the network.
- External appliance which is connected directly to M2M device (e.g. light, temperature sensor, motor).

In addition the demonstration setup shall include the following software components:

- Measurement software which enables to see e.g. LwM2M messages and CoAP messages.
 - The software shall enable
 - GUI to trigger the chosen test cases
 - To see the flow of information between client and Server, e.g. on LwM2M protocol level and on CoAP transport level.

Appendix C. LwM2M Configurations

When a Test Case references a certain Configuration of that Annex, it means the features of that configuration are the minimum ones which must be supported by LwM2M Clients & Servers to pass that test; a richer configuration (superset) may be used instead, as long as it still contains the minimum features.

C.1 Basic Configuration 1

PSK Mode support, no SMS support

```
{
  "bn": "?",
  "e": [
    {
      "n": "0/0", "sv": "<LwM2M Server URI>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a>",
      "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11/0", "v": 0,
      "n": "3/0/16", "sv": "U",
    }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger
 Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

C.2 Basic Configuration 2

PSK Mode & SMS supports

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/0/0", "sv": "<LwM2M Server URI>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a>",
      "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "0/0/6", "v": 3,
      "n": "0/0/7", "sv": "<KIc,KID,SPI,TAR>", // opaque in Base64
      "n": "0/0/8", "sv": "<SMS Secret Keys>", // opaque in Base64
      "n": "0/0/9", "sv": "<MSISDN>",
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11/0", "v": 0,
      "n": "3/0/16", "sv": "US",
    }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger
 Device Object ID:3 implements the mandatory Executable Resource ID: - Reboot

C.3 Configuration 3

This configuration is a superset of Configuration C.1.

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/0/0", "sv": "<LwM2M Server URI>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a> ",
      "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/2", "v": 1,
      "n": "1/0/3", "v": 10,
      "n": "1/0/5", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11", "v": 0,
      "n": "3/0/16", "sv": "U",
    ]
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger
 Server Object ID:1 implements the optional Executable Resource ID:4 - DisableDevice Object ID:3
 implements the mandatory Executable Resource ID:4 - Reboot

C.4 Configuration 4

Configuration 4 is a superset of Configuration 3 and can be used instead ; it implements the optional Resources related to external or internal battery which the voltage could vary.

```
{
  "bn": "/",
  "e": [
    { "n": "0/0/0", "sv": "<LwM2M Server URI>" },
    { "n": "0/0/1", "bv": false },
    { "n": "0/0/2", "v": 0 },
    { "n": "0/0/3", "sv": "<PSK Identity>" }, // opaque in Base64
    { "n": "0/0/4", "sv": "<n/a>" },
    { "n": "0/0/5", "sv": "<Secret Key>" }, // opaque in Base64
    { "n": "0/0/10", "v": 1 },
    { "n": "1/0/0", "v": 1 },
    { "n": "1/0/1", "v": 86400 },
    { "n": "1/0/2", "v": 1 },
    { "n": "1/0/3", "v": 10 },
    { "n": "1/0/5", "v": 86400 },
    { "n": "1/0/6", "bv": false },
    { "n": "1/0/7", "sv": "U" },
    { "n": "3/0/0", "sv": "<Manufacturer Name>" },
    { "n": "3/0/1", "sv": "<Model Name>" },
    { "n": "3/0/2", "sv": "<Serial Number>" },
    { "n": "3/0/3", "sv": "<Firmware Version>" },
    { "n": "3/0/6/0", "v": 1 }, // Internal Battery
    { "n": "3/0/6/1", "v": 2 }, // External Battery
    { "n": "3/0/7/0", "v": 0 }, // Internal Battery mV
    { "n": "3/0/7/1", "v": 0 }, // External Battery mV
    { "n": "3/0/8/0", "v": 0 }, // Internal Battery mA
    { "n": "3/0/8/1", "v": 0 }, // External Battery mA
    { "n": "3/0/9", "v": 0 }, // Internal Battery level
    { "n": "3/0/11", "v": 0 },
    { "n": "3/0/16", "sv": "U" },
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Server Object ID:1 implements the optional Executable Resource ID:4 - Disable

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

C.5 Configuration 5

This configuration is a superset of Configuration C.1 and can be used instead; it implements the full Connectivity Monitoring Object ID:4

```
{
  "bn": "/",
  "e": [
    { "n": "0/0/0", "sv": "<LwM2M Server URI>" },
    { "n": "0/0/1", "bv": false },
    { "n": "0/0/2", "v": 0 },
    { "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64 },
    { "n": "0/0/4", "sv": "<n/a>" },
    { "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64 },
    { "n": "0/0/10", "v": 1 },
    { "n": "1/0/0", "v": 1 },
    { "n": "1/0/1", "v": 86400 },
    { "n": "1/0/2", "v": 1 },
    { "n": "1/0/3", "v": 10 },
    { "n": "1/0/5", "v": 86400 },
    { "n": "1/0/6", "bv": false },
    { "n": "1/0/7", "sv": "U" },
    { "n": "3/0/0", "sv": "<Manufacturer Name>" },
    { "n": "3/0/1", "sv": "<Model Name>" },
    { "n": "3/0/2", "sv": "<Serial Number>" },
    { "n": "3/0/3", "sv": "<Firmware Version>" },
    { "n": "3/0/11", "v": 0 },
    { "n": "3/0/16", "sv": "U" },
    { "n": "4/0/0", "v": "<Network Bearer>" },
    { "n": "4/0/1/0", "v": "<Available Network Bearer 0>" },
    { "n": "4/0/1/1", "v": "<Available Network Bearer 1>" },
    { "n": "4/0/2", "v": "<Radio Signal Strength>" },
    { "n": "4/0/3", "v": "<Link Quality>" },
    { "n": "4/0/4/0", "sv": "<IP Address 0>" },
    { "n": "4/0/4/1", "sv": "<IP Address 1>" },
    { "n": "4/0/5/0", "sv": "<Router IP Address 0>" },
    { "n": "4/0/5/1", "sv": "<Router IP Address 1>" },
    { "n": "4/0/6", "v": "<Link Utilization>" },
    { "n": "4/0/7/0", "sv": "<APN 0>" },
    { "n": "4/0/7/1", "sv": "<APN 1>" },
    { "n": "4/0/8", "v": "<Cell ID>" },
    { "n": "4/0/9", "v": "<SMNC>" },
    { "n": "4/0/10", "v": "<SMCC>" }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger
 Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

C.6 Configuration 6

Configuration 6 is a superset of Configuration 4 and can be used instead; the configuration C.4, has been completed with all the optional resources of the Device Object.

```
{ "bn": "/",
  "e": [
    { "n": "0/0", "sv": <LwM2M Server URI> },
    { "n": "0/0/1", "bv": false },
    { "n": "0/0/2", "v": 0 },
    { "n": "0/0/3", "sv": <PSK Identity> }, // opaque in Base64
    { "n": "0/0/4", "sv": <n/a > },
    { "n": "0/0/5", "sv": <Secret Key> }, // opaque in Base64
    { "n": "0/0/10", "v": 1 },
    { "n": "1/0/0", "v": 1 },
    { "n": "1/0/1", "v": 86400 },
    { "n": "1/0/2", "v": 1 },
    { "n": "1/0/3", "v": 10 },
    { "n": "1/0/5", "v": 86400 },
    { "n": "1/0/6", "bv": false },
    { "n": "1/0/7", "sv": "U" },
    { "n": "3/0/0", "sv": <Manufacturer Name> },
    { "n": "3/0/1", "sv": <Model Name> },
    { "n": "3/0/2", "sv": <Serial Number> },
    { "n": "3/0/3", "sv": <Firmware Version> },
    { "n": "3/0/6/0", "v": 1 }, // Internal Battery
    { "n": "3/0/6/1", "v": 2 }, // External Battery
    { "n": "3/0/7/0", "v": <Internal Battery Voltage mV> },
    { "n": "3/0/7/1", "v": <External Battery Voltage mV > },
    { "n": "3/0/8/0", "v": <Internal Battery Current mA> },
    { "n": "3/0/8/1", "v": <External Battery Current mA > },
    { "n": "3/0/9", "v": <Internal Battery Level> },
    { "n": "3/0/10", "v": <Memory Free> },
    { "n": "3/0/11", "v": <Error Code> },
    { "n": "3/0/13", "t": <Current Time > },
    { "n": "3/0/14", "sv": <UTC Offset> },
    { "n": "3/0/15", "sv": <TimeZone > },
    { "n": "3/0/16", "sv": "U" },
    { "n": "3/0/17", "sv": <Device Type> },
    { "n": "3/0/18", "sv": <Hardware Version> },
    { "n": "3/0/19", "sv": <Software Version> },
    { "n": "3/0/20", "sv": <Battery Status> },
    { "n": "3/0/21", "sv": <Memory Total> },
    { "n": "3/0/22", "ol": <ExtDevInfo> }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Server Object ID:1 implements the optional Executable Resource ID:4 - Disable

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

Device Object ID:3 implements the optional Executable Resource ID:5 - Factory Reset

Device Object ID:3 implements the optional Executable Resource ID:12 - Reset Error Code

C.7 Configuration 7

This configuration is a superset of Configuration C.1 and can be used instead ; it implements the full Location Object ID:6

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/0", "sv": "<LwM2M Server URI>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a> ",
      "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/2", "v": 1,
      "n": "1/0/3", "v": 10,
      "n": "1/0/5", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11", "v": 0,
      "n": "3/0/16", "sv": "U",
      "n": "6/0/0", "v": 43.5723,
      "n": "6/0/1", "v": 153.21760,
      "n": "6/0/2", "v": 140,
      "n": "6/0/3", "v": 100,
      "n": "6/0/4", "sv": "0", // opaque 3GPP-TS- 23.032 Base64 representation
      "n": "6/0/5", "t": 1367491215, // time
      "n": "6/0/6", "v": 3
    }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

C.8 Configuration 8

```

{"bn":"/",
 "e":[
  {"n":"0/0/0","sv":<LwM2M Server URI>},
  {"n":"0/0/1","bv":false},
  {"n":"0/0/2","v":0},
  {"n":"0/0/3","sv":<PSK Identity>}, // opaque in Base64
  {"n":"0/0/4","sv":<n/a >},
  {"n":"0/0/5","sv":<Secret Key>}, // opaque in Base64
  {"n":"0/0/10","v":1},
  {"n":"1/0/0","v":1},
  {"n":"1/0/1","v":86400},
  {"n":"1/0/2","v":1},
  {"n":"1/0/3","v":10},
  {"n":"1/0/5","v":86400},
  {"n":"1/0/6","bv":false},
  {"n":"1/0/7","sv":"U"},
  {"n":"3/0/0","sv":<Manufacturer Name>},
  {"n":"3/0/1","sv":<Model Name>},
  {"n":"3/0/2","sv":<Serial Number>},
  {"n":"3/0/3","sv":<Firmware Version>},
  {"n":"3/0/11","v":0},
  {"n":"3/0/16","sv":"U"},
  {"n":"5/0/0","sv":<Firmware Package>}, // opaque in Base64
  {"n":"5/0/1","sv":<Package URI>},
  {"n":"5/0/3","v":<State>},
  {"n":"5/0/5","v":<Update Result>},
  {"n":"5/0/6","sv":<Package Name>},
  {"n":"5/0/7","sv":<Package Version>},
  {"n":"5/0/8","v":<Firmware Update Protocol Support> }, // CoaP only if absent
  {"n":"5/0/9","v":<Firmware Update Delivery Method> }
 ]
 }

```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger
 Server Object ID:1 implements the optional Executable Resource ID:4 - Disable
 Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot
 Device Object ID:5 implements the mandatory Executable Resource ID:2 - Update

C.9 Configuration 9

This configuration is a superset of Configuration C.1 and can be used instead ; it implements the full Connectivity Statistics Object ID:7

```
{ "bn": "?",
  "e": [
    { "n": "0/0/0", "sv": "<LwM2M Server URI>" },
    { "n": "0/0/1", "bv": false },
    { "n": "0/0/2", "v": 0 },
    { "n": "0/0/3", "sv": "<PSK Identity>" }, // opaque in Base64
    { "n": "0/0/4", "sv": "<n/a>" },
    { "n": "0/0/5", "sv": "<Secret Key>" }, // opaque in Base64
    { "n": "0/0/10", "v": 1 },
    { "n": "1/0/0", "v": 1 },
    { "n": "1/0/1", "v": 86400 },
    { "n": "1/0/2", "v": 1 },
    { "n": "1/0/3", "v": 10 },
    { "n": "1/0/5", "v": 86400 },
    { "n": "1/0/6", "bv": false },
    { "n": "1/0/7", "sv": "U" },
    { "n": "3/0/0", "sv": "<Manufacturer Name>" },
    { "n": "3/0/1", "sv": "<Model Name>" },
    { "n": "3/0/2", "sv": "<Serial Number>" },
    { "n": "3/0/3", "sv": "<Firmware Version>" },
    { "n": "3/0/11", "v": 0 },
    { "n": "3/0/16", "sv": "U" },
    { "n": "7/0/0", "v": 0 },
    { "n": "7/0/1", "v": 0 },
    { "n": "7/0/2", "v": 0 },
    { "n": "7/0/3", "v": 0 },
    { "n": "7/0/4", "v": 0 },
    { "n": "7/0/5", "v": 0 },
    { "n": "7/0/8", "v": 0 }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

Device Object ID:7 implements the mandatory Executable Resource ID:6 - Start

Device Object ID:7 implements the mandatory Executable Resource ID:7 - Stop

C.10 Basic Connectivity Management Configuration 10

PSK Mode support, no SMS support

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/0/0", "sv": "<LwM2M Server URI>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a>",
      "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11/0", "v": 0,
      "n": "3/0/16", "sv": "U",
      "n": "10/0/4", "v": 3600,
      "n": "10/0/5", "v": 600,
      "n": "10/0/6", "v": 40,
      "n": "10/0/8", "v": 20.48,
      "n": "10/0/9", "v": 51.2,
      "n": "11/0/0", "sv": "<Profile Name>",
      "n": "11/0/3", "b": true,
      "n": "12/0/0", "sv": "<wlan identifier>",
      "n": "12/0/1", "v": 0,
      "n": "12/0/3", "v": 0,
      "n": "12/0/4", "sv": "<MAC address>",
      "n": "12/0/8", "v": 0,
      "n": "12/0/4", "v": 0,
      "n": "12/0/14", "v": 0,
      "n": "12/0/15", "v": 0,
    }
  ]
}
```

C.11 Basic Connectivity Management Configuration 11 supporting Object ID:10 version 1.1

PSK Mode support, no SMS support

```
{“bn”：“/”,
  “e”:[
    {“n”：“0/0/0”,“sv”:<LwM2M Server URI>},
    {“n”：“0/0/1”,“bv”:false},
    {“n”：“0/0/2”,“v”：0},
    {“n”：“0/0/3”,“sv”:<PSK Identity>}, // opaque in Base64
    {“n”：“0/0/4”,“sv”:<n/a> },
    {“n”：“0/0/5”,“sv”:<Secret Key>}, // opaque in Base64
    {“n”：“0/0/10”,“v”：1},
    {“n”：“1/0/0”,“v”：1},
    {“n”：“1/0/1”,“v”：86400},
    {“n”：“1/0/6”,“bv”：false},
    {“n”：“1/0/7”,“sv”：“U”},
    {“n”：“3/0/0”,“sv”:<Manufacturer Name>},
    {“n”：“3/0/1”,“sv”:<Model Name>},
    {“n”：“3/0/2”,“sv”:<Serial Number>},
    {“n”：“3/0/3”,“sv”:<Firmware Version>},
    {“n”：“3/0/11/0”,“v”：0},
    {“n”：“3/0/16”,“sv”：“U”},
    {“n”：“10/0/4”,“v”：3600},
    {“n”：“10/0/5”,“v”：600},
    {“n”：“10/0/6”,“v”：40},
    {“n”：“10/0/8”,“v”：20.48},
    {“n”：“10/0/9”,“v”：51.2},
    {“n”：“10/0/11”,“ov”：“11:0”},
    {“n”：“10/0/13”,“v”：3},
    {“n”：“10/0/14”,“v”：1},
    {“n”：“11/0/0”,“sv”:<Profile Name>},
    {“n”：“11/0/3”,“b”：true},
    {“n”：“12/0/0”,“sv”:<wlan identifier>},
    {“n”：“12/0/1”,“v”：0},
    {“n”：“12/0/3”,“v”：0},
    {“n”：“12/0/4”,“sv”:<MAC address>},
    {“n”：“12/0/8”,“v”：0},
    {“n”：“12/0/4”,“v”：0},
    {“n”：“12/0/14”,“v”：0},
    {“n”：“12/0/15”,“v”：0},
```

C.12 Configuration 12

This configuration is a superset of Configuration C.1 and can be used instead; it implements the Basic Portfolio Object ID:16 (Identity Resource with 4 Instances)

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/0/0", "sv": "<LwM2M Server URI>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a >",
      "n": "0/0/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/2", "v": 1,
      "n": "1/0/3", "v": 10,
      "n": "1/0/5", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11", "v": 0,
      "n": "3/0/16", "sv": "U",
      "n": "16/0/0/0", "sv": "Host Device ID #1",
      "n": "16/0/0/1", "sv": "Host Device Manufacturer #1",
      "n": "16/0/0/2", "sv": "Host Device Model #1",
      "n": "16/0/0/3", "sv": "Host Device Software Version #1"
    }
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

C.13 Bootstrap Server Contact Configuration

PSK Mode support, no SMS support

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/1/0", "sv": "<LwM2M Bootstrap Server URI>",
      "n": "0/1/1", "bv": true,
      "n": "0/1/2", "v": 0,
      "n": "0/1/3", "sv": "<PSK Identity>", // opaque in Base64
      "n": "0/1/4", "sv": "<n/a> ",
      "n": "0/1/5", "sv": "<Secret Key>", // opaque in Base64
      "n": "3/0/0", "sv": "<Manufacturer Name>",
      "n": "3/0/1", "sv": "<Model Name>",
      "n": "3/0/2", "sv": "<Serial Number>",
      "n": "3/0/3", "sv": "<Firmware Version>",
      "n": "3/0/11/0", "v": 0,
      "n": "3/0/16", "sv": "U"
    }
  ]
}
```

The Mandatory Object Server (ID:1) is supported by the Client but has no Instance in that Configuration

C.14 Multi-Servers Context Initial Configuration

PSK Mode support, no SMS support

```
{
  "bn": "",
  "e": [
    {
      "n": "0/1/0", "sv": "<LwM2M Bootstrap Server URI>",
      "n": "0/1/1", "bv": true,
      "n": "0/1/2", "v": 0,
      "n": "0/1/3", "sv": "<PSK Identity #0>", // opaque in Base64
      "n": "0/1/4", "sv": "<n/a> ",
      "n": "0/1/5", "sv": "<Secret Key #0>", // opaque in Base64

      // Server Account #1
      "n": "0/0/0", "sv": "<LwM2M Server URI #1>",
      "n": "0/0/1", "bv": false,
      "n": "0/0/2", "v": 0,
      "n": "0/0/3", "sv": "<PSK Identity #1>", // opaque in Base64
      "n": "0/0/4", "sv": "<n/a> ",
      "n": "0/0/5", "sv": "<Secret Key #1>", // opaque in Base64
      "n": "0/0/10", "v": 1,
      "n": "1/0/0", "v": 1,
      "n": "1/0/1", "v": 86400,
      "n": "1/0/2", "v": 1,
      "n": "1/0/3", "v": 10,
      "n": "1/0/5", "v": 86400,
      "n": "1/0/6", "bv": false,
      "n": "1/0/7", "sv": "U",

      // Server Account #2
      "n": "0/2/0", "sv": "<LwM2M Server URI #2>",
      "n": "0/2/1", "bv": false,
      "n": "0/2/2", "v": 0,
      "n": "0/2/3", "sv": "<PSK Identity #2>", // opaque in Base64
      "n": "0/2/4", "sv": "<n/a> ",
      "n": "0/2/5", "sv": "<Secret Key #2>", // opaque in Base64
      "n": "0/2/10", "v": 2,
      "n": "1/1/0", "v": 2,
      "n": "1/1/1", "v": 86400,
      "n": "1/1/2", "v": 1,
      "n": "1/1/3", "v": 10,
      "n": "1/1/5", "v": 86400,
      "n": "1/1/6", "bv": false,
      "n": "1/1/7", "sv": "U"
    }
  ]
}
```

```
// Access Control Object : Server #1 has full access, Server #2 no access
{"n":"2/0/0","v":1},
{"n":"2/0/1","v":0},
{"n":"2/0/2/1","v":15},           // Server #1 full access
{"n":"2/0/3","v":1},

// Access Control Object : Server #2 has full access, Server #1 no access
{"n":"2/0/0","v":1},
{"n":"2/0/1","v":1},
{"n":"2/0/2/2","v":15},         // Server #2 full access
{"n":"2/0/3","v":2},

// Access Control Object : Object Device Server #2 & #1 have full access (i.e. RWE)
{"n":"2/0/0","v":3},
{"n":"2/0/1","v":0},
{"n":"2/0/2/2","v":15},
{"n":"2/0/3","v":<MAX_ID>},

{"n":"3/0/0","sv":<Manufacturer Name>},
{"n":"3/0/1","sv":<Model Name>},
{"n":"3/0/2","sv":<Serial Number>},
{"n":"3/0/3","sv":<Firmware Version>},
{"n":"3/0/11/0","v":0},
{"n":"3/0/16","sv":"U"},

]
} // C14
```

C.15 Binary AppData Container Configuration

This configuration is a superset of Configuration C.1 and can be used instead; it implements the Binary AppData Container Object ID: 19 (Identity Resource with 4 Instances)

```
{“bn”：“?”,
  “e”:[
    {"n":"0/0/0","sv":<LwM2M Server URI>},
    {"n":"0/0/1","bv":false},
    {"n":"0/0/2","v":0},
    {"n":"0/0/3","sv":<PSK Identity>}, // opaque in Base64
    {"n":"0/0/4","sv":<n/a>},
    {"n":"0/0/5","sv":<Secret Key>}, // opaque in Base64
    {"n":"0/0/10","v":1},
    {"n":"1/0/0","v":1},
    {"n":"1/0/1","v":86400},
    {"n":"1/0/2","v":1},
    {"n":"1/0/3","v":10},
    {"n":"1/0/5","v":86400},
    {"n":"1/0/6","bv":false},
    {"n":"1/0/7","sv":"U"},
    {"n":"3/0/0","sv":<Manufacturer Name>},
    {"n":"3/0/1","sv":<Model Name>},
    {"n":"3/0/2","sv":<Serial Number>},
    {"n":"3/0/3","sv":<Firmware Version>},
    {"n":"3/0/11","v":0}, {"n":"3/0/16","sv":"U"},
    {"n":"19/0/0","sv":<InNlcnZpY2VJZCI6Ik1ldGVyIiwNCiJzZXJ2aWNIRGF0YSI6ew0KImN1cnJlbnRSZWFkaW5nIjojNDYuMyIsDQoic2lnbmFsU3RyZW5ndGgiOjE2LA0KImRhaWx5QWN0aXZpdHIUaW11Ijo1NzA2DQo=>, // opaque in Base64
    {"n":"19/1/0","sv":<n/a>},
  ]
}
```

Server Object ID: 1 implements the mandatory Executable Resource ID: 8 - Registration Update Trigger

Device Object ID: 3 implements the mandatory Executable Resource ID: 4 - Reboot

C.16 Event Log Configuration

This configuration is a superset of Configuration C.1 and can be used instead; it implements the Event Log Object ID: 20 (Identity Resource with 4 Instances)

```
{
  "bn": "/",
  "e": [
    {
      "n": "0/0", "sv": "<LwM2M Server URI>"
    },
    {
      "n": "0/0/1", "bv": false
    },
    {
      "n": "0/0/2", "v": 0
    },
    {
      "n": "0/0/3", "sv": "<PSK Identity>"
    }, // opaque in Base64
    {
      "n": "0/0/4", "sv": "<n/a>"
    },
    {
      "n": "0/0/5", "sv": "<Secret Key>"
    }, // opaque in Base64
    {
      "n": "0/0/10", "v": 1
    },
    {
      "n": "1/0/0", "v": 1
    },
    {
      "n": "1/0/1", "v": 86400
    },
    {
      "n": "1/0/2", "v": 1
    },
    {
      "n": "1/0/3", "v": 10
    },
    {
      "n": "1/0/5", "v": 86400
    },
    {
      "n": "1/0/6", "bv": false
    },
    {
      "n": "1/0/7", "sv": "U"
    },
    {
      "n": "3/0/0", "sv": "<Manufacturer Name>"
    },
    {
      "n": "3/0/1", "sv": "<Model Name>"
    },
    {
      "n": "3/0/2", "sv": "<Serial Number>"
    },
    {
      "n": "3/0/3", "sv": "<Firmware Version>"
    },
    {
      "n": "3/0/11", "v": 0
    }, {
      "n": "3/0/16", "sv": "U"
    },
    {
      "n": "20/0/4010", "v": 0
    },
    {
      "n": "20/0/4011", "sv": "0,100"
    },
    {
      "n": "20/0/4013", "v": 3
    },
    {
      "n": "20/0/4014", "sv": "61-7C-E3-01-C1-11-00-00-05-00-60-18-18-18-0C-00-01-41-06-00-02-00-40-0C-00-00-00-00"
    }, // opaque in Base64
  ]
}
```

Server Object ID: 1 implements the mandatory Executable Resource ID: 8 - Registration Update Trigger

Device Object ID: 3 implements the mandatory Executable Resource ID: 4 - Reboot

C.17 Basic Configuration 17

Note: The referenced certificates & keys are provided in a separate ZIP file.

This configuration is a modification of the “Basic Configuration 1” file, which supports certificate-based authentication.

```
{“bn”：“/”,
  “e”:[
    {“n”：“0/0/0”,“sv”：coaps://server.example.com}, // The certificate was created for server.example.com.
    {“n”：“0/0/1”,“bv”：false},
    {“n”：“0/0/2”,“v”：2}, // certificate mode
    {“n”：“0/0/3”,“sv”：<Client Certificate >}, // Contains the client.crt from ZIP archive
    {“n”：“0/0/4”,“sv”：<Server Certificate >}, // Contains the server.crt from ZIP archive
    {“n”：“0/0/5”,“sv”：<Client Private Key>}, // Contains the client.key from ZIP archive
    {“n”：“0/0/10”,“v”：1},
    {“n”：“1/0/0”,“v”：1},
    {“n”：“1/0/1”,“v”：86400},
    {“n”：“1/0/6”,“bv”：false},
    {“n”：“1/0/7”,“sv”：“U”},
    {“n”：“3/0/0”,“sv”：<Manufacturer Name>},
    {“n”：“3/0/1”,“sv”：<Model Name>},
    {“n”：“3/0/2”,“sv”：<Serial Number>},
    {“n”：“3/0/3”,“sv”：<Firmware Version>},
    {“n”：“3/0/11/0”,“v”：0},
    {“n”：“3/0/16”,“sv”：“U”},
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

C.18 Basic Configuration 18

Note: The referenced certificates & keys are provided in a separate ZIP file.

This configuration is a modification of the “Basic Configuration 17” file, which supports certificate-based authentication. However, the LwM2M Server URI is set to a different value.

```
{“bn”：“/”,
  “e”:[
    {“n”：“0/0/0”,“sv”：coaps://server-fail.example.com}, // The certificate was created for server.example.com.
    {“n”：“0/0/1”,“bv”：false},
    {“n”：“0/0/2”,“v”：2}, // certificate mode
    {“n”：“0/0/3”,“sv”：<Client Certificate >}, // Contains the client.crt from ZIP archive
    {“n”：“0/0/4”,“sv”： <Server Certificate> }, // Contains the server.crt from ZIP archive
    {“n”：“0/0/5”,“sv”： <Client Private Key>}, // Contains the client.key from ZIP archive
    {“n”：“0/0/10”,“v”：1},
    {“n”：“1/0/0”,“v”：1},
    {“n”：“1/0/1”,“v”：86400},
    {“n”：“1/0/6”,“bv”：false},
    {“n”：“1/0/7”,“sv”：“U”},
    {“n”：“3/0/0”,“sv”：<Manufacturer Name>},
    {“n”：“3/0/1”,“sv”：<Model Name>},
    {“n”：“3/0/2”,“sv”：<Serial Number>},
    {“n”：“3/0/3”,“sv”：<Firmware Version>},
    {“n”：“3/0/11/0”,“v”：0},
    {“n”：“3/0/16”,“sv”：“U”},
  ]
}
```

Server Object ID:1 implements the mandatory Executable Resource ID:8 - Registration Update Trigger

Device Object ID:3 implements the mandatory Executable Resource ID:4 - Reboot

Appendix D. Core Test Coverage

The following table aims at quantifying the Test Coverage of LwM2M 1.0 provided by the set of interoperable Test Cases defined in ETS 1.0.1.

The LwM2M Test Cases are distributed into 8 Functional Blocks (*BLOCK*) :

- Bootstrap Interface
- Registration Interface
- DM& SE Interface
- Information Reporting Interface
- Security
- Queue Mode
- Core Objects
- Multi-Server Context

<i>BLOCK</i>								
Test	<i>Cover</i>	<i>Pass</i>	<i>W</i>	Test	<i>Test IDs</i>	<i>Avail</i>	<i>Pass</i>	Weight
<i>Bootstrap Interface</i>	20,00%	0,0%	10	[0-99]		0,2	0	1,00
				Client Initiated Bootstrap Mode	0	1	0	0,20
				Server Initiated Bootstrap Mode		0	0	0,10
				Smartcard Bootstrap Mode		0	0	0,20

For each Functional Block, an individual Test Coverage qualification is determined (*Cover*) and the final LwM2M Enabler Test Coverage is obtained by recombining each individual Test Coverage results pondered by an weight (*W*) specific to each Functional Block.

A Functional Block is composed of several Test Cases (*Test IDs*) which contribute to the final Block Functional Test Coverage result according to the individual availability (*Avail : 0 or 1*) and the individual weight (*Weight*) of these tests inside the group.

In addition, this table can be used to determin the gap between a certain implementation (LwM2M Server or Client) and the LwM2M Specification, in regards to the functionalities of the enabler covered by this Test Specification.

In the above example, the single available Test Case from the Bootstrap Interface, will only contribute to 2% of the final Test Coverage status, since only 20% of the Function Block functionality is covered (*Cover*) for a Weight of 10% regarding the ponderation of that Functional Block in the final LwM2M Enabler Test Coverage result.

Functional Test Coverage : Client&Server TS 1.0 Interoperability

Test Coverage Rate (TCR)

Overall Spec TCR

55,66%

Implem Test Result

BLOCK				Test	Test IDs	Avail	Pass	Weight
Test	Cover	Pas s	W					
Bootstrap Interface	80,00%	0,0%	10	[0-99]		0,8	0	1,00
				Client Initiated Bootstrap Mode (3)	0	1	0	0,60
				Server Initiated Bootstrap Mode		0	0	0,10
				Smartcard Bootstrap Mode		1	0	0,20
						0	0	0,10

Registration Interface	80,0%	0,0%	15	[100-199]		0,8	0	1,00
				Initial Registration	101	1	0	0,2
				Registration Update	102	1	0	0,2
				Deregistration	103	1	0	0,2
				Registration Update Trigger	104	1	0	0,2
						0	0	0,2

DM & SE Interface	70,00%	0,00%	25	[200-299]		0,7	0	1,00
				Querying basic information in Plain Text format	201	1	0	0,02
				Querying basic information in Opaque format	202	1	0	0,02
				Querying basic information in TLV format	203	1	0	0,04
				Querying basic information in JSON format	204	1	0	0,02
				Setting basic information in Plain Text format	205	1	0	0,02
				Setting basic information in Opaque format	210	1	0	0,02
				Setting basic information in TLV format	215	1	0	0,04
				Setting basic information in JSON format	220	1	0	0,02
				<i>Advanced Update Information</i>		0	0	0,20
				Executable Resource: Rebooting the device	241	1	0	0,10
				Discover Command	260	1	0	0,10
				Create Object Instance	270	1	0	0,20
				<i>Create Multiple Resource Instance</i>	271	0	0	0,10
Delete Object Instance	290	1	0	0,10				

Information Reporting Interface	60,00%	0,00%	10	[300-399]		0,6	0	1,00
				Observation and Notification of parameter values	301	1	0	0,2
				Cancel Observations using Reset Operation	302	1	0	0,2

				Cancel observations using Observe with Cancel parameter	303	1	0	0,2
				<i>Advanced Observation</i>		0	0	0,4

Security	30,00%	0,00%	15	[400-449] UDP Channel Security – Pre-shared Key Mod	401	0,3	0	1,00
				<i>Advanced Security</i>		1	0	0,3
						0	0	0,7

Queue Mode	0,00%	0,00%	5	[450-499]		0,0	0	1,00
						0	0	1,0

<i>Core Objects</i>	<i>41,88%</i>	<i>0,00%</i>	<i>14</i>
---------------------	---------------	--------------	-----------

Multi-Server Context	30,00%	0,00%	6	[950-999] Multi-Servers Registration	950	0,3	0	1,00
				Multi-Servers & Attributes	951	1	0	0,1
								0,2
								0,7

BLOCK									
Test	Cover	Pass	W	Test	Test IDs	Avail	Pass	Weight	
<i>Core Objects</i>	<i>41,88%</i>	<i>0,00%</i>	<i>14</i>						
Core Security Object	0,00%	0,00%	2	[500-549]		0	0	0,0	
						0	0	1,0	
Core Server Object	60,00%	0,00%	2	[550-599]					
				Access Check to the Resources	551	1	0	0,2	
				Check of the Disable (De-Registration) capability	555	1	0	0,2	
				Check of the Update Registration capability	556	1	0	0,2	
				Delayed Report Notification	560	0	0	0,0	
				Create Object Instance	565	0	0	0,4	
				Delete Object Instance	566	0	0	0,0	
Core Access Control Object	0,00%	0,00%	2	[600-649]		0	0	1,00	
								1,0	
								0,0	
								0,0	
Core Device Object	80,00%	0,00%	2	[650-699]		0,8	0	1,00	
				Check Access to the Resources	651	1	0	0,10	
				Querying the firmware version from the client	652	1	0	0,10	
				Check of the "Reboot" capability	655	1	0	0,10	
				Check of the "Factory Reset" capability	656	1	0	0,10	
				Check of the "Error Code" functionality	657	1	0	0,10	
				Basic Observation and notification of Device Object Resources	660	1	0	0,10	
				Extended Observation and notification of Device Object Resources	661	1	0	0,10	
				Create Multiple Resource Instances	670	0	0	0,20	
				Create Object Instance	680	1	0	0,05	
				Delete Object Instance	685	1	0	0,05	
						0	0	0,00	
Core Connectivity Monitoring Object	45,00%	0,00%	1	[700-749]		0,4	0	1,00	
				Querying the readable resources of object	701	1	0	0,25	
				Setting the writable resources	705	1	0	0,05	
				Basic Observation and notification on Connectivity Monitoring Obj	710	1	0	0,15	
				Extended Observation and notification of Connectivity Monitoring t	711	0	0	0,20	
				Create Multiple Resource Instances	720	0	0	0,25	
				Create Object Instance	730	0	0	0,05	
				Delete Object Instance	735	0	0	0,05	
Core Firmware Object Update	35,00%	0,00%	3	[750-799]		0,3	0	1,00	
				Querying the readable resources	751	1	0	0,05	
				Setting the writable Resource Package	755	1	0	0,05	
				Setting the writable Resource Package URI	756	1	0	0,05	
				Basic Observation and notification on Firmware Update Object Res	760	1	0	0,05	
				Extended Observation and notification on Firmware Update Object	761	0	0	0,05	
				Successful Firmware update (via COAP)	770	1	0	0,15	
				Successful Firmware update (via Alt mechanism)	771	1	0	0,15	
				Error Case 1: firmware installation without downloaded package	772	1	0	0,05	
				Error Case 2: shortage of storage memory	773	1	0	0,05	
				Error Case 3: out of memory	774	1	0	0,05	
				Error Case 4: Connection lost during download (package URI)	775	1	0	0,05	
				Error Case 5: Package Integrity check failure	776	1	0	0,05	
				Error Case 6: unsupported package type	777	1	0	0,05	
				Error Case 7: invalid URI (package URI)	778	1	0	0,05	
				Error Case 8: Unsuccessful Firmware Update	779	1	0	0,05	
				Error Case 9: Unsupported protocol	780	0	0	0,05	

Core Object	Weight	Weight	Count	Code Range	Code	Weight	Weight	Weight
Core Location Object	50,00%	0,00%	1	[800-849]		0,5	0	1,00
				Querying the readable resources of object	801	1	0	0,15
				Setting the writable resources	805	1	0	0,05
				Basic Observation and notification on Location Object Instance	810	1	0	0,30
				Extended Observation and notification of Location Object Instance	811	0	0	0,35
				Create Multiple Resource Instances	820	0	0	0,05
				Create Object Instance	830	0	0	0,05
				Delete Object Instance	835	0	0	0,05
Core Connectivity Statistics Object	65,00%	0,00%	1	[900-949]		0,6	0	1,00
				Querying a Data Collection from Connectivity Object Instance	901	1	0	0,20
				Setting the writable resources	905	1	0	0,20
				Basic Observation and notification on Connectivity Monitoring Obj	910	1	0	0,25
				Extended Observation and notification of Connectivity Statistics Ol	911	0	0	0,20
				Create Multiple Resource Instances	920	0	0	0,05
				Create Object Instance	930	0	0	0,05
				Delete Object Instance	935	0	0	0,05

Appendix E. Enabler Validation Plan

The validation of this enabler will be made based on the successful execution of test cases defined in the [LWM2M-ETS].

A summary of test results will be the evidence that those test cases were executed. The lack of open Problem Reports will be the evidence the test cases were successful executed and that any issue found whether on test specifications, technical specifications or other was correctly addressed.

E.1 Entry Criteria for TestFest

The following tests need to be performed and passed by implementations by members wishing to participate in the TestFest. This ensures minimal requisite capability of the implementations. The tests are defined in the ETS [LWM2M-ETS] and any special comments are noted.

Test Case Id	Special Conditions
LightweightM2M-1.0-int-101	
LightweightM2M-1.0-int-102	
LightweightM2M-1.0-int-201	
LightweightM2M-1.0-int-203	

Table 1: Listing of Tests for Entry Criteria for TestFest

E.2 Enabler Validation Test Cases

The following table should list the set of tests that are used for enabler validation. Not applicable in this case as the enabler has been already agreed.

Test Case Id	ETR Requirement Id/Feature Key	ETR Status	Notes

Table 2: Enabler Validation Test Cases