

TECHNICAL REPORT

COMpanion Specification for Energy Metering

Yellow Book - 5th Edition

DLMS/COSEM

Conformance Testing Process

DLMS User Association



DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	1/44
-----------------------	------------	------------------------------	------

Table of Contents

Foreword.....	4
1 Scope	5
2 Referenced documents	5
3 Terms, definitions and abbreviations	5
3.1 Terms and definitions.....	5
3.2 Abbreviations	10
4 Conformance testing – overview	11
4.1 OSI conformance testing	11
4.2 DLMS/COSEM conformance testing	11
4.3 Main features of DLMS/COSEM conformance testing process.....	12
5 The conformance test plans.....	13
5.1 Scope of testing	13
5.2 IUT testing.....	14
5.3 Structure of the abstract test plans	14
5.4 Abstract test cases.....	15
5.5 Outcomes and verdicts	16
5.6 The HDLC based data link layer ATS.....	17
5.7 The DLMS/COSEM application layer ATS	17
5.8 The COSEM interface objects ATS	17
5.9 The Security Suite 0 (SYMSEC_0) ATS.....	18
5.10 Executable test suites and test cases.....	18
6 The DLMS/COSEM conformance test tool	18
6.1 Overview	18
6.2 CTT versions and editions	19
6.3 Operating system and hardware requirements.....	19
6.4 Licensing the CTT	19
6.5 Installing the CTT	19
7 The conformance assessment process	20
7.1 Overview	20
7.2 Preparation for testing.....	21
7.2.1 Preparation of the IUT	21
7.2.2 Preparation of the conformance test information	22
7.3 Test operations	22
7.3.1 The CTT user interface	22
7.3.2 Miscellaneous settings	23
7.3.3 The CTI file	23
7.3.4 The COSEM object definition file	24
7.3.5 Selection of the test cases	26
7.3.6 Connection of the IUT to CTT	27
7.3.7 Test sessions.....	29
7.3.8 Production of the Test Result	29
7.4 Repeatability of results.....	34
7.5 Requirements for test laboratories.....	34
8 The certification process	35

8.1	General	35
8.2	Initiation of the certification process.....	35
8.3	Submission of conformance test documents.....	35
8.4	Technical and administrative checks.....	35
8.5	The Certification	35
8.6	Scope and validity of the Certification.....	36
8.7	Disclaimer	37
9	The quality program	37
9.1	General	37
9.2	Validation of the Abstract Test Suites and CTT.....	37
9.3	Assistance provided to users.....	37
9.4	Maintenance	37
9.5	Use cases	38
9.5.1	Use case 1 – introducing a new standard OBIS code	38
9.5.2	Use case 2 – modification of an existing test	38
9.5.3	Use case 3 – adding a test for a new standard feature	38
9.5.4	Use case 4 – revision of the specification	38
	Annex A Certification template (with sample data) (informative)	39
	Annex B (normative) Conformance Test Plans	41
	Annex C (informative) Bibliography	42

List of Figures

Figure 1 – DLMS/COSEM conformance testing process.....	12
Figure 2 – DLMS/COSEM interface object model and communication profiles.....	13
Figure 3 – Test suite structure.....	15
Figure 4 – Structure of the HDLC based data link layer ATS.....	17
Figure 5 – Structure of the DLMS/COSEM application layer ATS	17
Figure 6 – Structure of the COSEM interface objects ATS.....	18
Figure 7 – Structure of the Security Suite 0 (SYMSEC_0) ATS.....	18
Figure 8 – Conformance assessment process overview	20
Figure 9 – Miscellaneous settings	23
Figure 10 – The CTI window (illustration).....	24
Figure 11 – COSEM object definition file cover sheet.....	25
Figure 12 – Selection of the test cases	27
Figure 13 – Communication settings.....	28
Figure 14 – Fragment of a sample Test Report	31
Figure 15 – Basic log.....	32
Figure 16 – Detailed log presenting COSEM APDUs in XML format.....	33
Figure 17 – The Traffic window	34
Figure 18 – The DLMS/COSEM compliant logo.....	36

List of Tables

Table 1 – Template for test cases	16
---	----

Foreword

Copyright

© Copyright 2001–2015 DLMS User Association.

This document is confidential. It may not be copied, nor handed over to persons outside the standardisation environment.

The copyright is enforced by national and international law. The "Berne Convention for the Protection of Literary and Artistic Works", which is signed by 166 countries worldwide and other treaties apply.

Acknowledgement

The actual document has been established by the DLMS UA Maintenance Working Group Conformance Testing Task Force.

Revision History

Versions kept within the DLMS UA WG Conformance testing.

Version	Date	Author	Status	Comment
Edition 1.0	2001-05-01	DLMS-UA	Released	In line with: - DLMS UA 1000-1 (Blue Book) Edition 4.0; - DLMS UA 1000-2 (Green Book) Edition 2.0.
Edition 2.0	2002-06-04	DLMS-UA	Released	In line with: - DLMS UA 1000-1 (Blue Book) Edition 4.0; - DLMS UA 1000-2 (Green Book) Edition 2.0; - CTT v 1.0
Edition 2.0 Amd. 1	2003-01-09	DLMS UA	Released	Brought in line with CTT v 1.01
Edition 3.0	2007-08-28	DLMS UA	Released	In line with: - DLMS UA 1000-1 (Blue Book) Edition 8.0; - DLMS UA 1000-2 (Green Book) Edition 6.0; - CTT v 2.0.
Edition 4.0	2010-12-15	DLMS UA	Released	In line with: - DLMS UA 1000-1 (Blue Book) Edition 10.0; - DLMS UA 1000-2 (Green Book) Edition 7.0; (except security) - CTT v 2.3.
Edition 5.0	2015-06-19	DLMS UA	Released	In line with: - DLMS UA 1000-1 (Blue Book) Edition 11.0; - DLMS UA 1000-2 (Green Book) Edition 7.0 + Amendment 3 - CTT 3.0

1 Scope

This document specifies the conformance testing process of metering equipment implementing the DLMS/COSEM specification for meter data exchange.

This document only focuses on testing and certifying the implementation of the DLMS/COSEM specification. Other functional and performance tests are outside the scope of this document.

This Edition 5.0 version applies to CTT 3.0, in line with Blue Book Edition 11 and Green Book Edition 7.0, + Amendment 3.

It cancels and replaces Edition 4.0, published in 2010.

2 Referenced documents

Reference No.	Title
DLMS UA 1000-1 Ed. 11.0: 2013	COSEM Interface Classes and OBIS Identification System, the Blue Book
DLMS UA 1000-2 Ed. 7.0:2009 Amendment 3:2013	DLMS/COSEM Architecture and Protocols, the Green Book
DLMS UA 1002 Ed. 1.0:2003	Glossary of terms, the White Book
DLMS UA 1001-3: ATS_DL V 5 Released: 2010-12-15	DLMS/COSEM conformance testing – Conformance test plans – Data link layer using HDLC protocol
DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3 Released: 2015-06-18	DLMS/COSEM conformance testing – Abstract Test Plans – DLMS/COSEM application layer – Symmetric key security suite 0 – COSEM interface objects
DLMS UA 1001-7	COSEM conformance testing – Object definition tables
IEC 62056-21	Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange
ITU-T X.290 (11/1998)	OSI conformance testing methodology and framework for protocol recommendations for IUT-T applications – General concepts

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document the following definitions apply:

NOTE Most of the following definitions have been taken from ITU-T Recommendation ITU-T X.290. Some definitions have been modified to better adapt to the DLMS/COSEM conformance assessment process.

3.1.1

abstract test case

A complete and independent specification of the actions required to achieve a specific test purpose, defined at the level of abstraction of a particular abstract test method. It includes a preamble and a postamble to ensure starting and ending in a stable state (i.e., a state which can be maintained almost indefinitely, such as the “idle” state or “data transfer” state) and involves one or more consecutive or concurrent connections.

NOTE 1 – The specification should be complete in the sense that it is sufficient to enable a verdict to be assigned unambiguously to each potentially observable outcome (i.e., sequence of test events).

NOTE 2 – The specification should be independent in the sense that it should be possible to execute the derived executable test case in isolation from other such test cases (i.e., the specification should always include the possibility of starting and finishing in the “idle” state – that is without any existing connections except permanent ones). For some test cases, there may be pre-requisites in the sense that execution might require some specific capabilities of the IUT, which should have been confirmed by results of the test cases executed earlier.

[ITU-T X.290 3.6.3]

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	5/44
-----------------------	------------	------------------------------	------

3.1.2**abstract test suite (ATS)**

A test suite composed of abstract test cases. [ITU-T X.290 3.6.16]

3.1.3**basic interconnection test (BIT)**

Limited testing of an IUT to determine whether or not there is sufficient conformance to the main features of the relevant protocol(s) for interconnection to be possible, without trying to perform thorough testing. [ITU-T X.290 3.5.5]

3.1.4**behaviour testing**

Testing the extent to which the dynamic conformance requirements are met by the IUT. [ITU-T X.290 3.5.8]

3.1.5**capabilities of an IUT**

That set of functions and options in the relevant protocol(s) and, if appropriate, that set of facilities and options of the relevant service definition which are supported by the IUT. [ITU-T X.290 3.4.5]

3.1.6**capability testing**

Testing to determine the capabilities of an IUT.

NOTE This involves checking all mandatory capabilities and those optional ones that are stated in the CTI as being supported, but not checking those optional ones which are stated in the CTI as not supported by the IUT.

[ITU-T X.290 3.5.6, modified]

3.1.7**conformance assessment process**

The complete process of accomplishing all conformance testing activities necessary to enable the conformance of an implementation to one or more OSI* Recommendations* to be assessed. It includes the production of the CTI documents, preparation of the real tester and the IUT, the execution of one or more test suites, the analysis of the results and the production of the appropriate protocol conformance test reports. [ITU-T X.290 3.5.10, modified]

3.1.8**conformance log**

A record of sufficient information necessary to verify verdict assignments as a result of conformance testing. [ITU-T X.290 3.7.15]

3.1.9**conformance test information (CTI)**

A statement made by the supplier or implementor of an IUT stating the capabilities and options that have been implemented and additional information necessary to select and parameterize the executable test cases.

NOTE 1 Part of this information on the implementation may be taken from the IUT itself.

NOTE 2 X.290 uses the terms Protocol Implementation Conformance Statement (PICS) and Protocol Implementation Extra Information for Testing (PIXIT).

3.1.10**conformance testing**

Testing the extent to which an IUT is a conforming implementation. [ITU-T X.290 3.5.9]

3.1.11**conforming implementation**

An IUT, which is shown to satisfy conformance requirements, consistent with the capabilities stated in the CTI.

NOTE In case of DLMS/COSEM, the capabilities are partly declared in the CTI and they are partly provided by the IUT.

[ITU-T X.290 3.4.10, modified]

3.1.12

executable test case

A realization of an abstract test case. [ITU-T X.290 3.6.4]

3.1.13

executable test case error

A test case error in the realization of an abstract test case.

3.1.14

executable test suite (ETS)

A test suite composed of executable test cases. [ITU-T X.290 3.6.17]

3.1.15

“fail” verdict

A verdict given when the observed outcome is syntactically invalid or inopportune with respect to the relevant Recommendation(s)* or the CTI. [ITU-T X.290 3.7.13, modified]

3.1.16

foreseen outcome

An outcome identified or categorized in the abstract test case specification. [ITU-T X.290 3.7.4]

3.1.17

idle testing state

A stable testing state in which there is no established connection of the relevant protocol(s) and in which the state of the IUT is independent of any previously executed test cases.

3.1.18

implementation under test (IUT)

That part of a real open system which is to be studied by testing, which should be an implementation of one or more OSI* protocols in an adjacent user/provider relationship.

NOTE In DLMS/COSEM IUT are DLMS/COSEM servers.

[ITU-T X.290 3.4.1]

3.1.19

“inapplicable” test

A test case, which cannot be performed because the necessary conditions are not available.

3.1.20

“inconclusive” verdict

A verdict given when the observed outcome is valid with respect to the relevant Recommendation(s)* but prevents the test purpose from being accomplished. [ITU-T X.290 3.7.14]

3.1.21

initial testing state

The testing state in which a test body starts.

NOTE This may be either a stable testing state or a transient state.

3.1.22

inopportune test event

A test event which, although syntactically correct, occurs or arrives at a point in an observed outcome when not allowed to do so by the protocol Recommendation*. [ITU-T X.290 3.7.11]

3.1.23**means of testing (MOT) (IUTs)**

The combination of equipment and procedures that can perform the derivation, selection, parameterization and execution of test cases, in conformance with a reference standardized ATS, and can produce a conformance log.

3.1.24**negative test**

Test to verify the correct response of the IUT on:

- DLMS/COSEM conformant information and services, which are not implemented;
- non conformant communication traffic

3.1.25**outcome**

A sequence of test events together with the associated input/output, either identified by an abstract test case specifier, or observed during test execution. [ITU-T X.290 3.7.3]

3.1.26**parameterized executable test case**

An executable test case, in which all appropriate parameters have been supplied with values in accordance with a specific CTI [ITU-T X.290 3.6.23, modified]

3.1.27**“pass” verdict**

A verdict given when the observed outcome satisfies the test purpose and is valid with respect to the relevant Recommendation(s)* and with respect to the CTI. [ITU-T X.290 3.7.12, modified]

3.1.28**positive test**

test to ensure the correct implementation of the capabilities of the IUT as defined by the supplier. A positive test has a described and defined response

3.1.29**postamble**

The test steps needed to define the paths from the end of the test body up to the finishing stable state for the test case. [X.290 3.6.9]

3.1.30**preamble**

The test steps needed to define the path from the starting stable state of the test case up to the initial state from which the test body will start. [ITU-T X.290 3.6.7]

3.1.31**protocol conformance test report (PCTR)**

A document written at the end of the conformance assessment process, giving the details of the testing carried out for a particular protocol, including the identification of the abstract test cases for which corresponding executable test cases were run and for each test case the test purpose and verdict. [ITU-T X.290 3.7.8]

3.1.32**repeatability (of results)**

Characteristic of a test case, such that repeated executions on the same IUT lead to the same verdict, and by extension a characteristic of a test suite. [ITU-T X.290 3.7.1]

3.1.33**semantically invalid test event**

A test event which is neither inopportune nor syntactically invalid, but which contains a semantic error with respect to the relevant protocol specification (e.g. a PDU containing a parameter value outside the negotiated range for that parameter).

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	8/44
-----------------------	------------	------------------------------	------

3.1.34**stable testing state**

A testing state which can be maintained, without prescribed Lower Tester behaviour, sufficiently long to span the gap between one test case and the next in a test session.

3.1.35**static conformance review**

A review of the extent to which the static conformance requirements are met by the IUT, by comparing the static conformance requirements expressed in the relevant Recommendation(s)* with the PICS and the results of any associated capability testing. [ITU-T X.290 3.5.7 modified]

3.1.36**syntactically invalid test event**

A test event which syntactically is not allowed by the protocol Recommendation*. [ITU-T X.290 3.7.10]

3.1.37**test body**

The set of test steps that are essential in order to achieve the test purpose and assign verdicts to the possible outcomes. [ITU-T X.290 3.6.8]

3.1.38**test case**

A generic, abstract or executable test case. [ITU-T X.290]

3.1.39**test case error**

The term used to describe the result of execution of a test case when an error is detected in the test case itself.

3.1.40**test event**

An indivisible unit of test specification at the level of abstraction of the specification (e.g. sending or receiving a single PDU). [ITU-T X.290 3.6.11]

3.1.41**test group**

A named set of related test cases. [ITU-T X.290 3.6.14]

3.1.42**test group objective**

A description of the common objective which the test purposes within a specific test group are designed to achieve.

3.1.43**test laboratory**

An organization that carries out conformance testing. This can be a third party, a user organization, an Administration*, or an identifiable part of the supplier organization. [ITU-T X.290 3.4.13]

3.1.44**test purpose**

A description of the objective which an abstract test case is designed to achieve. [ITU-T X.290 3.6.5]

3.1.45**test step (sub-test)**

A named subdivision of a test case, constructed from test events and/or other test steps, and used to modularize abstract test cases. [ITU-T X.290 3.6.10, modified]

3.1.46**test session**

The process of executing the Parameterized Executable Test Suite for a particular IUT and producing the conformance log.

3.1.47**test suite**

A complete set of test cases, possibly combined into nested test groups, that is necessary to perform conformance testing or basic interconnection testing for an IUT or protocol within an IUT [ITU-T X.290 3.6.12]

3.1.48**unforeseen test outcome**

An outcome not identified or categorized in the abstract test case specification. [ITU-T X.290 3.7.5]

3.1.49**valid test event**

A test event which is allowed by the protocol Recommendation*, being both syntactically correct and occurring or arriving in an allowed context in an observed outcome. [ITU-T X.290 3.7.9]

3.1.50**verdict**

Statement of “pass”, “fail” or “inconclusive” concerning conformance of an IUT with respect to a test case that has been executed and which is specified in the abstract test suite. [ITU-T X.290 3.7.6]

3.2 Abbreviations

Abbreviation	Explanation
AA	Application Association
APDU	Application Protocol Data Unit
ATS	Abstract Test Suite
COSEM	Companion Specification for Energy Metering
COSEM object	An instance of an interface class
CO	Connection oriented
CTI	Conformance Test Information
CTT	Conformance Test Tool
DLMS	Device Language Message Specification
ETS	Executable Test Suite
HDLC	High-level Data Link Control
IEC	International Electrotechnical Commission
IP	Internet Protocol
ITU	International Telecommunication Union
IUT	Implementation Under Test
LD	Logical Device
MOT	Means of Testing
OBIS	OBject Identification System
OSI	Open System Interconnection
PDU	Protocol Data Unit
SAP	Service Access Point
TCP	Transmission Control Protocol

4 Conformance testing – overview

4.1 OSI conformance testing

The concept and methodology of OSI conformance testing is described in the Recommendation ITU-T X.290.

The objective of conformance testing is to establish whether the Implementation Under Test (IUT) conforms to the relevant specification(s).

Practical limitations make it impossible to be exhaustive, and economic considerations may restrict testing still further.

The primary purpose of conformance testing is to increase the probability that different implementations are able to interwork. While conformance is a necessary condition, it is not on its own a sufficient condition to guarantee interworking capability. Even if two implementations conform to the same protocol specification, they may fail to interwork fully.

What conformance testing does do is give confidence that an implementation has the required capabilities and that its behaviour conforms consistently in representative instances of communication.

4.2 DLMS/COSEM conformance testing

The DLMS/COSEM specification, as a global standard for data exchange with utility metering equipment, includes standardized conformance tests to ensure that IUTs comply with applicable requirements. It is based on the principles developed for OSI conformance testing.

The main elements of the DLMS/COSEM specification are the following:

- the Blue Book, specifying COSEM interface object model, see DLMS UA 1000-1;
- the Green Book, specifying communication profiles, see DLMS UA 1000-2; and

NOTE The contents of the Blue Book and the Green Book are internationally standardized, see the Bibliography.

The DLMS/COSEM conformance testing process comprises the following, see Figure 1:

- the Yellow book – this document – specifying the conformance testing process;
- the conformance test plans, i.e. the Abstract Test Suites (ATSs);
- the Conformance Test Tool (CTT);
- the conformance assessment process;
- the certification process;
- the quality program.

The conformance test plans i.e. the Abstract Test Suites (ATS) describe, at the level of abstraction, the tests to be performed. See Clause 5.

The DLMS/COSEM Conformance Test Tool (CTT) implements the ATSs in the form of Executable Test Suites. See Clause 6.

The conformance assessment process consists of the phases of preparation for testing, test operations and the production of the conformance Test Result. See Clause 7.

The certification process consists of examining the conformance Test Results and the publication of the Certifications. See Clause 8.

The quality program includes handling comments and questions and, when necessary, initiating the maintenance of the DLMS/COSEM specification, the ATSs and/or the CTT. See Clause 9.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	11/44
-----------------------	------------	------------------------------	-------

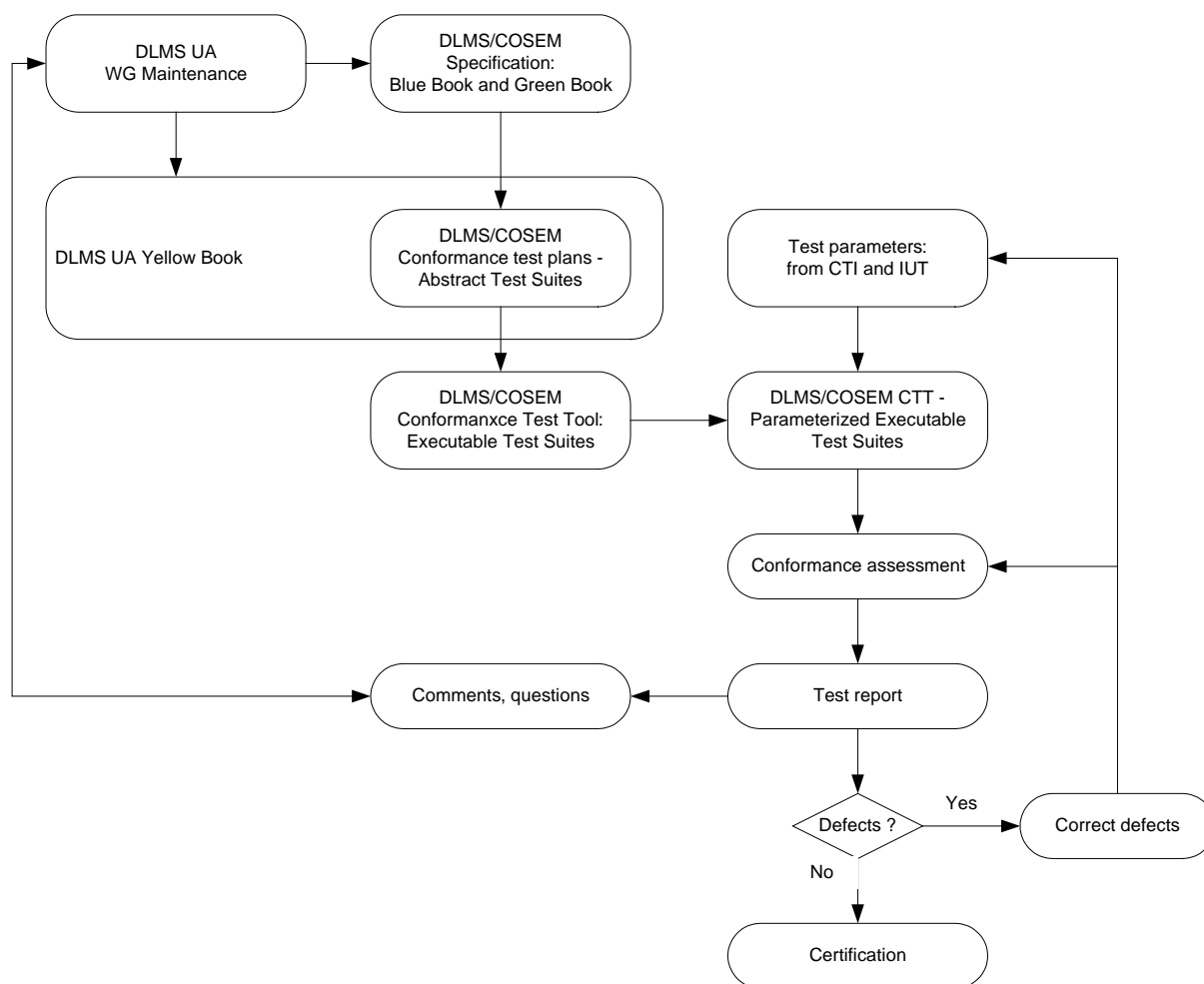


Figure 1 – DLMS/COSEM conformance testing process

4.3 Main features of DLMS/COSEM conformance testing process

The main features of the DLMS/COSEM conformance testing process are summarized below:

- it covers DLMS/COSEM servers implementing the COSEM interface object model and the DLMS/COSEM application layer, including the security suites. Conformance testing is limited to the server's functionality as presented at the communication interface(s). Other functions of the server are out of Scope;
- testing can be performed using either the 3-layer, CO, HDLC based profile or using the TCP/IP based profile. When the 3-layer, CO, HDLC based profile is used with direct HDLC connection, the implementation of the HDLC layer is also tested;
- the CTT can be used for self-testing or third party testing;
- to obtain a Certification, the manufacturer of the IUT shall possess a registered three-letter manufacturer ID; see <http://dlms.com/organization/flagmanufacturesids/index.html>;
- the CTT automatically generates the Test Result necessary for the Certification, see 7.3.8;
- the Certification is issued by the DLMS UA to the manufacturer, see Clause 8;
- the DLMS UA operates a Quality program to maintain the test plans and the CTT, see Clause 9.

5 The conformance test plans

5.1 Scope of testing

The communication model of DLMS/COSEM servers to be tested is shown in Figure 2.

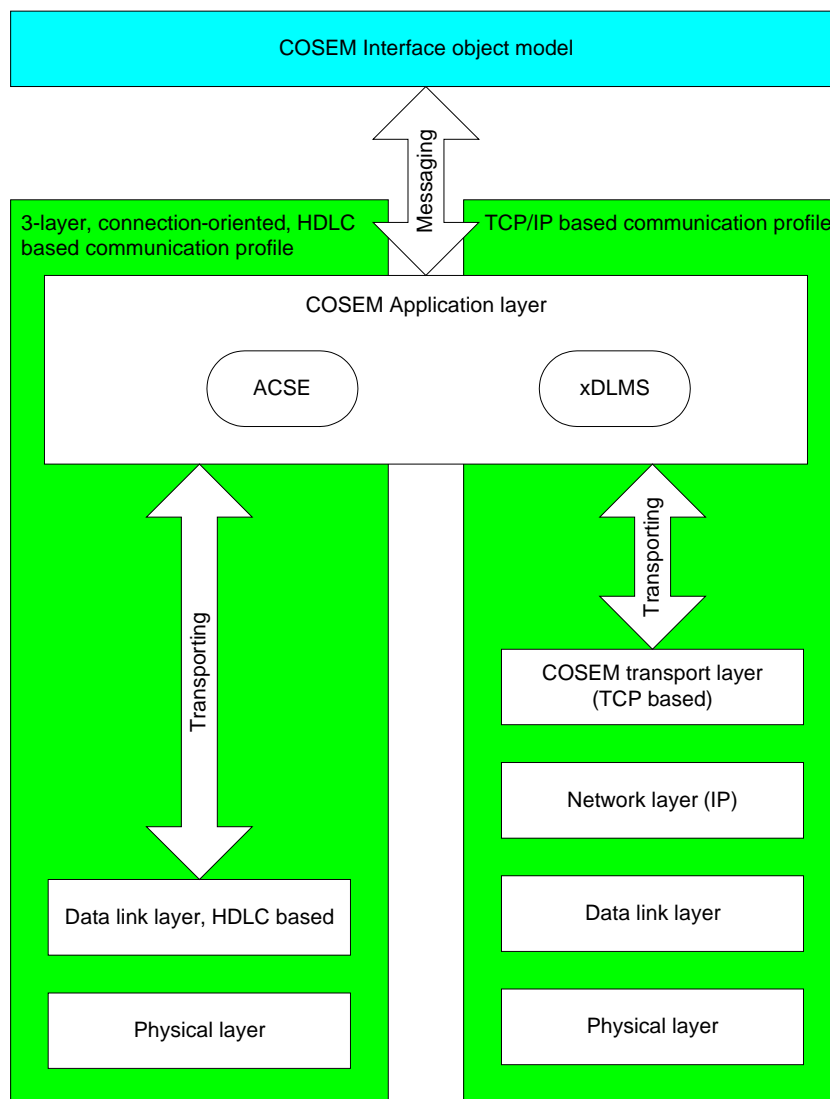


Figure 2 – DLMS/COSEM interface object model and communication profiles

The COSEM Interface object model, specified in DLMS UA 1000-1 and the DLMS/COSEM Application layer specified in DLMS UA 1000-2 are used in all IUTs.

The selection of the lower layers depends on the communication profile:

- in the 3-layer, connection-oriented, HDLC based communication profile, the DLMS/COSEM Application layer is supported by the data link layer using HDLC protocol, specified in DLMS UA 1000-2 Ed. 7.0:2009 Clause 8 and this is supported by the physical layer specified in DLMS UA 1000-2 Ed. 7.0:2009 Clause 5;
- in the TCP/IP based communication profile the DLMS/COSEM Application layer is supported by the DLMS/COSEM transport layer specified in DLMS UA 1000-2 Ed. 7.0:2009 Clause 7, and this is supported by a set of lower layers appropriate for the communication media.

See also 7.3.5.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	13/44
-----------------------	------------	------------------------------	-------

The conformance test plans cover:

- the data link layer using the HDLC protocol;
- the DLMS/COSEM Application layer;
- the COSEM Interface objects; and
- the Security Suite 0.

The TCP and IP layers, when used, are implicitly tested.

5.2 IUT testing

A single IUT is tested against a single test source.

For the purposes of testing, the IUT is considered as a black box. The test session consists of sending messages by the CTT to the IUT and observing the responses.

As access to protocol layer boundaries is not available, the interface object model and the protocol stack are tested in combination. Therefore, the following assumptions are made:

- for testing the data link layer using the HDLC protocol, it is assumed that the physical layer works correctly;
- for testing the DLMS/COSEM Application layer, it is assumed that the supporting layers work correctly;
- for testing the COSEM Interface object model, it is assumed that the protocol stack works correctly.

5.3 Structure of the abstract test plans

The abstract test plan comprises abstract test suites (ATs).

NOTE The remaining part of 5.3 and 5.4 applies strictly to the protocol layer test plans only.

The abstract test suites have a hierarchical structure (see Figure 3) in which an important level is the test case.

Each test case has a specified test purpose, such as verifying that the IUT has a certain required capability (e.g. the ability to support certain packet sizes) or exhibit a certain required behaviour (e.g. behave as required when a particular event occurs in a particular state).

Within a test suite, nested test groups are used to provide a logical ordering of the test cases.

Associated with each test group is a test group objective.

Test cases may be modularised by using named subdivisions called subtests. Test events are indivisible units of specification within a test step (e.g. the transfer of a single PDU to or from the IUT).

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	14/44
-----------------------	------------	------------------------------	-------

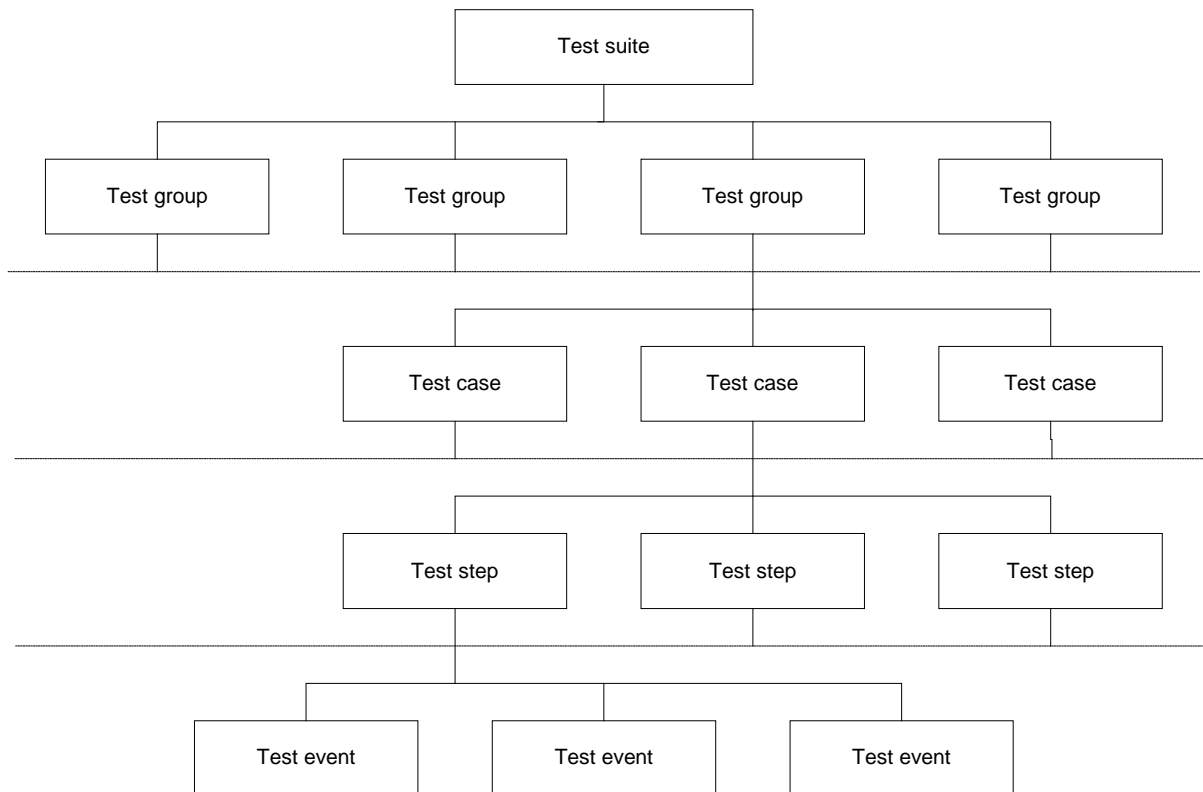


Figure 3 – Test suite structure

Test suites include test cases falling in the following categories (the list is not exhaustive):

- capability tests;
- tests of valid behaviour (positive tests);
- tests of syntactically invalid or inopportune behaviour (negative tests);
- tests related to each protocol state;
- PDU encoding variations;
- variations in values of individual parameters and/or combination of parameters.

5.4 Abstract test cases

An abstract test case is derived from a test purpose and from the relevant specifications. An abstract test case:

- has a *Test case* name, used as a reference and relating the test case to the test group and the test suite;
- gives the *References* pointing to the relevant clauses of the Blue Book DLMS UA 1000-1 and the Green Book DLMS UA 1000-2 constituting the base specification the test case is related to and derived from;
- describes the *Test purpose*;
- specifies the *Prerequisites*;
- specifies the *Expected result*, i.e. the expected behaviour of the IUT;
- specifies the test steps needed to define the path from the starting stable state of the test case up to the initial state from which the test body will start; this test sequence comprises the *Preamble*;

- specifies the sequences of foreseen test events necessary in order to achieve the test purpose. These sequences comprise the *Test body*. It may consist of one or more subtests;
- specifies the verdict to be assigned to each foreseen test outcome.
- specifies test steps needed to define the paths from the end of the test body up to the finishing stable state for the test case; this test sequence comprises the *Postamble*;

The abstract test cases are formatted using the template shown in Table 1.

Table 1 – Template for test cases

Test case	
References	
Test purpose	
Prerequisites	
Expected result	
Preamble	
Test body	Subtest 1
	...
	Subtest n:
Postamble	
Comments	

5.5 Outcomes and verdicts

The outcome is the sequence of test events of events observed during test execution.

A foreseen test outcome is one, which has been identified in the abstract test case i.e. the events which occurred during test execution matched a sequence of test events defined in the abstract test case. A foreseen test outcome always results in the assignment of a test verdict to the test case.

The test verdict will be PASSED, FAILED, INAPPLICABLE or INCONCLUSIVE:

- **PASSED** – Means that the observed test outcome gives evidence of conformance to the conformance requirement(s) on which the test purpose of the test case is focused, and is valid with respect to the relevant specification(s);
- **FAILED** – Means that the observed test outcome either demonstrates non-conformance with respect to (at least one of) the conformance requirement(s) on which the test purpose of the test case is focused, or contains at least one invalid test event, with respect to the relevant specification(s);
- **INAPPLICABLE** – Means that the test case cannot be run with the given CTI declarations and with the information taken from the IUT;
- **INCONCLUSIVE** – Means that the observed test outcome is such that neither a pass nor a fail verdict can be given.

An unforeseen test outcome is one, which has not been identified by the abstract test case, i.e. the events, which occurred during execution of the test case did not match any sequence of test events defined in the abstract test case. An unforeseen test outcome always results in the recording of a test case error or an abnormal test case termination for the test case.

A test case error is recorded if an error is detected either in the abstract test case itself, (i.e. an abstract test case error) or in its realization, (i.e. an executable test case error).

An abnormal test case termination is recorded if the execution of the test case is prematurely terminated by the test system for reasons other than test case error.

The results of executing the relevant individual test cases will be recorded in the conformance Test Results.

5.6 The HDLC based data link layer ATS

The HDLC based data link layer ATS is specified in DLMS UA 1001-3: ATS_DL V 5. Its structure is shown on Figure 4.

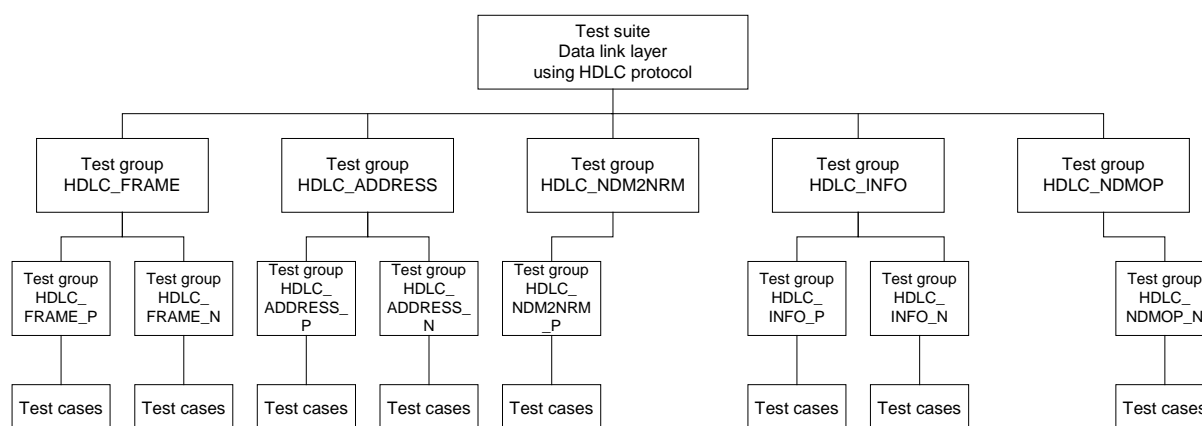


Figure 4 – Structure of the HDLC based data link layer ATS

5.7 The DLMS/COSEM application layer ATS

The DLMS/COSEM application layer ATS is specified in DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3. Its structure is shown on Figure 5.

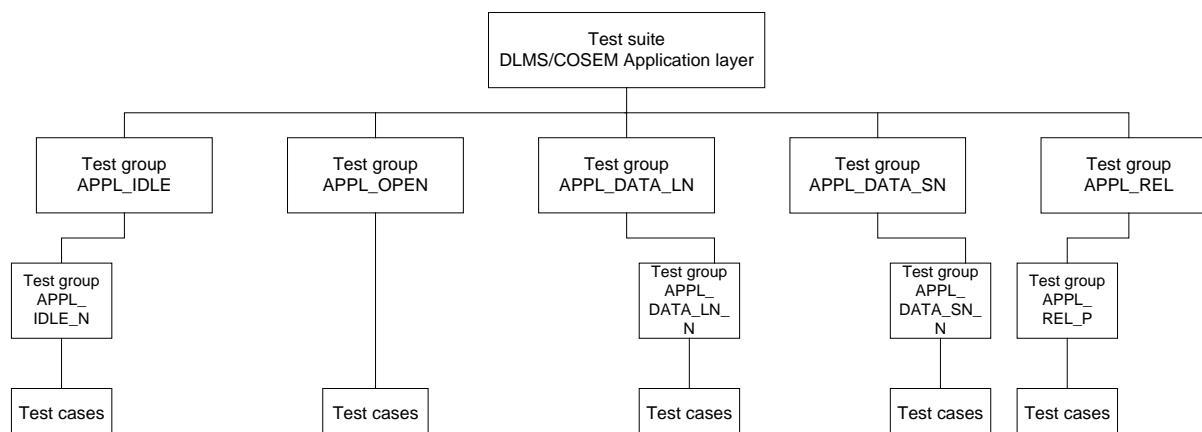


Figure 5 – Structure of the DLMS/COSEM application layer ATS

5.8 The COSEM interface objects ATS

The COSEM interface objects ATS is specified in DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3. Its structure is shown on Figure 6.

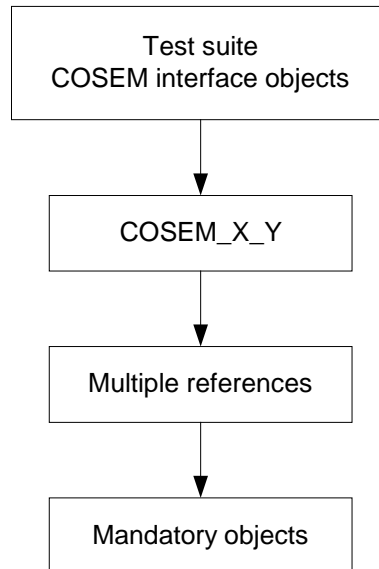


Figure 6 – Structure of the COSEM interface objects ATS

5.9 The Security Suite 0 (SYMSEC_0) ATS

The Security Suite 0 (SYMSEC_0) ATS is specified in DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3. Its structure is shown on Figure 5.

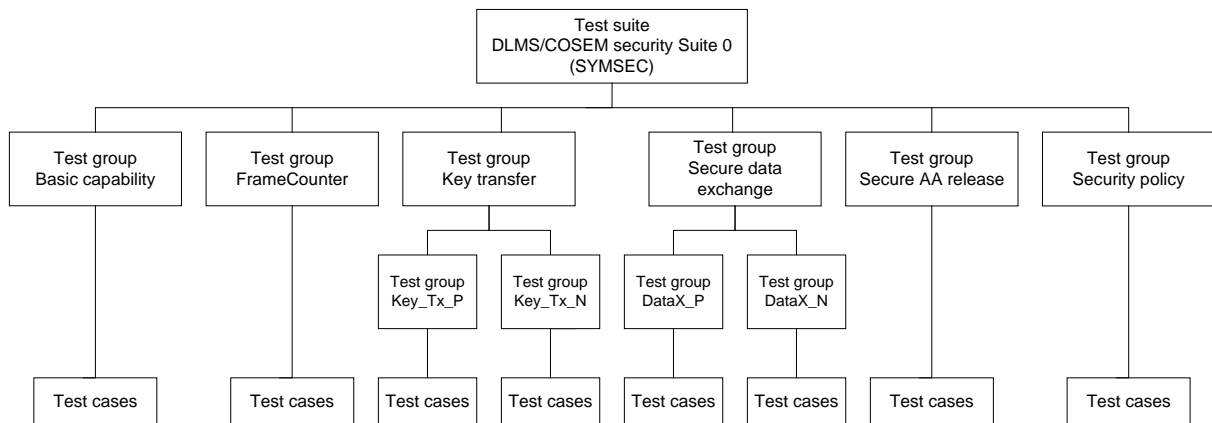


Figure 7 – Structure of the Security Suite 0 (SYMSEC_0) ATS

5.10 Executable test suites and test cases

The executable test suites and test cases are derived from the relevant ATSs. There is one ETS for each ATS and one executable test case for each abstract test case.

In each executable test case, the sequences of test events and the verdict assignments are the same as in the corresponding ATS. The depth of analysing the PDUs is as specified in the ATS.

6 The DLMS/COSEM conformance test tool

6.1 Overview

The DLMS/COSEM conformance test tool (CTT) is a Means Of Testing (MOT) as defined in 3.1.23, i.e. an implementation of the abstract test suites (ATS) in the form of executable test suites (ETS).

It is a computer program running on a PC under 64 bit Windows (version 7 and 8.1) acting as a test DLMS/COSEM client whereas the Implementation Under Test (IUT) acts as a DLMS/COSEM server, see 6.3. The CTT is available in two editions:

- the standard edition produces the Test Result files (Report, Log and Line Traffic) necessary for Certification;
- the extended edition provides a detailed log by decoding the messages exchanged between the CTT and the IUT and thereby it facilitates interpreting the log.

The CTT can perform the following:

- the selection of test options, see 7.3.2;
- the parametrization of the test cases taking information from a Conformance Test Information (CTI) file and the IUT itself, see 7.3.3 and 7.3.4;
- the selection of test cases to be performed, see 7.3.5;
- running test sessions, i.e. automatically executing the test cases selected; see 7.3.7;
- the generation of the Test Result, see 7.3.8.

6.2 CTT versions and editions

CTT 2.X is suitable for testing IUTs implementing Blue Book Ed. 10.0 and Green Book Ed. 6.0 (except the authentication mechanism using HLS).

CTT 3.0 is suitable for testing IUTs implementing Blue Book Ed 11.0 and Green Book Ed. 7.0 + Amendment 3. It is available now and replaces CTT 2.X.

CTT 3.1 BB12_GB8 (X) – under development – will be suitable for testing IUTs implementing Blue Book Ed. 12.0 and Green Book Ed. 8.0.

A new version of the CTT obsoletes all earlier version of the CTT after a transition period.

Earlier versions of the CTT can be used for re-testing earlier implementations.

The CTT standard edition allows performing all tests and produces the Report, Log and Line Traffic files.

The CTT Extended edition produces a more detailed log file and it allows logging and viewing:

- in the case of the 3-layer, CO, HDLC based profile the HDLC frames in a decoded form;
- in the case of the TCP/IP profile the wrapper frames in a decoded form;
- the COSEM APDUs in XML format.

6.3 Operating system and hardware requirements

The CTT runs on a host computer under 64 bit versions of Windows 7 and Windows 8.1.

6.4 Licensing the CTT

The CTT can be licensed to any member of the DLMS UA.

The “Rules for availability and use” are published at the homepage of the DLMS UA, at www.dlms.com under the “Conformance” menu.

6.5 Installing the CTT

The CTT can be downloaded from www.eurodcs.com and installed following the process described by EuroDCS.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	19/44
-----------------------	------------	------------------------------	-------

7 The conformance assessment process

7.1 Overview

The conformance assessment process is the complete process of accomplishing all conformance testing activities necessary to enable the conformance of the IUT to be assessed.

The test can be performed by the manufacturer or by another (third) party:

- when the test is performed by the manufacturer itself, the CTT licensee and the manufacturer are the same: this is known as self-testing;
- when the test is performed by another (third) party, the CTT licensee and the manufacturer are different: this is known as third party testing. Any CTT licensee may act as a third party test laboratory.

An overview of the conformance assessment process is given in Figure 8.

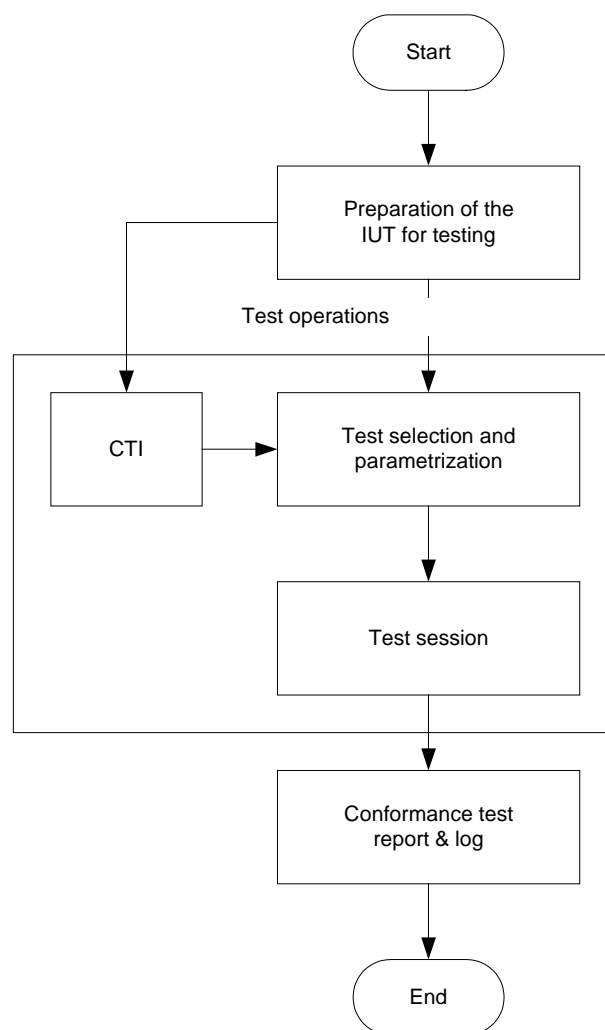


Figure 8 – Conformance assessment process overview

The preparation for testing phase involves:

- the preparation of the IUT, see 7.2.1;
- the preparation of Conformance Test Information (CTI) file, see 7.2.2;

The test operations include:

- selection and parameterization of the test cases, see 7.3.3, 7.3.4 and 7.3.5;
- connection of the IUT to the CTT, see 7.3.6;
- running the test, see 7.3.7;
- production of the Test Result, see 7.3.8.

In the following, the elements of the conformance test process are described and the use of the CTT is explained.

7.2 Preparation for testing

7.2.1 Preparation of the IUT

The configuration of the IUT for the test is the responsibility of the manufacturer. To facilitate system integration, it is advisable that the test is performed on a configuration that is representative for the intended application(s) so that all required features for which DLMS/COSEM compliance is claimed are tested.

The following provides a guideline:

- if the IUT supports more than one logical device, then at least two logical devices should be configured;
- if the IUT supports more than one application context, authentication security mechanism and xDLMS context then the set of AAs declared shall cover each context and mechanism declared. These AAs may be in the same logical device or spread across the logical devices;
- an AA between the same client and the server may be declared several times with different application contexts, authentication mechanisms, DLMS contexts and security contexts as needed;
- if the IUT is a complete – fully integrated or modular – meter, then the mandatory Management Logical Device (Server SAP = 0x01) shall be present and it shall support an AA with the public client (Client SAP = 0x10, Server SAP = 0x01, no ciphering, no authentication security);
- if the IUT is a communication module, then the Management Logical Device (Server SAP = 0x01) does not have to be present: it is assumed that it is present in the base meter. However a Logical Device of the communication module shall support an AA with the public client, see above;
- the set of xDLMS services and capabilities (i.e. the conformance block) should be representative for the intended application;
- the set of security features should be representative for the intended application;
- the set of interface objects available should be representative for the intended application;
- the AAs shall provide access to the objects and attributes to be tested, with appropriate access rights;
- if load profiles with selective access are to be tested, then a sufficient amount of data should be present. The conditions are specified in the COSEM objects test plan;

it is the responsibility of the manufacturer to restrict access rights to attributes, so that the CTT cannot unduly modify them. This can be done by providing interface class and/or instance related extra information in the CTI.

For testing IUTs supporting Security Suite 0, additional requirements are specified in the Security Suite 0 (SYMSEC_0) ATS.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	21/44
-----------------------	------------	------------------------------	-------

For testing IUTs supporting push operation, additional requirements are specified in the COSEM objects ATS.

See also 8.6, Scope and validity of the Certification.

7.2.2 Preparation of the conformance test information

For the parametrization of the executable test cases, the following information is necessary:

- information on the manufacturer;
- information on the IUT:
 - logical devices;
 - application associations;
 - authentication security mechanisms;
 - xDLMS context;
 - security context and security material;
 - media / energy types supported;
 - COSEM interface objects.

Part of this information is obtained by CTT from the IUT itself during the test session through the negotiation of the application context, the authentication mechanism and the xDLMS context, as well as by reading the **object_list attribute** of the “Association SN /LN” objects and, in the case of SN referencing, the **access_rights_list** attribute of “Association SN” object (supported from version 1 of that IC).

Another part of this information has to be declared in the CTI.

The CTI file identifies the manufacturer and the IUT and contains specific information necessary for testing. It can be prepared using the CTI template provided by the CTT. A Help, explaining the syntax and the contents of the CTI is provided.

The CTI template is shown in DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3, Annex A.

7.3 Test operations

7.3.1 The CTT user interface

The CTT Main Menu comprises five items:

- the File menu allows saving and loading test results and to exit the CTT application;
- the Run menu allows to run a test session and to abort it, see 7.3.7;
- the View menu allows opening the Test Plans i.e. the Abstract Test Suites and the Line Traffic window;
- the Settings menu allows setting the communication parameters and selecting some other parameters and choices (Miscellaneous), see 7.3.6 and 7.3.2.
- the Help menu provides information on the CTT and its use.

The CTT also provides a number of panes:

- the REPORT pane displays the Test Report;
- the LOG pane displays a listing of the actions executed by the CTT;
- the CTI pane displays the CTI file;
- the TEST CASES pane allows selecting the test cases;

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	22/44
-----------------------	------------	------------------------------	-------

- the Traffic window – to be opened from the View menu – shows the messages sent (shown in green) and received (shown in red) by the CTT. Time stamps are provided so that the Line Traffic can be correlated with the Log. In the case of the 3-layer, CO, HDLC profile it shows the HDLC frames. In the case of the TCP/IP profile, it shows the TCP streams.

Right-clicking in the REPORT, LOG, CTI panes and in the Traffic window opens a contextual menu.

7.3.2 Miscellaneous settings

The **Miscellaneous** tab in the **Settings** menu – see Figure 9 – offers the following possibilities:

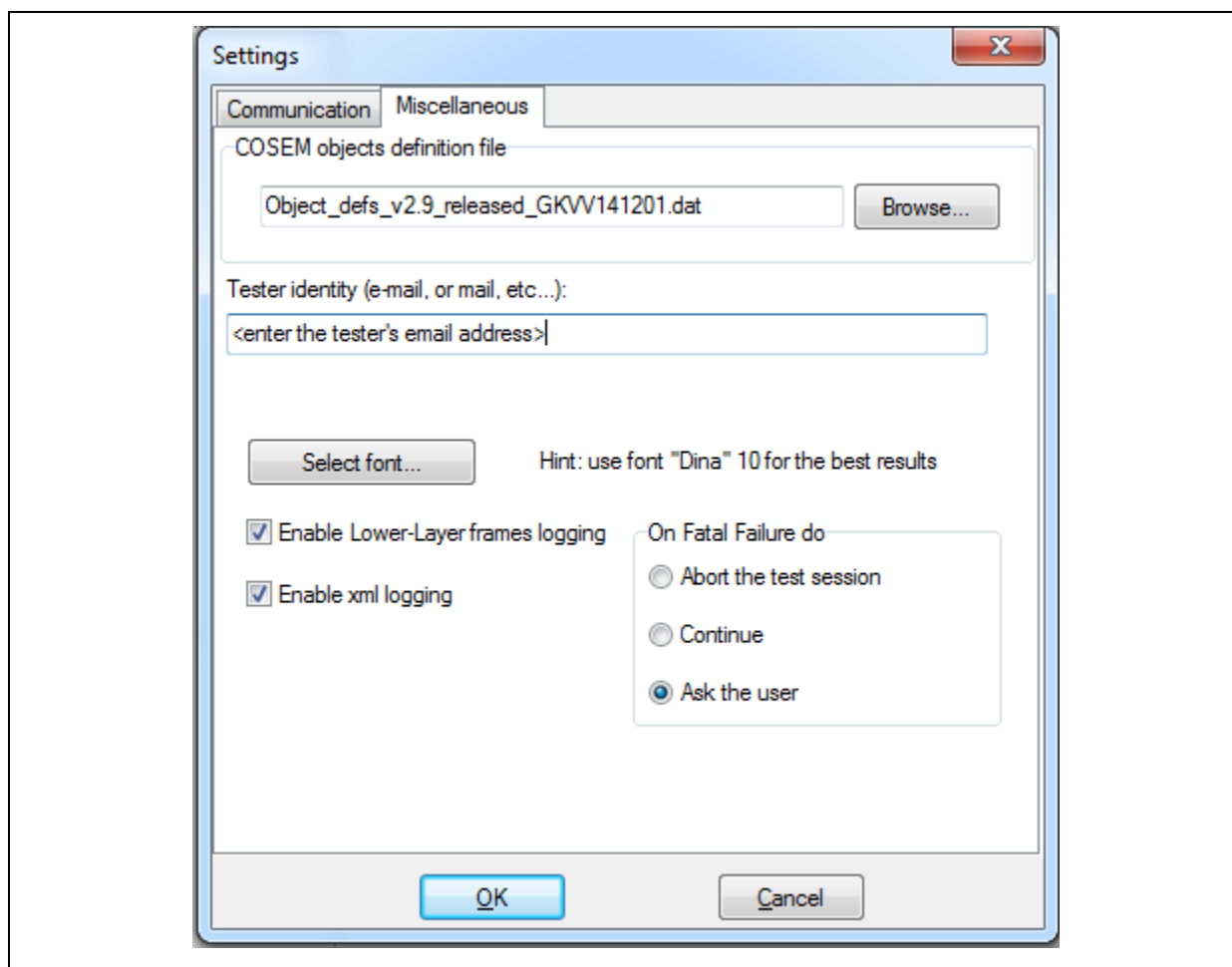


Figure 9 – Miscellaneous settings

- the COSEM object definition .dat file – see 7.3.4 – to be used by the CTT during the COSEM object tests shall be selected;

NOTE The selections cannot be OK-d if this field is empty.

- the tester identity e.g. an email address. This will appear in the Test Report;
- the font used in the Test Result documents can be chosen;
- logging of lower layer frames and COSEM APDUs (in XML format) can be enabled. This feature is available in the extended edition, see 6.2;
- the options on the occurrence of a fatal failure can be chosen, see 7.3.7.

7.3.3 The CTI file

The CTI file can be edited, saved and loaded from the CTI pane.

Figure 10 shows the CTI template loaded, with the contextual menu open, that allows loading and saving CTI files and to edit the file.

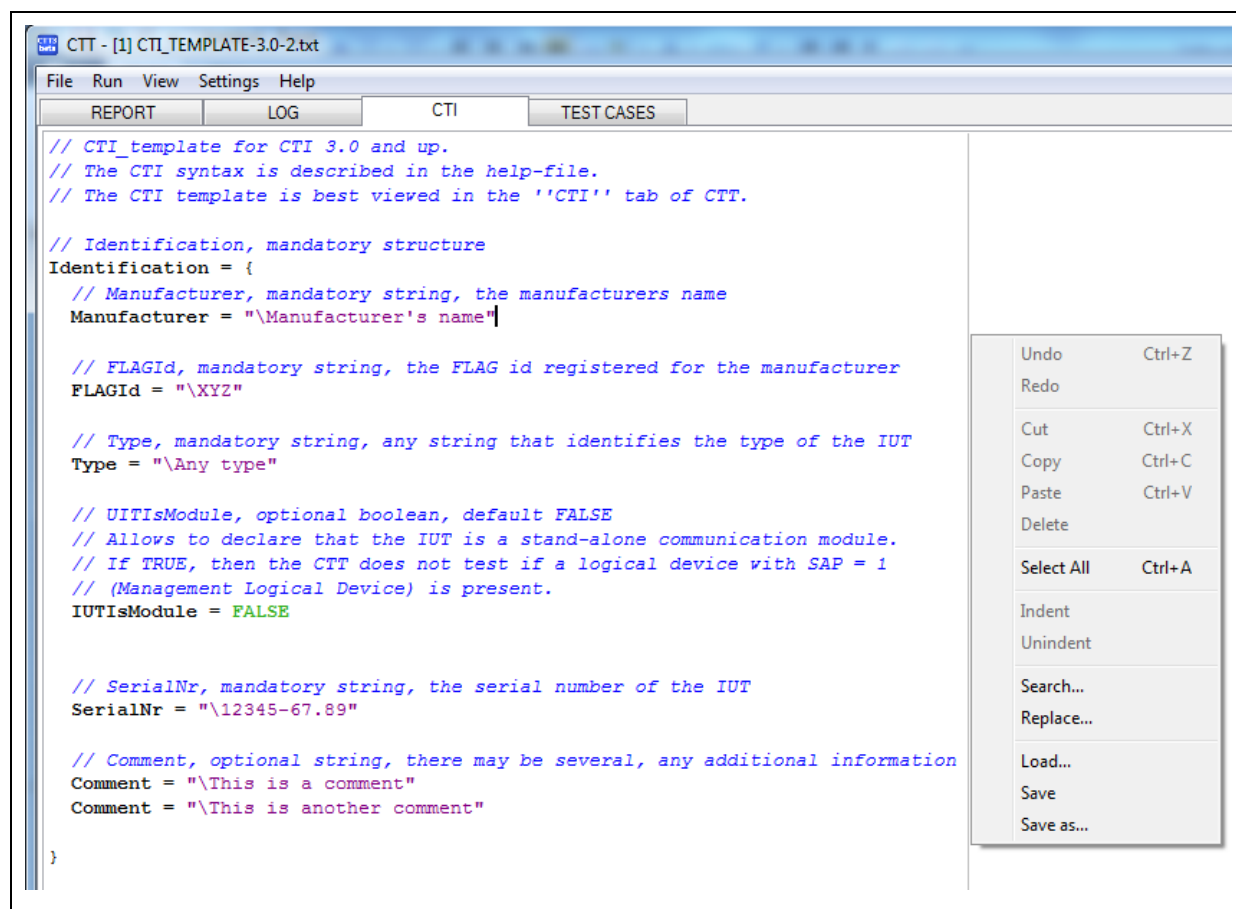


Figure 10 – The CTI window (illustration)

The CTI file content and file syntax is described in the Help Menu.

The CTI allows declaring some DoNotTest options:

- ATTRIBUTES_TYPE_CHOICES: when this element is present, then CTT does not check the type of COSEM object attributes specified as a CHOICE type;
- ATTRIBUTES_VALUES: when this element is present, then CTT does not check the values (ranges, and sub-ranges) of COSEM object attributes.

If the purpose of the test session is to obtain a Certification, then DoNotTest has to be omitted or empty.

7.3.4 The COSEM object definition file

The COSEM object definition file DLMS UA 1001-7 contains all information for testing the COSEM objects including:

- the valid OBIS codes that – together with the COSEM interface class – provide the semantical meaning of all data. In some cases, an OBIS code may be used with alternative interface classes;
- the valid data type of the attributes in the cases where the Blue Book DLMS UA 1000-1 allows choices.

NOTE The COSEM object definition file currently covers abstract, electricity and gas related COSEM objects.

Figure 11 shows the COSEM object definition file cover sheet (version 2.9).

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	24/44
-----------------------	------------	------------------------------	-------


 dlms device language message specification	COSEM conformance testing
COSEM conformance testing - Object definition tables	
Author:	DLMS UA WG Maintenance Gyozo Kmethy, Victoria Varju
Version:	V2.9
Filename:	Object_defs_v2.9_released_GKVV141201.xlsx
Revision date:	01/12/2014
Status:	Released
Digital signature of .dat file	389E-C19C-6A7F-0C42-32AE-5C1C-1DBA-2106
Copyright:	© Copyright DLMS UA 1997-2014
Classification:	DLMS User Association use only
Replaces:	Object_defs_v2.8_released_141023.xlsx
Date:	22nd October 2014
<p>Object definitions are valid combinations of Logical names, interface classes and the list of data types for attributes which are declared in the relevant interface class definitions as type "CHOICE".</p> <p>This version contains object definitions of abstract, electricity and gas related objects.</p> <p>The textual name of each object is described in a modular way.</p> <p>For the changes in the various version, see the "Change log" sheet.</p> <p>A special tool converts the worksheets to a .dat file for use by the Conformance Test Tool.</p> <p>CTT 2.X reports all objects found with their logical name, textual name and in case of "CHOICE" attributes, with data type. The access rights to all attributes are also reported.</p> <p>References: [1] DLMS UA 1000-1 Ed. 12.0 2014-09-10, COSEM Interface Classes and OBIS Object Identification System "Blue Book" [2] EN 13757-1 Communication system for meters and remote reading of meters - Part 1: Data exchange</p>	
DLMS User Association	DLMS UA 1001-7 Ed. 2.9: 2014-12-01

Figure 11 – COSEM object definition file cover sheet

The COSEM object definition file is updated whenever new COSEM objects (OBIS codes and/or interface classes) are defined by the DLMS UA.

The various versions are publicly available at www.dlms.com as excel files under the CONFORMANCE and the DOCUMENTATION menu.

The CTT uses a .dat file generated from the excel file. The various versions can be downloaded by licensed CTT users from <http://www.eurodcs.com/>. The .dat file has to be copied to the same folder where the CTT.exe file is located, see 6.5.

NOTE The excel file is not used by the CTT.

The Test Report contains the file name and the hash value of the excel file.

It is recommended using always the latest version. Earlier versions can be used for re-testing whenever it is necessary.

7.3.5 Selection of the test cases

The TEST CASES window allows selecting the test cases. The test suites are:

- HDLC, see DLMS UA 1001-3: ATS_DL V 5;
- Application layer (APPL), COSEM object (COSEM) and Symmetric key security (SYMSEC_0) see DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3;

The test cases can be selected by test suites and within each test suite one by one, see Figure 12.

During a test session, the CTT executes only those test cases that have been selected and that are applicable based on the IUT configuration and the CTI declarations.

Test cases, which are not selected, are marked in the Report and the Log as “SKIPPED”.

If the purpose of the test session is to obtain a Certification, then all test cases shall be selected. The CTT will automatically run all test cases applicable with the given CTI file and IUT configuration. The tests that cannot be run are marked as “INAPPLICABLE”.

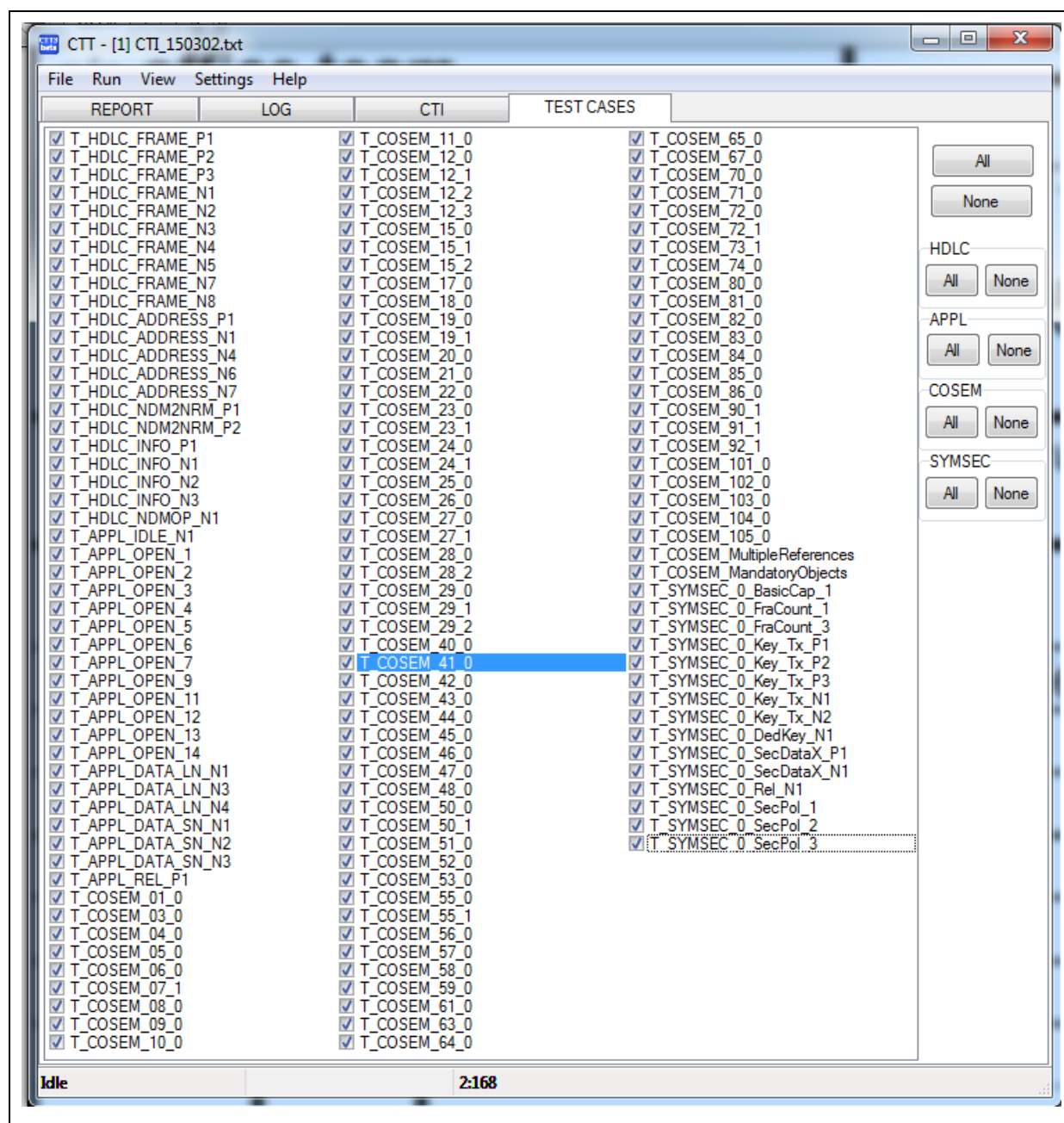


Figure 12 – Selection of the test cases

7.3.6 Connection of the IUT to CTT

The connection of the IUT to CTT depends on the communication profile used:

- when the 3-layer, CO, HDLC based communication profile is used, the IUT can be connected directly or via an optical probe to a communication port of the host computer running the CTT. This port shall not be shared by any other applications during running time. If an optical probe is used, it shall be checked if the probe is echoing or not.

Battery operated meters can also be tested using the wake-up sequence specified in IEC 62056-21.

- to connect the IUT via a pair of modems, the connection (com) – MODEM – MODEM – IUT has to be established before testing, i.e. the dialling of (and the connection to) the remote modem has to be done before testing;

- when the TCP/IP based communication profile is used over the GPRS network, then the connection OS(com) – GPRS-MODEM – GPRS-MODEM – IUT has to be established before testing.

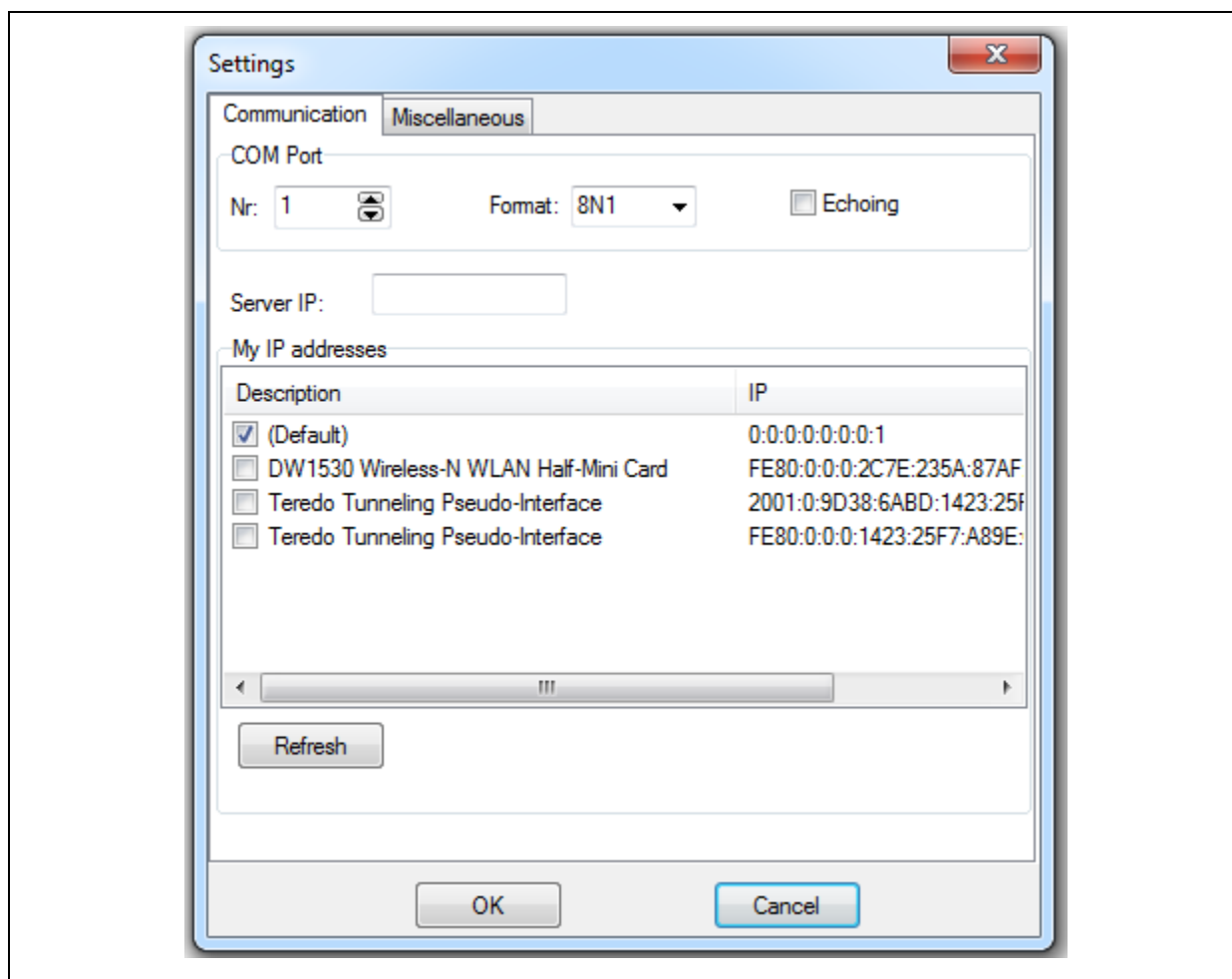


Figure 13 – Communication settings

The communication parameters can be set from the **Settings** menu, **Communication** tab as shown in Figure 13:

- COM port: here, the communication port used in the case of the the 3-layer, CO, HDLC based profile can be selected;
- Format: refers to the physical layer of the 3-layer, CO, HDLC based communication profile and it shall be set to 8N1: asynchronous transmission with 1 start bit, 8 data bits, no parity and 1 stop bit;
- Echoing: this shall be set if the optical probe is used and it is echoing;
- Server IP: when the TCP/IP based profile is used, the IP address of the IUT shall be entered here;
 NOTE The way to obtain the IP address depends on the provider and the SIM card and the process is out of the Scope of this document.
- “My IP addresses” holds the IP address of the PC running the CTT. It is needed only if the IUT has to establish the TCP connection to the CTT i.e. when the IUT is pushing. It shall be configured in the IUT.

7.3.7 Test sessions

If the IUT supports more than one communication profile, then a test session shall be performed for each communication profile for which compliance is claimed.

If the IUT supports more than one communication interface, then a test session may be performed on each interface.

A test session can be started from the Run menu in two ways:

- “Run”: the test session runs on the Logical Devices and AAs enabled, with the test cases, media and “Do not test” options selected. At the end of the test session a “Test Result” .zip file can be created. Test cases not selected are marked in the Test Report and Log as “SKIPPED”;
- “Run for Certification”: for this, all Logical Devices and AAs must be enabled, all tests cases shall be selected and there shall be no “Do not test” options specified. At the end of the test session a “Test Result” .zip file can be created. Test cases that could not be run on the IUT with the given IUT configuration and CTI declarations are marked in the Test Report and the Log as “INAPPLICABLE”.

The progress of the test can be followed via the REPORT and LOG panes and the Traffic window (to be opened from the View menu).

In the case when a failure occurs in a test case that may affect running other test cases the CTT raises a fatal failure. CTT allows the user to choose one of a predefined set of options:

- “Abort the test session”: if this choice is taken, the test session is aborted;
- “Continue”: if this choice is taken, the test session is continued;
- “Ask the user”: if this choice is taken, the test session is suspended and a dialog box is displayed. If the operator chooses to continue the test session, it is resumed. If the operator chooses not to continue, then the test session is aborted.

In all cases, the reason for raising the fatal failure is logged.

In some cases, an EXCEPTION can occur during the test session. These may be caused by inappropriate settings or by abnormal data received from the IUT. The Help file provides more information on this.

The test session can be aborted any time from the **Run** menu.

7.3.8 Production of the Test Result

7.3.8.1 Overview

The Test Result comprises five files that can be produced at the end of the test session from the **File / Save Test-Result**:

- _CTI.txt is the CTI file;
- _Report.txt is a text file holding the content the "REPORT" pane;
- _Log.txt is a text file holding the content of the "LOG" pane;
- _Traffic.rtf is a rich text file holding the content of the Traffic window;
- _hash.txt is a text file containing a digest of the 4 other files.

The files are best viewed from CTT itself or using a suitable Large Text Viewer program.

An existing Test Result file can be loaded using **File / Load Test-Result**. The files of the zip archive are displayed in their respective panes and windows. The integrity of the archive is verified using the _hash file.

7.3.8.2 The Report

The main elements of the Report are the following:

- 1) General information on the test session:
 - date of testing;
 - CTT version;
 - information on the licensee;
 - information on the tester;
 - whether the test session was started as “Run for Certification”;
- 2) Identification of the IUT;
- 3) A summary of results and the features supported (aggregated over all logical devices and AAs):
 - number of tests executed for each test suite, and the summary of the verdicts;
 - the communication profile supported;
 - the application context names supported;
 - the ACSE and xDLMS features supported;
 - the security features supported;
 - information on the Logical Devices found;
 - the COSEM interface classes tested;
 - the COSEM interface classes found but not tested;
- 4) the result of HDLC tests (when applicable);
- 5) the result of the Application layer test cases;
- 6) the result of the COSEM test cases;
- 7) the result of the SYMSEC_0 test cases;
- 8) the CTI file;
- 9) the name of the Object definition file used and its hash value.

A fragment of a sample report is shown in Figure 14.

```

*****
DLMS Conformance test report
27-MAY-2015 17:17:04
CTT 3.0 extended edition, 64bits (100)
Licensed to: i-cube (21-Jul-05)
Tester: Christian
RUN FOR CERTIFICATION
*****

*****
* Identification *
*****

Identification = {
  Manufacturer = "\i-cube"
  FLAGid = "\ICU"
  Type = "\simulation"
  SerialNr = "000102"
  Comment = 1234
  Comment = "\comment 2"
}

*****
* Summary *
*****

TYPE      TOTAL      SKIPPED      INAPPLICABLE      INCONCLUSIVE      PASSED      FAILED
----      -
HDLC
APPL      52          0           10              0                42          0
COSEM     988          0            1              0               986          1
SYMSEC    55           0           10              0                44          1

Communication profile supported:      TCP

Application context names supported:  LONG_NAMES, LONG_NAMES_WITH_CIPHERING

Security mechanisms supported:        NO_SECURITY, HIGH_LEVEL_SECURITY_GMAC

Features supported:
ACTION, ACTIVATE_SECURITY_POLICY
GENERAL_BLOCK_TRANSFER
GENERAL_GLO_CIPHERING, GET
MULTIPLE_REFERENCES, RLRQ_RLRE
SELECTIVE_ACCESS
SERVICE_SPECIFIC_BLOCK_TRANSFER, SET

Logical device(s) found:
SAP = 1 is "4943553030303030" (ICU00000)
SAP = 2 is "4943553030303032" (ICU00002)
SAP = 3 is "4943553030303033" (ICU00003)

Tested COSEM classes:
1,3,4,5,6,7(1),8,9,10,11,12(3),15(2),17,18
19(1),20,21,22,23(1),24,24(1),25,26,27,28
28(2),29,29(1),29(2),40,41,42,43,44,45,46
47,48,50(1),51,52,53,55(1),56,57,58,59,61
63,64,65,70,71,72,73(1),74,80,81,82,83,84
85,86,90(1),91(1),92(1),101,102,103,104,105

COSEM classes found but not tested:  (none)

*****
* HDLC Tests *
*****

*****
* APPL Tests *
*****

T_APPL_IDLE_N1

```

Figure 14 – Fragment of a sample Test Report

7.3.8.3 The Log

The Log displays the listing of all actions executed by CTT, see Figure 15.

```

STARTED 27-May-15 17:35:47
0'00.001 Starting server listening on port 4059
0'00.013 HDLC Tests
0'00.014 WARNING: All HDLC tests skipped, communication profile is TCP
0'00.017 APPL Tests
0'00.020 Starting T_APPL_IDLE_N1
0'00.021 T_APPL_IDLE_N1 0/0, ServerSAP = 1, ClientSAP = 16, LONG_NAMES, NO_SECURITY
0'00.569 VERDICT: Data exchange in IDLE state PASSED

0'00.572 Starting T_APPL_OPEN_1
0'00.574 T_APPL_OPEN_1 0/0, ServerSAP = 1, ClientSAP = 16, LONG_NAMES, NO_SECURITY
0'00.576 SubTest 1.Establish an AA using the parameters declared
0'01.744 VERDICT: 1.Establish an AA using the parameters declared PASSED

0'01.746 SubTest 2.Check that the AA has been established
0'01.858 VERDICT: 2.Check that the AA has been established PASSED

0'01.860 SubTest 3.Release the AA
0'01.962 VERDICT: 3.Release the AA PASSED

0'01.965 T_APPL_OPEN_1 0/1, ServerSAP = 1, ClientSAP = 1, LONG_NAMES_WITH_CIPHERING,
HIGH_LEVEL_SECURITY_GMAC
0'01.967 SubTest 1.Establish an AA using the parameters declared
0'03.259 VERDICT: 1.Establish an AA using the parameters declared PASSED

0'03.261 SubTest 2.Check that the AA has been established
0'03.374 VERDICT: 2.Check that the AA has been established PASSED

0'03.376 SubTest 3.Release the AA
0'03.480 VERDICT: 3.Release the AA PASSED

```

Figure 15 – Basic log

The CTT extended edition provides a more detailed log. Figure 16 shows the detailed log presenting COSEM APDUs in XML format.


```

STARTED 27-May-15 17:17:04
0'00.001 Starting server listening on port 4059
0'00.002 HDLC Tests
0'00.008 WARNING: All HDLC tests skipped, communication profile is TCP
0'00.008 APPL Tests
0'00.010 Starting T_APPL_IDLE_N1
0'00.011 T_APPL_IDLE_N1 0/0, ServerSAP = 1, ClientSAP = 16, LONG_NAMES, NO_SECURITY
REQUEST:
    <GetRequest>
    <GetRequestNormal>
    <InvokeIdAndPriority Value="C1" />
    <AttributeDescriptor>
    <ClassId Value="000F" />
    <InstanceId Value="0000280000FF" />
    <AttributeId Value="01" />
    </AttributeDescriptor>
    </GetRequestNormal>
    </GetRequest>
WRAPPER[1] Sent 000100100001000DC001C1000F0000280000FF0100
+400 WRAPPER[1] Rec 0001000100100003D80101
RESPONSE:
    <ExceptionResponse>
    <StateError Value="ServiceNotAllowed" />
    <ServiceError Value="OperationNotPossible" />
    </ExceptionResponse>
0'00.552 VERDICT: Data exchange in IDLE state PASSED

0'00.555 Starting T_APPL_OPEN_1
0'00.557 T_APPL_OPEN_1 0/0, ServerSAP = 1, ClientSAP = 16, LONG_NAMES, NO_SECURITY
0'00.559 SubTest 1.Establish an AA using the parameters declared
REQUEST:
    <AssociationRequest>
    <ApplicationContextName Value="LN" />
    <InitiateRequest>
    <ProposedDlmsVersionNumber Value="06" />
    <ProposedConformance>
    <ConformanceBit Name="Action" />
    <ConformanceBit Name="EventNotification" />
    <ConformanceBit Name="SelectiveAccess" />
    <ConformanceBit Name="Set" />
    <ConformanceBit Name="Get" />
    <ConformanceBit Name="DataNotification" />
    <ConformanceBit Name="MultipleReferences" />
    <ConformanceBit Name="BlockTransferWithAction" />
    <ConformanceBit Name="BlockTransferWithSetOrWrite" />
    <ConformanceBit Name="BlockTransferWithGetOrRead" />
    <ConformanceBit Name="Attribute0SupportedWithGet" />
    <ConformanceBit Name="PriorityMgmtSupported" />
    <ConformanceBit Name="Attribute0SupportedWithSet" />
    <ConformanceBit Name="GeneralBlockTransfer" />
    <ConformanceBit Name="GeneralProtection" />
    </ProposedConformance>
    <ProposedMaxPduSize Value="FFFF" />
    </InitiateRequest>
    </AssociationRequest>
WRAPPER[1] Sent
000100100001001F601DA109060760857405080101BE10040E01000000065F1F040060FE9FFFFF
+530 WRAPPER[1] Rec
00010001001000346132A109060760857405080101A203020100A305A103020100890760857405080200BE10040E08
00065F1F0400601A9D05000007
RESPONSE:

```

Figure 16 – Detailed log presenting COSEM APDUs in XML format

7.3.8.4 Traffic

The Traffic window– see Figure 17 – can be opened from the **View** menu. As the test progresses, it shows the frames sent to (in green) and the received from (in red) the IUT.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	33/44
-----------------------	------------	------------------------------	-------

```

TCPClient 0'00.001 +88851 Start Listening
TCPClient 0'00.022 +21 Connecting
TCPClient 0'00.147 +124 Connected to 127.0.0.1:4058
TCPClient 0'00.149 +1 000100100001000DC001C1000F0000280000FF0100
TCPClient 0'00.549 +400 0001000100100003D80101
TCPClient 0'00.558 +8 Disconnecting
TCPClient 0'00.559 +0 Disconnected
TCPClient 0'00.560 +1 Waiting TCP disconnect to connect delay 500 ms
TCPClient 0'01.060 +499 Connecting
TCPClient 0'01.185 +124 Connected to 127.0.0.1:4058
TCPClient 0'01.191 +5
000100100001001F601DA109060760857405080101BE10040E01000000065F1F040060FE9FFFFF
TCPClient 0'01.721 +530
00010001001000346132A109060760857405080101A203020100A305A103020100890760857405080200BE10040E08
00065F1F0400601A9D05000007
TCPClient 0'01.746 +24 000100100001000DC001C1000F0000280000FF0100
TCPClient 0'01.836 +90 000100010010000CC401C10009060000280000FF
TCPClient 0'01.864 +27 00010010000100056203800100
TCPClient 0'01.944 +80 00010001001000056303800100
TCPClient 0'01.966 +21 Disconnecting
TCPClient 0'01.967 +0 Disconnected
TCPClient 0'01.986 +19 Waiting TCP disconnect to connect delay 485 ms
TCPClient 0'02.471 +485 Connecting
TCPClient 0'02.596 +124 Connected to 127.0.0.1:4058
TCPClient 0'02.602 +5
000100010001005A6058A109060760857405080103A60A040843545430303030308A0207808B0760857405080205AC
0D800BAC93A248C8FC1CA5C20B27BE230421211F3026ECF3DAC94C41B52F6118D3FAAF68BA253187948CF66D01FD93
264BDCC9
TCPClient 0'03.112 +510
000100010001006B6169A109060760857405080103A203020100A305A10302010EA40A040849435530303030308802
0780890760857405080205AA1280106A495F8D185723E288A794A19AE64024BE230421281F3026ECF3DFBA974DD84A
A9262FA3FD4349F7A268D0B41B4BBCDA8448715093
TCPClient 0'03.145 +32
000100010001003CDB084354543030303030313026ECF3E01ACF25914E9576DC0DA746172438C8242066F456E2D73F
4EAB5264C69B2608DD37694CCF5BEB4FA71D51424D
TCPClient 0'03.235 +90
0001000100010035DB0849435530303030302A3026ECF3E1E385CBA72F04AF871CE9C0760AC57172D7F69120791D8F
C5E0A2C5E74BB43B53D2E1560671
TCPClient 0'03.267 +32 000100010001000DC001C1000F0000280000FF0100
TCPClient 0'03.347 +79 000100010001000CC401C10009060000280000FF
TCPClient 0'03.382 +34
000100010001002A6228800100BE230421211F3026ECF3E1B2CC46BB00C666E7B147F7C2005397473D1862BC5D0257
F64359

```

Figure 17 – The Traffic window

7.4 Repeatability of results

In order to achieve the objective of credible conformance testing, it is clear that the result of executing a test case on an IUT should be the same whenever it is performed. Experience shows that it may not be possible to execute a complete conformance test suite and observed test outcomes which are identical to those obtained on another occasion.

Nevertheless, at the test case level, every effort has been made to minimize the possibility that a test case produces different test outcomes on different occasions.

7.5 Requirements for test laboratories

Conformance assessment may be performed by a manufacturer (self-testing), a third party or a user.

If the test is done by the manufacturer, the test laboratory should be an identifiable part of the manufacturer's organisation.

8 The certification process

8.1 General

The purpose of the certification process is to obtain a “DLMS/COSEM compliant” Certification. This clause describes the necessary steps.

8.2 Initiation of the certification process

The certification process may be initiated by any member of the DLMS UA.

The manufacturer of the device to be certified shall be a member of the DLMS UA and shall possess a three-letter manufacturer ID.

For more information on the manufacturer IDs (FLAG ID) see www.dlms.com, ORGANIZATION menu.

8.3 Submission of conformance test documents

The Test Results generated by the CTT (as .zip files) shall be submitted to certification@dlms.com. The DLMS UA registers and examines them to confirm their authenticity and to verify that all technical and administrative acceptance conditions are met. If so, it prepares the Certification. The DLMS UA does not publish the test results; these may be obtained from the manufacturer.

The DLMS UA maintains the right to discuss the contents of the conformance test result with the organization having initiated the certification process.

8.4 Technical and administrative checks

The technical verification and acceptance criteria are the following:

- the Test result .zip file includes the Report, the Log, the Line traffic, the CTI and the hash files and the hash value is correct;
- a recent version of the COSEM object definitions file has been used;
- the Logical Device Name of the Management Logical Device is syntactically correct and the Three Letter Manufacturer ID (AKA FLAG ID) matches the one registered for the manufacturer;
- the test session has been started with “Run for Certification”;
- there are no FAILED test cases;
- the reasons for INCONCLUSIVE test results are acceptable. For example selective access – which is not mandatory – is not available,

The administrative checks before issuing the Certification are the following:

- the test result has been generated by a test laboratory having purchased the CTT. This is verified by checking the license owner’s name reported;
- the manufacturer and the tester – if different – are active members of the DLMS UA;
- in the case of third party testing, the license fee is paid;
- there was no Certification issued for the same type.

8.5 The Certification

A Certification is issued if the IUT passed all applicable tests. It is prepared using the data taken from the Test Result(s) and contains the following elements:

- a unique Certification number assigned by the DLMS UA;
- the identification of the IUT;
- the identification of the Management Logical Device (bound to SAP = 1);

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	35/44
-----------------------	------------	------------------------------	-------

- the identification of the manufacturer as declared in the CTI;
- the identification of the CTT version;
- the identification of the licensee;
- the version of the COSEM Object definitions file used;
- the media (energy types) used for COSEM object testing;
- for each test performed:
 - the communication profile and the opening mode (when applicable);
 - the application contexts;
 - the security suite;
 - the date and time of testing; and
 - the hash value of the Test Result
- an indication that the Certification is only valid for the functions successfully tested;
- an indication that the test is executed on one specimen of the product and that the test results may not be applicable for other test specimens;
- any remarks added by the DLMS UA, as seen fit;
- date of issue and signature;
- a summary of features for each test session extracted from the Test Report.

The Certification with example data is included as Annex A.

The Certification is always issued to the manufacturer as given in the Test Report. The Certifications are published on the website of the DLMS UA at www.dlms.com, CONFORMANCE menu.

The data in a Certification published cannot be changed. If the data of the manufacturer or the IUT change, a new Certification is necessary.

The Certification obtained entitles the manufacturer to place the “DLMS/COSEM compliant mark” on its products and documentation. See Figure 18.



Figure 18 – The DLMS/COSEM compliant logo

The test results are filed by the DLMS UA.

8.6 Scope and validity of the Certification

The DLMS/COSEM Certification certifies that the IUT as identified by the manufacturer/test house passed the tests applicable for the given configuration.

The supporting evidence is the conformance Test Result.

DLMS/COSEM compliance can be claimed only for the features tested.

The DLMS UA does not control if the meters manufactured are identical to the IUT tested.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	36/44
-----------------------	------------	------------------------------	-------

The Certification remains valid as long as no design or manufacturing changes in communication hardware and firmware with essential influence on the implementation have been made. If changes have been made, a re-test is necessary. This is left to the judgement of the manufacturer.

8.7 Disclaimer

The DLMS UA takes all possible effort to ensure that the conformance test plans and the CTT are line with the DLMS/COSEM specification and provide a reasonable depth of testing.

The Certification does not mean however that an absolute proof of conformance is given.

9 The quality program

9.1 General

An important element of the DLMS/COSEM conformance testing process is the quality program. It includes:

- validation of the conformance test plans and the CTT;
- the support provided to users;
- maintenance.

9.2 Validation of the Abstract Test Suites and CTT

The validation of the ATSS and its implementation, the CTT has been done in several steps:

1. the test plans have been written by experts from different members of the DLMS UA WG Maintenance based on DLMS UA 1000-1 and DLMS UA 1000-2.
2. the executable Test Suite has been validated by running them against several implementations.

9.3 Assistance provided to users

The DLMS UA, upon request, provides support to the users of the tool. For this purpose, test results can be sent to the DLMS UA: certification@dlms.com.

9.4 Maintenance

The DLMS UA maintains the conformance testing process to eliminate any problems with the tool found during testing, to enhance tests and to accommodate changes in the DLMS/COSEM specification. The procedure is the following:

1. a proposal, together with a justification is made to modify or to add a test. This can be initiated by any member of the DLMS UA or by the DLMS UA itself;
2. the request is investigated by the DLMS UA;
3. if the request is accepted, the relevant abstract test cases are amended by the DLMS UA;
4. the new abstract test cases are validated by the DLMS UA;
5. the new abstract test cases are implemented;
6. the amended ATSS are published;
7. a new version of the CTT is made available to the licensed tool users.

This process is supported by the DLMS UA website.

In the following, the process is illustrated by use cases.

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	37/44
-----------------------	------------	------------------------------	-------

9.5 Use cases

9.5.1 Use case 1 – introducing a new standard OBIS code

A manufacturer needs a new standard OBIS code to support a new functionality in the metering equipment.

The proposal is submitted to the DLMS UA. The DLMS UA checks if the proposal is in line and can fit with the DLMS/COSEM specification. If approved, the COSEM Object definition tables DLMS UA 1001-7 are amended and a new .dat file is made available for download. See also 7.3.4.

9.5.2 Use case 2 – modification of an existing test

If it is found that despite of careful validation, a test case is not fully compliant with the DLMS/COSEM specification or if it is necessary to enhance a test, then a proposal may be submitted to the DLMS UA and the process described above is followed.

9.5.3 Use case 3 – adding a test for a new standard feature

A manufacturer implements a feature described in the DLMS/COSEM specification, but which is not yet covered in the CTT.

The manufacturer submits the proposed test plan to the DLMS UA and the process described above is followed.

9.5.4 Use case 4 – revision of the specification

The DLMS UA initiates a revision or amendment of the DLMS/COSEM specification (e.g. introducing a new protocol stack).

The conformance requirements and the test plans are prepared together with the standard, but at least upon the acceptance of the new standard.

The DLMS UA initiates the maintenance of the tool.

Annex A

Certification template (with sample data) (informative)



DLMS User Association

Bahnhofstrasse 28
CH-6304 Zug
Switzerland

Tel. +36 28 514 065
Fax +36 28 514 066
dlms@dlms.com

Certification No. 1XXX

This is to certify that the metering equipment identified as:

Type: **SAMPLE**
Mgmt. Logical Device Name: **ICU00002**
Manufactured by: **I-Cube**

has successfully passed the DLMS/COSEM Conformance test, under the following conditions:

- CTT version: CTT 3.0 extended edition, 64bits (100)
- Licensed to: I-Cube
- COSEM object definitions file version: Object_defs_v2.9_released_GKVV141201.dat
- Media identifier(s): Abstract, Electricity

Tests	Comm. profile & opening mode	Application contexts	Security suite	Date and time	Test Result hash value
Test 1	TCP	LN, LN with ciphering	0	2015-05-28	481CF68B3A28C86EED7BCD9908281C50 4312FA91D8E1EFC09B703858D00B9AA8C
Test 2	HDLC, Mode_E	LN, LN with ciphering	0	2015-05-28	<hash value >
Test 3	HDLC, Direct	LN, LN with ciphering	0	2015-05-28	<hash value >

The authenticity of the test report(s) has been verified by the DLMS User Association and the metering equipment identified above is listed on its web site at: www.dlms.com.

With this, the manufacturer is entitled to display the DLMS/COSEM Compliant mark – shown below – on its product duly identified and on its product literature.



The test reports are filed by the DLMS UA. Copies are available from the manufacturer.

This Certificate is only valid for the functions successfully tested; see the Summary. The test has been executed on one specimen of the product, as identified by the Management Logical Device Name reported. Results may not be applicable for other test specimens.

Date: Zug, the

Paul Fuchs
General Secretary

DLMS User Association

dlms@dlms.com

Page 1 of 3

Certification No. 1XXX

Summary of features tested for Test 1

This section provides an overview of the features tested, aggregated over all logical devices and Application Associations in one test session. Full information is available in the Test Result.

* Identification *

```
Identification = {
  Manufacturer = "\i-cube"
  FLAGid = "\ICU"
  Type = "\simulation"
  SerialNr = "000102"
  Comment = 1234
  Comment = "\comment 2"
}
```

* Summary *

TYPE	TOTAL	SKIPPED	INAPPLICABLE	INCONCLUSIVE	PASSED	FAILED
----	-----	-----	-----	-----	-----	-----
HDLC		All (Communication profile not HDLC)				
APPL	52	0	10	0	42	0
COSEM	988	0	1	0	987	0
SYMBEC	55	0	10	0	45	0

Communication profile: TCP

Application context names: LONG_NAMES, LONG_NAMES_WITH_CIPHERING

Security mechanisms: NO_SECURITY, HIGH_LEVEL_SECURITY_GMAC

ACSE and xDLMS features: ACTION, GENERAL_BLOCK_TRANSFER, GET
MULTIPLE_REFERENCES, RLRQ_RLRE
SELECTIVE_ACCESS
SERVICE_SPECIFIC_BLOCK_TRANSFER, SET

Security features: ACTIVATE_SECURITY_POLICY
GENERAL_GLO_CIPHERING

Logical device(s) found: SAP = 1 is "4943553030303030" (ICU000000)
SAP = 2 is "4943553030303032" (ICU000002)
SAP = 3 is "4943553030303033" (ICU000003)

COSEM classes tested: 1, 3, 4, 5, 6, 7 (1), 8, 9, 10, 11, 12 (3), 15 (2), 17, 18
19 (1), 20, 21, 22, 23 (1), 24, 24 (1), 25, 26, 27, 28
28 (2), 29, 29 (1), 29 (2), 40, 41, 42, 43, 44, 45, 46
47, 48, 50 (1), 51, 52, 53, 55 (1), 56, 57, 58, 59, 61
63, 64, 65, 70, 71, 72, 73 (1), 74, 80, 81, 82, 83, 84
85, 86, 90 (1), 91 (1), 92 (1), 101, 102, 103, 104, 105

COSEM classes found but not tested: (none)

Annex B (normative) Conformance Test Plans

The Conformance test plans – Abstract Test Suites – used in CTT 3.0 are the following:

- DLMS/COSEM conformance testing – Conformance test plans - Data link layer using HDLC protocol: DLMS UA 1001-3: ATS_DL V 5;
- DLMS/COSEM conformance testing – Abstract Test Plans – DLMS/COSEM application layer – COSEM interface objects – Symmetric key security suite: DLMS UA 1001-6: ATS_AL_COSEM_SYMSEC_0 V 1.3.

The Conformance test plans are attached to this document.

Annex C (informative) Bibliography

ETSI ETR 021: 1991, *Advanced testing methods (ATM); Tutorial on protocol conformance testing (especially OSI standards and profiles) ETR/ATM-1002*

IEC 60870-5-6:2006, *Telecontrol equipment and systems – Part 5-6: Guidelines for conformance testing for the IEC 60870-5 companion standards*

IEC 61850-10: 2005, *Communication networks and systems in substations – Part 10: Conformance testing*

IEC 62056-1-0, *Electricity metering data exchange – The DLMS/COSEM suite – Part 1 0: Smart metering standardisation framework*

IEC 62056-21, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange*

IEC 62056-46, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data link layer using HDLC protocol*

IEC 62056-5-3, *Electricity metering data exchange – The DLMS/COSEM suite – Part 5-3: DLMS/COSEM application layer*

IEC 62056-6-1, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-1: Object Identification System (OBIS)*

IEC 62056-6-2, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-2: COSEM interface classes 3*

IEC 62056-7-6, *Electricity metering data exchange – The DLMS/COSEM suite – Part 7-6: The 3-layer, connection-oriented HDLC based communication profile*

IEC 62056-8-3, *Electricity metering data exchange – The DLMS/COSEM suite – Part 8-3: Communication profile for PLC S-FSK neighbourhood networks*

IEC 62056-9-7, *Electricity metering data exchange – The DLMS/COSEM suite – Part 9-7: Communication profile for TCP-UDP/IP networks*

EN 13757-1:2014, *Communication system for meters – Part 1: Data exchange*

EN 13757-3:2004, *Communication systems for and remote reading of meters – Part 3: Dedicated application layer*

NOTE This standard is referenced in the “M-Bus client setup” interface class version 0.

EN 13757-3:2013, *Communication systems for and remote reading of meters – Part 3: Dedicated application layer*

NOTE This standard is referenced in the M-Bus client setup interface class version 1.

ITU-T X.291:1992, *OSI conformance testing methodology and framework for protocol recommendations for IUT-T applications – Abstract test suite specification*

ITU-T X.293:1995, *OSI conformance testing methodology and framework for protocol recommendations for IUT-T applications – Test realization*

DLMS User Association	2015-06-19	DLMS UA 1001-1 Ed. 5.0 V 1.0	42/44
-----------------------	------------	------------------------------	-------

INDEX

- 3-layer, CO, HDLC based communication profile, 13, 27
- Abnormal test case termination, 16, 17
- Abstract Test Case, 5, 15, 16
- Abstract Test Case error, 16
- Abstract Test Suite, 6, 10, 11, 14, 18, 37
- Access right, 21
- Application Association, 10, 22
- Application context, 21, 36
- Ask the user, 29
- Authentication security mechanism, 21, 22
- base specification, 15
- Basic interconnection test (BIT), 6
- Behaviour, 14
- Behaviour test, 6, 15
- Black box, 14
- Capabilities, required, 11
- Capability, 6, 14
- Capability test, 6, 15
- Certification, 26, 35
- Certification process, 11, 35
- Certification, scope and validity, 36
- Communication interface, 12, 29
- Communication model, 13
- Communication port, 28
- Communication profile, 11, 29, 36
- Companion Specification, 10
- Conformance assessment process, 5, 6, 11, 20
- Conformance log, 6
- Conformance test information, 21
- Conformance test information (CTI), 6
- Conformance test plan, 11, 13, 14, 37
- Conformance test result, 17, 36
- Conformance Test Tool, 11, 18
- Conformance testing, 6, 11, 12, 20, 34, 37
- Conforming implementation, 6
- COSEM interface object, 17, 18
- COSEM interface object model, 11, 12, 13
- COSEM object definition file, 23, 24, 36
- CTI file, 23
- CTT, installation, 19
- CTT, licensing, 19
- Data link layer using HDLC protocol, 13
- DLMS/COSEM application layer, 12
- DLMS/COSEM compliant, 35
- DLMS/COSEM compliant mark, 36
- DLMS/COSEM conformance testing, 11
- DLMS/COSEM transport layer, 13
- DoNotTest options, 24
- Executable Test Case, 7
- Executable Test Case error, 7, 16
- Executable Test Suite, 7, 10, 11, 18
- Expected result, 15
- Extra information, 21
- FAILED, 16
- Foreseen test event, 16
- Idle testing state, 7
- Implementation Under Test (IUT), 7, 11
- Inapplicable, 7
- INAPPLICABLE, 16, 26
- INCONCLUSIVE, 16
- Inconclusive (verdict), 7
- Initial testing state, 7
- Inopportune test event, 7
- Intended application, 21
- Interface object, 21
- Interworking, 11
- Invalid test event, 16
- IP address, 28
- IUT, identification, 35
- IUT, preparation, 21
- Line traffic, 33
- Load profile, 21
- Log, 32
- Logging, basic, 32
- Logging, detailed, 33
- Logical device, 21
- Maintenance, 37
- Management Logical Device, 21
- Manufacturer, 35
- Manufacturer identification, 36
- Means Of Testing, 8, 18
- Negative test, 8, 15
- Observed test outcome, 8, 16, 34
- OSI conformance testing, 11
- Parameterized executable test case, 8
- Pass (verdict), 8
- PASSED, 16
- Physical layer, 14
- Positive test, 8, 15
- Postamble, 16
- Preamble, 15
- Protocol conformance test report (PCTR), 8
- Protocol Data Unit, 10
- Protocol stack, 14
- Quality program, 11, 12, 37
- Repeatability of results, 8, 34
- Run, 29
- Run for Certification, 29
- Security context, 22
- Security features, 21
- Security material, 22
- Security suite, 21, 36
- Selective access, 21
- Self-testing, 12, 20, 34
- Semantically invalid test event, 8
- Service Access Point, 10
- SKIPPED, 26
- Stable testing state, 9
- Static conformance review, 9
- Supporting layer, 14
- Syntactically invalid test event, 9
- TCP/IP based communication profile, 13, 28
- Template, test case, 16
- Test, 16
- Test body, 9, 16
- Test case, 9, 14, 15, 34
- Test case error, 9, 16
- Test case name, 15
- Test event, 9, 14, 16
- Test group, 9, 14, 15
- Test group objective, 9, 14
- Test laboratory, 9, 34
- Test outcome, 16
- Test outcome, foreseen, 7, 16
- Test outcome, unforeseen, 10, 16
- Test purpose, 9, 14, 15
- Test result, hash value, 36
- Test session, 10, 26
- Test step, 14
- Test step (sub-test), 9
- Test suite, 10, 15, 34
- Test tool version, 36
- Third party testing, 12, 20, 34
- Three-letter manufacturer ID, 12, 35
- Valid test event, 10
- Validation of the conformance test plan, 37
- Verdict, 10, 16
- Verdict, FAIL, 7
- xDLMS context, 21, 22
- xDLMS services, 21

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Project: **DLMS/COSEM conformance testing –
Conformance test plans –
Data link layer using HDLC protocol**

Author:	DLMS User Association – WG Maintenance – Gyozo Kmethy
Version:	V5.0
Status:	Released
Revision Date:	2010-12-15
Copyright:	© Copyright 2000-2010 DLMS UA
Classification:	DLMS User Association use only
Filename:	ATS_DL_V5_Released_101215.doc
Replace Doc:	V 4.1
Comment:	In line with Green Book Edition 7.0

Table of contents

Foreword	4
1 Scope	4
2 Introduction	5
2.1 Referenced documents	5
2.2 Terms, Definitions and Abbreviations	6
2.3 Revision History	8
2.4 Main changes compared to DLMS UA 1001-3 Version 4.1	8
3 The HDLC based data link layer test suite	9
3.1 Test strategy	9
3.2 Capabilities supported / not supported	9
3.3 Test groups	10
3.3.1 Overview	10
3.3.2 Test group HDLC_FRAME	10
3.3.3 Test group HDLC_ADDRESS	11
3.3.4 Test group HDLC_NDM2NRM	11
3.3.5 Test group HDLC_INFO	12
3.3.6 Test group HDLC_NDMOP	12
3.4 Information for testing	12
3.5 Test options	13
3.6 General test conditions	13
3.7 Test cases	14
3.7.1 Test cases of test group HDLC_FRAME	14
3.7.2 Test cases of test group HDLC_ADDRESS	18
3.7.3 Test cases of test group HDLC_NDM2NRM	21
3.7.4 Test cases of test group HDLC_INFO	22
3.7.5 Test cases of test group HDLC_NDMOP	23
INDEX	24

List of Tables

Table 1 – Capabilities supported / not supported	9
Table 2 – Test groups	10
Table 3 – Test group HDLC_FRAME_P: Positive tests	11
Table 4 – Test group HDLC_FRAME_N: Negative tests	11
Table 5 – Test group HDLC_ADDRESS_P: Positive tests	11
Table 6 – Test group HDLC_ADDRESS_N: Negative tests	11
Table 7 – Test group HDLC_NDM2NRM_P: Positive tests	12
Table 8 – Test group HDLC_INFO_P: Positive tests	12
Table 9 – Test group HDLC_INFO_N: Negative tests	12
Table 10 – Test group HDLC_NDMOP_N: Negative tests	12
Table 11 – HDLC_FRAME_P1: Connection and disconnection of the HDLC layer	14
Table 12 – HDLC_FRAME_P2: InterFrameTimeout	14
Table 13 – HDLC_FRAME_P3: InactivityTimeout	15
Table 14 – HDLC_FRAME_N1: HDLC frame flags missing	15
Table 15 – HDLC_FRAME_N2: HDLC frame too short	16
Table 16 – HDLC_FRAME_N3: Frame format type sub-field check	16
Table 17 – HDLC_FRAME_N4: Frame length sub-field check	16
Table 18 – HDLC_FRAME_N5: Control field check	17
Table 19 – HDLC_FRAME_N7: HCS field check	17
Table 20 – HDLC_FRAME_N8: FCS field check	18
Table 21 – HDLC_ADDRESS_P1: Correct addresses using the address structure(s) specified	18
Table 22 – HDLC_ADDRESS_N1: Two-bytes source address	19
Table 23 – HDLC_ADDRESS_N4: Unknown destination addresses	19
Table 24 – HDLC_ADDRESS_N6: One byte destination address when two or four bytes are expected	20
Table 25 – HDLC_ADDRESS_N7: Three bytes or five bytes destination address	20
Table 26 – HDLC_NDM2NRM_P1: MaximumInformationFieldLength parameter	21
Table 27 – HDLC_NDM2NRM_P2: WindowSize parameter	21
Table 28 – HDLC_INFO_P1: I frame exchange	22
Table 29 – HDLC_INFO_N1: Too long information field	22
Table 30 – HDLC_INFO_N2: Wrong N(R) sequence number	23
Table 31 – HDLC_INFO_N3: Wrong N(S) sequence number	23
Table 32 – HDLC_NDMOP_N1: I frame in NDM	23

Foreword

Copyright

© Copyright 1997-2010 DLMS User Association.

This document is confidential. It may not be copied, nor handed over to persons outside the standardisation environment.

The copyright is enforced by national and international law. The "Berne Convention for the Protection of Literary and Artistic Works", which is signed by 121 countries world-wide, and other treaties apply.

1 Scope

This document specifies an abstract test suite (ATS) for testing implementations of the Data link layer using HDLC protocol. It is based on the Green Book [2], [10] and [15].

To perform tests, the DLMS UA Conformance Test Tool is available.

DLMS/COSEM conformance testing – Conformance test plans – Data link layer using HDLC protocol

2 Introduction

2.1 Referenced documents

Ref.	Title
[1]	DLMS UA 1000-1:2010-08, Ed. 10.0: <i>Blue Book</i>
[2]	DLMS UA 1000-2:2009-12, Ed. 7.0: <i>Green Book</i>
[3]	DLMS UA 1001-7:2010-11 Ed. 2.1, <i>COSEM conformance testing - Object definition tables</i> ¹
[4]	IEC 60559:1989, <i>Binary floating-point arithmetic for microprocessor systems</i>
[5]	IEC 61334-4-41 Ed. 1.0: 1996, <i>Distribution automation using distribution line carrier systems - Part 4: Data communication protocols - Section 41: Application protocol - Distribution line message specification</i>
[6]	IEC 61334-6 Ed.1.0: 2000, <i>Distribution automation using distribution line carrier systems – Part 6: A-XDR encoding rule</i>
[7]	IEC 62053-23 Ed. 1.0: 2003, <i>Electricity metering equipment (a.c.) – Particular requirements - Static meters for reactive energy (classes 2 and 3)</i>
[8]	IEC 62056-21 Ed. 1.0: 2002, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct Local Data Exchange (3rd edition of IEC 61107)</i>
[9]	IEC 62056-31 Ed. 1.0: 1999, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 31: Using local area networks on twisted pair with carrier signalling</i>
[10]	IEC 62056-46 Ed.1.1: 2007, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data Link Layer using HDLC Protocol</i>
[11]	IEC 62056-47 Ed. 1.0: 2006, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 47: COSEM transport layers for IPv4 networks</i>
[12]	IEC 62056-53 Ed. 2.0: 2006, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 53: COSEM Application Layer</i>
[13]	IEC 62056-61 Ed.2.0: 2006, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: Object identification system (OBIS)</i>
[14]	IEC 62056-62 Ed.2.0: 2006, <i>Electricity metering – Data exchange for meter reading, tariff and load control – Part 62: Interface Objects</i>
[15]	ISO/IEC 13239:2002, Ed.3, <i>Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures</i>
[16]	ITU Recommendation X.217 (04/95), ISO/IEC 8649:1996 – <i>Information technology - Open Systems Interconnection – Service definition for the association control service element</i>
[17]	ITU Recommendation X.227 (04/95), ISO/IEC 8650:1996 – <i>Information technology - Open Systems Interconnection Connection-oriented protocol for the association control service element: Protocol specification</i>
[18]	ANSI C12.19: 1997 / IEEE 1377: 1998, <i>Utility industry end device data tables</i>
¹⁾ For the latest version, see www.dlms.com	

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

2.2 Terms, Definitions and Abbreviations

Abbreviation	Explanation
AA	Application Association
AARE	Application Association Response
AARQ	Application Association ReQuest
ACSE	Application Control Service Element
AL	Application Layer
ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
ASE	Application Service Element
ATS	Abstract Test Suite
A-XDR	Adapted Extended Data Representation
base_name	The short_name corresponding to the first attribute ("logical_name") of a COSEM object
CHAP	Challenge Handshake Authentication Protocol
Class_id	Interface class identification code
COSEM	Companion Specification for Energy Metering
COSEM object	An instance of an interface class
CTI	Conformance Test Information
CtoS	Client to Server challenge
CTT	Conformance Test Tool
DHCP	Dynamic Host Control Protocol
DLMS	Device Language Message Specification
DNS	Domain Name Server
EAP	Extensible Authentication Protocol
ETS	Executable Test Suite
GMT	Greenwich Mean Time
GPS	Global Positioning System
HLS	High Level Security
IANA	Internet Assigned Numbers Authority
IC	Interface Class
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Organization for Standardisation
ITU	International Telecommunication Union
IUT	Implementation under test
IPCP	Internet Protocol Control Protocol
LCP	Link Control Protocol
LLS	Low Level Security
LN	Logical Name
LSB	Least Significant Bit
m	mandatory, used in conjunction with attribute and method definitions

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Abbreviation	Explanation
MD5	Message Digest Algorithm 5
MSB	Most Significant Bit
NDM	Normal Disconnected Mode (HDLC)
NRM	Normal Response Mode (HDLC)
o	optional, used in conjunction with attribute and method definitions
OBIS	OBject Identification System
PAP	Password Authentication Protocol
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation eXtra Information for Testing
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
SAP	Service Access Point
SHA-1	Secure Hash Algorithm
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SN	Short Name
StoC	Server to Client Challenge
UTC	Universal Time Co-ordinated
<p>NOTE In this document, the Camel Notation is used, except for the name of the test groups. This is in line with the notation of the test tool.</p> <p>Example: MaximumInformationFieldLength.</p>	

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

2.3 Revision History

Versions kept within the DLMS User Association Working Group Conformance Testing.

Version	Date	Author	Status	Comment
2.0	2002-05-14	G. Kmethy	Released	Brought in line with Version 1.0 of the CTT.
4.1	2007-09-10	G. Kmethy	Released	In line with Green Book Edition 6.0 Implemented in CTT 2
5.0	2010-12-15	G. Kmethy	Released	In line with Green Book Edition 7.0

2.4 Main changes compared to DLMS UA 1001-3 Version 4.1

Item	Clause	Type of change	Description
1.		General	References updated to Green Book Edition 7.0.
2.		General	References to CTT V1.x have been removed.
3.	3.1	Technical	A sentence has been added, clarifying that in the case of MODE_E opening all HDLC test are marked as skipped.
4.	Table 15	Technical	The description of the test body has been amended.

3 The HDLC based data link layer test suite

3.1 Test strategy

This ATS defines tests for testing implementations of the Data link layer using HDLC protocol in DLMS/COSEM server devices.

The tests are performed using direct connection to the IUT over an electrical or optical port. The data link cannot be tested when the IUT uses Mode E defined in [8], because a finishing stable testing case at the end of test cases cannot be maintained. Therefore, all test cases are marked as “SKIPPED” in this case.

The test strategy follows the various modes of the HDLC layer. In each test case, appropriate HDLC frames are sent to the IUT and the responses are observed.

The test is performed using one HDLC connection. For details, see 3.6.

3.2 Capabilities supported / not supported

This ATS supports a subset of the capabilities specified in [2].

Table 1 – Capabilities supported / not supported

Capability	Supported	
	Yes	No
HDLC command and response frames		
I	x	
RR	x	
RNR		x
SNRM	x	
DISC	x	
UA	x	
DM	x	
FRMR	x	
UI		x
Addressing schemes		
One byte addressing	x	
Two-bytes addressing	x	
Four-bytes addressing	x	
CALLING Physical Device address		x
HDLC parameters		
Handling of MaxInfoFieldLength	x	
Handling of WindowSize	x	
WindowSize >1		x
Transporting long SDUs	x	

3.3 Test groups

3.3.1 Overview

Table 2 provides an overview of the test groups.

Table 2 – Test groups

Test group name	Test group purpose
HDLC_FRAME	To test the handling of the elements of the HDLC frames as well as time-outs
HDLC_ADDRESS	To test address management
HDLC_NDM2NRM	To test the mode change from Normal Disconnected Mode (NDM) to Normal Response Mode (NRM)
HDLC_INFO	To test information exchange in NRM
HDLC_NDMOP	To test the behaviour in NDM

These test groups are further split to a group of positive tests and negative tests as appropriate. Each positive and negative test group comprises test cases focusing on a particular aspect.

Each test case is complete in the sense that it is sufficient to enable a test verdict to be assigned unambiguously to each potentially observable test outcome (i.e. sequence of test events).

Each test case is independent in the sense that it is possible to execute the derived executable test case in isolation from other such test cases: the preamble moves the IUT to NRM or NDM as appropriate, and at the end of each test case, the IUT is moved back to NDM.

Test cases may consist of one or more subtests, which in turn may consist of one or more test events.

The verdict is given for each subtest of each test case. However, a test case is aborted as soon as a subtest fails.

The following tables provide an overview of each test group.

3.3.2 Test group HDLC_FRAME

The purpose of the test group HDLC_FRAME is to verify that the IUT handles the elements of HDLC frames properly. The time-outs are also tested in this test group.

The test cases in this test group are performed by sending SNRM frames and observing the response.

NOTE 1 Handling the address fields is tested in the test group HDLC_ADDRESS, see 3.3.3.

NOTE 2 Handling sequence numbers is tested in test group HDLC_INFO, see 3.3.5.

The positive tests are basic interconnection tests to verify that the HDLC based data link layer can be connected and disconnected and the IUT provides the expected responses.

Table 3 – Test group HDLC_FRAME_P: Positive tests

Test case reference	Test case name	Description
HDLC_FRAME_P1	Connection and disconnection of the HDLC layer	Table 11
HDLC_FRAME_P2	InterFrameTimeout	Table 12
HDLC_FRAME_P3	InactivityTimeout	Table 13

During the negative tests, various errors are introduced into the SNRM frame – except into the address fields – to verify that the IUT correctly checks all elements of the frame and discards erroneous frames or provides the correct responses as specified in [2] and [15].

Table 4 – Test group HDLC_FRAME_N: Negative tests

Test case reference	Test case name	Description
HDLC_FRAME_N1	HDLC frame flags missing	Table 14
HDLC_FRAME_N2	HDLC frame too short	Table 15
HDLC_FRAME_N3	Frame format type sub-field check	Table 16
HDLC_FRAME_N4	Frame length sub-field check	Table 17
HDLC_FRAME_N5	Control field check	Table 18
HDLC_FRAME_N7	HCS field check	Table 19
HDLC_FRAME_N8	FCS field check	Table 20

3.3.3 Test group HDLC_ADDRESS

The purpose of test group HDLC_ADDRESS is to verify that the IUT handles the address fields correctly; i.e. it responds only to frames with valid and correct addresses and that it discards frames with erroneous or inappropriate address fields. All test cases use SNRM frames.

An IUT may be able to handle more than one addressing scheme. The addressing schemes supported must be declared in the CTI. If two-bytes addressing or four-bytes addressing is supported, the lower HDLC address must be declared in the CTI. See also 3.4.

NOTE If the upper or lower HDLC address is $>7F_H$, then only four byte addressing is possible.

Table 5 – Test group HDLC_ADDRESS_P: Positive tests

Test case reference	Test case name	Description
HDLC_ADDRESS_P1	Correct addresses using the address structure(s) specified	Table 21

Table 6 – Test group HDLC_ADDRESS_N: Negative tests

Test case reference	Test case name	Description
HDLC_ADDRESS_N1	Two-bytes source address	Table 22
HDLC_ADDRESS_N4	Unknown destination addresses	Table 23
HDLC_ADDRESS_N6	One byte destination address when two or four bytes are expected	Table 24
HDLC_ADDRESS_N7	Three bytes or five bytes destination address	Table 25

3.3.4 Test group HDLC_NDM2NRM

The purpose of this test group is to verify that the IUT correctly handles the SNRM mode setting command and HDLC parameters.

Table 7 – Test group HDLC_NDM2NRM_P: Positive tests

Test case reference	Test case name	Description
HDLC_NDM2NRM_P1	MaximumInformationFieldLength parameter	Table 26
HDLC_NDM2NRM_P2	WindowSize parameter	Table 27

3.3.5 Test group HDLC_INFO

The purpose of this test group is to verify that the IUT is correctly handling information exchange using I frames and the handling of sequence numbering is correct. This is also a basic interconnection test.

Table 8 – Test group HDLC_INFO_P: Positive tests

Test case reference	Test case name	Description
HDLC_INFO_P1	I frame exchange	Table 28

Table 9 – Test group HDLC_INFO_N: Negative tests

Test case reference	Test case name	Description
HDLC_INFO_N1	Too long information field	Table 29
HDLC_INFO_N2	Wrong N(R) sequence number	Table 30
HDLC_INFO_N3	Wrong N(S) sequence number	Table 31

3.3.6 Test group HDLC_NDMOP

Table 10 – Test group HDLC_NDMOP_N: Negative tests

Test case reference	Test case name	Description
HDLC_NDMOP_N1	I frame in NDM	Table 32

3.4 Information for testing

For the parameterisation of the test cases, the following information shall be declared in the Conformance Test Information (CTI) file provided by the CTT:

- on the logical device level:
 - the value of the ServerSAP: mandatory, the ServerSAP of the relevant logical device;

NOTE The IUT may contain one or more logical devices. These are numbered in the CTI from 0 up to the required number.
- for application associations:
 - the value of the ClientSAP: mandatory, the ClientSAP of the relevant AA;

NOTE Each logical device may contain one or more AAs. These are numbered from 0 up to the required number.
- for the physical layer:
 - if the port is echoing or not: mandatory;
 - the OpeningMode: Mandatory, MODE_E, WAKEUP_MODE_E or DIRECT_HDLC;
 - the HDLCBaud: necessary if the opening mode is DIRECT_HDLC;
- for the HDLC based data link layer:
 - InactivityTimeout: mandatory, the value of the inactivity timeout as specified in [2];
 - InterFrameTimeout: mandatory, the value of inter-frame timeout as specified in [2];

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

- ResponseTimeout: mandatory, the value of the response-timeout as specified in [2];
- DISCToNDMTimeout: mandatory, specifies time needed by the IUT to go the NDM after having received a DISC frame;
- AddressingSchemes: mandatory, the addressing scheme(s) supported by the IUT: one or more of one-byte-addressing, two-bytes-addressing, four-bytes-addressing;
- ServerLowerMACAddress: optional, it is needed only if the IUT supports two-byte and/or four-bytes addressing. It specifies the physical address of the IUT.

3.5 Test options

To facilitate testing, the following option is provided:

- DoNotTest_LOG0_IS_MGMT_LOG_DEVICE: when selected, it is not tested that Logical device 0 is the Management Logical Device (ServerSap = 1) and that an Association with no security is the Public Association (ClientSAP = 0x10);

If the purpose of the test is to obtain a Certificate, this option has to be deselected (must not be selected).

3.6 General test conditions

Unless otherwise specified, in all frames sent by the CTT:

- the opening and closing flags are present;
 - the Frame format field is correct (Format type = 3, with correct frame length sub-field);
 - the address structure is as declared in the CTI. Unless otherwise specified, if more than one address structures are supported, the test is performed only with the shortest (e.g. one-byte or two-bytes);
 - the default addresses are the following:
 - the server upper HDLC address is the address of Logical Device 0 declared in the CTI;
 - the server lower HDLC address is the value specified, if any, in the CTI;
 - the client address is the address declared for the first Association declared in Logical device 0, being in line with the application context used for the test run;
- NOTE If the purpose of the test campaign is to obtain a Certificate, then Logical Device 0 shall be the Management Logical Device – Server_SAP = 0.
- the Poll bit of command frames is set to “1”;
 - if SNRM frame is sent without an information field: all HDLC parameters are default;
 - the FCS and the HCS fields are correct.

The “response received” condition is met if a response is received from the IUT before the expiry of the ResponseTimeout specified in [2] 8.4.5.5.1, 8.4.5.6.1 and declared in the CTI.

If the requirement is that the IUT shall not respond, special care has been taken – as far as possible – to ensure that there are no multiple reasons for this.

If the requirement is that a command frame is not actioned, it is verified – as far as possible – that the IUT did not change its state.

At the beginning and at the end of the test it is checked that the device is there.

After sending a DISC to the IUT to move the IUT to NDM state, the DISCToNDMTimeout value specified in the CTI is observed before a new frame can be sent.

3.7 Test cases

3.7.1 Test cases of test group HDLC_FRAME

Table 11 – HDLC_FRAME_P1: Connection and disconnection of the HDLC layer

Test case	HDLC_FRAME_P1: Connection and disconnection of the HDLC layer
References	[2] 8.4.3, 8.4.5.3.1, 8.4.5.3.3, 8.4.5.6.1, 8.4.5.6.3, [15] 5.5.3.3 a)
Test purpose	To verify if the HDLC data link layer can be put to NRM mode and back to NDM mode.
Expected result	The IUT actions the mode setting commands and sends appropriate responses.
Preamble	<p>Make sure that the IUT is in NDM.</p> <p>Send a DISC frame. Wait until a response is received or the ResponseTimeout expires.</p> <p>NOTE It is assumed, that the IUT sends a response within ResponseTimeout. If the ResponseTimeout expires, this may be for example because the meter is not connected or not responsive for any other reason. Then during the next exchange, we get a FAILED verdict.</p> <p>According to [2] 8.4.5.6.3, the data link layer shall be disconnected after InactivityTimeout expires. However, this may be very long, therefore, we proceed as soon as the response is received or after ResponseTimeout expires.</p> <p>This note applies to all similar preambles.</p>
Test body	<p><u>Subtest 1: Move the IUT to NRM</u></p> <p>Send an SNRM frame. Expect a UA frame with or without an information field, with the Final bit set to "1".</p> <p>NOTE The information field may contain one or more HDLC parameters. Handling of HDLC parameters is tested in test group HDLC_NDM2NRM, see Table 7.</p> <p>If there is no response, or the response frame is <> UA, or the Final bit is <> 1, the verdict is FAILED.</p>
	<p><u>Subtest 2: Check that the IUT is in NRM</u></p> <p>Send an RR frame. Expect an RR frame.</p> <p>If there is no response, or the frame received is <> RR, or the Final bit is <> 1, or N(R) is not 0, then the verdict is FAILED.</p>
	<p><u>Subtest 3: Move the IUT to NDM</u></p> <p>Send a DISC frame. Expect a UA frame with or without an information field.</p> <p>If there is no response, or the response frame is <> UA, then the verdict is FAILED.</p>
	<p><u>Subtest 4: Check that the IUT is in NDM</u></p> <p>Send a DISC frame. Expect a DM frame.</p> <p>If there is no response, or the response frame is <> DM, then the verdict is FAILED.</p>
Postamble	–
Comments	–

Table 12 – HDLC_FRAME_P2: InterFrameTimeout

Test case	HDLC_FRAME_P2: InterFrameTimeout
References	[2] 8.4.5.6.4
Test purpose	To verify that the IUT correctly implements the InterFrameTimeout mechanism.
Expected result	The IUT assumes the end of the frame after the expiry of the InterFrameTimeout.
Preamble	Put the IUT to NRM by sending an SNRM frame.
Test body	<p>Start from a good DISC frame, but strip the trailing flag.</p> <p>Send this frame to the IUT and wait for inter-frame time-out + 10%, then send an RR frame. Expect an RR response.</p> <p>If there is no response or the response frame is <> RR, the verdict is FAILED.</p>

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_FRAME_P2: InterFrameTimeout
Postamble	Move back the IUT to NDM by sending a DISC frame.
Comments	–

Table 13 – HDLC_FRAME_P3: InactivityTimeout

Test case	HDLC_FRAME_P3: InactivityTimeout
References	[2] 8.4.5.6.3
Test purpose	To verify that the IUT correctly implements the InactivityTimeout mechanism.
Expected result	The IUT disconnects the HDLC layer if no characters are received during the period defined by the InactivityTimeout specified in the CTI.
Preamble	Put the IUT to NRM by sending an SNRM frame.
Test body	Check that the IUT HDLC layer is disconnected after the expiry of the InactivityTimeout. Wait InactivityTimeout + 10%. Send a DISC frame. If there is no response or the response is <> DM, then the verdict is FAILED.
Postamble	–
Comments	–

Table 14 – HDLC_FRAME_N1: HDLC frame flags missing

Test case	HDLC_FRAME_N1: HDLC frame flags missing
References	[2] 8.4.1.2, 8.4.4.2.3, 8.4.5.6.1
Test purpose	To verify that the IUT is hunting for and recognising opening and closing flags.
Expected result	The IUT discards frames not properly bounded by opening and closing flags and does not action such frames.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received or the ResponseTimeout expires.
Test body	<u>Subtest 1: Opening flag missing</u> Start from a correct SNRM frame and strip the opening flag. Send this frame to the IUT. The IUT shall discard this frame and shall not respond. If a response is received, the verdict is FAILED.
	<u>Subtest 2: Closing flag missing</u> Start from a correct SNRM frame and strip the closing flag. Send this frame to the IUT. The IUT shall discard this frame and shall not respond. If a response is received, the verdict is FAILED.
	<u>Subtest 3: Both flags missing</u> Start from a correct SNRM frame and strip both flags. Send this frame to the IUT. The IUT shall discard this frame and shall not respond. If a response is received, the verdict is FAILED.
	<u>Subtest 4: Check that the IUT is in NDM</u> Send a DISC frame. Expect a DM frame. If there is no response, or the response frame is <> DM, then the verdict is FAILED.
Postamble	–
Comments	–

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Table 15 – HDLC_FRAME_N2: HDLC frame too short

Test case	HDLC_FRAME_N2: HDLC frame too short
References	[2] 8.4.4.1, 8.4.4.2.3, 8.4.5.6.1.
Test purpose	To verify that the IUT rejects SNRM frames, which are too short, i.e. they do not contain the necessary elements.
Expected result	The IUT discards and does not respond to frames, which are shorter than 7 octets between the flags with one byte addressing, shorter than 8 octets with two-bytes addressing or shorter than 10 octets with four-bytes addressing.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received or the ResponseTimeout expires.
Test body	Start from a correct SNRM frame but omit one, two, three bytes of the valid HDLC frame following the length byte. The opening and closing flags shall be present and the frame length sub-field shall be correct. The IUT shall discard and shall not respond to any incomplete frames. Before sending the next frame, wait until ResponseTimeout expires. If there is a response to any of such incomplete frames, the verdict is FAILED. Finally, send the complete SNRM frame. Expect a UA frame with or without an information field. If there is no response, or the response frame is <> UA, then the verdict is FAILED.
Postamble	Move back the IUT to NDM by sending a DISC frame.
Comments	–

Table 16 – HDLC_FRAME_N3: Frame format type sub-field check

Test case	HDLC_FRAME_N3: Frame format type sub-field check
References	[2] 8.4.1.3
Test purpose	To verify that the IUT checks the Format type sub-field of the Frame format field and responds only to frames of Frame format type 3 (binary 1010).
Expected result	The IUT discards any other frame types; it does not respond and does not action the mode setting command.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.
Test body	Start from a correct SNRM frame but insert a wrong value in the Format type sub-field. Send this frame to the IUT. The IUT shall discard this frame and shall not respond. If a response is received, the verdict is FAILED. Check that the IUT is in NDM. Send a DISC frame. Expect a DM frame. If there is no response, or the response frame is <> DM, then the verdict is FAILED.
Postamble	–
Comments	–

Table 17 – HDLC_FRAME_N4: Frame length sub-field check

Test case	HDLC_FRAME_N4: Frame length sub-field check
References	[2] 8.4.1.3
Test purpose	To verify that the IUT checks the Frame length sub-field of the Frame format field and only actions and responds to frames with correct value in this field.
Expected result	The IUT discards frames with incorrect length; it does not respond and does not action the mode setting command.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_FRAME_N4: Frame length sub-field check
Test body	<p>Start from a correct SNRM frame but insert a wrong value in the Frame length sub-field.</p> <p>Send this frame to the IUT. The IUT shall discard this frame and shall not respond.</p> <p>If a response is received, the verdict is FAILED.</p> <p>Check that the IUT is in NDM.</p> <p>Send a DISC frame. Expect a DM frame.</p> <p>If there is no response, or the response frame is <> DM, then the verdict is FAILED.</p>
Postamble	–
Comments	–

Table 18 – HDLC_FRAME_N5: Control field check

Test case	HDLC_FRAME_N5: Control field check
References	[2] 8.4.1.5, 8.4.3.1, 8.4.3.9 case 1, 8.4.5.5.4, [15] Clause 5.5.3.4.2, 5.6.4
Test purpose	To verify that the IUT correctly checks and interprets the command identifiers carried by the Control field.
Expected result	The IUT rejects commands, which are not defined or not implemented.
Preamble	Move the IUT to NRM by sending an SNRM frame.
Test body	<p><u>Subtest 1: Unknown command identifier</u></p> <p>Start from a correct SNRM frame but insert an unknown command identifier in the control field.</p> <p>Send this frame to the IUT. The IUT shall respond with an FRMR frame with or without an information field containing diagnostic information.</p> <p>If there is no response or the frame is <> FRMR, then the verdict is FAILED.</p> <p><u>Subtest 2: Check that the HDLC layer can be initialised</u></p> <p>Send an SNRM frame to check if the device is there. Expect a UA frame with or without an information field.</p> <p>If there is no response, or the response frame is <> UA, then the verdict is FAILED.</p>
Postamble	Move back the IUT to NDM by sending a DISC frame.
Comments	-

Table 19 – HDLC_FRAME_N7: HCS field check

Test case	HDLC_FRAME_N7: HCS field check
References	[2] 8.4.1.6, 8.4.5.5.2 [15] 4.2.6
Test purpose	<p>To verify that the IUT checks the HCS and discards frames with a wrong HCS and it takes no action.</p> <p>NOTE Frames that do not have an information field or have an empty information field, e.g. as with some supervisory frames, do not contain an HCS and FCS, only an FCS.</p>
Expected result	The IUT does not respond and does not change its operating status.
Preamble	<p>Make sure that the IUT is in NDM.</p> <p>Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.</p>
Test body	<p>Start from a good SNRM frame with an information field, but insert a wrong HCS.</p> <p>Send this frame to the IUT. The IUT shall discard this frame and shall not respond.</p> <p>If a response is received, the verdict is FAILED.</p> <p>Check that the IUT is in NDM.</p> <p>Send a DISC frame. Expect a DM frame.</p> <p>If there is no response, or the response frame is <> DM, then the verdict is FAILED.</p>
Postamble	–

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_FRAME_N7: HCS field check
Comments	-

Table 20 – HDLC_FRAME_N8: FCS field check

Test case	HDLC_FRAME_N8: FCS field check
References	[2] 8.4.1.8, 8.4.5.5.2
Test purpose	To verify that the IUT checks the FCS and discards frames with a wrong FCS and it takes no action. NOTE Frames that do not have an information field or have an empty information field, e.g., as with some supervisory frames, do not contain an HCS and FCS, only an FCS.
Expected result	The IUT does not respond and does not change its operating status.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.
Test body	Start from a good SNRM with an information field, but insert a wrong FCS. Send this frame to the IUT. The IUT shall discard this frame and shall not respond. If a response is received, the verdict is FAILED. Check that the IUT is in NDM. Send a DISC frame. Expect a DM frame. If there is no response, or the response frame is <> DM, then the verdict is FAILED.
Postamble	–
Comments	–

3.7.2 Test cases of test group HDLC_ADDRESS

Table 21 – HDLC_ADDRESS_P1: Correct addresses using the address structure(s) specified

Test case	HDLC_ADDRESS_P1: Correct addresses using the address structure(s) specified
References	[2] 8.4.2.2
Test purpose	To verify that the IUT can handle all addressing schemes specified in the CTI.
Expected result	The IUT can be moved to NRM and back the NDM and it sends correct responses.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.
Test body	<u>Subtest 1: One byte address</u> If one byte addressing is not supported, mark this test as "INAPPLICABLE". Send an SNRM frame including a one-byte destination address. Expect a UA frame with or without an information field. If there is no response, or the response frame is <> UA, then the verdict is FAILED. Send a DISC frame. Expect a UA frame with or without an information field. If there is no response, or the response frame is <> UA, then the verdict is FAILED.
	<u>Subtest 2: Two-bytes addressing</u> If two-bytes addressing is not supported, mark this test as "INAPPLICABLE". Send an SNRM frame including a two-bytes destination address. Expect a UA frame with or without an information field. If there is no response, or the response frame is <> UA, then the verdict is FAILED. Send a DISC frame. Expect a UA frame with or without an information field. If there is no response, or the response frame is <> UA, then the verdict is FAILED.

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_ADDRESS_P1: Correct addresses using the address structure(s) specified
	<p><u>Subtest 3: Four-bytes addressing</u></p> <p>If four-bytes addressing is not supported, mark this test as “INAPPLICABLE”.</p> <p>Send an SNRM frame including a four-bytes destination address.</p> <p>Expect a UA frame with or without an information field.</p> <p>If there is no response, or the response frame is <> UA, then the verdict is FAILED.</p> <p>Send a DISC frame. Expect a UA frame with or without an information field.</p> <p>If there is no response, or the response frame is <> UA, then the verdict is FAILED.</p>
Postamble	–
Comments	This is a basic capability test; to make sure that the error cases are testable.

Table 22 – HDLC_ADDRESS_N1: Two-bytes source address

Test case	HDLC_ADDRESS_N1: Two-bytes source address
References	[2] 8.4.2.5, first bullet point.
Test purpose	To verify that the IUT correctly checks the length of the source address of frames received.
Expected result	The IUT does not change its operating status and does not respond.
Preamble	<p>Make sure that the IUT is in NDM.</p> <p>Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.</p>
Test body	<p>Start from a correct SNRM with correct addresses, but the client address expressed on two bytes.</p> <p>Send this frame to the IUT. The IUT shall discard this frame and shall not respond.</p> <p>If a response is received, the verdict is FAILED.</p> <p>Check that the IUT is in NDM.</p> <p>Send a DISC frame with the correct addresses: client address expressed on one byte. Expect a DM frame.</p> <p>If there is no response, or the response frame is <> DM, then the verdict is FAILED.</p>
Postamble	–
Comments	–

Table 23 – HDLC_ADDRESS_N4: Unknown destination addresses

Test case	HDLC_ADDRESS_N4: Unknown destination addresses
References	[2] 8.4.2.3
Test purpose	To verify that the IUT correctly checks the value of the destination address fields.
Expected result	The IUT shall discard frames with unknown destination addresses and shall not respond.
Preamble	Move the IUT to NRM by sending an SNRM frame using default addresses.
Test body	<p><u>Subtest 1: Unknown destination address on one byte</u></p> <p>If the IUT does not support one-byte addressing, mark this test as “INAPPLICABLE”.</p> <p>Send an SNRM frame with one byte destination address. The address shall be unknown (taken from the reserved range). The IUT shall not respond.</p> <p>If there is a response, the verdict is FAILED.</p>
	<p><u>Subtest 2: Unknown destination addresses on two bytes</u></p> <p>If the IUT does not support two-bytes addressing, mark this test as “INAPPLICABLE”.</p> <p>Send an SNRM frame with two-bytes destination address, Both the upper and the lower HDLC address shall be unknown (taken from the reserved range). The IUT shall not respond.</p> <p>If there is a response, the verdict is FAILED.</p>

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_ADDRESS_N4: Unknown destination addresses
	<p><u>Subtest 3: Unknown destination addresses on four bytes</u></p> <p>If the IUT does not support four-bytes addressing, mark this test as “INAPPLICABLE”.</p> <p>Send an SNRM frame with four-bytes destination address, Both the upper and the lower HDLC address shall be unknown (taken from the reserved range). The IUT shall not respond.</p> <p>If there is a response, the verdict is FAILED.</p>
	<p><u>Subtest 4: Check that the IUT is still there.</u></p> <p>Send a DISC frame (with correct addresses) and expect a UA frame.</p> <p>If there is no response, or the response is <> UA, the verdict is FAILED.</p>
Postamble	–
Comments	–

Table 24 – HDLC_ADDRESS_N6: One byte destination address when two or four bytes are expected

Test case	HDLC_ADDRESS_N6: One byte destination address when two or four bytes are expected
References	[2] 8.4.2.2, 8.4.2.5 Table 7, case 1 and 2
Test purpose	To verify that the IUT correctly checks the destination addresses.
Expected result	The IUT discards frames with a one-byte address, when two-bytes or four-bytes HDLC addresses are expected.
Preamble	<p>If one byte addressing is supported, mark this test as “INAPPLICABLE”.</p> <p>Make sure that the IUT is in NDM.</p> <p>Send a DISC frame using two bytes addressing with the lower HDLC address declared in the CTI.</p> <p>Wait until a response is received, or the ResponseTimeout expires.</p>
Test body	<p>Send an SNRM frame with one byte destination address.</p> <p>The IUT shall not respond. If a response is received, the verdict is FAILED.</p> <p>Check that the IUT is in NDM.</p> <p>Send a DISC frame using default addresses, with the lower HDLC address declared in the CTI.</p> <p>Expect a DM frame. If there is no response, or the response frame is <> DM, then the verdict is FAILED.</p>
Postamble	–
Comments	–

Table 25 – HDLC_ADDRESS_N7: Three bytes or five bytes destination address

Test case	HDLC_ADDRESS_N7: Three bytes or five bytes destination address
References	[2] 8.4.2.1, 8.4.2.5, Table 7, case 7.
Test purpose	To verify that the IUT discards frames with an illegal address length.
Expected result	The IUT does not action frames with illegal address lengths and does not respond to them.
Preamble	<p>Make sure that the IUT is in NDM.</p> <p>Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.</p>
Test body	<p><u>Subtest 1: Illegal address length – three bytes</u></p> <p>Start from a correct SNRM, but with a three-bytes address.</p> <p>Send this frame to the IUT. The IUT shall discard this frame and shall not respond.</p> <p>If a response is received, the verdict is FAILED.</p> <p>Check that the device is still there.</p>

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_ADDRESS_N7: Three bytes or five bytes destination address
	Send a DISC frame and expect a UA or DM frame. If there is no response, the verdict is FAILED.
	<u>Subtest 2: Illegal address length – five bytes</u> Start from a correct SNRM, but with a five-bytes address. Send this frame to the IUT. The IUT shall discard this frame and shall not respond. If a response is received, the verdict is FAILED. Check that the device is still there. Send a DISC frame and expect a UA or DM frame. If there is no response, the verdict is FAILED.
Postamble	–
Comments	–

3.7.3 Test cases of test group HDLC_NDM2NRM

Table 26 – HDLC_NDM2NRM_P1: MaximumInformationFieldLength parameter

Test case	HDLC_NDM2NRM_R1: MaximumInformationFieldLength parameter
References	[2] 8.4.5.3.2
Test purpose	To verify that the IUT correctly handles the MaximumInformationFieldLength parameter. NOTE The ServerMaximumInformationFieldLengthTransmit parameter is the same as the ClientMaximumInformationFieldLengthReceive.
Expected result	The IUT shall answer with a UA frame. This shall include the MaximumInformationFieldLength parameter. Default values do not have to be present.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.
Test body	Send an SNRM frame with the information field containing the MaximumInformationFieldLength -transmit and -receive parameters set to their maximum possible values (2030 bytes). If there is no response, or a response is received, but the frame is <> UA or if the value of the parameter returned is >2030, the verdict is FAILED.
Postamble	Move back the IUT to NDM by sending a DISC frame.
Comments	–

Table 27 – HDLC_NDM2NRM_P2: WindowSize parameter

Test case	HDLC_NDM2NRM_P2: WindowSize parameter
References	[2] 8.4.5.3.2
Test purpose	To verify that the IUT correctly handles the window size parameter and that the encoding is correct.
Expected result	The IUT shall answer with a UA frame. This shall include the window size parameter. Default values do not have to be present.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.
Test body	NOTE This is for the purpose of this test only, the CTT supports only window size = 1. Send an SNRM frame with the information field containing with window-size -transmit and -receive = 4. If there is no response, or a response is received, but the frame type <> UA or if the window size parameter value is not encoded on four bytes, the verdict is FAILED.

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Test case	HDLC_NDM2NRM_P2: WindowSize parameter
Postamble	Move back the IUT to NDM by sending a DISC frame.
Comments	–

3.7.4 Test cases of test group HDLC_INFO

Table 28 – HDLC_INFO_P1: I frame exchange

Test case	HDLC_NDM2NRM_P2: I frame exchange
References	[2] 8.4.5.4.4
Test purpose	To verify that the IUT is able to handle HDLC level segmentation
Expected result	The IUT handles segmenting correctly.
Preamble	–
Test body	<p><u>Subtest 1: Send an I frame including an AARQ</u></p> <p>Move the IUT to NRM by sending an SNRM frame.</p> <p>Send an I frame containing a correct AARQ APDU. Expect an I frame response.</p> <p>If there is no response or the response is not an I frame, the verdict is FAILED.</p> <p>Close the association which may have been built and disconnect the data link layer by sending a DISC.</p>
	<p><u>Subtest 2: Send an I frame including an AARQ APDU in segments</u></p> <p>Move the IUT to NRM by sending an SNRM frame.</p> <p>Send an I frame containing a correct AARQ APDU in several segments with Seg = TRUE and Poll = TRUE and monitor response. For each segment, the IUT shall respond with an RR frame with correct sequence numbering. Send the last segment with Seg = FALSE.</p> <p>If, following the last segment, there is no response or the response is not an I frame, the verdict is FAILED.</p> <p>Close the association which may have been built and disconnect the data link layer by sending a DISC.</p>
Postamble	–
Comments	–

Table 29 – HDLC_INFO_N1: Too long information field

Test case	HDLC_INFO_N1: Too long information field
References	[2] 8.4.3.9 case 2
Test purpose	To verify that the IUT correctly handles a situation when an I frame is received with an information field exceeding the MaximumInformationFieldLength that can be accommodated by the secondary station.
Expected result	The IUT responds with an FRMR frame. If the IUT does not respond, it is also acceptable.
Preamble	<p>Move the IUT in NRM by sending an SNRM.</p> <p>The value of MaximumInformationFieldLength receive of the IUT is extracted from the UA response. If it is >= 2030, then the test is marked as "INAPPLICABLE".</p> <p>NOTE In this case HDLC_NDM2NRM_P1 will fail.</p>
Test body	<p>Send an I frame with an information field exceeding the MaximumInformationFieldLength that can be accommodated by the secondary station. Expect an FRMR frame with or without information field.</p> <p>If there is a response, but the frame type is <> FRMR, the verdict is FAILED. Check if the IUT HDLC layer still can be initialised.</p> <p>Send an SNRM frame and expect a UA frame. If there is no response or the response frame is <> UA, the verdict is FAILED.</p>
Postamble	Move the IUT to NDM by sending a DISC.
Comments	–

DLMS/COSEM conformance testing – Conformance test plans –
Data link layer using HDLC protocol

Table 30 – HDLC_INFO_N2: Wrong N(R) sequence number

Test case	HDLC_INFO_N2: Wrong N(R) sequence number
References	[2] 8.4.3.9 case 3
Test purpose	To verify that the IUT is correctly checking sequence numbers and responds with an FRMR frame if an RR frame with an invalid N(R) is received from the primary station.
Expected result	The IUT responds with an FRMR frame.
Preamble	Move the IUT in NRM by sending an SNRM.
Test body	Send an RR frame but with N(R) = 1 instead of 0. Expect an FRMR frame with or without information field. If there is no response or if the response frame is <> FRMR, the verdict is FAILED. Check if the IUT HDLC layer still can be initialised. Send an SNRM frame and expect a UA frame. If there is no response or the response frame is <> UA, the verdict is FAILED.
Postamble	Move the IUT to NDM by sending a DISC.
Comments	–

Table 31 – HDLC_INFO_N3: Wrong N(S) sequence number

Test case	HDLC_INFO_N: Wrong N(S) sequence number
References	[2] 8.4.3.4, [15] 5.4.3.3.2, 5.6.2
Test purpose	To verify that the IUT is correctly checking sequence numbers.
Expected result	The IUT asks for the “lost” I frame.
Preamble	Move the IUT in NRM by sending an SNRM.
Test body	Send an I frame containing an AARQ, but with N(S) = 1 in the first segment. Expect an RR frame with N(R) = 0. If the frame received is <> RR or if the N(R) is not 0, the verdict is FAILED.
Postamble	Move the IUT to NDM by sending a DISC.
Comments	–

3.7.5 Test cases of test group HDLC_NDMOP

Table 32 – HDLC_NDMOP_N1: I frame in NDM

Test case	HDLC_NDMOP_N1: I frame in NDM
References	[2] 8.4.3.8
Test purpose	To verify that the IUT discards I frames received in the NDM mode.
Expected result	The IUT does not respond.
Preamble	Make sure that the IUT is in NDM. Send a DISC frame. Wait until a response is received, or the ResponseTimeout expires.
Test body	Send an I frame containing an AARQ. Expect a DM frame or no response. If there is a response but the frame received <> DM, the verdict is FAILED. Check NDM state: Send a DISC frame. Expect a DM frame. If no response is received or the frame received is <> DM, the verdict is FAILED.
Postamble	–
Comments	–

DLMS/COSEM conformance testing – Conformance test plans – Data link layer using HDLC protocol

INDEX

AARE, 6
 AARQ, 6
 ACSE, 6
 Address management, 10
 Address structure, 13
 APDU, 6
 Application Association, 6, 12
 Application context, 13
 ASE, 6
 A-XDR, 6
 base_name, 6
 Certificate, 13
 CHAP, 6
 Class_id, 6
 ClientSAP, 12
 Companion Specification, 6
 Conformance Test Information, 12
 Connection and disconnection of the HDLC layer,
 11, 14
 Control field check, 11, 17
 Correct addresses using the address structure(s)
 specified, 11, 18
 COSEM, 6
 DHCP, 6
 DISCToNDMTimeout, 13
 DLMS, 6
 DNS, 6
 EAP, 6
 Elements of the HDLC frames, 10
 Extended, 6
 FCS field check, 11, 18
 Flag, 13
 Frame format type sub-field check, 11, 16
 Frame length sub-field check, 11, 16
 Global Positioning System, 6
 GMT, 6
 GPS, 6
 HCS field check, 11, 17
 HDLC frame flags missing, 11, 15
 HDLC frame too short, 11, 16
 HDLC_ADDRESS_N, 11
 HDLC_ADDRESS_P, 11
 HDLC_FRAME_N, 11
 HDLC_FRAME_P, 11
 HDLC_INFO_N, 12
 HDLC_INFO_P, 12
 HDLC_NDM2NRM_P, 12
 HDLC_NDMOP_N, 12
 HDLCBaud, 12
 High Level Security, 6
 HLS, 6
 I frame exchange, 12, 22
 I frame in NDM, 23
 I frames sent by the primary station, 12
 IANA, 6
 IC, 6
 IETF, 6
 InactivityTimeout, 11, 12, 15
 Information exchange, 10
 Interface class, 6
 InterFrameTimeout, 11, 12, 14
 IP, 6
 IPCP, 6
 LCP, 6
 LLS, 6
 LN, 6
 Logical device, 12
 Logical device 0, 13
 Logical name, 6
 Low Level Security, 6
 LSB, 6
 Management Logical Device, 13
 MaximumInformationFieldLength parameter, 12, 21
 MD5, 7
 Message Digest Algorithm 5, 7
 MSB, 7
 Normal Disconnected Mode, 10
 Normal Response Mode, 10
 OBIS, 7
 One byte destination address when two or four bytes
 are expected, 11, 20
 OpeningMode, 12
 PAP, 7
 Parameterisation, 12
 PDU, 7
 Physical layer, 12
 Poll bit, 13
 PPP, 7
 Protocol Data Unit, 6
 PSTN, 7
 Public Switched Telephone Network, 7
 Response received, 13
 ResponseTimeout, 12, 13
 SAP, 7
 Secure Hash Algorithm, 7
 ServeLowerMACAddress, 13
 Server to Client Challenge, 7
 ServerSAP, 12
 Service Access Point, 7
 Service Element, 6
 SHA-1, 7
 Short Message Service, 7
 Short name, 7
 SMS, 7
 SMTP, 7
 SN, 7
 StoC, 7
 Test conditions, 13
 Test group HDLC_ADDRESS, 11, 18
 Test group HDLC_FRAME, 10, 14
 Test group HDLC_INFO, 12, 22
 Test group HDLC_NDM2NRM, 11, 21
 Test group HDLC_NDMOP, 12, 23
 Test groups, 10
 Test options, 13
 Test strategy, 9
 Three bytes or five bytes destination address, 11, 20
 Time-out, 10
 Too long information field, 12, 22
 Two-bytes source address, 11, 19
 Unknown destination addresses, 11, 19
 Verdict, 10
 WindowSize parameter, 12, 21
 Wrong N(R) sequence number, 12, 23
 Wrong N(S) sequence number, 12, 23

Project: DLMS/COSEM conformance testing –
Abstract Test Plans for CTT 3.0 –
DLMS/COSEM application layer –
COSEM interface objects –
Symmetric key security suite 0

Author:	DLMS User Association – WG Maintenance
Version:	V 1.3
Status:	Released
Revision Date:	2015-06-18
Copyright:	© Copyright 2000-2015 DLMS UA
Classification:	DLMS User Association use only
Filename:	ATS_AL_COSEM_SYMSEC_0_V1.3_150618.docx
Replace Doc:	ATS_AL_SYMSEC_0_COSEM_V1.1_Beta2_150428.docx
Comment:	In line with: <ul style="list-style-type: none">- Green Book Edition Edition 7.0 Amendment 3; and- Blue Book Edition 11.0.

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	1/76
-----------------------	------------	----------------------	------

CONTENTS

Foreword	5
Revision history	5
1 Scope.....	6
2 Referenced documents.....	6
3 Definitions abbreviations and notation	6
3.1 Definitions	6
3.2 Abbreviations	6
3.3 Notation	7
4 The conformance assessment process	7
5 Procedures.....	7
6 Abstract Test Suite DLMS/COSEM application layer.....	14
6.1 Capabilities supported / not supported	14
6.2 Test cases.....	15
6.2.1 Test group APPL_IDLE.....	15
6.2.2 Test group APPL_OPEN.....	15
6.2.3 Test group APPL_DATA: xDLMS data services	25
6.2.4 Test group APPL_REL.....	29
7 Abstract Test Suite COSEM objects	30
7.1 Interface classes supported.....	30
7.2 Test algorithm	32
7.3 Interface class specific tests.....	34
7.3.1 Data (class_id = 1) and Register (class_id = 3).....	34
7.3.2 Extended register (class_id = 4)	34
7.3.3 Demand register (class_id = 5)	34
7.3.4 Profile generic (class_id = 7)	34
7.3.4.1 General.....	34
7.3.4.2 Availability of selective access	34
7.3.4.3 Selective access by range.....	35
7.3.4.4 Selective access by entry.....	35
7.3.5 PushTimeout Script table (class_id = 9).....	36
7.3.6 Association SN (class_id = 12) and Association LN (class_id = 15)	36
7.3.7 Register monitor (class_id = 21)	36
7.3.8 Push setup (class_id= 40)	36
7.3.8.1 General.....	36
7.3.8.2 Test scenario	36
7.3.8.3 Parameterisation of the “Push setup” and “Security setup” objects	36
7.3.8.4 The push test process.....	37
7.3.9 Security setup (class_id = 64).....	38
7.3.10 Dummy attributes	38

7.3.11	Multiple references test	39
7.3.12	Check mandatory objects	39
7.3.13	Interpretation of and reporting some attributes	41
8	Abstract Test Suite for symmetric key security suite 0: SYMSEC_0	43
8.1	Requirements for the IUT for testing security	43
8.2	Capabilities supported / not supported	43
8.3	Security personalisation	44
8.4	Overview of the SYMSEC_0 test cases	44
8.5	Test cases	45
8.5.1	Test group SYMSEC_0_BasicCap: Basic capability test	45
8.5.2	Test group SYMSEC_0_FraCount: Frame counter	46
8.5.3	Test group SYMSEC_0_GlobalKeyTx: Global key transfer	47
8.5.4	Test group SYMSEC_0_DedKey_N1: Dedicated-key transfer	51
8.5.5	Test group SYMSEC_0_SecDataX: Secure data exchange	52
8.5.6	Test group SYMSEC_0_SecRel: Secure AA release	55
8.5.7	Test group SYMSEC_0_SecPol: Security policy	56
Annex A	(informative) Conformance Test Information (CTI) template	59
Table 1	– Make sure that the IUT AL is in the IDLE state	8
Table 2	– Establish a confirmed AA with the parameters declared	8
Table 3	– Check that the AA is in the Associated state	11
Table 4	– Release AA	11
Table 5	– Read attribute	12
Table 6	– Write attribute	12
Table 7	– Invoke method	13
Table 8	– Raise fatal failure	13
Table 9	– Capabilities supported / not supported	14
Table 10	– APPL_IDLE_N1: Data exchange in IDLE state	15
Table 11	– APPL_OPEN_1: Establish an AA with the parameters declared	16
Table 12	– APPL_OPEN_2: Client user identification	16
Table 13	– APPL_OPEN_3, HLS authentication, Pass 3 and Pass 4	17
Table 14	– APPL_OPEN_4: Protocol version	18
Table 15	– APPL_OPEN_5: Application context	18
Table 16	– APPL_OPEN_6: Titles, qualifiers and invocation identifiers	19
Table 17	– APPL_OPEN_7: Authentication functional unit	21
Table 18	– APPL_OPEN_9: xDLMS InitiateRequest: dedicated-key	23
Table 19	– APPL_OPEN_11: xDLMS InitiateRequest: quality-of-service	23
Table 20	– APPL_OPEN_12: xDLMS InitiateRequest: dlms-version-number	24
Table 21	– APPL_OPEN_13: xDLMS InitiateRequest: conformance-block	24
Table 22	– APPL_OPEN_14: xDLMS InitiateRequest: client-max-receive-pdu-size	25
Table 23	– APPL_DATA_LN_N1: Get-Request with errors	25

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	3/76
-----------------------	------------	----------------------	------

Table 24 – APPL_DATA_LN_N3: Set-Request with errors	26
Table 25 – APPL_DATA_LN_N4: Unsupported service	27
Table 26 – APPL_DATA_SN_N1: ReadRequest with errors	27
Table 27 – APPL_DATA_SN_N2: WriteRequest with errors	28
Table 28 – APPL_DATA_SN_N3: Unsupported service	28
Table 29 – APPL_REL_P1	29
Table 30 – Interface classes supported.....	30
Table 31 – Push operation test	37
Table 32 – COSEM mandatory objects.....	39
Table 33 – Attributes to be interpreted	42
Table 34 – Capabilities supported	43
Table 36 – SYMSEC_0_BasicCap_1: Basic security capability test.....	45
Table 37 – SYMSEC_0_FraCount_1: Message replay protection	46
Table 38 – SYMSEC_0_FraCount_3: Send frame counter.....	47
Table 39 – SYMSEC_0_Key_Tx_P1: Transfer and restore GUEK	47
Table 40 – SYMSEC_0_Key_Tx_P2: Transfer and restore GAK.....	48
Table 41 – SYMSEC_0_Key_Tx_P3: Transfer and restore GUEK and GAK	49
Table 42 – SYMSEC_0_Key_Tx_N1: Global key transfer, wrong key_id.....	49
Table 43 – SYMSEC_0_Key_Tx_N2: GUEK transfer, wrong wrapping	50
Table 44 – SYMSEC_0_DedKey_N1: Dedicated-key negative tests	51
Table 45 – SYMSEC_0_SecDataX_P1: Write and read STA1 and STA2 using global and dedicated ciphering.....	52
Table 46 – SYMSEC_0_SecDataX_N1: Write and read STA1 using incorrect ciphering	54
Table 47 – SYMSEC_REL_N1: Release an AA using ciphered application context with insufficiently protected RLRQ.....	55
Table 48 – SYMSEC_0_SecPol_1: Activate security policy (1).....	56
Table 49 – SYMSEC_0_SecPol_2: Activate security policy (2).....	57
Table 50 – SYMSEC_0_SecPol_3: Activate security policy (3).....	58

Foreword

Copyright

© Copyright 1997-2015 DLMS User Association.

This document is confidential. It may not be copied, nor handed over to persons outside the standardisation environment.

The copyright is enforced by national and international law. The "Berne Convention for the Protection of Literary and Artistic Works", which is signed by 166 countries world-wide and other treaties apply.

Revision history

Versions kept within the DLMS User Association Working Group Conformance Testing.

Version	Date	Author	Status	Comment
V 1.0	2014-12-01	G. Kmethy V. Victoria	For review	Released with CTT3 beta version.
V 1.1	2015-04-28	G. Kmethy C. Aymon	For review	Released with CTT3 beta version 2
V 1.2	2015-05-28	G. Kmethy V. Victoria	With changes	Finalized based on the results of beta version 2. Clauses 7 and 8 swapped to follow the order of running the test suites.
V 1.3	2015-06-18	G. Kmethy V. Victoria	For review	Push setup test plan aligned with the CTT implementation. <ul style="list-style-type: none">- Subtest 1 is without protection forced and Subtest 2 is with protection forced. Subtest 2 is not applicable if security policy is not writeable.- "AA filter" line added to all test cases;- Push test assumptions updated, internal procedures specified and called.

1 Scope

This document specifies Abstract Test Suites (ATs) for testing the DLMS/COSEM application Layer, the COSEM interface objects and for Symmetric Key Security Suite 0 (SYMSEC_0).

This document is in line with Green Book Edition 7 and its Amendment 3 [2_7_3] and Blue Book Edition 11 [1_11] and – in relation to the “G3-PLC setup” interface classes – Blue Book Edition 12 [1_12].

It is implemented in the DLMS UA Conformance Test Tool Version 3: CTT 3.0.

2 Referenced documents

[1_11]	DLMS UA 1000-1:2013, Ed. 11.0: <i>Blue Book</i>
[1_12]	DLMS UA 1000-1:2013, Ed. 12.0: <i>Blue Book</i>
[2_7_3]	DLMS UA 1001-7, DLMS UA 1000-2:2009, <i>Green Book</i> Edition 7: 2009 and Amendment 3:2013
[3]	Object definition tables
[4_5]	DLMS UA 1001-1:2015, Ed. 5.0, <i>Yellow Book</i>

3 Definitions abbreviations and notation

3.1 Definitions

See [4_5]

3.2 Abbreviations

Abbreviation	Explanation
AA	Application Association
AARE	A-Associate Response – an APDU of the ACSE
AARQ	A-Associate Request – an APDU of the ACSE
A E A+E	Authentication Encryption Authentication and encryption
AFU	Authentication Functional Unit (of the AARQ / AARE APDUs)
ATS	Abstract Test Suite
CtoS	Client's challenge to the server during HLS authentication
CTT	Conformance Test Tool
DEK	Dedicated key
DLMS	Device Language Message Specification
COSEM	Companion Specification for Energy Metering
GAK	Global Authentication Key
GUEK	Global Unicast Encryption Key
HDLC	High-level Data Link Control
HLS	High Level Security authentication
HLS_MD5	HLS using MD5, authentication mechanism_id (3)
HLS_SHA-1	HLS using SHA-1, authentication mechanism_id (4)

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	6/76
-----------------------	------------	----------------------	------

HLS_GMAC	HLS using GMAC, authentication mechanism_id (5)
IUT	Implementation Under Test
LD	Logical Device
LDN	Logical Device Name
LLS	Low Level Security authentication, authentication mechanism_id (1)
RLRE	A-Release Response – an ACSE APDU
RLRQ	A-Release Request – an ACSE APDU
SAP	Service Access Point
SH	Security Header
StoC	Server's challenge to the client during HLS authentication
W7	Windows 7 Operating System
W8.1	Windows 8.1 Operating System

3.3 Notation

In the Abstract Test Suites, the following notation is used:

- elements of the COSEM ASN.1 syntax are written in bold;
- to name elements of COSEM APDUs a dotted notation is used, where the elements of the APDU are separated by a dot: e.g. **AARQ.application-context-name** means the AARQ application-context-name field;
- the security services and the keys used are listed in curly brackets. For example {A+E, GUEK, GAK} means that the security services are authentication and encryption, using the GUEK and the GAK;
- COSEM object names are in quotation marks e.g. "Data", "Profile generic";
- COSEM object attribute and method names are separated from the interface class name by a dot and are written in **bold** e.g. "Data".**value**, "Association LN".**object_list**.

4 The conformance assessment process

See [4_5] Clause 7.

5 Procedures

This clause contains the description of the procedures used in the ATSs and cross-referenced from the Abstract Test Cases.

NOTE The specification of these procedures allows presenting the test plans in a concise and coherent way.

Any failure in a procedure causes the (sub)test that called the procedure to fail with the reason of the failure of the procedure.

Example: if a subtest calls the procedure **Read attribute** and it fails with access-denied then the verdict for the (sub)test is FAILED, read access-denied.

The failures of the following procedures are fatal failures and will be handled as described in Table 8:

- **Establish a confirmed AA with the parameters declared;** and
- **Release AA**

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	7/76
-----------------------	------------	----------------------	------

In some cases the failure of a procedure is the expected behaviour. In these cases the failure of the procedure will be ignored. These cases are noted in the test cases with "FAILED if the procedure succeeds".

Table 1 – Make sure that the IUT AL is in the IDLE state

Procedure	Make sure that the IUT AL is in the IDLE state
References	[2_7_3] 9.3.3, 9.4.1, 9.4.5.
Prerequisites	–
Expected result	The IUT AL is in the IDLE state.
	If the supporting layer is connected, disconnect it.
Comments	

Table 2 – Establish a confirmed AA with the parameters declared

Procedure	Establish a confirmed AA with the parameters declared
References	[1_11] 4.4.2, 4.4.3, 4.4.4, 4.4.5. [2_7_3] 9.2.3, 9.2.4, 9.3.2, 9.4.2.2.2, 9.4.2.2.3, 9.4.4, 9.4.6.1, 9.5, 11.
Prerequisites	For each AA, the application context, the authentication mechanism, the xDLMS context, the security context and security material are declared. In the case when the application context is ciphered or the authentication mechanism is HLS_GMAC the IUT is provisioned with the GUEK and GAK and these keys are declared. If the client user identification mechanism is used, then one user – the test user – is declared and this user is configured in the ITU. NOTE The "Association SN / LN". add_user / remove_user methods are not tested. In the case when the authentication mechanism is LLS, HLS_MD5 or HLS_SHA-1 the secret is declared. In the case when the authentication mechanism is HLS_GMAC the system title of the IUT is declared.
Expected result	The IUT establishes the AA proposed and sends AARE with appropriate information.
	<p>Pass 1: Send AARQ with the parameters of the AA as declared with AARQ.protocol-version absent</p> <p><u>If the AA requires client user identification:</u></p> <ul style="list-style-type: none"> AARQ.calling-AE-invocation-id carries the client user_id declared. <p><u>If the authentication mechanism is Lowest Level Security (no security):</u></p> <ul style="list-style-type: none"> the Authentication Functional Unit (AFU) is absent. <p><u>If the authentication mechanism is Low Level Security (LLS):</u> The AFU is present with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and indicating authentication; AARQ.mechanism-name is LLS; AARQ.calling-authentication-value is the password declared. <p><u>If the authentication mechanism is High Level Security (HLS):</u></p> <ul style="list-style-type: none"> AARQ.calling-AP-title is absent in the case of HLS_MD5 and HLS_SHA-1 (unless the application context is with ciphering) and it is present and carries the CTT system title (CTT00000) in the case of HLS_GMAC; The AFU is present with:

Procedure	Establish a confirmed AA with the parameters declared
	<ul style="list-style-type: none"> – AARQ.sender-acse-requirements present and indicating authentication; – AARQ.mechanism-name as declared for the given AA; – AARQ.calling-authentication-value carries an appropriate challenge CtoS. <p><u>If the application context is no ciphering:</u></p> <ul style="list-style-type: none"> • AARQ.calling-AP-title is absent (unless HLS_GMAC authentication is used); • AARQ.user-information contains InitiateRequest without ciphering; <ul style="list-style-type: none"> • InitiateRequest.dedicated-key is absent; • InitiateRequest.response-allowed is absent (default TRUE); • InitiateRequest.quality-of-service is absent; • InitiateRequest.proposed-dlms-version-number is present with value = 6; • InitiateRequest.proposed-conformance contains all possible services and capabilities applicable in the given application context declared; • InitiateRequest.client-max-receive-pdu-size is 0xFFFF. <p><u>If the application context is with ciphering:</u></p> <ul style="list-style-type: none"> • AARQ.calling-AP-title carries the system title of the CTT (CTT00000); • AARQ.user-information contains a correctly ciphered InitiateRequest using {A+E, GUEK, GAK}. <ul style="list-style-type: none"> • InitiateRequest.dedicated-key contains a dedicated-key when its use is declared; • the other elements of InitiateRequest are the same as in the case of an application context without ciphering. <p>Save the unciphered InitiateRequest. This information is used in the RLRQ when releasing the AA takes place using a RLRQ / RLRE exchange with the user-information field ciphered.</p> <p><u>Pass 2: Expect AARE. Check the fields of AARE received</u></p> <ul style="list-style-type: none"> • if AARE.protocol-version is present, then its value shall be {version 1 (0)}; • AARE.application-context-name shall carry the application context name proposed; • AARE.result shall be accepted; • AARE.result-source-diagnostic shall be: <ul style="list-style-type: none"> – acse-service-user.null (0) in the case of no-security or LLS authentication; or – acse-service-user.authentication-required (14) in the case of HLS authentication; • AARE.responding-AP-title: <ul style="list-style-type: none"> – if the application context is with ciphering or in the case of HLS_GMAC it shall be present and it shall carry the IUT system title which shall match the CTI declaration; – it shall be absent otherwise. <p><u>If the authentication-mechanism is Lowest Level Security (no security):</u></p> <p>The AFU shall be absent with:</p> <ul style="list-style-type: none"> • AARE.responder-acse-requirements absent; • AARE.mechanism-name absent; and • AARE.responding-authentication-value absent. <p><u>If the authentication-mechanism is Low Level Security (LLS):</u></p> <p>The AFU may be absent with:</p> <ul style="list-style-type: none"> • AARE.responder-acse-requirements absent; • AARE.mechanism-name absent; and

CTT 3.0: ATS DLMS/COSEM Application layer – ATS COSEM interface objects –
ATS SYMSEC_0

Procedure	Establish a confirmed AA with the parameters declared
	<ul style="list-style-type: none"> • AARE.responding-authentication-value absent. <p>However, if the AFU is present then:</p> <ul style="list-style-type: none"> • AARE.responder-acse-requirements shall be present and shall indicate authentication; • AARE.mechanism-name shall be present and shall be equal to the one proposed; • AARE.responding-authentication-value shall be absent. <p><u>If the mechanism name is HLS:</u> The AFU shall be present and:</p> <ul style="list-style-type: none"> • AARE.responder-acse-requirements shall be present and shall indicate authentication; • AARE.mechanism-name shall be present and shall be equal to the one proposed; • AARE.responding-authentication-value shall be present and shall carry the challenge StoC, a string of length 8 to 64 octets, which shall be different from CtoS. <p><u>If the application context is no ciphering:</u></p> <ul style="list-style-type: none"> • AARE.user-information shall contain InitiateResponse without ciphering. <p><u>If the application context is with ciphering:</u></p> <ul style="list-style-type: none"> • AARE.user-information shall contain a correctly ciphered InitiateResponse using {A+E, GUEK, GAK}. <p><u>Check the fields of InitiateResponse:</u></p> <ul style="list-style-type: none"> • ignore InitiateResponse.negotiated-quality-of-service; • check that InitiateResponse.negotiated-dlms-version-number = 6; • check that InitiateResponse.negotiated-conformance is equal to the logical AND between the conformance block proposed and the conformance block declared for that AA; • check that InitiateResponse.server-max-receive-pdu-size is ≥ 12; • check that InitiateResponse.vaa-name is 0x0007 with LN referencing and 0xFA00 with SN referencing. <p>If the authentication mechanism is Lowest Level Security (no security) or Low Level Security (LLS) the procedure ends here.</p> <p><u>If the authentication mechanism is HLS proceed to Pass 3: Process StoC as requested by the authentication mechanism to obtain f(StoC), using the HLS secret declared:</u></p> <ul style="list-style-type: none"> • in the case of HLS_MD5 or HLS_SHA-1 as specified in [2_7_3] 9.2.3.4: $f(\text{StoC}) = \text{MD5}(\text{StoC} \parallel \text{HLS secret}) \text{ in the case of authentication mechanism (3)}$ $f(\text{StoC}) = \text{SHA-1}(\text{StoC} \parallel \text{HLS secret}) \text{ in the case of authentication mechanism (4)}$ • in the case of HLS_GMAC, (authentication mechanism (5)) as specified in [2_7_3] 9.2.4.8.4 using GUEK and GAK; $f(\text{StoC}) = \text{SC} \parallel \text{FC} \parallel \text{T} = \text{SC} \parallel \text{FC} \parallel \text{GMAC}(\text{SC} \parallel \text{AK} \parallel \text{StoC})$ <p>Do Invoke method current "Association SN" / "Association LN".reply_to_HLS_authentication with method invocation parameter f(StoC) and using {A+E, GUEK, GAK}.</p> <p><u>Pass 4:</u> Expect that the IUT returns f (CtoS). Check that f(CtoS) is correct.</p>
Comments	

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	10/76
-----------------------	------------	----------------------	-------

Table 3 – Check that the AA is in the Associated state

Procedure	Check that the AA is in the Associated state
References	[1_11] 4.4.3, 4.4.4. [2_7_3] 9.3.2, 9.4.4.
Prerequisites	The current “Association SN / LN”. logical_name is readable.
Expected result	COSEM object attributes could be read.
	Do Read attribute current “Association SN / LN”. logical_name . Expect the correct logical name.
Comments	

Table 4 – Release AA

Procedure	Release AA
References	[2_7_3] 9.3.3, 9.4.5, 9.5, 10.2.6.2, 10.3.6.1, 12.7, 12.8.
Prerequisites	The use of RLRQ / RLRE is declared.
Expected result	The AA is gracefully released.
	<p><u>If RLRQ is not supported then:</u></p> <ul style="list-style-type: none"> disconnect the supporting layer. <p><u>If RLRQ is supported:</u></p> <ul style="list-style-type: none"> if the application context is no ciphering, send RLRQ with RLRQ.reason = normal. Expect an RLRE with RLRE.reason = normal; if the application context is with ciphering, send RLRQ with: <ul style="list-style-type: none"> RLRQ.reason = normal; and RLRQ.user-information carrying the same InitiateRequest as in the AARQ having established the AA and protected the same way as in the AARQ. Expect RLRE with: <ul style="list-style-type: none"> RLRE.reason = normal; RLRE.user-information shall carry the same InitiateResponse as in the AARE. The protection shall be the same as in the RLRQ. If the profile is HDLC, then disconnect the supporting layer.
Comments	

Table 5 – Read attribute

Procedure	Read attribute
References	[2_7_3] 9.3.6, 9.3.11, 9.3.12, 9.4.6.3, 9.4.6.7, 9.5, 14.
Prerequisites	–
Expected result	The COSEM object attribute(s) could be read.
	<p>Send:</p> <ul style="list-style-type: none"> • ReadRequest in the case of SN referencing; or • Get-Request in the case of LN referencing with the appropriate parameters. <p>Expect ReadResponse / Get-Response.</p> <p><u>Protection on the request:</u></p> <ul style="list-style-type: none"> • if the application context is no ciphering, then no protection is applied; • if the application context is with ciphering, then protection is applied as follows: <ul style="list-style-type: none"> • if the protection is specified in the test case, then that protection is applied; • if the protection is not specified in the test case then protection is applied as required by the prevailing security context, i.e. <ul style="list-style-type: none"> • security_policy: SP_{min}; • security_keys valid for the AA; and • general-ciphering is used if supported and service-specific ciphering is used otherwise. <p><u>Protection on the response:</u></p> <p>The procedure checks that the response is protected at least as required by SP_{min}.</p>
Comments	

Table 6 – Write attribute

Procedure	Write attribute
References	[2_7_3] 9.3.7, 9.3.11, 9.3.13, 9.4.6.4, 9.4.6.8, 9.5, 14.
Prerequisites	-
Expected result	The COSEM object attributes could be written.
	<p>Send:</p> <ul style="list-style-type: none"> • WriteRequest in the case of SN referencing; or • Set-Request in the case of LN referencing with the appropriate parameters. <p>Expect WriteResponse / Set-Response.</p> <p><u>Protection:</u> as in Read attribute.</p>
Comments	

Table 7 – Invoke method

Procedure	Invoke method
References	[2_7_3] 9.3.8, 9.3.11, 9.3.12, 9.3.13, 9.4.6.5, 9.4.6.7, 9.4.6.8, 9.5.
Prerequisites	-
Expected result	The COSEM object method could be invoked.
	<p>Send:</p> <ul style="list-style-type: none"> • WriteRequest when return parameters are not expected or ReadRequest when return parameters are expected in the case of SN referencing; • Action-Request in the case of LN referencing with the appropriate parameters. <p>Expect WriteResponse / ReadResponse / Action-Response.</p> <p><u>Protection</u>: as in Read attribute.</p>
Comments	

Table 8 – Raise fatal failure

Procedure	Raise fatal failure
References	–
Prerequisites	–
Expected result	–
	<p>According to the settings specified:</p> <ul style="list-style-type: none"> • if “Abort the test session” has been selected (default), the test session is aborted; • if “Continue” has been selected, the test session is continued; • if “Ask the user” has been selected, the test session is suspended and a dialog box is displayed. If the operator chooses to continue the test session, it is resumed. If the operator chooses not to continue, then the test session is aborted. <p>In all cases, the reason of raising the fatal failure is logged.</p>
Comments	

6 Abstract Test Suite DLMS/COSEM application layer

6.1 Capabilities supported / not supported

The ATS DLMS/COSEM application layer supports a subset of the capabilities specified in [2_7_3].

Table 9 – Capabilities supported / not supported

Capability	Supported	
	Yes	No
Association Control Service Element		
COSEM-OPEN, confirmed, AARQ / AARE APDUs	x	
COSEM-OPEN, unconfirmed		x
RLRQ / RLRE APDUs	x	
COSEM-ABORT		x
Application contexts		
Logical_Name_Referencing_No_Cphering ::= context_id(1)	x	
Short_Name_Referencing_No_Cphering ::= context_id(2)	x	
Logical_Name_Referencing_With_Ciphering ::= context_id(3)	x	
Short_Name_Referencing_With_Ciphering ::= context_id(4)	x	
Authentication contexts		
COSEM_lowest_level_security_mechanism_name ::= mechanism_id(0)	x	
COSEM_low_level_security_mechanism_name ::= mechanism_id(1)	x	
COSEM_high_level_security_mechanism_name ::= mechanism_id(2)		x
COSEM_high_level_security_mechanism_name_using_MD5 ::= mechanism_id(3)	x	
COSEM_high_level_security_mechanism_name_using_SHA-1 ::= mechanism_id(4)	x	
COSEM_high_level_security_mechanism_name_using_GMAC ::= mechanism_id(5)	x	
DLMS version number		
DLMS-version-number = 6	x	
DLMS conformance block Confirmed services only ¹⁾		
general-protection (1)	x	
general-block-transfer (2) ²⁾	x	
read (3, SN)	x	
write (4, SN)	x	
unconfirmed-write (5, SN)		x
attribute0-supported-with-SET (8, LN)		x
priority-mgmt-supported (9, LN)		x
attribute0-supported-with-GET (10, LN)	x	
block-transfer-with-get-or-read (11)	x	
block-transfer-with-set or write (12)	x	
block-transfer-with-action (13, LN)		x

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	14/76
-----------------------	------------	----------------------	-------

Capability	Supported	
multiple-references (14, SN and LN)	x	
information-report (15, SN) ²⁾		x
data-notification (16)	x	
parameterized-access (18, SN)	x	
get (19, LN)	x	
set (20, LN)	x	
selective-access (21, LN)	x	
event-notification (22, LN) ²⁾		x
action (23, LN) ³⁾		x
¹⁾ For details, see 6.2.3		
²⁾ The CTT can handle these APDUs but does not have particular tests related to them.		
³⁾ Action service is implicitly tested during the COSEM object tests; see 6.2.3.		

6.2 Test cases

6.2.1 Test group APPL_IDLE

The purpose of the test group APPL_IDLE is to verify that the IUT does not respond to xDLMS service requests which are not allowed when the AL is in the IDLE state.

Table 10 – APPL_IDLE_N1: Data exchange in IDLE state

Test case	APPL_IDLE_N1: Data exchange in IDLE state
References	[2_7_3] 9.4.1, 9.5.
Test purpose	To verify that the IUT does not respond to xDLMS service requests which are not allowed when the AL is in the IDLE state.
AA filter	AA with the public client.
Prerequisites	–
Expected result	The IUT shall answer with ConfirmedServiceError (SN referencing) or with ExceptionResponse (LN referencing). A no response is also accepted.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
Test body	Do Check that the AA is in the Associated state. Failed if the procedure succeeds.
Postamble	–
Comments	

6.2.2 Test group APPL_OPEN

The purpose of the test group APPL_OPEN is to verify that the IUT is able to establish confirmed application associations and handles error cases correctly.

Table 11 – APPL_OPEN_1: Establish an AA with the parameters declared

Test case	APPL_OPEN_1: Establish an AA with the parameters declared
References	[1_11] 4.4.2, 4.4.3, 4.4.4 [2_7_3] 9.3.2, 9.4.2.2.2, 9.4.2.2.3, 9.4.4, 9.5, 11.
Test purpose	To verify that the IUT is able to establish an AA with the application context, authentication mechanism and xDLMS context declared.
AA filter	Each AA declared.
Prerequisites	–
Expected result	The IUT establishes the proposed AA and sends back AARE with appropriate information.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
	<u>Subtest 1: Establish an AA using the parameters declared</u> Do Establish a confirmed AA with the parameters declared. FAILED if the procedure fails.
	<u>Subtest 2: Check that the AA has been established</u> Do Check that the AA is in the Associated state. FAILED if the procedure fails.
	<u>Subtest 3: Release the AA</u> Do Release AA. FAILED if the procedure fails.
Postamble	–
Comments	

Table 12 – APPL_OPEN_2: Client user identification

Test case	APPL_OPEN_2: Client user identification
References	[1_11] 4.4.2 [2_7_3] 9.3.2, 9.4.4.1, 9.5.
Test purpose	To verify that the IUT correctly handles the client user identifier during AA establishment.
AA filter	Each AA that supports the client user identification mechanism.
Prerequisites	A (test) ClientUser is declared. An IllegalClientUser may be also declared. If not declared, the CTT uses a default value (the ClientUser declared + 1). The “Association SN / LN”. user-list holds {user_id, user_name} of the test client user.
Expected result	The AA cannot be established without a client user or by the illegal current user.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
Test body	<u>Subtest 1: Try to establish an AA without the client user ID present</u> Do Establish a confirmed AA with the parameters declared but without AARQ.calling_AE_invocation_id present. FAILED if the procedure succeeds or if AARE.result is not rejected-permanent or AARE.result-source-diagnostic is not acse-service-user.calling-AE-invocation-identifier-not-recognized (6).
	<u>Subtest 2: Try to establish an AA with an illegal client user ID</u> Do Establish a confirmed AA with the parameters declared but with AARQ.calling_AE_invocation_id carrying an illegal client user ID. Verdict as in subtest 1.
Postamble	–
Comments	

Table 13 – APPL_OPEN_3, HLS authentication, Pass 3 and Pass 4

Test case	APPL_OPEN_3, HLS authentication, Pass 3 and Pass 4
References	[1_11] 4.4.3, 4.4.4 [2_7_3] 9.2.3.4, 9.2.4.8.4, 9.3.2, 9.4.2.2.3, 9.4.4, 9.5.
Test purpose	To verify that the IUT accepts only correctly processed StoC.
AA filter	Each AA that uses a ciphered application context and HLS authentication.
Prerequisites	–
Expected result	After AA establishment Pass 2, the access of the client is restricted to current "Association SN / LN". reply_to_HLS_authentication . The AA is established only if the client processes StoC correctly and if the IUT returns a correctly processed CtoS.
Preamble	–
Test body	<p><u>Subtest 1: Check that after Pass 2, the access is restricted to reply_to_HLS_authentication method</u></p> <p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared but perform only Passes 1 and 2.</p> <p>Do Check that the AA is in the Associated state.</p> <p>FAILED if the procedure succeeds.</p>
	<p><u>Subtest 2: Check that the AA is not established if f(StoC) is incorrect</u></p> <p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared but perform only Passes 1 and 2.</p> <p>Do Invoke method current "Association SN / LN".reply_to_HLS_authentication using {A+E, GUEK, GAK} but with an incorrectly processed StoC:</p> <ul style="list-style-type: none"> in the case of HLS_MD5 instead of <ul style="list-style-type: none"> f(StoC) = MD5(StoC HLS secret) (correct) send f(StoC) = MD5(HLS secret StoC) (incorrect); in the case of HLS_SHA-1 instead of: <ul style="list-style-type: none"> f(StoC) = SHA-1(StoC HLS secret) (correct) send f(StoC) = SHA-1(HLS secret StoC) (incorrect); in the case of HLS_GMAC: <ul style="list-style-type: none"> in f(StoC) = SC FC GMAC(SC AK StoC) replace the first SC = 0x10 (authenticated only) by 0x20 (encrypted only). <p>FAILED if the procedure succeeds.</p>
	<p><u>Subtest 3: Check that the AA is not established if in Pass 3 DEK is used</u></p> <p>If DEK is not supported then mark this subtest INAPPLICABLE.</p> <p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared, but in Pass 3 use {A+E, DEK, GAK} instead of {A+E, GUEK, GAK}.</p> <p>FAILED if the procedure succeeds.</p>
Postamble	–
Comments	

Table 14 – APPL_OPEN_4: Protocol version

Test case	APPL_OPEN_4: Protocol version
References	[2_7_3] 9.1.2.2, 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.5, 11.6.
Test purpose	To verify that the IUT correctly monitors AARQ.protocol-version .
AA filter	An AA with the public client.
Prerequisites	–
Expected result	The IUT shall establish the proposed AA only if AARQ.protocol-version is absent, or if it is present and its value is {version 1 (0)}. Otherwise, it shall reject the proposed AA and shall send back AARE with correct diagnostic information. NOTE The case when the AARQ.protocol-version is absent is tested in APPL_OPEN_1.
Preamble	–
Test body	<u>Subtest 1: protocol-version present and containing the default value</u> Do Make sure that the IUT AL is in the IDLE state . Do Establish a confirmed AA with the parameters declared but send AARQ with AARQ.protocol-version present with default value {version 1 (0)}. FAILED if the procedure fails.
	<u>Subtest 2: protocol-version present but not containing the default value</u> Do Make sure that the IUT AL is in the IDLE state . Do Establish a confirmed AA with the parameters declared but send AARQ with AARQ.protocol-version present, and not containing the default value. FAILED if the procedure succeeds or if AARQ.result is not rejected-permanent or AARQ.result-source-diagnostic is not acse-service-provider.no-common-acse-version (2).
Postamble	–
Comments	

Table 15 – APPL_OPEN_5: Application context

Test case	APPL_OPEN_5: Application context
References	[2_7_3] 9.1.2.2, 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.5, 11.6.
Test purpose	To verify that the IUT correctly monitors AARQ.application-context-name .
AA filter	An AA with the public client.
Prerequisites	–
Expected result	The IUT shall establish the proposed AA only if the value of AARQ.application-context-name matches the value declared for the given AA. Otherwise, it shall reject the proposed AA and shall send back AARE with correct diagnostic information.
Preamble	Do Make sure that the IUT AL is in the IDLE state .
Test body	<u>Unknown application context</u> Do Establish a confirmed AA with the parameters declared but with AARQ.application-context-name carrying an unknown application context name. FAILED if procedure succeeds or if AARQ.result is not rejected-permanent or if AARE.result-source-diagnostic is not acse-service-user.application-context-name-not-supported (2).
Postamble	–
Comments	

Table 16 – APPL_OPEN_6: Titles, qualifiers and invocation identifiers

Test case	APPL_OPEN_6: Titles, qualifiers and invocation identifiers
References	[2_7_3] 9.1.3, 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.6.
Test purpose	<p>To verify that IUT correctly processes the AARQ fields where relevant and ignores them otherwise:</p> <ul style="list-style-type: none"> • AARQ.called-AP-title (not used); • AARQ.called-AE-qualifier (not used); • AARQ.called-AP-invocation-id (not used); • AARQ.called-AE-invocation-id (not used); • AARQ.calling-AP-title (carries the system title of the client if the application context is with ciphering or if the authentication-mechanism is HLS_GMAC); • AARQ.calling-AE-qualifier (not used); • AARQ.calling-AP-invocation-id (not used); • AARQ.calling-AE-invocation-id (carries the client user ID if the client user identification mechanism is supported)
AA filter	See in the subtests.
Prerequisites	–
Expected result	<p>The IUT correctly handles the fields of the AARQ:</p> <ul style="list-style-type: none"> • it ignores the fields AARQ.called-AP-title, AARQ.called-AE-qualifier, AARQ.called-AP-invocation-id, AARQ.called-AE-invocation-id and AARQ.calling-AP-invocation-id; • it correctly handles the AARQ.calling-AP-title field; • it ignores the AARQ.calling-AE-invocation-id field when client user identification is not supported.
Preamble	–
Test body	<p><u>Subtest 1: Unused AARQ fields are present with a dummy value</u></p> <p>This subtest is performed in a public AA (i.e. with lowest level security), supporting client user identification or not.</p> <p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> • AARQ.called-AP-title: dummy; • AARQ.called-AE-qualifier: dummy; • AARQ.called-AP-invocation-id: dummy; • AARQ.called-AE-invocation-id: dummy; • AARQ.calling-AP-title: absent; • AARQ.calling-AE-qualifier: absent; • AARQ.calling-AP-invocation-id: dummy; • AARQ.calling-AE-invocation-id: carries the client user ID declared if the client user identification mechanism is supported and absent otherwise. <p>FAILED if the procedure fails or if any of the following elements are present:</p> <ul style="list-style-type: none"> • AARE.responding-AP-title; • AARE.responding-AE-qualifier; • AARE.responding-AP-invocation-id; • AARE.responding-AE-invocation-id.
	<p><u>Subtest 2: AARQ.calling-AP-title too short</u></p> <p>This subtest is performed in an AA with ciphering or with HLS_GMAC. If there is no such AA available, the subtest is INAPPLICABLE.</p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> • AARQ.called-AP-title: dummy; • AARQ.called-AE-qualifier: dummy;

CTT 3.0: ATS DLMS/COSEM Application layer – ATS COSEM interface objects –
ATS SYMSEC_0

Test case	APPL_OPEN_6: Titles, qualifiers and invocation identifiers
	<ul style="list-style-type: none"> • AARQ.called-AP-invocation-id: dummy; • AARQ.called-AE-invocation-id: dummy; • AARQ.calling-AP-title: carrying the CTT system title truncated to 7 bytes; • AARQ.calling-AE-qualifier: absent; • AARQ.calling-AP-invocation-id: dummy; • AARQ.calling-AE-invocation-id: carries the client user ID declared if the client user identification mechanism is supported and empty otherwise. <p>FAILED if the procedure succeeds or if any of the following conditions is true:</p> <ul style="list-style-type: none"> • AARE.result-source-diagnostic is not acse-service-user with no-reason-given (1), calling-AP-title-not recognized (3), authentication-failure (13) or authentication-required (14) or if any of the following elements are present: <ul style="list-style-type: none"> • AARE.responding-AE-qualifier; • AARE.responding-AP-invocation-id ; • AARE.responding-AE-invocation-id.
	<p><u>Subtest 3: AARQ.calling-AP-title too long</u></p> <p>This subtest is performed in an AA with ciphering or with HLS_GMAC. If there is no such AA available, the subtest is INAPPLICABLE.</p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> • AARQ.called-AP-title: dummy; • AARQ.called-AE-qualifier: dummy; • AARQ.called-AP-invocation-id: dummy; • AARQ.called-AE-invocation-id: dummy; • AARQ.calling-AP-title: carrying the CTT system title with an extra byte added at the end; • AARQ.calling-AE-qualifier: absent; • AARQ.calling-AP-invocation-id: dummy; • AARQ.calling-AE-invocation-id: carries the client user ID declared if the client user identification mechanism is supported and empty otherwise. <p>FAILED if the procedure succeeds or if any of the following conditions is true:</p> <ul style="list-style-type: none"> • AARE.result-source-diagnostic is not acse-service-user with no-reason-given (1), calling-AP-title-not recognized (3), authentication-failure (13) or authentication-required (14) or if any of the following elements are present: <ul style="list-style-type: none"> • AARE.responding-AE-qualifier; • AARE.responding-AP-invocation-id; • AARE.responding-AE-invocation-id.
	<p><u>Subtest 4: AARQ.calling-AE-invocation-id present when client user identification is not supported</u></p> <p>This subtest is performed in an AA that does not support client user identification. If there is no such AA available, the subtest is INAPPLICABLE.</p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> • AARQ.called-AP-title: dummy; • AARQ.called-AE-qualifier: dummy; • AARQ.called-AP-invocation-id: dummy; • AARQ.called-AE-invocation-id: dummy; • AARQ.calling-AP-title: the CTT system title if the AA is with ciphering or with HLS_GMAC and empty otherwise; • AARQ.calling-AE-qualifier: absent; • AARQ.calling-AP-invocation-id: dummy; • AARQ.calling-AE-invocation-id: dummy. <p>FAILED if the procedure fails or if any of the following elements are present:</p> <ul style="list-style-type: none"> • AARE.responding-AE-qualifier; • AARE.responding-AP-invocation-id;

Test case	APPL_OPEN_6: Titles, qualifiers and invocation identifiers
	<ul style="list-style-type: none"> AARE.responding-AE-invocation-id. <p>The AARE.responding-AP-title shall be present if the AA is with ciphering or with HLS_GMAC and shall be absent otherwise.</p>
Postamble	–
Comments	

Table 17 – APPL_OPEN_7: Authentication functional unit

Test case	APPL_OPEN_7: Authentication functional unit
References	[2_7_3] 9.3.2, 9.4.2.1, 9.4.2.2.3, 9.4.4.1, 9.5, 12.3, 12.6.
Test purpose	To verify that the IUT correctly handles the AARQ and AARE Authentication Functional Unit (AFU).
AA filter	All AAs declared, except the ones using no security authentication.
Prerequisites	For each AA, the authentication mechanism supported is declared, together with the relevant security material as specified in [2_7_3] 9.2.3.
Expected result	The IUT rejects the proposed AA if the fields of the AARQ AFU are not correct. The IUT sends back AARE and its fields provide suitable diagnostic information.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
Test body	
AFU_000	<p><u>Subtest 1: AFU absent</u></p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ without:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements; AARQ.mechanism-name; AARQ.calling-authentication-value. <p>FAILED if the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-mechanism-name-not-recognised (11); or authentication-mechanism-name-required (12); or authentication-failure (13); or authentication-required (14).
AFU_F11	<p><u>Subtest 2: Wrong ACSE requirement</u></p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present but wrong (bit 0 not set); AARQ.mechanism-name present and correct; AARQ.calling-authentication-value: in the case of LLS, the correct password, in the case of HLS CtoS of correct length. <p>FAILED if the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-mechanism-name-required (12); or authentication-failure (13); or authentication-required (14).
AFU_100	<p><u>Subtest 3: Mechanism-name and calling-authentication-value absent</u></p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and correct; AARQ.mechanism-name absent; and AARQ.calling-authentication-value absent. <p>FAILED is the procedure succeeds or AARE.result-source-diagnostic is not acse-service-</p>

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	21/76
-----------------------	------------	----------------------	-------

Test case	APPL_OPEN_7: Authentication functional unit
	<p>user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-mechanism-name-not-recognised (11); or authentication-mechanism-name-required (12); or authentication-failure (13); or authentication-required (14).
AFU_101	<p><u>Subtest 4: Mechanism-name absent</u></p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and correct; AARQ.mechanism-name absent; AARQ.calling-authentication-value: in the case of LLS, the correct password, in the case of HLS CtoS of correct length; <p>FAILED is the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-mechanism-name-not-recognised (11); or authentication-mechanism-name-required (12); or authentication-failure (13); or authentication-required (14).
AFU_110	<p><u>Subtest 5: Calling-authentication-value absent</u></p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and correct; AARQ.mechanism-name present and correct; AARQ.calling-authentication-value absent. <p>FAILED is the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-failure (13); or authentication-required (14).
AFU_1F1	<p><u>Subtest 6: Mechanism-name present but wrong</u></p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and correct; AARQ.mechanism-name present but wrong: <ul style="list-style-type: none"> if the AA uses LLS authentication then AARQ.mechanism-name = mechanism_id(0); if the AA uses HLS authentication then AARQ.mechanism-name = mechanism_id(1); AARQ.calling-authentication-value: in the case of LLS, the correct password, in the case of HLS CtoS of correct length. <p>FAILED is the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-mechanism-name-not-recognised (11); or authentication-mechanism-name-required (12); or authentication-failure (13); or authentication-required (14).
AFU_11F	<p><u>Subtest 7: Calling-authentication-value present but wrong</u></p> <p>If the AA uses LLS authentication then:</p> <p>Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and correct; AARQ.mechanism-name present and correct;

Test case	APPL_OPEN_7: Authentication functional unit
	<ul style="list-style-type: none"> AARQ.calling-authentication-value containing an incorrect password. <p>If the AA uses HLS authentication then: Do Establish a confirmed AA with the parameters declared but send AARQ with:</p> <ul style="list-style-type: none"> AARQ.sender-acse-requirements present and correct; AARQ.mechanism-name present and correct; AARQ.calling-authentication-value containing a too short challenge. <p>FAILED if the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user with:</p> <ul style="list-style-type: none"> no-reason-given (1); or authentication-failure (13). <p>NOTE authentication-required would be inappropriate in the case of HLS, because that information is given after sending a correct CtoS.</p>
Postamble	–
Comments	

Table 18 – APPL_OPEN_9: xDLMS InitiateRequest: dedicated-key

Test case	APPL_OPEN_9: xDLMS InitiateRequest: dedicated-key
References	[2_7_3] 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.6.
Test purpose	To verify that the IUT rejects the AA proposed if the application context is no ciphering but the dedicated-key is present.
AA filter	An AA with application context no ciphering.
Prerequisites	–
Expected result	The IUT shall not establish the proposed AA and shall send back AARE with correct diagnostic information.
Preamble	Do Make sure that the IUT AL is in the IDLE state .
Test body	Do Establish a confirmed AA with the parameters declared but with AARQ.user-information.InitiateRequest.dedicated-key present. FAILED if the procedure succeeds.
Postamble	–
Comments	

Table 19 – APPL_OPEN_11: xDLMS InitiateRequest: quality-of-service

Test case	APPL_OPEN_11: xDLMS InitiateRequest: quality-of-service
References	[2_7_3] 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.6.
Test purpose	To verify that the IUT ignores InitiateRequest.quality-of-service .
AA filter	An AA with the public client.
Prerequisites	–
Expected result	The IUT shall establish the proposed AA and send back a correct AARE .
Preamble	Do Make sure that the IUT AL is in the IDLE state .
Test body	Do Establish a confirmed AA with the parameters declared but include any value in AARQ.user-information.InitiateRequest.quality-of-service . FAILED if the procedure fails.

Postamble	–
Comments	

Table 20 – APPL_OPEN_12: xDLMS InitiateRequest: dlms-version-number

Test case	APPL_OPEN_12: xDLMS InitiateRequest: dlms-version-number
References	[2_7_3] 9.1.2.3.1, 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.5, 11.6.
Test purpose	To verify that the IUT is correctly checking the value of the proposed-dlms-version-number.
AA filter	An AA with the public client.
Prerequisite	–
Expected result	The IUT shall accept only AAs with proposed-dlms-version-number \geq 6. The negotiated-dlms-version-number shall be = 6.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
Test body	<u>Subtest 1: proposed-dlms-version-number = 5</u> Do Establish a confirmed AA with the parameters declared but with AARQ.user-information carrying InitiateRequest.proposed-dlms-version-number = 5 . FAILED if the procedure succeeds or AARE.result-source-diagnostic is not acse-service-user no-reason given or AARE.user-information does not carry confirmedServiceError with initiate-error [1] – initiate [6] - dlms-version-too-low (1). <u>Subtest 2: proposed-dlms-version-number = 7</u> Do Establish a confirmed AA with the parameters declared but with AARQ.user-information carrying InitiateRequest.proposed-dlms-version-number = 7 . FAILED if the procedure fails or the negotiated DLMS version is not equal to 6.
Postamble	–
Comments	

Table 21 – APPL_OPEN_13: xDLMS InitiateRequest: conformance-block

Test case	APPL_OPEN_13: xDLMS InitiateRequest: conformance-block
References	[2_7_3] 9.1.2.3, 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.4.6.1, 9.5, 11.2, 11.3, 11.6.
Test purpose	To verify that the IUT negotiates the conformance block correctly.
AA filter	An AA with the public client.
Prerequisites	–
Expected result	The IUT shall establish the proposed AA only if the negotiated conformance block is consistent with the application context name and it is sufficient. If the proposed AA is accepted, the IUT shall send back AARE with AARE.user-information carrying InitiateResponse containing the negotiated conformance block. If the conformance block is not properly negotiated – the IUT is expected to make a logical AND between the conformance block proposed and its own conformance block supported – or if the negotiated conformance block is not sufficient, the AA is rejected.
Preamble	–
Test body	<u>Subtest 1: Propose all services and capabilities in line with the application context</u> Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared but with proposed-conformance containing all possible services and capabilities in line with the application context declared. FAILED if the procedure fails. <u>Subtest 2: Propose a mix of LN and SN services and capabilities</u> Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared but with proposed-

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	24/76
-----------------------	------------	----------------------	-------

Test case	APPL_OPEN_13: xDLMS InitiateRequest: conformance-block
	conformance containing all possible SN and LN services and capabilities. FAILED if the procedure fails or if the negotiated conformance contains a service /capability incompatible with the application context.
	<u>Subtest 3: Propose an insufficient conformance block</u> Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared but with proposed-conformance containing an insufficient conformance block: <ul style="list-style-type: none"> in the case of the application context using SN referencing, propose only GET; in the case of the application context using LN referencing, propose only Read. FAILED if the procedure succeeds.
Postamble	–
Comments	

Table 22 – APPL_OPEN_14: xDLMS InitiateRequest: client-max-receive-pdu-size

Test case	APPL_OPEN_14: xDLMS InitiateRequest: client-max-receive-pdu-size
References	[2_7_3] 9.1.2.3.1, 9.3.2, 9.4.2, 9.4.3, 9.4.4, 9.5, 11.2, 11.3, 11.6.
Test purpose	To verify that the IUT correctly handles client-max-receive-pdu-size.
AA filter	An AA with the public client.
Prerequisites	–
Expected result	The IUT shall not establish the proposed AA if the client-max-receive-pdu-size is below 12.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
Test body	Do Establish a confirmed AA with the parameters declared , but with AARQ.user-information.InitiateRequest.client-max-receive-pdu-size = 11 . FAILED if the procedure succeeds or if AARE.user-information does not carry ConfirmedServiceError with InitiateError [1] - initiate [6] – pdu-size-too-short (3).
Postamble	–
Comments	

6.2.3 Test group APPL_DATA: xDLMS data services

The purpose of the test group APPL_DATA is to verify that the xDLMS services for data exchange are properly implemented and that error cases are appropriately handled.

This test group covers only error cases.

Table 23 – APPL_DATA_LN_N1: Get-Request with errors

Test case	APPL_DATA_LN_N1: Get-Request with errors
References	[2_7_3] 9.1.2.3, 9.3.6, 9.4.6.1, 9.4.6.3, 9.5, 14.
Test purpose	To verify that the IUT properly handles erroneous Get-Request APDUs.
AA filter	An AA with LN referencing no ciphering and that supports the GET service.
Prerequisites	There is such an AA available. If not, mark this test INAPPLICABLE.
Expected result	The IUT does not serve the request and sends back an appropriate error message.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	<u>Subtest 1: Get-Request with unknown tag</u> Do Read attribute "Association SN/LN". logical_name but with non-existing tag in the Get-

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	25/76
-----------------------	------------	----------------------	-------

Test case	APPL_DATA_LN_N1: Get-Request with errors
	Request. FAILED if the response is not ExceptionResponse or not Get-Response-Normal with Get-Data-Result = Data-Access-Result. Do Check that the AA is in the Associated state.
	<u>Subtest 2: Get-Request with missing elements</u> Do Read attribute "Association SN/LN". logical_name but omit some elements in the Get-Request . The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
	<u>Subtest 3: Get-Request for a non-existing object (illegal logical name)</u> Do Read attribute "Association SN/LN". logical_name but replace the logical name with a dummy value. The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
	<u>Subtest 4: Get-Request for a non-existing attribute</u> Do Read attribute "Association SN/LN". logical_name but replace the attribute-id with the value 0x40. The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
Postamble	–
Comments	

Table 24 – APPL_DATA_LN_N3: Set-Request with errors

Test case	APPL_DATA_LN_N3: Set-Request with errors
References	[2_7_3] 9.1.2.3, 9.3.7, 9.4.6.1, 9.4.6.4, 9.5, 14.
Test purpose	To verify that the IUT properly handles erroneous Set-Request APDUs.
AA filter	An AA with LN referencing no ciphering and that supports the SET service.
Prerequisites	There is such an AA available. If not, mark this test INAPPLICABLE.
Expected result	The IUT does not serve the request and sends back an appropriate error message.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	<u>Subtest 1: Set-Request with unknown tag</u> Do Write attribute "Association SN/LN". logical_name but insert a non-existing tag in the Set-Request . FAILED if the response is not ExceptionResponse or not Set-Response-Normal with Data-Access-Result not success. Do Check that the AA is in the Associated state.
	<u>Subtest 2: Set-Request with missing elements</u> Do Write attribute "Association SN/LN". logical_name but but omit some elements of the Set-Request . The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
	<u>Subtest 3: Set-Request for a non-existing object (illegal logical name)</u> Do Write attribute "Association SN/LN". logical_name but replace the logical name with a dummy value. The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
	<u>Subtest 4: Set-Request for a non-existing attribute</u>

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	26/76
-----------------------	------------	----------------------	-------

CTT 3.0: ATS DLMS/COSEM Application layer – ATS COSEM interface objects –
ATS SYMSEC_0

	Do Write attribute "Association SN/LN". logical_name but replace the attribute-id with the value 0x40. The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state .
Postamble	–
Comments	

Table 25 – APPL_DATA_LN_N4: Unsupported service

Test case	APPL_DATA_LN_N4: Unsupported service
References	[2_7_3] 9.4.6.1, 9.5.
Test purpose	To verify that the IUT ignores service requests not defined or not implemented.
AA filter	An AA with LN referencing no ciphering.
Prerequisites	There is such an AA available. If not, mark this test INAPPLICABLE.
Expected result	The IUT may not respond, or it may respond with ConfirmedServiceError or with ExceptionResponse .
Preamble	Do Make sure that the IUT AL is in the IDLE state . Do Establish a confirmed AA with the parameters declared .
Test body	Do Read attribute "Association SN/LN". logical_name but use a ReadRequest service with base_name 0xFA00. FAILED if there is a response but it is not ConfirmedServiceError or ExceptionResponse .
Postamble	–
Comments	

Table 26 – APPL_DATA_SN_N1: ReadRequest with errors

Test case	APPL_DATA_SN_N1: ReadRequest with errors
References	[2_7_3] 9.1.2.3, 9.3.11, 9.3.12, 9.4.6.1, 9.4.6.7, 9.5, 14.
Test purpose	To verify that the IUT properly handles erroneous ReadRequest APDUs.
AA filter	An AA with SN referencing no ciphering.
Prerequisites	There is such an AA available. If not, mark this test INAPPLICABLE.
Expected result	The IUT does not serve the request and sends back an appropriate error message.
Preamble	Do Make sure that the IUT AL is in the IDLE state . Do Establish a confirmed AA with the parameters declared .
Test body	<u>Subtest 1: ReadRequest with unknown tag</u> Do Read attribute "Association SN/LN". logical_name but insert a non-existing tag for VariableAccessSpecification . NOTE The correct tags are [2] variable-name, [4] parameterized-access, [5] block-number-access, and [6] read-data-block-access. FAILED if the response is not ReadResponse.data-access-error . Do Check that the AA is in the Associated state . <u>Subtest 2: ReadRequest with missing elements</u> Do Read attribute "Association SN/LN". logical_name but omit some elements. The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state . <u>Subtest 3: ReadRequest for a non-existing variable-name</u> Do Read attribute "Association SN/LN". logical_name but replace the variable-name with a dummy value. The verdict is the same as in Subtest 1.

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	27/76
-----------------------	------------	----------------------	-------

Test case	APPL_DATA_SN_N1: ReadRequest with errors
	Do Check that the AA is in the Associated state.
Postamble	–
Comments	

Table 27 – APPL_DATA_SN_N2: WriteRequest with errors

Test case	APPL_DATA_SN_N2: WriteRequest with errors
References	[2_7_3] 9.1.2.3, 9.3.11, 9.3.13, 9.4.6.1, 9.4.6.8, 9.5, 14.
Test purpose	To verify that the IUT properly handles erroneous WriteRequest APDUs.
AA filter	An AA with SN referencing no ciphering and that supports the Write service.
Prerequisites	There is such an AA available. If not, mark this test INAPPLICABLE.
Expected result	The IUT does not serve the request and sends back an appropriate error message.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	<u>Subtest 1: WriteRequest with unknown tag</u> Do Write attribute “Association SN/LN”. logical_name but insert a non-existing tag for VariableAccessSpecification. NOTE The correct tags are [2] variable-name, [4] parameterized-access and [7] write-data-block-access. FAILED if the response is not WriteResponse.data-access-error . Do Check that the AA is in the Associated state.
	<u>Subtest 2: WriteRequest with missing elements</u> Do Write attribute “Association SN/LN”. logical_name but but omit some elements. The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
	<u>Subtest 3: WriteRequest for a non-existing variable-name</u> Do Write attribute “Association SN/LN”. logical_name but replace the variable_name with a dummy value (e.g. use 0xFA01). The verdict is the same as in Subtest 1. Do Check that the AA is in the Associated state.
Postamble	–
Comments	

Table 28 – APPL_DATA_SN_N3: Unsupported service

Test case	APPL_DATA_SN_N3: Unsupported service
References	[2_7_3] 9.4.6.1, 9.5.
Test purpose	To verify that the IUT ignores services not defined or not implemented.
AA filter	An AA with SN referencing no ciphering.
Prerequisites	There is such an AA available. If not, mark this test INAPPLICABLE.
Expected result	The IUT may not respond, or it may respond with ConfirmedServiceError .
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	Do Read attribute “Association SN/LN”. logical_name but use a Get-Request-Normal APDU. FAILED if there is a response but it is not ConfirmedServiceError .
Postamble	–

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	28/76
-----------------------	------------	----------------------	-------

Comments	
----------	--

6.2.4 Test group APPL_REL

The purpose of this test group is to verify that the AA release is correctly implemented.

Table 29 – APPL_REL_P1

Test case	APPL_REL_P1
References	[2_7_3] 9.1.2.2, 9.3.3, 9.4.2, 9.5, 10.2.6.2, 10.3.6.1.
Test purpose	To verify that the AA can be gracefully released.
AA filter	An AA with the public client.
Prerequisites	–
Expected result	The AA can be released and the IUT AL returns to IDLE state.
Preamble	Do Establish a confirmed AA with the parameters declared .
Test body	<p><u>Subtest 1: Graceful release in the 3-layer, CO, HDLC based communication profile</u></p> <p>NOTE In this communication profile, AAs are mapped to data link layer connections, and they are released by disconnecting the supporting data link layer.</p> <p>If the IUT does not support this communication profile, mark this test as “INAPPLICABLE”.</p> <p>Do Release AA.</p> <p>Connect the application layer's supporting layer again.</p> <p>Do Check that the AA is in the Associated state.</p> <p>FAILED if the procedure succeeds.</p>
	<p><u>Subtest 2: Graceful release in the TCP/IP based profile by disconnecting the supporting layer</u></p> <p>If the IUT does not support this communication profile, mark this test as “INAPPLICABLE”.</p> <p>Disconnect the TCP layer.</p> <p>Connect the TCP layer again.</p> <p>Do Check that the AA is in the Associated state.</p> <p>FAILED if the procedure succeeds.</p>
	<p><u>Subtest 3: Graceful release in the TCP/IP based profile using RLRQ / RLRE</u></p> <p>If the IUT does not support releasing AAs by RLRE/RLRQ, mark this test as “INAPPLICABLE”.</p> <p>Do Release AA.</p> <p>Do Check that the AA is in the Associated state.</p> <p>FAILED if the procedure succeeds.</p>
Postamble	–
Comments	

7 Abstract Test Suite COSEM objects

7.1 Interface classes supported

CTT 3 supports all interface classes included in Blue Book Edition 11.

Table 30 – Interface classes supported

Interface class name	class_id	version(s)	IC specific test clause
Data	1	0	7.3
Register	3	0	7.3
Extended register	4	0	7.3.2
Demand register	5	0	7.3.3
Register activation	6	0	–
Profile generic	7	1	7.3.4
Clock	8	0	–
Script table	9	0	7.3.5
Schedule	10	0	–
Special days table	11	0	–
Association SN	12	0, 1, 2, 3	7.3.6
Association LN	15	0, 1, 2	7.3.6
SAP Assignment	17	0	–
Image transfer	18	0	–
IEC Local port setup	19	0, 1	–
Activity calendar	20	0	–
Register monitor	21	0	7.3.7
Single action schedule	22	0	–
IEC HDLC setup	23	0, 1	–
IEC Twisted pair (1) setup	24	0, 1	–
M-BUS slave port setup	25	0	–
Utility tables	26	0	–
PSTN modem configuration	27	0	–
Modem configuration	27	1	–
Auto answer	28	0, 2	–
PSTN auto dial	29	0	–
Auto connect / Auto dial	29	1, 2	–
Push setup	40	0	7.3.8
TCP-UDP setup	41	0	–
IPv4 setup	42	0	–
Ethernet setup / MAC address setup	43	0	–
PPP setup	44	0	–
GPRS modem setup	45	0	–

**CTT 3.0: ATS DLMS/COSEM Application layer – ATS COSEM interface objects –
ATS SYMSEC_0**

Interface class name	class_id	version(s)	IC specific test clause
SMTP setup	46	0	–
GSM diagnostic	47	0	–
IPv6 setup	48	0	–
S-FSK Phy&MAC setup NOTE The use of version 0 is deprecated.	50	0, 1	–
S-FSK Active initiator	51	0	–
S-FSK MAC synchronization timeouts	52	0	–
S-FSK MAC counters	53	0	–
S-FSK IEC 6334-4-32 LLC setup	55	0	–
IEC 6334-4-32 LLC setup	55	1	–
S-FSK Reporting system list	56	0	–
ISO/IEC 8802-2 LLC Type 1 setup	57	0	–
ISO/IEC 8802-2 LLC Type 2 setup	58	0	–
ISO/IEC 8802-2 LLC Type 3 setup	59	0	–
Register table	61	0	–
Status mapping	63	0	–
Security setup	64	0	7.3.9
Parameter monitor	65	0	–
Sensor manager	67	0	–
Disconnect control	70	0	–
Limiter	71	0	–
M-Bus client	72	0, 1	–
Wireless Mode Q channel	73	0	–
M-Bus master port setup	74	0	–
61334-4-32 LLC SCS setup	80	0	–
PRIME NB OFDM PLC Physical layer counters	81	0	–
PRIME NB OFDM PLC MAC setup	82	0	–
PRIME NB OFDM PLC MAC functional parameters	83	0	–
PRIME NB OFDM PLC MAC counters	84	0	–
PRIME NB OFDM PLC MAC network administration data	85	0	–
PRIME NB OFDM PLC Application identification	86	0	–
NOTE Version 0 of G3-PLC setup interface classes specified in Blue Book Edition 11 have been deprecated because a new Edition of ITU-T G.9903 has been published in April 2014. Therefore version 1 of these ICs specified in Blue Book Edition 12 have to be supported			
G3 NB OFDM PLC MAC layer counters	90	1	–
G3 NB OFDM PLC MAC setup	91	1	–
G3 NB OFDM PLC 6LoWPAN adaptation layer setup	92	1	–
ZigBee® SAS startup	101	0	–
ZigBee® SAS join	102	0	–
ZigBee® SAS APS fragmentation	103	0	–

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	31/76
------------------------------	-------------------	-----------------------------	--------------

Interface class name	class_id	version(s)	IC specific test clause
ZigBee® network control	104	0	–
ZigBee® tunnel setup	105	0	–

7.2 Test algorithm

The general algorithm of testing COSEM interface objects is the following:

- for each logical device:
 - for each application association:
 - build the AA and read the object_list;
 - sort the list according to class_id and version;
 - for each object, execute the corresponding COSEM_X_Y attribute test;
 - perform Multiple references test;
 - check presence and report value of mandatory elements:
 - attributes of the current “Association SN / LN” object, including a textual report of the object list;
 - attributes of the “SAP assignment” object;
 - Logical Device Name.

The information for testing is obtained from the IUT itself and from the Conformance Test Information (CTI) file. See Annex A.

Each COSEM_X_Y_attribute test case tests each attribute of the object under test along the following pattern:

NOTE X is the Class_Id and Y is the version of the object instance.

For each Attribute do begin

The algorithm has three branches: "Attribute readable", "Attribute not readable" and "Attribute writeable". The branches that will be performed depend on the access_right which may be no_access, read_only, write_only or read_write. If a FAILED or an INCONCLUSIVE verdict is given, the rest of the algorithm is not performed.

NOTE From this point of view the access rights requesting authentication do not make a difference.

In the case of LN referencing, access_rights are provided by the object_list attribute of the Association LN object. In the case of SN referencing, access rights are provided by the access_rights_list attribute of the Association SN object. If this is not supported by the IUT the CTT uses the access rights declared in the CTI.

```

if Attribute readable then begin
  Read Attribute
  If an error message is received then the verdict is FAILED.

  if Attribute is logical_name then begin

```

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	32/76
-----------------------	------------	----------------------	-------

```
If the data_type is not octet_string of 6, or the combination of
the logical name and the interface class_id is not valid, then
the verdict is FAILED.

end else if Attribute is of type CHOICE then begin

    test type of attribute

        For such attributes, ranges or enumerated values are not
        specified in the interface class definition.

        If the type is not correct then the verdict is FAILED.

    end

else begin

    For all other attributes, the interface class definition
    specifies the data_type to be used.
    Ranges (Min. / Max.) may be defined.
    In the case of type enumerated, the values allowed are
    specified.

    ATTRIBUTE_VALUES testing tests if the value is within the range
    or among the enumerated values specified.

    check type of Attribute
    If the type is not correct then the verdict is FAILED.

    check the value
    If the value is not in the range specified or not one of the
    enumerated values specified, then the verdict is FAILED.

end

end else begin

    Attribute not readable
    Try to read Attribute and expect an error message.
    If data are delivered or the error message is not correct then
    the verdict is FAILED.

end

if Attribute writable then begin
    Write tests are not performed in the case of attributes of
    "Profile generic" objects, irrespectively of the access_rights
    specified.

    if CTI contains WriteTestData for Attribute then begin
        Write the specified data
        If an error message is received then the verdict is FAILED.

    end else if Attribute readable then begin
        Rewrite the data that has been read
        If an error message is received then the verdict is FAILED.

    end else
```

```
    The verdict is INCONCLUSIVE "WRITE DATA NOT AVAILABLE"  
end
```

```
The verdict is PASSED.
```

```
end
```

For each object instance, the following is reported:

- the class_id, the version and the logical name;
- for each attribute, the access right and the verdict.

In addition, the following information is recorded in the log:

- for logical_names, reference to the appropriate row of the COSEM object definitions tables [3];
- the data type of attributes of type CHOICE.

7.3 Interface class specific tests

7.3.1 Data (class_id = 1) and Register (class_id = 3)

Value: the data types allowed for the *value* attribute depend on the instance / logical name. They are listed in [3].

7.3.2 Extended register (class_id = 4)

Value: as in the case of “Data” objects.

Status: The data types allowed are defined in the interface class definition.

7.3.3 Demand register (class_id = 5)

Value: as in the case of “Data” objects.

Status: The data types allowed are defined in the interface class definition.

current_average_value and the *last_average_value*: The data types allowed depend on the instance / logical name. They are listed in [3].

7.3.4 Profile generic (class_id = 7)

7.3.4.1 General

The following elements are specific:

- write tests are not performed since writing into certain attributes may reset the buffer;
- “Profile generic” objects may have no entries or only one entry in use. Therefore, if an array of 0 or 1 is returned when the buffer is read, it is acceptable;
NOTE Examples are profiles holding readout lists.
- when reading the buffer, null_data may be returned in any column. See [1_11] 4.3.6;
- if the sort_method is not FIFO or LIFO, and the capture_objects and sort_object attributes are available for reading, it is checked that the sort_object is among the capture_objects. If these attributes are not readable, the test shall be marked as “INAPPLICABLE”.

7.3.4.2 Availability of selective access

The availability of selective access is specified by the conformance block:

- in the case of LN referencing, bit 21, selective access;

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	34/76
-----------------------	------------	----------------------	-------

- in the case of SN referencing, bit 18 parameterized access.

7.3.4.3 Selective access by range

If, for the given instance, selective access by range is supported, then it is checked that the following conditions are met:

- the *buffer* attribute is readable;
- the *capture_objects* attribute is readable;
- there are at least two *capture_objects* (two columns in the buffer);
- one of the *capture_objects* is the time attribute of the clock;

NOTE The test algorithm always uses the time attribute of the Clock object as the restricting_object. Therefore, this must be among the *capture_objects*. It can be in any of the columns.

- the number of entries in use is at least 5.

The CTT checks first the number of entries, then the availability of selective access by range.

The test is marked as “INCONCLUSIVE” if not all conditions are met. If all these conditions are met, the test is performed using the following pattern:

- skip this subtest if the number of entries is above 1 000;
- set the *from_value* and *to_value* parameters of the *restricting_object* so that this time interval covers the whole time interval marked by the first and the last time stamp found during the full reading of the buffer, and select all columns. Expect the whole buffer: number of elements in the array >0 and number of elements in the structure equal to the number of *capture_objects*;
- set the *from_value* and *to_value* parameters of the *restricting_object* so that the time interval is outside the time interval marked by the first and the last time stamp found during the full reading of the buffer, and select all columns. Expect an array of 0 elements or an appropriate error message;
- set the *from_value* and *to_value* parameters of the *restricting_object* so that the time interval covers part of the time interval marked by the first and the last time stamp found during the full reading of the buffer, and select all columns. Expect part of the buffer: number of elements in the array >0 and number of elements in the structure equal to the number of *capture_objects*;
- set the *from_value* and *to_value* parameters of the *restricting_object* so that the time interval covers part of the time interval marked by the first and the last time stamp found during the full reading of the buffer, and select two columns. Expect part of the buffer: number of elements in the array >0 and number of elements in the structure 2.

7.3.4.4 Selective access by entry

If, for the given instance selective access by entry is supported, it is checked if the following conditions are met:

- the *buffer* attribute is readable;
- the *capture_objects* attribute is readable;
- there are at least two *capture_objects* (two columns in the buffer);
- the number of entries in use is at least 5.

The CTT checks first the number of entries, then the availability of selective access by entry.

The test is marked as “INCONCLUSIVE” if not all these conditions are met. If all these conditions are met, the test shall be performed using the following pattern:

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	35/76
-----------------------	------------	----------------------	-------

- select one line and one column. Expect a single cell of the buffer: array of 1 element and structure of 1 element;
- skip this subtest if the number of entries is above 1 000. Select all entries and all columns using wild-cards. Expect the whole buffer: number of elements in the array >0 and number of elements in the structure equal to the number of capture_objects;
- select just one part of entries and columns. Expect part of the buffer: number of elements in the array >0 and number of elements in the structure >0.

7.3.5 PushTimeout Script table (class_id = 9)

In the case of the “Script table” interface class, the action_specification structure may be undefined or “dummy”, with all elements being of value 0. To avoid reporting a failure, for the service_id not only the values 1 and 2 as specified, but also the value 0 shall be accepted.

7.3.6 Association SN (class_id = 12) and Association LN (class_id = 15)

CTT 3 accepts all values specified in the IC specification:

- For Association SN: 1 to 3;
- For Association LN: 1 to 4.

The selective access feature is not tested.

The *reply_to_HLS_authentication* method is used in the case of HLS authentication.

7.3.7 Register monitor (class_id = 21)

It is tested that all thresholds are of the same type. Then if the *monitored_value* class_id, logical_name and attribute_index identify an attribute, which is defined as CHOICE, it is tested that the type of the "threshold" elements is one of the types defined in the COSEM Object definitions tables [3].

NOTE In the case of “Data”, “Register” and “Extended register” only attribute 2 can be of CHOICE type. In the case of “Demand register” attributes 2 and 3 can be of CHOICE type. (Status attributes are never monitored).

7.3.8 Push setup (class_id= 40)

7.3.8.1 General

The push operation is tested during testing “Push setup” object instances.

7.3.8.2 Test scenario

The prerequisites of testing push operation are that the IUT uses the TCP/IP profile, and the “Push setup” objects allow the CTT to parameterize and trigger the push operation from the current AA.

Once this is done, the CTT disconnects the TCP connection and with this it releases the AA. The IUT builds a new TCP connection and the push operation occurs, i.e. the DataNotification APDU is sent in a pre-established AA.

Once the DataNotification AA has been received or the PushTimeout declared in the CTI has expired, either the IUT or the CTT closes the TCP connection, the CTT opens a new TCP connection and re-establishes the AA.

7.3.8.3 Parameterisation of the “Push setup” and “Security setup” objects

The “Push setup” objects, and when cryptographic protection is required, a push “Security setup” (SecuritySetupInstanceIdForPush) object shall be visible in the same AA.

The “Push setup” object shall be pre-configured as follows:

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	36/76
-----------------------	------------	----------------------	-------

- **push_object_list** shall be readable and not empty. To elicit GBT, the set of attributes values shall be long enough;
- **send_destination_and_method** shall be readable and writeable;
- **communication_window**, **randomisation_start_interval**, **number_of_retries** and **repetition_delay** shall be writeable;
- the **push** method shall be accessible.

7.3.8.4 The push test process

The test of the push process is done as part of testing the “Push setup” object instances.

Table 31 – Push operation test

Test case	Push operation test
References	[1_11] 4.4.8. [2_7_3] 9.3.5, 9.3.9, 9.4.6.11, 9.5.
Test purpose	To verify that the push operation works as expected.
Prerequisites	The “Push setup” objects allow the CTT to parameterize and trigger the push operation
Expected result	The IUT pushes the data referenced by push_object_list .
Preamble	–
	Internal procedures
	Initialize push Do Write attribute “Push setup”. send_destination_and_method with { TCP , Destination , message } where destination is the IP address of the CTT to be used by the IUT and message = A-XDR; Do Write attribute “Push setup”. communication_window with an empty array; Do Write attribute “Push setup”. randomisation_start_interval with the value 0; Do Write attribute “Push setup”. number_of_retries with the value 0; Do Write attribute “Push setup”. repetition_delay with the value 0; Do Invoke method “Push setup”. push .
	Wait for DataNotification Close the TCP connection. NOTE With this the current AA is also released. It is re-established at the end of the subtest. Wait for the IUT to connect the TCP layer and expect DataNotification to arrive within PushTimeout. Close the TCP connection according to the value of DataNotificationToDisconnectDelay declared in the CTI.
Test body	<u>Subtest 1: Push without protection forced</u> INAPPLICABLE if: <ul style="list-style-type: none"> • the communication profile is not TCP; • if CTI.CanPush = FALSE. Do Initialize push . Do Wait for DataNotification Do Establish a confirmed AA with the parameters declared . NOTE The AA is re-established now.

Test case	Push operation test
	<p>FAILED if any of these steps fail. PASSED if DataNotification has been received and the APDU is syntactically correct.</p>
	<p><u>Subtest 2: Push with protection forced</u></p> <p>INAPPLICABLE if:</p> <ul style="list-style-type: none"> the communication profile is not TCP; if CTI.CanPush = FALSE; the application context is not ciphered; or the push “Security setup” object (CTI.SecuritySetupInstanceIdForPush) is not declared and visible in the current AA; or push “Security setup”.security_policy is not writeable. <p><u>Activate push security policy</u></p> <p>Do Read attribute push “Security setup”.security_policy. Save the value. If security_policy read is not {A+E}, then Do Invoke method push “Security setup”.security_activate with enum = 3 using {A+E}.</p> <p>Do Initialize push.</p> <p>Do Wait for DataNotification</p> <p>Do Establish a confirmed AA with the parameters declared. NOTE The AA is re-established now.</p> <p><u>Reset security_policy</u></p> <p>Do Write attribute push “Security setup” security_policy, with the value read at the beginning of subtest 2, using {A+E}.</p> <p>FAILED If any of these steps fail. PASSED if DataNotification has been received, the protection of the DataNotification APDU was {A+E} and the unciphered APDU is syntactically correct.</p>
Postamble	–
Comments	

7.3.9 Security setup (class_id = 64)

The *security_activate* and *global_key_transfer* methods are used during the SYMSEC_0 test suite. Their availability shall be declared in the CTI.

7.3.10 Dummy attributes

It may happen that some attributes of some objects are not fully parameterised in the IUT. The possible cases are listed below:

- the *register_assignment* attribute of a “Register activation” object. In this case, all elements of *object_definition* structures may be 0;
- the *mask_list* attribute of a “Register activation” object. In this case, all elements of *register_act_mask* structures may be 0;

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	38/76
-----------------------	------------	----------------------	-------

- the *sort_object* attribute of a “Profile generic” object, when the sort method is fifo or lifo. In this case, all elements of the *capture_object_definition* structure may be 0;
- the *scripts* attribute of a “Script table” object. In this case, all elements of *action_specification* structures may be 0;
- the *entries* attribute of a “Special day table” object. In this case, all elements of *spec_day_entry* structures may be 0;
- the *season_profile* attribute of an “Activity calendar” object. In this case, all elements of a *season* structure may be 0;
- the *week_profile_table* attribute of an “Activity calendar” object. In this case, all elements of a *week_profile* structure may be 0;
- the *day_profile_table* attribute of an “Activity calendar” object. In this case, all elements of a *day_profile* structure may be 0;
- the *monitored_value* attribute of a “Register monitor” object. In this case, all elements of a *value_definition* structure may be 0;
- the *actions* attribute of a “Register monitor” object. In this case, all elements of an *action_set* structure may be 0;
- the *executed_script* attribute of a “Single action schedule” object. In this case, all elements of a *script* structure may be 0.

Where any element of the structures listed is of type enumerated and the enumeration does not include 0 then the value 0 is also accepted.

7.3.11 Multiple references test

For LN referencing, see [2_7_3] 9.1.2.3.9.

Three Multiple references tests are performed. In each case, the values read together are compared to the values read separately.

- Read 10 *value* attributes of instances of classes 1, 3 or 4;
- Read a short (1 to 10 bytes) and a long (500 to 1000 bytes) attribute;
- Read the first 4 attributes of the first instance that has at least 4 attributes.

The Clock interface class shall be always excluded from the multiple references test. It shall be possible to exclude other objects in the CTI ExtraInformation declarations by class or by instance.

7.3.12 Check mandatory objects

This test is performed at the beginning of the COSEM object tests in each AA.

Table 32 – COSEM mandatory objects

Test case	COSEM mandatory objects
References	[1_11] 4.1.8, 4.4.2, 4.4.3, 4.4.4, 4.4.5. [2_7_3] 9.3.2, 9.4.4.
Test purpose	To verify that the mandatory COSEM objects are present with appropriate attribute values. This test is performed in all AAs.
Prerequisites	–
Expected result	The attribute values read are as expected for the current Association SN / Association LN object.
Preamble	All readable attributes of the ‘Association SN / Association LN’ objects have been already read

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	39/76
-----------------------	------------	----------------------	-------

CTT 3.0: ATS DLMS/COSEM Application layer – ATS COSEM interface objects –
ATS SYMSEC_0

Test case	COSEM mandatory objects
	and saved.
Test body	<p><u>Subtest 1: Current association elements</u></p> <p><u>If the application context is SN then check attributes of "Association SN object"</u></p> <p><u>security_setup_reference:</u> INAPPLICABLE if the version is < 2 or if the authentication context is no ciphering. FAILED if the value read is not the logical name of one of the "Security setup" objects visible in the current AA.</p> <p><u>user_list:</u> INAPPLICABLE if the version is < 3 or if the client user identification feature is not supported or the attribute is not readable. FAILED if the attribute is readable but the client user declared is not on the list.</p> <p><u>current_user:</u> INAPPLICABLE if the version is < 3 or if the client user identification feature is not supported or the attribute is not readable. FAILED if the attribute is not readable or if the value does not match the value declared.</p> <p><u>If the application context is LN then check attributes of "Association LN object"</u></p> <p><u>Check attributes of "Association LN object"</u></p> <p><u>associated_partners_id:</u> FAILED if the value does not match the value declared.</p> <p><u>application_context_name:</u> INAPPLICABLE if not readable. FAILED if the value does not match the value declared</p> <p><u>xDLMS_context_info:</u> INAPPLICABLE if not readable. FAILED if the value does not match the value declared.</p> <p><u>authentication_mechanism_name:</u> INAPPLICABLE if not readable. FAILED if the value does not match the value declared.</p> <p><u>association_status:</u> INAPPLICABLE if not readable. FAILED if the value is not (2) associated.</p> <p><u>security_setup_reference:</u> INAPPLICABLE if the version is < 1 or if the authentication context is no ciphering. FAILED if the value read is not the logical name of one of the "Security setup" objects visible in the current AA.</p> <p><u>user_list:</u> INAPPLICABLE if the version is <2 or if the client user identification feature is not supported or the attribute is not readable. FAILED if the attribute is readable but the client user declared is not on the list.</p>

Test case	COSEM mandatory objects
	<p><u>current user:</u> INAPPLICABLE if the version is <2 or if the client user identification feature is not supported. FAILED if the attribute is not readable or if the value does not match the value declared.</p>
	<p><u>Subtest 2: SAP assignment</u></p> <p><i>Requirements (recapitulation from [1_11] 4.1.8): In each AA, the Logical Device Name object shall be present. In the public AA (AA between the public client and the Management LD), a SAP assignment object shall be present if there is more than one Logical Device. When the SAP assignment object is present, the LDN object does not have to be present.</i></p> <p><u>If the AA is with the public client and there are multiple LDs declared:</u></p> <p>Check “SAP Assignment”.sap_assignment_list:</p> <ul style="list-style-type: none"> • all logical devices declared are present with correct SAPs and LDNs; • there are no overlapping SAPs; • there are no overlapping LDNs; • the LDNs meet the requirements of [1_11] 4.1.8.2. <p>FAILED if the SAP assignment object is not found or the requirements above are not met.</p>
	<p><u>Subtest 3: Logical Device Name</u></p> <p>Check that LDN values match the value declared.</p>
Postamble	–
Comments	

7.3.13 Interpretation of and reporting some attributes

Generally, only the communication behaviour of the interface objects is tested. The attribute values read from the IUT are not interpreted, except in the cases below.

The logical name attribute of each interface object is interpreted: the combination of the class_id and the logical name shall be correct.

In addition, in the case of the following interface objects (if available) the value of some attributes is interpreted and reported, see Table 33.

Table 33 – Attributes to be interpreted

Interface object	Attribute	Use
SAP_Assignment	SAP_assignment_list	Provides the list of logical devices available.
Logical Device Name	value	Identifies the logical device tested.
Association SN	object_list	Contains the list of objects to be tested.
	access_rights_list (from version 1)	Provides the access rights to the attributes and methods.
	security_setup_reference (from version 2)	Provides the logical name of the “Security setup” object holding and managing the security context of the AA.
	user_list (from version 3)	Hold the list of the legitimate client users.
	current_user (from version 3)	Holds the current user identifier and name.
Association LN	object_list	Contains the list of objects to be tested including the access rights to the attributes and methods
	associated_partners_id	Identifies the associated partners.
	application_context_name	Identifies the application context.
	xDLMS_context_info	Identifies the xDLMS context.
	authentication_mechanism_name	Identifies the authentication mechanism.
	security_setup_reference (from version 1)	Provides the logical name of the “Security setup” object holding and managing the security context of the AA.
	user_list (from version 2)	Hold the list of the legitimate client users.
	current_user (from version 2)	Holds the current user identifier and name.
Security setup	security_policy	Holds the security policy.
	security_suite	Holds the security suite identifier.
	client_system_title	Holds the system title of the client,
	server_sytem_title	Holds the system title of the server

The CTT reports all logical devices found in the SAP Assignment object, if present, identified by their SAPs and Logical Device Names.

The CTT interprets and reports the *value* attribute of the Logical Device Name object (if present).

The first three characters of the Logical Device Name – either read from the SAP assignment object or from a Logical Device Name object – shall match the value declared. Each logical device name within the physical device shall be unique.

At the end of testing the objects, the attributes of the AA just tested are reported.

NOTE The association with the logical name 0.x.40.0.0.255 (current Association) does not have to be present in the object list.

8 Abstract Test Suite for symmetric key security suite 0: SYMSEC_0

8.1 Requirements for the IUT for testing security

For testing the implementation of the security suite 0 the IUT shall meet the following requirements for testing, beyond the mandatory elements specified in [1_11] clause 4.1.8;

- IUTs implementing security suite 0 shall support “Security setup” version 0 or 1 and “Association SN” version 2, 3 or 4 or “Association LN” version 1, 2 or 3;
- the **security_setup_reference** attribute of the “Association SN / LN” object managing the AA tested shall be readable;
- in each “Security setup” object, the **security_policy** shall be initially set to SP_{min} declared in the CTI and it shall be readable and possibly writeable;
- AAs using a ciphered application context shall support all protection levels equal to or stronger – i.e. supporting more security services – than SP_{min};
- in each AA supporting an application context with ciphering Security Test Attributes (STAs) shall be made available:
 - STA1 shall be the value attribute of “Data” objects with data type octet-string (length >=8) with access right read_and_write;
 - Optionally, STAs shall be the value attribute of “Data” objects with data type octet-string (length >=8) with access right authenticated_read_and_write;

8.2 Capabilities supported / not supported

The SYMSEC_0 ATS supports a subset of the capabilities specified in [2_7_3].

Table 34 – Capabilities supported

Capability	Supported		Reference
	Yes	No	
Application contexts			[2_7_3] 9.4.2.2.2
Logical_Name_Referencing_With_Ciphering ::= context_id(3)	x		Idem
Short_Name_Referencing_With_Ciphering ::= context_id(4)	x		Idem
DLMS context			
Dedicated-key	x		[2_7_3] 9.3.2, 9.2.4.7.3.4
Security suite			[2_7_3] 9.2.4.2, 9.2.4.4
(0)	x		
Security policy			[1_11] 4.4.7 [2_7_3] 9.2.4.3
• security is not imposed	x		
• all messages to be authenticated	x		
• all messages to be encrypted	x		
• all messages to be authenticated and encrypted	x		
Security material			[2_7_3] 9.2.4.5
Key encryption Key (Master key)	x		[2_7_3] 9.2.4.7.3.2, Table 15
Global unicast encryption key (GUEK)	x		[2_7_3] 9.2.4.7.3.3, Table 15
Global broadcast encryption key (GBEK)		x	[2_7_3] 9.2.4.7.3.3, Table 15
(Global) Authentication key (GAK).	x		[2_7_3] 9.2.4.7.3.3, Table 15

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	43/76
-----------------------	------------	----------------------	-------

Capability	Supported		Reference
The GAK is not mandatory. If not used, it shall be declared as an empty string in the CTI.			
Dedicated (unicast) encryption key	x		[2_7_3] 9.2.4.7.3.4, Table 15
Frame counter	x		[2_7_3] 9.2.4.8.3.4.5
Optional methods			[1_11] 4.4.7
security_activate	x		
global_key_transfer	x		

8.3 Security personalisation

The IUT shall be provisioned with master key, GUEK and GAK (optional).

These keys have to be declared then in the CTI or the CTI uses default keys in which case the IUT has to be configured with those keys.

8.4 Overview of the SYMSEC_0 test cases

The Abstract Test Suite SYMSEC_0 specifies test cases to verify the correct implementation of security suite 0.

Table 35 provides an overview and comparison of the SYMSEC_0 test cases.

Table 35 – Overview of the SYMSEC_0 test cases

Test group and test cases
Test group Basic capability test
Table 36 – SYMSEC_0_BasicCap_1: Basic security capability test
Test group AES-GCM frame / invocation counter tests
Table 37 – SYMSEC_0_FraCount_1: Message replay protection
Table 38 – SYMSEC_0_FraCount_3: Send frame counter
Test group Symmetric key management tests
Table 39 – SYMSEC_0_Key_Tx_P1: Transfer and restore GUEK
Table 40 – SYMSEC_0_Key_Tx_P2: Transfer and restore GAK
Table 41 – SYMSEC_0_Key_Tx_P3: Transfer and restore GUEK and GAK
Table 42 – SYMSEC_0_Key_Tx_N1: Global key transfer, wrong key_id
Table 43 – SYMSEC_0_Key_Tx_N2: GUEK transfer, wrong wrapping
Table 44 – SYMSEC_0_DedKey_N1: Dedicated-key negative tests
Test group Secure message exchange tests
Table 45 – SYMSEC_0_SecDataX_P1: Write and read STA1 and STA2 using global and dedicated ciphering
Table 46 – SYMSEC_0_SecDataX_N1: Write and read STA1 using incorrect ciphering
Test group Secure AA release tests
Table 47 – SYMSEC_REL_N1: Release an AA using ciphered application context with insufficiently protected RLRQ
Test group Access right / security policy tests
Table 48 – SYMSEC_0_SecPol_1: Activate security policy (1)
Table 49 – SYMSEC_0_SecPol_2: Activate security policy (2)
Table 50 – SYMSEC_0_SecPol_3: Activate security policy (3)

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	44/76
-----------------------	------------	----------------------	-------

8.5 Test cases

8.5.1 Test group SYMSEC_0_BasicCap: Basic capability test

The purpose of the test group SYMSEC_0_BasicCap is to verify that basic security capabilities are correctly implemented:

- an AA that uses an application context with ciphering can be established;
- the attribute STA1 can be read and written;
- when available, the attribute STA2 can be read and written; and
- the AA can be securely released.

Table 36 – SYMSEC_0_BasicCap_1: Basic security capability test

Test case	SYMSEC_0_BasicCap_1: Basic security capability test
References	[1_11] 4.4.3, 4.4.4, 4.4.7, [2_7_3] 9.2.4.6, 9.3.2, 9.3.3.
Test purpose	To verify that the IUT supports basic security capabilities.
AA filter	All AAs with ciphering using security suite 0.
Prerequisites	–
Expected result	The AA can be established, STA1, and when available, STA2 can be written and read and the AA can be securely released.
Preamble	Do Make sure that the IUT AL is in the IDLE state.
Test body	<p><u>Subtest 1: Establish AA using an application context with ciphering</u> Do Establish a confirmed AA with the parameters declared. Do Read attribute “Association SN/ LN” security_setup_reference using {A+E}. Do Read attribute “Security setup” security_policy using {A+E}. Save the values.</p> <p><u>Subtest 2: Write STA1 using security_policy read in Subtest 1)</u> Create and save an arbitrary value R. Do Write attribute STA1 with R using the security policy read in Subtest 1).</p> <p><u>Subtest 3: Read STA1 using security_policy read in Subtest 1)</u> Do Read attribute STA1 using the security policy read in Subtest 1). FAILED if the value read is not R.</p> <p><u>Subtest 4: Write STA1 using {A+E}</u> Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E}.</p> <p><u>Subtest 5: Read STA1 using {A+E}</u> Do Read attribute STA1 using {A+E}. FAILED if the value read is not R.</p> <p><u>Subtest 6: Write STA2 using {A+E}</u> INAPPLICABLE if STA2 is not available. Create and save an arbitrary value R. Do Write attribute STA2 with R using {A+E}.</p> <p><u>Subtest 7: Read STA2 using {A+E}</u> INAPPLICABLE if STA2 is not available. Do Read attribute STA2 using {A+E}. FAILED if the value read is not R.</p> <p><u>Subtest 8: Release AA</u> Do Release AA.</p>

Test case	SYMSEC_0_BasicCap_1: Basic security capability test
Postamble	–
Comments	

8.5.2 Test group SYMSEC_0_FraCount: Frame counter

The purpose of this test group is to verify that the invocation field of the initialization vector is correctly managed, in particular that replayed APDUs are not accepted by the IUT.

Table 37 – SYMSEC_0_FraCount_1: Message replay protection

Test case	SYMSEC_0_FraCount_1: Message replay protection
References	[1_11] 6.2.30, [2_7_3] 9.2.4.8.3.4.5.
Test purpose	To verify that the IUT correctly checks the receive frame counter related to the key used.
AA filter	All AAs with ciphering using security suite 0.
Prerequisites	–
Expected result	If the CTT sends a ciphering APDU with the same key and the same frame counter more than once, the IUT shall reject it.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	<p><u>Subtest 1: Message replay protection with GUEK</u></p> <p>Create and save an arbitrary value R1. Do Write attribute STA1 with R1 using {A+E, GUEK, GAK}. Save FC1, the value of FC used. Create and save an arbitrary value R2. Do Write attribute STA1 with R2 using {A+E, GUEK, GAK} but using FC = FC1. FAILED if the procedure succeeds. Do Read attribute STA1. FAILED if the value read is not R1. Do Release AA.</p> <p><u>Subtest 2: Message replay protection with DEK</u></p> <p>If no AA is available using dedicated-ciphering, then this subtest is INAPPLICABLE. Do Establish a confirmed AA with the parameters declared. Create and save an arbitrary value R1. Do Write attribute STA1 with R1 using {A+E, DEK, GAK}. Save FC1, the value of FC used. Create and save an arbitrary value R2. Do Write attribute STA1 with R2 using {A+E, DEK, GAK} using FC = FC1. FAILED if the procedure succeeds. Do Read attribute STA1. FAILED if the value read is not R2. Do Release AA.</p>
Postamble	–
Comments	

Table 38 – SYMSEC_0_FraCount_3: Send frame counter

Test case	SYMSEC_0_FraCount_3: Send frame counter
References	[1_11] 6.2.30, [2_7_3] 9.2.4.8.3.4.5.
Test purpose	To verify that the IUT updates its send frame counter correctly. NOTE There is no COSEM object to hold the IUT send FCs.
AA filter	All AAs with ciphering using security suite 0.
Prerequisites	–
Expected result	The values of the IUT send FCs are increased with each ciphered APDUs sent by the IUT.
Preamble	–
Test body	<p><u>Subtest 1: GUEK send FC</u></p> <p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared.</p> <p>Do Read attribute STA1 using {A+E, GUEK, GAK}.</p> <p>Save FC1 = the value of FC received.</p> <p>Do Read attribute STA1 using {A+E, GUEK, GAK}.</p> <p>Save FC2 = the value of FC received.</p> <p>FAILED if $FC2 \leq FC1$.</p>
	<p><u>Subtest 2: DEK send FC</u></p> <p>If no AA is available using dedicated-ciphering, then this subtest is INAPPLICABLE.</p> <p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared.</p> <p>Do Read attribute STA1 using {A+E, DEK, GAK}.</p> <p>Save FC1 = the value of FC received.</p> <p>Do Read attribute STA1 using {A+E, DEK, GAK}.</p> <p>Save FC2 = the value of FC received.</p> <p>FAILED if $FC2 \leq FC1$.</p>
Postamble	–
Comments	

8.5.3 Test group SYMSEC_0_GlobalKeyTx: Global key transfer

The purpose of this test group is to verify that global keys can be transferred, the old keys are inactivated and the new keys become active. At the end of the tests, the original keys are restored.

Table 39 – SYMSEC_0_Key_Tx_P1: Transfer and restore GUEK

Test case	SYMSEC_0_Key_Tx_P1: Transfer and restore GUEK
References	[1_11] 4.4.7, [2_7_3] 9.2.4.7.3.
Test purpose	To verify that a new GUEK can be successfully transferred.
AA filter	An AA with ciphering using security suite 0.
Prerequisites	–
Expected result	After the new GUEK (the test GUEK) has been successfully transferred the original GUEK is inactivated.
Preamble	<p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared.</p> <p>Do Read attribute “Association SN/ LN” security_setup_reference using {A+E}.</p> <p>Create and save an arbitrary value R.</p>

Test case	SYMSEC_0_Key_Tx_P1: Transfer and restore GUEK
	Do Write attribute STA1 with R using {A+E}.
Test body	<p><u>Subtest 1: Transfer test GUEK</u></p> <p>Do Invoke method “Security setup”.global_key_transfer with key_id (0) and the correctly wrapped test GUEK using {A+E}.</p> <p>FAILED if the procedure fails.</p> <p>NOTE The new key shall be active after the method invocation has been returned with action-result = success.</p>
	<p><u>Subtest 2: Check that the original GUEK has been inactivated</u></p> <p>Do Read attribute STA1 using {A+E, GUEK, GAK}.</p> <p>FAILED if the procedure succeeds.</p>
	<p><u>Subtest 3: Check that the test GUEK has been activated.</u></p> <p>Do Read attribute STA1 using {A+E, test GUEK, GAK}.</p> <p>FAILED if the procedure fails or the value read is not R.</p>
	<p><u>Subtest 4: Restore the original GUEK</u></p> <p>Do Invoke method “Security setup”.global_key_transfer with key_id (0) and the correctly wrapped original GUEK, using {A+E, test GUEK, GAK} when global ciphering is used and using {A+E, DEK, GAK} when dedicated ciphering is used.</p> <p>FAILED if the procedure fails.</p>
Postamble	–
Comments	

Table 40 – SYMSEC_0_Key_Tx_P2: Transfer and restore GAK

Test case	SYMSEC_0_Key_Tx_P1: Transfer and restore GAK
References	[1_11] 4.4.7, [2_7_3] 9.2.4.7.3.
Test purpose	To verify that a new GAK can be successfully transferred. If GAK is not available then the test is INAPPLICABLE.
AA filter	An AA with ciphering using security suite 0.
Prerequisites	–
Expected result	After the new GAK (the test GAK) has been successfully transferred the original GAK is inactivated.
Preamble	<p>Do Make sure that the IUT AL is in the IDLE state.</p> <p>Do Establish a confirmed AA with the parameters declared.</p> <p>Do Read attribute “Association SN/ LN” security_setup_reference using {A+E}.</p> <p>Create and save an arbitrary value R.</p> <p>Do Write attribute STA1 with R using {A+E}.</p>
Test body	<p><u>Subtest 1: Transfer test GAK</u></p> <p>Do Invoke method “Security setup”.global_key_transfer with key_id (2) and the correctly wrapped test GAK using {A+E}.</p> <p>FAILED if the procedure fails.</p> <p>NOTE The new key shall be active after the method invocation has been returned with action-result = success.</p>
	<p><u>Subtest 2: Check that the original GAK has been inactivated</u></p> <p>Do Read attribute STA1 using {A+E, GUEK, GAK}</p> <p>FAILED if the procedure succeeds.</p>
	<p><u>Subtest 3: Check that the test GAK has been activated</u></p> <p>Do Read attribute STA1 using {A+E, GUEK, test GAK}.</p> <p>FAILED if the procedure fails or the value read is not R.</p>
	<u>Subtest 4: Restore the original GAK</u>

Test case	SYMSEC_0_Key_Tx_P1: Transfer and restore GAK
	Do Invoke method “Security setup”. global key transfer with key_id (2) and the correctly wrapped original GAK, using {A+E, GUEK, test GAK} when global ciphering is used and using {A+E, DEK, test GAK} when dedicated ciphering is used. FAILED if the procedure fails.
Postamble	–
Comments	

Table 41 – SYMSEC_0_Key_Tx_P3: Transfer and restore GUEK and GAK

Test case	SYMSEC_0_Key_Tx_GUEK_GAK_P1: Transfer and restore GUEK and GAK
References	[1_11] 4.4.7, [2_7_3] 9.2.4.7.3.
Test purpose	To verify that a new GUEK and a new GAK can be successfully transferred together. If GAK is not available then the test is INAPPLICABLE.
AA filter	An AA with ciphering using security suite 0.
Prerequisites	–
Expected result	After the new GUEK (the test GUEK) and the new GAK (the test GAK) has been successfully transferred the original GUEK and GAK is inactivated.
Preamble	Do Make sure that the IUT AL is in the IDLE state . Do Establish a confirmed AA with the parameters declared . Do Read attribute “Association SN/ LN” security_setup_reference using {A+E}. Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E}.
Test body	<u>Subtest 1: Transfer test GUEK and test GAK</u> Do Invoke method “Security setup”. global key transfer with key_id (0, 2) and the correctly wrapped test GUEK and test GAK using {A+E}. FAILED if the procedure fails. NOTE The new keys shall be active after the method invocation has been returned with action-result = success.
	<u>Subtest 2: Check that the original GUEK has been inactivated</u> Do Read attribute STA1 using {A+E, GUEK, test GAK}. FAILED if the procedure succeeds.
	<u>Subtest 3: Check that the original GAK has been inactivated by</u> Do Read attribute STA1 using {A+E, test GUEK, GAK}. FAILED if the procedure succeeds.
	<u>Subtest 4: Check that the test GUEK and test GAK have been activated</u> Do Read attribute STA1 using {A+E, test GUEK, test GAK}. FAILED if the procedure fails or the value read is not R.
	<u>Subtest 5: Restore the original GUEK and GAK</u> Do Invoke method “Security setup”. global key transfer with key_id (0, 2) and the correctly wrapped original GAK, using {A+E, test GUEK, test GAK} when global ciphering is used and using {A+E, DEK, test GAK} when dedicated ciphering is used. FAILED if the procedure fails.
Postamble	–
Comments	

Table 42 – SYMSEC_0_Key_Tx_N1: Global key transfer, wrong key_id

Test case	SYMSEC_0_Key_Tx_GUEK_N1: Global key transfer, wrong key_id
References	[1_11] 4.4.7, [2_7_3] 9.2.4.7.3.3

DLMS User Association	2015-06-18	DLMS UA 1001-6 V 1.3	49/76
-----------------------	------------	----------------------	-------

Test case	SYMSEC_0_Key_Tx_GUEK_N1: Global key transfer, wrong key_id
Test purpose	To verify that the IUT correctly handles the key_id parameter of "Security setup". global_key_transfer and does not accept a wrong key_id. This test is performed in one AA with ciphering.
AA filter	An AA with ciphering using security suite 0.
Prerequisites	–
Expected result	The key transfer fails.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Read attribute "Association SN/ LN". security_setup_reference using {A+E}. Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E}.
Test body	<u>Subtest 1: Transfer test GUEK, but using an invalid key_id</u> Do Invoke method "Security setup". global_key_transfer with key_id (0x80) and a correctly wrapped test GUEK using {A+E}. FAILED if procedure succeeds.
	<u>Subtest 2: Check that the GUEK has not been inactivated</u> Do Read attribute STA1 using {A+E, GUEK, GAK}. FAILED if the procedure fails or the value read is not R.
Postamble	–
Comments	

Table 43 – SYMSEC_0_Key_Tx_N2: GUEK transfer, wrong wrapping

Test case	SYMSEC_0_Key_Tx_GUEK_N1: Transfer GUEK, with wrong wrapping
References	[1_11] 4.4.7, [2_7_3] 9.2.4.7.3.3
Test purpose	To verify that the IUT correctly handles "Security setup". global_key_transfer.key_data.key_wrapped and does not accept a key wrapped with an incorrect master key.
AA filter	An AA with ciphering using security suite 0.
Prerequisites	–
Expected result	The key transfer fails.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Read attribute "Association SN/ LN" security_setup_reference using {A+E}. Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E}.
Test body	<u>Subtest 1: Transfer test GUEK wrapped with incorrect master key</u> Do Invoke method "Security setup". global_key_transfer with key_id (0) and a correctly wrapped test GUEK but using a wrong master-key, using {A+E}. FAILED if the procedure succeeds.
	<u>Subtest 2: Check that the GUEK has not been inactivated</u> Do Read attribute STA1 using {A+E, GUEK, GAK}. FAILED if the procedure fails.
Postamble	–
Comments	

8.5.4 Test group SYMSEC_0_DedKey_N1: Dedicated-key transfer

The purpose of this test group is to verify that if a dedicated-key is not transferred or not correctly transferred, then dedicated-key ciphering APDUs cannot be used to access COSEM objects.

Table 44 – SYMSEC_0_DedKey_N1: Dedicated-key negative tests

Test case	SYMSEC_0_DedKey_N1: Dedicated-key negative tests
References	[2_7_3] 9.2.4.6, 9.3.2
Test purpose	Verify that incorrect usages of dedicated-key are rejected.
AA filter	An AA with ciphering using security suite 0 and dedicated key.
Prerequisites	–
Expected result	If the InitiateRequest is not properly protected, or if the dedicated-key is not correct, the proposed AA is rejected and services using dedicated-key ciphering cannot be used.
Preamble	–
Test body	<p><u>Subtest 1: Transfer DEK with InitiateRequest not ciphered</u> Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared but with InitiateRequest not ciphered. FAILED if the procedure succeeds. NOTE This test is essentially the same as the one specified in Table 18 – APPL_OPEN_9: xDLMS InitiateRequest: dedicated-key, but it is performed on an AA that supports the usage of dedicated keys.</p> <p><u>Subtest 2: Transfer DEK with InitiateRequest using {E}</u> Do Establish a confirmed AA with the parameters declared but with InitiateRequest encrypted only. FAILED if the procedure succeeds.</p> <p><u>Subtest 3: Transfer DEK with InitiateRequest using {A}</u> Do Establish a confirmed AA with the parameters declared but with InitiateRequest authenticated only. FAILED if the procedure succeeds.</p> <p><u>Subtest 4: Transfer too short DEK</u> Do Establish a confirmed AA with the parameters declared but transfer a dedicated-key of length 15 octets. FAILED if the procedure succeeds.</p> <p><u>Subtest 5: Transfer too long DEK</u> Do Establish a confirmed AA with the parameters declared but transfer a dedicated-key of length 17 octets. FAILED if the procedure succeeds.</p> <p><u>Subtest 6: Transfer DEK of length zero</u> Do Establish a confirmed AA with the parameters declared but transfer a dedicated-key of length 0. NOTE The usage flag of the dedicated-key element clearly indicates that the dedicated-key is present, so a length of 0 octets is a wrong length; it cannot be interpreted as if the dedicated-key was absent. FAILED if the procedure succeeds.</p> <p><u>Subtest 7: Use ded-ciphering services without transferring DEK first</u> Do Establish a confirmed AA with the parameters declared but without transferring a dedicated-key. Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E, GUEK, GAK}. Do Read attribute STA1 using {A+E, GUEK, GAK}, but the tag of the APDU shall indicate</p>

Test case	SYMSEC_0_DedKey_N1: Dedicated-key negative tests
	dedicated ciphering. FAILED if the procedure succeeds.
	Subtest 8: Use an old DEK in a new AA Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Save DEK1, the dedicated-key transferred. Do Release AA. Do Establish a confirmed AA with the parameters declared but transferring a different dedicated-key DEK2. Do Read attribute STA1 using {A+E, DEK1, GAK}. FAILED if the procedure succeeds.
Postamble	–
Comments	

8.5.5 Test group SYMSEC_0_SecDataX: Secure data exchange

The purpose of the test group SYMSEC_0_SecDataX_P is to verify that COSEM object attributes and methods can be accessed with APDUs correctly ciphered as requested by the access rights and the security policy prevailing and cannot be accessed with APDUs that are not correctly or insufficiently protected.

Table 45 – SYMSEC_0_SecDataX_P1: Write and read STA1 and STA2 using global and dedicated ciphering

Test case	SYMSEC_0_SecDataX_P1: Write and Read STA1 and STA2 using global and dedicated ciphering
References	[2_7_3] 9.2.4.6.
Test purpose	To verify that data exchange takes place in line with the access rights and security policy in force.
AA filter	An AA with ciphering using security suite 0, supporting dedicated keys and in which STA2 is available. If not, the test is performed in an AA using ciphering but not supporting dedicated keys or in which STA2 is not available. In this case, the related subtests are INAPPLICABLE.
Prerequisites	–
Expected result	Accessing STA1 with service requests with protection equal to or higher than SP_{min} should be possible. Accessing STA2 with authenticated and encrypted service requests should be possible.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	<u>Subtests with global ciphering</u>
Write STA1 {A}	<u>Subtest 1: Write STA1 using {A} with global ciphering</u> INAPPLICABLE if SP_{min} is not 0 or 1. Create and save an arbitrary value R. Do Write attribute STA1 with R using {A, GUEK, GAK}. FAILED if the procedure fails.
Read STA1 {A}	<u>Subtest 2: Read STA1 using {A} with global ciphering</u> INAPPLICABLE if SP_{min} is not 0 or 1. Do Read attribute STA1 using {A, GUEK, GAK}. FAILED if the procedure fails or the value read is not R.

CTT 3.0: ATS DLMS/COSEM Application layer – ATS COSEM interface objects –
ATS SYMSEC_0

Test case	SYMSEC_0_SecDataX_P1: Write and Read STA1 and STA2 using global and dedicated ciphering
Write STA1 {E}	<u>Subtest 3: Write STA1 using {E} with global ciphering</u> INAPPLICABLE if SP_{min} is not 0 or 2. Create and save an arbitrary value R. Do Write attribute STA1 with R using {E, GUEK, GAK}. FAILED if the procedure fails.
Read STA1 {E}	<u>Subtest 4: Read STA1 using {E} with global ciphering</u> INAPPLICABLE if SP_{min} is not 0 or 2. Do Read attribute STA1 using {E, GUEK, GAK}. FAILED if the procedure fails or the value read is not R.
Write STA1 {A+E}	<u>Subtest 5: Write STA1 using {A+E} with global ciphering</u> Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E, GUEK, GAK}. FAILED if the procedure fails.
Read STA1 {A+E}	<u>Subtest 6: Read STA1 using {A+E} with global ciphering</u> Do Read attribute STA1 using {A+E, GUEK, GAK}. FAILED if the procedure fails or value read is not R.
Write STA2 {A+E}	<u>Subtest 7: Write STA2 using {A+E} with global ciphering</u> Create and save an arbitrary value R. Do Write attribute STA2 with R using {A+E, GUEK, GAK}. FAILED if the procedure fails.
Read STA2 {A+E}	<u>Subtest 8: Read STA2 using {A+E} with global ciphering</u> Do Read attribute STA2 using {A+E, GUEK, GAK}. FAILED if the procedure fails or if the value read is not R.
Read STA2 {E}	<u>Subtest 9: Read STA2 using {E} with global ciphering</u> INAPPLICABLE if SP_{min} is not 0 or 2. Do Read attribute STA2 using {E, GUEK, GAK}. FAILED if the procedure succeeds.
Read STA2 {A}	<u>Subtest 10: Read STA2 using {A} with global ciphering</u> INAPPLICABLE if SP_{min} is not 0 or 1. Do Read attribute STA2 using {A, GUEK, GAK}. FAILED if the procedure fails or if the value read is not R.
	<u>Subtests with dedicated ciphering</u>
Write STA1 {A+E}	<u>Subtest 11: Write STA1 using {A+E} with dedicated ciphering</u> INAPPLICABLE if dedicated ciphering is not supported. Create and save an arbitrary value R. Do Write attribute STA1 with R using {A+E, DEK, GAK}. FAILED if the procedure fails.
Read STA1 {A+E}	<u>Subtest 12: Read STA1 using {A+E} with dedicated ciphering</u> INAPPLICABLE if dedicated ciphering is not supported. Do Read attribute STA1 using {A+E, DEK, GAK}. FAILED if the procedure fails or the value read is not R.
Write STA2 {A+E}	<u>Subtest 13: Write STA2 using {A+E} with dedicated ciphering</u> INAPPLICABLE if dedicated ciphering is not supported. Create and save an arbitrary value R. Do Write attribute STA2 with R using {A+E, DEK, GAK}. FAILED if the procedure fails.

Test case	SYMSEC_0_SecDataX_P1: Write and Read STA1 and STA2 using global and dedicated ciphering
Read STA2 {A+E}	<u>Subtest 14: Read STA2 using {A+E} with dedicated ciphering</u> INAPPLICABLE if dedicated ciphering is not supported. Do Read attribute STA2 using {A+E, DEK, GAK}. FAILED if the procedure fails or the value read is not R.
Read STA2 {E}	<u>Subtest 15: Read STA2 using {E} with dedicated ciphering</u> INAPPLICABLE if dedicated ciphering is not supported or if SP_{min} is not 0 or 2. Do Read attribute STA2 using {E, DEK, GAK}. FAILED if the procedure succeeds.
Read STA2 {A}	<u>Subtest 16: Read STA2 using {A} with dedicated ciphering</u> INAPPLICABLE if dedicated ciphering is not supported or if SP_{min} is not 0 or 1. Do Read attribute STA2 using {A, DEK, GAK}. FAILED if the procedure fails or the value read is not R.
Postamble	–
Comments	

Table 46 – SYMSEC_0_SecDataX_N1: Write and read STA1 using incorrect ciphering

Test case	SYMSEC_0_SecDataX_N1: Write and read STA1 using incorrect ciphering
References	[2_7_3] 9.2.4.6.
Test purpose	To verify that the IUT correctly handles the fields of the ciphering APDUs and the AES-GCM authenticated decryption function is implemented correctly.
AA filter	An AA with ciphering using security suite 0.
Prerequisites	–
Expected result	STA1 cannot be read if the protection is not correct.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.
Test body	<u>Subtest 1: Read STA1 using {A+E, test GUEK, GAK}</u> This subtest is only performed if the AA supports GAK. Do Read attribute STA1 using {A+E, test GUEK, GAK}. FAILED if the procedure succeeds.
	<u>Subtest 2: Read STA1 using {A+E, GUEK, test GAK}</u> This subtest is only performed if the AA supports GAK. Do Read attribute STA1 using {A+E, GUEK, test GAK}. FAILED if the procedure succeeds.
	<u>Subtest 3: Read STA1 using {A+E, GUEK, GAK} with dedicated ciphering</u> This subtest is only performed if the AA supports dedicated-keys else it is INAPPLICABLE. Do Read attribute STA1 using {A+E, GUEK, GAK}. The tag of the APDU shall show dedicated ciphering. FAILED if the procedure succeeds.
	<u>Subtest 4: Read STA1 using {A+E} but SH indicating GBEK</u> Do Read attribute STA1 using {A+E} but in the Security Header bit 6 indicating GBEK. FAILED if the procedure succeeds.
	<u>Subtest 5: Read STA1 using {A+E} but SH indicating security suite 15</u> Do Read attribute STA1 using {A+E} but in the Security Header bits 0..3 set (indicating security suite 15). FAILED if the procedure succeeds.

Test case	SYMSEC_0_SecDataX_N1: Write and read STA1 using incorrect ciphering
	Subtest 6: Read STA1 using {A+E} but one bit of the ciphertext flipped Do Read attribute STA1 using {A+E} but with one bit of the ciphertext flipped. FAILED if the procedure succeeds.
	Subtest 7: Read STA1 using {A+E} but last byte of authentication tag missing Do Read attribute STA1 using {A+E}, but last byte of the authentication tag missing.
Postamble	–
Comments	

8.5.6 Test group SYMSEC_0_SecRel: Secure AA release

The purpose of the test group SYMSEC_0_SecRel is to verify that the AA is not released if the RLRQ is insufficiently protected.

Table 47 – SYMSEC_REL_N1: Release an AA using ciphered application context with insufficiently protected RLRQ

Test case	SYMSEC_REL_N1: Release an AA using ciphered application context with insufficiently protected RLRQ
References	[2_7_3] 9.1.2.2, 9.3.3, 9.4.2, 9.4.5, 9.5, 10.2.6.2, 10.3.6.1.
Test purpose	To verify that the AA using ciphered application context is not released if the RLRQ is insufficiently protected. This test is performed in each AA with ciphering and supporting RLRQ/RLRE.
AA filter	All AAs using ciphering with security suite 0 and supporting RLRQ/RLRE.
Prerequisites	–
Expected result	The AA remains in the Associated state.
Preamble	–
Test body	<p>Subtest 1: Send RLRQ with user-information containing InitiateRequest with no ciphering Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Release AA but send RLRQ with user-information containing InitiateRequest with no ciphering. FAILED if the procedure succeeds. Do Check that the AA is in the Associated state.</p> <p>Subtest 2: Send RLRQ with user-information containing InitiateRequest using {E} Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Release AA but send RLRQ with user-information containing InitiateRequest using {E} FAILED if the procedure succeeds. Do Check that the AA is in the Associated state.</p> <p>Subtest 3: Send RLRQ with user-information containing InitiateRequest using {A} Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Release AA but send RLRQ with user-information containing InitiateRequest using {A}. FAILED if the procedure succeeds. Do Check that the AA is in the Associated state.</p> <p>Subtest 4: Send RLRQ with user-information containing InitiateRequest using {A+E, DEK, GAK} If dedicated key is not supported this subtest is INAPPLICABLE. Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared.</p>

	Do Release AA but send RLRQ with user-information containing authenticated and encrypted InitiateRequest using {A+E, DEK, GAK}. FAILED if the procedure succeeds. Do Check that the AA is in the Associated state .
Postamble	–
Comments	

8.5.7 Test group SYMSEC_0_SecPol: Security policy

The purpose of the test group is to verify that security policies can be activated and that data exchange takes place in line with the security policy in force.

Table 48 – SYMSEC_0_SecPol_1: Activate security policy (1)

Test case	SYMSEC_0_SecPol_1: Activate security policy (1)
References	[1_11] 4.4.7 [2_7_3] 9.2.4.3
Test purpose	Activate security policy (1): all messages to be authenticated. Verify that data exchange takes place in line with the security policy in force.
AA filter	All AAs using ciphering with security suite 0.
Prerequisites	For this test the current security policy shall be (0).
Expected result	After setting the security policy, data can be accessed only in line with the security policy in force.
Preamble	Do Make sure that the IUT AL is in the IDLE state . Do Establish a confirmed AA with the parameters declared . Do Read attribute "Security setup". security_policy using {A+E}. If it is not (0), the test is INAPPLICABLE.
Test body	<u>Subtest 1: Write STA1 with no ciphering</u> Create and save random value R. Do Write attribute STA1 with R with no ciphering. FAILED if the procedure fails.
	<u>Subtest 2: Strengthen security policy to (1) all messages to be authenticated</u> Do Invoke method "Security setup". security_activate with data ::= enum (1), with no ciphering. FAILED if the procedure fails.
	<u>Subtest 3: Read STA1 with no ciphering</u> Do Read attribute STA1 with no ciphering. FAILED if the procedure succeeds.
	<u>Subtest 4: Read STA1 using {A}</u> Do Read attribute STA1 using {A}. FAILED if the procedure fails or if the value read is not R.
Postamble	<u>Reset security_policy to SP_{min}</u> If security_policy attribute is not writeable then skip the postamble. Do Write attribute "Security setup" security_policy , with SP _{min} declared, using {A+E}.
Comments	

Table 49 – SYMSEC_0_SecPol_2: Activate security policy (2)

Test case	SYMSEC_0_SecPol_2: Activate security policy (2)
References	[1_11] 4.4.7 [2_7_3] 9.2.4.3
Test purpose	Activate security policy (2): all messages to be encrypted. Verify that data exchange takes place in line with the security policy in force.
AA filter	All AAs using ciphering with security suite 0.
Prerequisites	For this test, the security_policy shall be (0).
Expected result	After setting the security policy, data can be accessed only in line with the security policy in force.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Read attribute "Security setup".security_policy using (A+E). If it is not (0) the test is INAPPLICABLE.
Test body	<u>Subtest 1: Write STA1 with no ciphering</u> Create and save random value R. Do Write attribute STA1 with R with no ciphering. FAILED if the procedure fails.
	<u>Subtest 2: Strengthen security policy to (2) all messages to be encrypted</u> Do Invoke method "Security setup".security_activate with data::= enum (2) with no ciphering. FAILED if the procedure fails.
	<u>Subtest 3: Read STA1 with no ciphering</u> Do Read attribute STA1 with no ciphering. FAILED if the procedure succeeds.
	<u>Subtest 3: Read STA1 using {A}</u> Do Read attribute STA1 using {A}. FAILED if the procedure succeeds.
	<u>Subtest 4: Read STA1 using {E}</u> Do Read attribute STA1 using {E}. FAILED if the procedure fails or the value read is not R.
Postamble	<u>Reset security_policy to to SP_{min}</u> If security_policy attribute is not writeable then skip the postamble. Do Write attribute "Security setup" security_policy, with SP _{min} declared, using {A+E}.
Comments	

Table 50 – SYMSEC_0_SecPol_3: Activate security policy (3)

Test case	SYMSEC_0_SecPol_3: Activate security policy (3)
References	[1_11] 4.4.7 [2_7_3] 9.2.4.3
Test purpose	Activate security policy (3): all messages to be authenticated and encrypted. Verify that data exchange takes place in line with the security policy in force.
AA filter	All AAs using ciphering with security suite 0.
Prerequisites	For this test, the security_policy shall be 0, 1 or 2.
Expected result	After setting the security policy, data can be accessed only in line with the security policy in force.
Preamble	Do Make sure that the IUT AL is in the IDLE state. Do Establish a confirmed AA with the parameters declared. Do Read attribute "Security setup".security_policy using {A+E}
Test body	<u>Subtest 1: Write STA1 using {A+E}</u> Create and save random value R. Do Write attribute STA1 with R using {A+E}. FAILED if the procedure fails.
	<u>Subtest 2: Strengthen security policy to (3) all messages to be authenticated and encrypted</u> Do Invoke method "Security setup".security_activate with data::= enum (3), using {A+E}. FAILED if the procedure fails.
	<u>Subtest 3: Read STA1 with no ciphering</u> Do Read attribute STA1 with no ciphering. FAILED if the procedure succeeds.
	<u>Subtest 4: Read STA1 using {A}</u> Do Read attribute STA1 using {A}. FAILED if the procedure succeeds.
	<u>Subtest 5: Read STA1 using {E}</u> Do Read attribute STA1 using {E}. FAILED if the procedure succeeds.
	<u>Subtest 6: Read STA1 using {A+E}</u> Do Read attribute STA1 using {A+E}. FAILED if the the procedure fails or if the value read is not R.
	<u>Subtest 7: Try to decrease the strength of the security policy</u> Do Invoke method "Security setup" security_activate with data ::= 0 using {A+E}. FAILED if the procedure succeeds.
	<u>Subtest 8: Verify that the security policy did not change</u> Do Read attribute "Security setup".security_policy using {A+E}. FAILED if the value is not enum (3).
Postamble	<u>Reset security_policy to to SP_{min}</u> If security_policy attribute is not writeable then skip the postamble. Do Write attribute "Security setup" security_policy, with SP _{min} declared, using {A+E}.
Comments	

Annex A
(informative)
Conformance Test Information (CTI) template

```
// CTI_template for CTI 3.0 and up.  
// The CTI syntax is described in the help-file.  
// The CTI template is best viewed in the ''CTI'' tab of CTT.  
  
// Identification, mandatory structure  
Identification = {  
    // Manufacturer, mandatory string, the manufacturers name  
    Manufacturer = "\Manufacturer's name"  
  
    // FLAGId, mandatory string, the FLAG id registered for the manufacturer  
    FLAGId = "\XYZ"  
  
    // Type, mandatory string, any string that identifies the type of the IUT  
    Type = "\Any type"  
  
    // IUTIsModule, optional boolean, default FALSE  
    // Allows to declare that the IUT is a stand-alone communication module.  
    // If TRUE, then the CTT does not test if a logical device with SAP = 1
```

```
// (Management Logical Device) is present.
IUTIsModule = FALSE

// SerialNr, mandatory string, the serial number of the IUT
SerialNr = "\12345-67.89"

// Comment, optional string, there may be several, any additional information
Comment = "\This is a comment"
Comment = "\This is another comment"
}

// TestOptions, mandatory structure, options chosen for the test session
TestOptions = {
    // CommunicationProfile, mandatory enum, the communication profile used for testing
    // HDLC or TCP
    CommunicationProfile = TCP

    // ReferencingMethod, mandatory if the CTI declares both LN and SN associations, else
    // optional. It selects the referencing method used for testing.
    // LONG_NAMES or SHORT_NAMES
    ReferencingMethod = SHORT_NAMES
}
```



```
// DoNotTest, optional, set of enum, specify a set of checks to avoid
// when present ATTRIBUTES_TYPE_CHOICES, then do not check the type of attributes having a CHOICE type.
// when present ATTRIBUTES_VALUES, then do not check the values (ranges, and sub-ranges) of attributes.
// If the purpose of the test session is to obtain a Certification, then DoNotTest has to be omitted or
// empty.
DoNotTest = [ATTRIBUTES_VALUES]
}

// IMPORTANT NOTE
// *****
// All the elements listed below (until 'LogicalDevice') can be declared at three levels: in an
// 'Association[]' element, in a 'LogicalDevice[]' element or globally. During the test, when a value
// is needed, the test process first looks in the Association[], then in the parent LogicalDevice element[],
// and finally at the global level. It uses the declaration first found. If no declaration is made,
// and a default is available, then the default value is used, else an exception is raised.

// ClassExtraInfo, optional structure, specifies extra information related to an
// interface class. There may be several ClassExtraInfo declarations.
ClassExtraInfo = {
    // ClassId, mandatory integer, the class id of the interface class.
    ClassId = 8
```

```
// Version, mandatory integer, the version of the interface class
Version = 0

// AttributeExtraInfo, mandatory structure, information for the attribute concerned.
// there may be several AttributeInfo declarations, for different attributes
AttributeExtraInfo = {
    // AttributeId, mandatory integer, identifies the attribute concerned.
    AttributeId = 2

    // AccessRights, optional enum, specifies the access right to the attribute concerned, it overrides the
    // object-list specified access rights. One of NO_ACCESS, READ_ONLY, WRITE_ONLY, READ_WRITE
    // AUTHENTICATED_READ_ONLY, AUTHENTICATED_WRITE_ONLY, AUTHENTICATED_READ_AND_WRITE
    AccessRights = READ_WRITE

    // Selective AccessSelectors, optional set of integer, specifies the selector allowed for the attribute
    //concerned.
    // It overrides the object-list specified selectors.
    // Selective access is tested only on "Profile generic" buffer attribute
    SelectiveAccessSelectors = [1,2]

    // WriteTestData, optional string, value to be written to the attribute when performing write test.
    // The syntax is described in the CTT help file
```

```
WriteTestData = '\<Data><OctetString Value = "000000000000000000000000"></Data>'
}
}

// InstanceExtraInfo, optional structure, specifies extra information related to an instance.
// There may be several InstanceExtraInfo declarations
InstanceExtraInfo = {
    // LogicalName, mandatory string, logical name of the instance
    LogicalName = "0-0:25.9.0.255"

    // ClassId, mandatory integer, class id of the instance
    ClassId = 40

    // BaseName, optional integer, base name of the concerned instance. Needed in
    // some test cases when using the short names referencing method
    BaseName = 0xE800

    // CanPush, optional boolean, default FALSE, used only on instances of class "Push setup".
    // It indicates that a push can be performed by invoking the push method.
    CanPush = TRUE

    // SecuritySetupInstanceIdForPush, optional string, Used only on instances of class "Push setup"
```

```
// that ''CanPush''. Identifies the "Security setup" object that defines the security context in which the
//push is performed by this instance. The security_policy attribute of the "Security setup" must be readable.
// The security_activate method is invoked to activate security A+E on the Data-Notification APDU.
SecuritySetupInstanceIdForPush = "0-0:43.0.1.255"

// AttributeExtraInfo, optional structure, information for the attribute concerned.
// There may be several AttributeExtraInfo declarations, for different attributes.
// The content is the same as AttributeInfo in ClassExtraInfo
AttributeExtraInfo = {
    //
}

// MediaIdentifiers, mandatory, set of enum, specifies the media / energy types supported by the AA(s).
// It determines the possible values of value group A for objects that may be abstract or
// energy type related. Set of ABSTRACT, ELECTRICITY, HCA, COOLING, HEAT, GAS, COLDWATER, HOTWATER
MediaIdentifiers = [ABSTRACT, ELECTRICITY]

// SystemTitle, string of length 8, The - unique - System Title of the IUT, starting with the FLAG ID,
// Mandatory when ciphered contexts or HLS mechanism (5) GMAC is declared.
SystemTitle = "\XYZ00000"
```

```
// MasterKey, optional string of length 16, needed only when ciphered contexts or HLS_GMAC are used
// Default = 00112233445566778899AABBCCDDEEFF
MasterKey = "11223344556677881122334455667788"

// GUEK, optional string of length 16, needed only when ciphered contexts or HLS_GMAC are used
// Default = 000102030405060708090A0B0C0D0E0F
GUEK = "88776655443322118877665544332211"

// GAK, optional string of length 16, needed only when ciphered contexts or HLS_GMAC are used
// Default = D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF
GAK = "A0A1A2A3A4A5A6A7A8A9AAABACADAEAF"

// SecurityPolicyMinimum, optional enum, The minimum level of security-policy required by the IUT.
// The default is NO_SECURITY. One of NO_SECURITY, AUTHENTICATION, ENCRYPTION,
// AUTHENTICATION_AND_ENCRYPTION
SecurityPolicyMinimum = AUTHENTICATION

// SecurityActivateMethodSupported, optional boolean, default TRUE, declares if the security
// activate method of the security setup object can be invoked.
// When FALSE, several test cases are INAPPLICABLE.
SecurityActivateMethodSupported = TRUE
```

```
// GlobalKeyTransferMethodSupported, optional boolean, default TRUE, declares if the global key transfer method
// of the security setup object can be invoked. When FALSE, several test cases are INAPPLICABLE.
GlobalKeyTransferMethodSupported = TRUE

// SecurityPolicyWriteable, optional boolean, declares if the attribute security_policy of the
// security setup object is writeable or not, default TRUE.
// If TRUE, the value of "Security setup".security_policy is restored to its initial value at the
// end of any test that causes its modification.
// If FALSE, then the security_policy is not restored to its previous value and therefore the IUT cannot be
//used for repeated testing.
SecurityPolicyWritable = FALSE

// UseDedicatedKey, optional boolean, default FALSE.
// Indicates if the AA(s) supports the usage of dedicated keys.
// If TRUE, a dedicated-key is generated and sent to the IUT when establishing the AA,
// and ded-ciphered APDUs are used.
UseDedicatedKey = TRUE

// STA1 (security test attribute 1), structure, mandatory for SYMSEC tests, specifies a readable
// and writable octet-string attribute. During the test, STA1 is written with ASCII values of the
// form STA1_XXX, where XXX is random.
STA1 = {
```

```
// LogicalName, mandatory string
LogicalName = "0000806403FF"

// ClassId, mandatory, integer
ClassId = 1

// AttributeId, mandatory string
AttributeId = 2
}

// STA2, (security test attribute 2), optional structure, specifies a readable and writable
// octet-string attribute with AUTHENTICATED_READ_AND_WRITE access-rights. During the test,
// STA2 is written with ASCII values of the form STA2_XXX, where XXX is random.
STA2 = {
    // Has the same members as STA1
}

// ClientUser, optional integer, if defined, the value is put in the Calling-AE-Invocation-Id of the AARQ.
ClientUser = 1

// IllegalClientUser, optional integer, value of an illegal client user.
// If not specified then the value ClientUser+1 is assumed to be an illegal client user.
```

```
IllegalClientUser = 12

// Secret, mandatory string for AA(s) using either LLS or HLS_SHA1 or HLS_MD5.
// In the case of LOW_LEVEL_SECURITY AA(s), specifies Password needed to establish the association.
// In the case of HIGH_LEVEL_SECURITY_MD5 and HIGH_LEVEL_SECURITY_SHA1 associations,
// specifies the HLS secret necessary to calculate f(Stoc) and f(CtoS).
Secret = "\The secret"

// DefaultAccessRights, optional structure, specifies the default access rights (of attributes). It
// can be used when access rights are not specified in the Association object or using extra information.
DefaultAccessRights = {
    // Attributes, mandatory enum, default access rights to attributes, one of NO_ACCESS,
    // READ_ONLY, WRITE_ONLY, READ_WRITE, AUTHENTICATED_READ_ONLY, AUTHENTICATED_WRITE_ONLY,
    // AUTHENTICATED_READ_AND_WRITE
    Attributes = READ_WRITE
}

// ConformanceBlock, mandatory set of enum, specifies the set of xDLMS services and
// capabilities advertized by the IUT in its AARE.InitiateResponse.NegotiatedConformance
// member when the AARQ.InitiateRequest.ProposedConformance proposes the full set of services and
// capabilities for the referencing method used. Set of GENERAL_PROTECTION, GENERAL_BLOCK_TRANSFER,
// READ, WRITE, UNCONFIRMED_WRITE, ATTRIBUTE0_SUPPORTED_WITH_SET, PRIORITY_MGMT_SUPPORTED,
```



```
// ATTRIBUTE0_SUPPORTED_WITH_GET, BLOCK_TRANSFER_WITH_GET_OR_READ, BLOCK_TRANSFER_WITH_SET_OR_WRITE,
// BLOCK_TRANSFER_WITH_ACTION, MULTIPLE_REFERENCES, INFORMATION_REPORT, DATA_NOTIFICATION,
// PARAMETRIZED_ACCESS, GET, SET, SELECTIVE_ACCESS, EVENT_NOTIFICATION, ACTION
ConformanceBlock = [
    ACTION,
    SELECTIVE_ACCESS,
    SET,
    GET,
    DATA_NOTIFICATION,
    MULTIPLE_REFERENCES,
    BLOCK_TRANSFER_WITH_SET_OR_WRITE,
    BLOCK_TRANSFER_WITH_GET_OR_READ,
    GENERAL_BLOCK_TRANSFER,
    GENERAL_PROTECTION]

// ServerMaxReceivePduSize, mandatory integer [12..], the maximum length of the APDU the IUT supports,
// is only used in test APPL_OPEN_6 to avoid sending a too long APDU to the IUT.
ServerMaxReceivePduSize = 500

// RLRQSupported, optional, boolean, default FALSE, indicates if the AA(s) can be released using the
// RLRQ/RLRE APDU exchange.
RLRQSupported = TRUE
```

```
// PushTimeout, mandatory when push is performed, integer, the delay (ms), awaited for the DataNotification
// after having triggered the push.
PushTimeout = 60000

// DataNotificationToDisconnectDelay, optional integer, the delay (ms) between the reception of
// the DataNotification over a connection and the closing of that connection. By default, the delay
// is 0, i.e. the connection is closed immediately after receiving the DataNotification. If set to -1,
// then the connection is not closed, i.e. it has to be closed by the IUT, else the connection is
// closed after the specified delay.
DataNotificationToDisconnectDelay = 100

// PreEstablished, optional, boolean, default FALSE, indicates that the AA(s) is(are) Pre-Established
// and will be tested only during the COSEM tests.
PreEstablished = FALSE

// LogicalDevice, mandatory array of struct, specifies a logical device. The numbering of logical
// Array index starts from 0.
LogicalDevice[0] = {

    // Enabled, optional boolean, allows encluding/excluding all the AAs of the LD from the testing process.
    // Default TRUE
```

```
Enabled = TRUE

// Name, mandatory string, logical device name, 4-16 octets starting with the FLAG ID.
// They must be all different for each LD declared.
Name = "\XYZaaabbbcccd"

// ServerSAP, mandatory integer, the SAP of the Logical Device.
// The SAP of LogicalDevice [0] shall be 1 except if IUTIsModule is true.
// The SAP of each LD must be different.
ServerSAP = 1

// Association, mandatory array of struct, specifies the AA(s) of this LD
// Array index starts from 0
Association [0] = {
    // Enabled, optional boolean, allows including/excluding the AA from the testing process.
    // Default = TRUE
    Enabled = TRUE

    // ClientSAP, mandatory integer, The client SAP. It must be different for each AA in the LD using
    // the same referencing method. In the Management Logical Device (Server SAP = 1), an association
    // with ClientSAP = 0x10 (public client) must be declared
    ClientSAP = 0x10
```

```
// ApplicationContextName, mandatory enum, the application context name. One of SHORT_NAMES,  
// LONG_NAMES, SHORT_NAMES_WITH_CIPHERING, LONG_NAMES_WITH_CIPHERING  
ApplicationContextName = SHORT_NAMES  
  
// AuthenticationMechanismName, mandatory enum, the authentication mechanism. One of NO_SECURITY,  
// LOW_LEVEL_SECURITY, HIGH_LEVEL_SECURITY_MD5, HIGH_LEVEL_SECURITY_SHA1, HIGH_LEVEL_SECURITY_GMAC  
AuthenticationMechanismName = NO_SECURITY  
}  
  
Association [1] = {  
    // Another association  
    // ...  
}  
}  
  
LogicalDevice [1] = {  
    // Another logical device  
    // ...  
}  
  
// HDLCProfile, optional structure, required if the TestOptions.CommunicationProfile = HDLC.
```

```
HDLCProfile = {

    // PhysicalLayer, mandatory structure, parameters of the physical layer.
    PhysicalLayer = {

        // HDLCBaud, mandatory (if DIRECT_HDLC) integer, the baud supported by the IUT.
        HDLCBaud = 9600

        // OpeningMode, mandatory enum, the opening mode of the HDLC layer, one of MODE_E, DIRECT_HDLC,
        // WAKEUP_MODE_E
        OpeningMode = MODE_E

        // ModeEIdString, optional (if MODE_E) string, Mode E device address string.
        ModeEIdString = "\aString"

        // ModeEDelay1, optional integer, during MODE_E opening: delay in ms awaited after having received the
        // ACK from the IUT, default 0.
        ModeEDelay1 = 0

        // ModeEDelay2, optional integer, during MODE_E opening: delay in ms awaited after having sent the
        // ACK to the IUT, default 300.
        ModeEDelay2 = 300
    }
}
```

```
// ModeEDelay3, optional integer, during MODE_E opening: delay in ms awaited after having switched
// the baud, default 1000.
ModeEDelay3 = 1000

// LastOutputToWakeUpDelay, optional integer, used only if OpeningMode is WAKEUP_MODE_E, is the delay
// (in ms) awaited between the last HDLC frame before a wake-up procedure and the wake-up procedure
LastOutputToWakeUpDelay = 30000
}

// DataLinkLayer, mandatory structure, parameters of the data link layer.
DataLinkLayer = {

// AddressingSchemes, mandatory set of enum, the addressing schemes supported by the IUT.
// Set of ONE_BYTE_ADDRESSING, TWO_BYTES_ADDRESSING, FOUR_BYTES_ADDRESSING
AddressingSchemes = [ONE_BYTE_ADDRESSING, TWO_BYTES_ADDRESSING]

// ServerLowerMacAddress, mandatory when TWO_BYTES_ADDRESSING and FOUR_BYTES_ADDRESSING are declared,
// integer, the physical address of the IUT
ServerLowerMacAddress = 0x11

// InformationFieldLength, optional integer [32..2030], the lengths( end and transmit) of the information
```

```
//field proposed during HDLC parameter negotiation. By default, nothing is proposed, and the default is
// then 128.
InformationFieldLength = 200

// InactivityTimeout, mandatory integer, inactivity timeout of the IUT in ms
InactivityTimeout = 120000

// InterframeTimeout, mandatory integer, inter frame timeout of the IUT in ms.
InterFrameTimeout = 100

// ResponseTimeout, mandatory integer, response timeout in ms
ResponseTimeout = 2000

// DISCToNDMTimeout, mandatory integer, time needed by the IUT to go to the NDM state after having received
// a DISC frame, in ms.
DISCToNDMTimeout = 1000
}
}

// TCPProfile, optional structure, required if the TestOptions.CommunicationProfile = TCP
TCPProfile = {
```

```
// ServerTCPPort, mandatory integer, the port where the IUT listen to, the value 4059 has been assigned
// by the IANA for DLMS/COSEM.
ServerTCPPort = 4059

// ConnectTimeout, mandatory integer, time (ms) allowed to perform a TCP connection with the IUT.
ConnectTimeout = 5000

// ResponseTimeout, mandatory integer, time (in ms) allowed for a response.
ResponseTimeout = 2000

// DisconnectToConnectDelay, mandatory integer, delay (ms) between a disconnection and a reconnection of the
// TCP layer.
DisconnectToConnectDelay = 2000
}
// END CTI_template
```