EXCERPT FROM

Companion Specification for Energy Metering

COSEM

Conformance Test Process

DLMS User Association
Table of Contents

1. Foreword ................................................................. 5

2. Scope ........................................................................... 6

3. Introduction ................................................................... 7
   3.1 Referenced documents ........................................... 7
   3.2 Terms, Definitions and Abbreviations .................... 7

4. DLMS/COSEM conformance testing .............................. 8
   4.1 What is OSI conformance testing? .......................... 8
   4.2 DLMS/COSEM conformance testing ....................... 8

5. The DLMS/COSEM conformance test plans ......................... 9
   5.1 Test plans and test tool ........................................ 9
   5.2 Physical layer tests ............................................... 10
   5.3 Data link layer tests ............................................. 11
   5.4 COSEM application layer tests ............................ 12
   5.5 COSEM objects tests .......................................... 13

6. The DLMS/COSEM conformance test tool ......................... 15
   6.1 General ............................................................... 15
   6.2 Main features ....................................................... 16
   6.3 PICS and PIXIT ..................................................... 17
       6.3.1 The PICS and PIXIT files ............................... 17
       6.3.2 PICS contents ............................................... 17
       6.3.3 PIXIT contents .............................................. 18
   6.4 Conformance Test Executor .................................... 18
       6.4.1 General Requirements ................................. 18
       6.4.2 Handling the tool – the Graphical User Interface (GUI) 19
       6.4.3 Preparation and execution of conformance tests .... 20
       6.4.4 Tool Validation ............................................. 21
       6.4.5 Handling of CTT copyright aspects ................. 22
       6.4.6 Preparation of implementations for conformance testing 22
       6.4.7 PICS template and example ........................... 22
       6.4.8 PIXIT template and example ......................... 25

7. DLMS/COSEM compliance certification process .................. 28
   7.1 Licensing the CTT .................................................. 28
   7.2 Performing the conformance test ............................ 28
   7.3 Verification and acceptance of test report ................. 28
   7.4 The “DLMS/COSEM compliant” mark ....................... 29
   7.5 Testing different configurations ............................. 29
   7.6 Disclaimer .......................................................... 29
   7.7 Arbitration .......................................................... 29

8. List of certified equipment .............................................. 30

9. The DLMS/COSEM conformance test maintenance process ... 31
   9.1 Maintenance: use case 1 – introducing a new standard OBIS code 31
   9.2 Maintenance: use case 2 – modification of an existing test ... 31
9.3 Maintenance: use case 3 – adding a test for a new standard feature ...........................................31
9.4 Maintenance: use case 4 – revision of the specification .................................................................32

10. Physical layer test plan ..................................................................................................................33
10.1 Scope ...........................................................................................................................................33

11. Data Link Layer test plan .............................................................................................................34
11.1 Foreword .....................................................................................................................................34
11.2 Scope ...........................................................................................................................................34

12. COSEM application layer test plan .............................................................................................35
12.1 Foreword .....................................................................................................................................35
12.2 Scope ...........................................................................................................................................35

13. Interface objects test plan ............................................................................................................36
13.1 Foreword .....................................................................................................................................36
13.2 Scope ...........................................................................................................................................36

14. List of standardised OBIS codes ...................................................................................................37
14.1 Objective .......................................................................................................................................37
14.2 Explanations .................................................................................................................................37

Figures

Figure 1 - Generic building blocks of a device .......................................................................................6
Figure 2 - CTT architecture and environment .......................................................................................16
1. Foreword

Copyright


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.

Acknowledgement

The actual document has been established by the WG CT of the DLMS UA.
2. Scope

This document describes the DLMS/COSEM conformance testing process developed by the DLMS UA. The main elements of the conformance test process are:

- the conformance test plans;
- the conformance test tool (CTT);
- the certification process;
- the maintenance of a list of compliant equipment;
- the conformance test maintenance process.

The functionality of a DLMS/COSEM device (e.g. meter), which is available at its communication interface(s) and the way the functions and the data can be accessed from the outside is specified in [1] (Blue Book) and [2] (Green Book). The DLMS/COSEM specification follows a Client/Server approach. The complex functionality of the device is divided into generic building blocks. The COSEM specifications follow a three-step approach as illustrated in Figure 1.

Step 1: The meter model and data identification (data model), based on object modelling technique;

Step 2: The mapping of the model into protocol data units (PDU).

Step 3: The transportation of the bits and bytes through the communication channel.

This document describes the necessary means to test if a DLMS/COSEM server can be operated with a DLMS/COSEM client, both conforming to the specifications described above.
3. Introduction

3.1 Referenced documents

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Reference No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>DLMS UA 1000-1</td>
<td>COSEM Identification System and Interface Objects</td>
</tr>
<tr>
<td></td>
<td>DLMS UA 1001-1</td>
<td>COSEM Conformance Test Process (this document)</td>
</tr>
<tr>
<td>[4]</td>
<td>DLMS UA 1001-3</td>
<td>Conformance Test Specification - Data link layer (this document, clause 11)</td>
</tr>
<tr>
<td>[5]</td>
<td>DLMS UA 1001-4</td>
<td>Conformance Test Specification – Application layer (this document, clause 12)</td>
</tr>
</tbody>
</table>

Note 1: for the actual version of these documents, see the relevant documents or check the DLMS UA website at http://www.dlms.com/

Note 2: for IEC and CEN standards containing the DLMS/COSEM specification see the Bibliography section of this document.

3.2 Terms, Definitions and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AARE</td>
<td>Application Association Response</td>
</tr>
<tr>
<td>COSEM</td>
<td>Companion Specification for Energy Metering</td>
</tr>
<tr>
<td>CT</td>
<td>Conformance Test (for DLMS/COSEM)</td>
</tr>
<tr>
<td>CTT</td>
<td>Conformance Test Tool (for DLMS/COSEM)</td>
</tr>
<tr>
<td>DLMS</td>
<td>Device Language Message Specification (Distribution Line Message Specification)</td>
</tr>
<tr>
<td>DLMS UA</td>
<td>DLMS User Association</td>
</tr>
<tr>
<td>DUT</td>
<td>Device Under Test</td>
</tr>
<tr>
<td>EuroDCS</td>
<td>EuroDCS Energiedaten AG (handling the commercial aspects of the CTT)</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HDLC</td>
<td>High-level Data Link Control</td>
</tr>
<tr>
<td>IC</td>
<td>Interface Classes</td>
</tr>
<tr>
<td>IUT</td>
<td>Implementation under test</td>
</tr>
<tr>
<td>OBIS</td>
<td>Object Identification System</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>PICS</td>
<td>Protocol Implementation Conformance Statement</td>
</tr>
<tr>
<td>PIXIT</td>
<td>Protocol Implementation eXtra Information for Testing</td>
</tr>
</tbody>
</table>

© Copyright 2001 - 2002 DLMS User Association
4. DLMS/COSEM conformance testing

4.1 What is OSI conformance testing?

The concept and methodology of OSI Conformance testing is described in the ISO/IEC 9646 series of standards.

Conformance testing is a verification that an implementation meets the formal requirements of an OSI standard. During the test phase the implementation is referred to as the Implementation Under Test (IUT).

The primary objective of conformance testing is to increase the probability that different product implementations actually interoperate.

No amount of testing can give a full guarantee of successful interworking. The exhaustive testing of every possible aspect of protocol behaviour is unrealistic and impractical for technical and economical reasons. Conformance testing can however give a reasonable degree of confidence that an implementation which passes the test, will comply with the requirements in communication with other systems. As such, conformance testing can be regarded as a prerequisite for interworking.

4.2 DLMS/COSEM conformance testing

To enable claiming compliance of server implementations to the DLMS/COSEM specification, the DLMS User Association provides a conformance test process. The main elements of this process are:

- the conformance test plans, based on the DLMS/COSEM specification, which describe the tests to be performed;
- the conformance test tool (CTT) to allow performing the tests;
- the certification process, with a “DLMS UA Compliant” mark;
- a list of compliant equipment;
- a conformance test maintenance process.
5. The DLMS/COSEM conformance test plans

5.1 Test plans and test tool

The test plans, developed by the DLMS UA Working Group conformance testing, provide a textual description of the tests to be executed.

As on the servers there are no Points of Control and Observation (PCO-s) provided, and the direct access to the individual layers is not possible, the test plans are based on specifying complete messages, HDLC frames including the necessary APDU-s, to be sent by the test tool to the IUT and specifying the expected response. The testing of a higher layer assumes that the lower layers are operating correctly.

A distinction is made between positive and negative test cases:

- Positive tests cases are tests, in which the test client sends an implemented, syntactically and semantically valid service request in the correct state of the server.
- Negative test cases are tests, in which the test client sends a service request, which is not implemented, syntactically or semantically not correct, or which is not sent in the correct state of the server.

A client which is conformant to the specification will not send syntactically invalid messages or inopportune messages to the Server.

For the result, a distinction is made between:

- 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 with an observed outcome;
- syntactically invalid test event: a test event which syntactically is not allowed by the protocol recommendation;
- 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.

A verdict is a statement of “pass”, “fail” or “inconclusive” concerning conformance of an IUT with respect to a test case which has been executed and which is specified in the abstract test suite.

A PASS verdict is given when the observed outcome satisfies the test purpose and is valid with respect to the standard and with respect to the PICS.

A FAIL verdict is given when the observed outcome is syntactically invalid or inopportune with respect to the standards or the PICS.

An INCONCLUSIVE verdict is given when the observed outcome is valid with respect to the relevant standard but prevents the test purpose from being accomplished.

The test plan structures the test cases in a hierarchical way. They include abstract test cases, test groups and test suites.

In the conformance test tool, each test case is translated into a test script, which is a formal description of the test case. These are also organised in a hierarchical way, into sub-tests, tests groups and test suites.

An example for the Application layer is shown in the table below:
The test suites are structured in a hierarchical way:

The test scripts include the instructions to import the necessary information from the PICS and PIXIT files and to generate the verdict. When the PICS and PIXIT files are loaded, the tests become executable.

The conformance test consists of a collection of test cases drawn from a standard test suite in such a way as to test each COSEM object and communication service for which support is claimed (positive test). Further, an appropriate response is expected in cases of:

- erroneous objects or services;
- objects and services that are not implemented (negative test).

### 5.2 Physical layer tests

The physical layer test plan includes the following tests. For detailed test specifications, see [3].
5.3 Data link layer tests

The HDLC test suite contains the following tests. For detailed test specifications, see [4].

<table>
<thead>
<tr>
<th>Test groups</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDLC_1: Incorrect frame field values</td>
<td>Subtest 0: make sure that we have a connected device</td>
</tr>
<tr>
<td></td>
<td>Subtest 1: Frame is not properly bounded by two flags</td>
</tr>
<tr>
<td></td>
<td>Subtest 2: Frame is too short</td>
</tr>
<tr>
<td></td>
<td>Subtest 4: Unknown HDLC addresses</td>
</tr>
<tr>
<td></td>
<td>Subtest 5: ALL_STATION &amp; NO_STATION addresses</td>
</tr>
<tr>
<td></td>
<td>Subtest 6: ALL_STATION address with poll = 1</td>
</tr>
<tr>
<td></td>
<td>Subtest 7: Unknown value in format type sub-field of the Frame format field</td>
</tr>
<tr>
<td></td>
<td>Subtest 8: Wrong value in frame length sub-field of the Frame format field</td>
</tr>
<tr>
<td></td>
<td>Subtest 9: Unknown command identifier in control field</td>
</tr>
<tr>
<td></td>
<td>Subtest 10: Wrong N(R) sequence number in control field</td>
</tr>
<tr>
<td></td>
<td>Subtest 11: Wrong N(S) sequence number in control field</td>
</tr>
<tr>
<td></td>
<td>Subtest 12: Too long information field</td>
</tr>
<tr>
<td>HDLC_2: Mode change to NRM</td>
<td>Subtest 1: SNRM without information field</td>
</tr>
<tr>
<td></td>
<td>Subtest 2: Check NRM by sending a RR frame</td>
</tr>
<tr>
<td></td>
<td>Subtest 3: SNRM with information field = Client.MaxInfoLengthReceive parameter negotiation</td>
</tr>
<tr>
<td></td>
<td>Subtest 4: SNRM with information field = ClientWindowSizeReceive parameter negotiation</td>
</tr>
<tr>
<td></td>
<td>Subtest 5: Window size length encoding</td>
</tr>
<tr>
<td>HDLC_3: Wrong command frames, I frames in NDM mode</td>
<td>Subtest 1: I frame in NDM</td>
</tr>
<tr>
<td>HDLC_4: Mode change to NDM</td>
<td>Subtest 2: DISC frame in NDM</td>
</tr>
<tr>
<td>HDLC_5: I frame exchange</td>
<td>Subtest 1: Send small frames with Seg = true and Poll = true and monitor responses</td>
</tr>
<tr>
<td>HDLC_6: Inter-frame time-out</td>
<td></td>
</tr>
</tbody>
</table>
### 5.4 COSEM application layer tests

The COSEM application layer test suite contains the following tests. For detailed tests specifications, see [5].

<table>
<thead>
<tr>
<th>Test groups</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDLC_7: Long delay without any data exchange (Inactivity time-out)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test groups</th>
<th>Test cases</th>
</tr>
</thead>
</table>

#### Appl_01: Connection establishment : Protocol-version
- Subtest 1: Protocol version not present
- Subtest 2: Protocol version present and containing the default value
- Subtest 3: Protocol version present but not containing the default value

#### Appl_02: Connection establishment : Application-context-name
- Subtest 1: Using the wrong context
- Subtest 2: Uses an unknown application context

#### Appl_03: Called and calling – titles, qualifiers and identifiers
- Subtest 1: Parameters not included
- Subtest 2: All parameters included with dummy values

#### Appl_04: Connection establishment : authentication parameter
- Subtest 1: NO_SECURITY association with parameters
- Subtest 2: LOW_SECURITY association without using authentication
- Subtest 3: Associate using the specified parameters
- Subtest 4: Associate using the wrong sender ACSE requirement
- Subtest 5: Associate using the wrong mechanism name
- Subtest 6: Associate using the wrong calling authentication value

#### Appl_06: Connection establishment: implementation information

#### Appl_07: DLMS-Initiate.request: Dedicated Key

#### Appl_08: DLMS-Initiate.request: quality of service

#### Appl_09: DLMS-Initiate.request: dlms-version-number
- Subtest 1: ProposedDlmsVersionNumber = 1
- Subtest 2: ProposedDlmsVersionNumber = 255
- Subtest 3: ProposedDlmsVersionNumber = PIXIT declaration minus 1
- Subtest 4: ProposedDlmsVersionNumber = PIXIT declaration plus 1

#### Appl_10: DLMS-Initiate.request: conformance
- Subtest 1: Request all services
- Subtest 2: Request services that are not implemented

#### Appl_11: DLMS-Initiate.request: client-max-receive-pdu-size
- Subtest 1: Propose a non null size < 12
- Subtest 2: Propose the largest possible valid size

#### Appl_12: Graceful releasing of an association
### 5.5 COSEM objects tests

The testing of the COSEM interface objects is based on the information obtained by reading the `object_list` attribute of the Association LN object or the `object_list` attribute and the (optional) `get_attributes&methods` method of the Association SN object of the logical device under test. The following information is necessary:

- for each interface class, the version implemented;
- the COSEM object instances implemented (logical names);
- the attributes and methods implemented in each object (methods are not tested in the first version of the CTT);
- the access rights to each attributes and methods;
- the availability and parameters of selective access to attributes.

If necessary, this information can be provided by supplying the `ClassExtraInfo` and/or the `Instance-ExtraInfo` in the PICS.

The COSEM objects test suite contains the following tests. For detailed test specifications see [6].

<table>
<thead>
<tr>
<th>Test groups</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSEM_01_0_Data</td>
<td></td>
</tr>
<tr>
<td>COSEM_03_0_Register</td>
<td></td>
</tr>
<tr>
<td>COSEM_04_0_Extended Register</td>
<td></td>
</tr>
<tr>
<td>COSEM_05_0-Demand Register</td>
<td></td>
</tr>
<tr>
<td>COSEM_07_1_Profile Generic</td>
<td></td>
</tr>
<tr>
<td>COSEM_08_0_Clock</td>
<td></td>
</tr>
<tr>
<td>COSEM_09_0_Script Table</td>
<td></td>
</tr>
<tr>
<td>COSEM_11_0_Special Days</td>
<td></td>
</tr>
<tr>
<td>COSEM_12_0_Association SN</td>
<td></td>
</tr>
<tr>
<td>COSEM_12_1_Association SN</td>
<td></td>
</tr>
<tr>
<td>COSEM_15_0_Association LN</td>
<td></td>
</tr>
<tr>
<td>COSEM_17_0_SAP Assignment</td>
<td></td>
</tr>
<tr>
<td>COSEM_19_0_IEC Local Port Setup</td>
<td></td>
</tr>
<tr>
<td>COSEM_20_0_Activity Calendar</td>
<td></td>
</tr>
<tr>
<td>COSEM_21_0_Register Monitor</td>
<td></td>
</tr>
<tr>
<td>COSEM_23_0_IEC_HDLC_Setup</td>
<td></td>
</tr>
<tr>
<td>COSEM_23_1_IEC_HDLC_Setup</td>
<td></td>
</tr>
<tr>
<td>COSEM_27_0_PSTN Modem Configuration</td>
<td></td>
</tr>
<tr>
<td>COSEM_28_0_PSTN Auto Answer</td>
<td></td>
</tr>
<tr>
<td>Test groups</td>
<td>Test cases</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>COSEM_29_0_PSTN Auto Dial</td>
<td></td>
</tr>
<tr>
<td>COSEM_Multiple references</td>
<td></td>
</tr>
</tbody>
</table>
6. The DLMS/COSEM conformance test tool

6.1 General

The CTT is an application that automatically performs pre-defined test cases on devices implementing the DLMS/COSEM specification.

The CTT runs on a PC to which the IUT is connected via a serial line. Before each test run, the CTT takes as input two machine-readable text files, the Protocol Implementation Conformance Statement (PICS) and the Protocol Implementation eXtra Information (PIXIT) that describe the implementation and relevant device parameters necessary for the test. The PICS and the PIXIT is provided by the manufacturer or the vendor of the device and they can be edited using the tool. For testing COSEM interface objects the information is obtained by reading the object_list attribute of the Association LN object or the object_list attribute and the (optional) get_attributes&methods method of the Association SN object of the logical device under test.

The CTT handles, interprets and runs sets of test routines.

The CTT allows the selection of test groups (individual tests for a given layer) and test suites (all tests for a given layer) to be performed. For a valid conformance test, all tests must be selected. The CTT will automatically select the tests, which can be performed, based on the information provided in the PICS, PIXIT and the Association SN/LN objects. Tests, which can not be performed, will be marked as skipped (e.g. HDLC tests in case of direct connection according to IEC 62056-21).

The server sends the messages defined by the test plan to the server and analyses the responses. Based on the result, it assigns the verdict for each test case.

During the test, the line traffic and the log can be viewed and they can be saved for further analysis.

At the end of the test run, the CTT generates a test report showing which test groups have run and the result of each test group. The test report is a digitally signed text file, allowing to verify its authenticity.

The CTT provides fully automatic testing. The test routines are published but can not be modified.

Figure 2 shows graphically the elements involved in the testing process.
6.2 Main features

- The purpose of the CTT is to test if a server can be operated with a client conforming to the DLMS/COSEM standards ("Blue Book" [1] and "Green Book" [2]).
- The CTT is limited to the testing of the server. It includes the testing of all standardised COSEM objects, their attributes and methods. Manufacturer-specific interface classes are not tested. Manufacturer specific instances of interface classes are tested to the extent possible. The CTT also includes the testing of all services of the protocol layers transporting the model.
- The CT is limited to the server’s functionality as presented at the communication interface. Other functions of the server are not covered by the tests.
- The test can be performed through direct connection via an optical port, a current loop, or an RS232 port. If IEC 62056-21 mode E (with baud rate switching) opening is used, HDLC tests are skipped (because the server falls back to the initial state after each disconnection of the data link layer). The tests are implicitly done during testing the higher layers. Multi-drop configurations are not tested.
- The test tool consists of abstract test cases, defined by the WG CT and implemented as test routines, which can not be modified by the operator. The test routines are readable by a machine and by a human being.
- The test tool includes positive and negative tests. Positive tests cases are tests, in which the test client sends an implemented, syntactically and semantically valid service request in the correct state of the server. Negative test cases are tests, in which the test client sends a service request, which is not implemented, syntactically or semantically not correct, or which is not sent in the correct state of the server. For positive tests, the response shall be a syntactically and semantically valid service response as defined in the standard. For negative test cases, the response shall be a valid error message as defined in the standard.
- The CT can be performed by the manufacturer or by third parties, with a minimal human intervention. To support fault-finding during the test process, the test cases can be selected individually by the operator.
- The information concerning the implementation and the extra information necessary to execute the tests are provided in the form of a human- and machine readable PICS and PIXIT files. The test tool provides the necessary tools to edit the PICS and PIXIT files.
- Testing of interface classes is based on the information obtained from the Association LN/Association SN objects.
- To monitor the execution of the test, a line traffic monitor is provided.
- A test log is provided, listing the actions executed by the CTT. Fatal error messages are displayed in the Log.
- The result of the CT consists of the test report. The test report identifies the CTT version and the licence owner, the manufacturer, the device under tests, and displays the result of the tests. It includes the rele-
vant sections of the PICS and PIXIT files. The test report is readable by a human being. The authenticity
of the test report is provable using a digital signature.

6.3 PICS and PIXIT

6.3.1 The PICS and PIXIT files

The Protocol Implementation Conformance Statement (PICS) is a document made by the vendor of a DLMS/COSEM implementation. It is based on the PICS pro-forma, which is part of the test plans. It states the capabilities and options, which have been implemented and the features, which have been omitted. Thus, the implementation can be tested for conformance against the relevant requirements listed in the PICS, and against those requirements only.

The Protocol Implementation eXtra Information for Testing (PIXIT) is a statement made by the vendor of a DLMS/COSEM implementation. It is based on the PIXIT pro-forma, which is part of the test plans. It provides additional information about the IUT necessary for the test laboratory to conduct the tests. This includes addressing information, specific values to complement the range of values stated in the PICS etc.

The PICS and PIXIT files have the following common characteristics:

1. They are text files.
2. There is one PICS and one PIXIT file for each physical device tested.
3. The PICS and PIXIT file is both human and computer readable.
4. Relevant parts of the PICS and PIXIT files are included in the test report.

6.3.2 PICS contents

A DLMS/COSEM PICS is a public document that is available for use by any interested party. A DLMS/COSEM PICS includes the following information:

1. Manufacturer identification (mandatory);
2. Physical device identification (mandatory);
3. For all physical devices, identification of all logical devices implemented:
4. For all logical devices, the following information must be provided:
   - ClassExtraInfo (optional): gives extra information for each interface classes implemented, including their class Id, version and information on the attributes: AttributeId, AccessRights, WriteTestData, SelectiveAccessSelectors. The ClassExtraInfo is used for providing information necessary for testing the interface classes, when this information cannot be obtained by reading the Association object. It can also be used to provide information on the values to be used during write tests.
   - InstanceExtraInfo (optional): gives extra information on object instances, including the Logical name, and information on the attributes. The use of InstanceExtraInfo is the same as the use of ClassExtraInfo, but more specific.
   - The ClassExtraInfo and InstanceExtraInfo declaration may be valid globally for the physical device, for a given logical device or for a given association.
   - Manufacturer (optional): as above;
   - Device (optional): as above;
     - Association (mandatory): Mandatory, specifies the parameters of an association. At least two Associations must be specified. Association[0] must have a NO_SECURITY MechanismName and Association[1] must have a LOW_LEVEL_SECURITY MechanismName.
     - ContextName (mandatory): specifies the association context name
     - MechanismName (mandatory): Specifies the authentication mechanism name
• Conformance (mandatory): specifies the list of COSEM data communication services implemented (contents of the COSEM Conformance block);
• OBIS field identifiers (mandatory): specifies the list of standard OBIS identifiers to be used for testing. Possible values are Abstract, Electricity (current), HCA, Cooling, Gas, Cold water, Hot water (future).

See the PICS template in Annex A.

6.3.3 PIXIT contents

A DLMS/COSEM PIXIT includes the following information:

1. Manufacturer identification (mandatory);
2. Device identification (mandatory, same as in the PICS);
3. Physical layer information (mandatory):
   • PortType (mandatory): defines the port type used for testing – optical or RS 232;
   • EchoingPort (mandatory): TRUE or FALSE
   • OpeningMode (mandatory): defines the protocol opening mode: IEC 62056-21 Mode E or direct HDLC (IEC 62056-46)
   • ModeEIdstring (optional): defines the string using during mode E opening
   • HDLCBaud (optional): defines the baud rate for direct HDLC
   • Remark: if Mode E is chosen the data link layer is not testable and will be indicated as 'skipped' in the test report.
4. Data link layer information: defines the HDLC based data link layer parameters and addresses:
   • Inactivity time out (mandatory);
   • InterframeTimeout (mandatory);
   • ResponseTimeout (mandatory);
   • DISCToNDM timeout (optional): defines the time between the receipt of a DISC frame and the transition to NDM state (default 0).
   • AddressingSchemes (optional): defines the addressing scheme used in the server: one byte addressing, two bytes addressing, four bytes addressing;
   • ServerLowerMacAddress (mandatory): specifies the physical device address.
5. Application layer information:
   • LogicalDevice: specifies the parameters for each logical device.
     • ClientMacAddress (mandatory): specifies the MAC layer address of the Client;
     • ServerUpperMacAddress (mandatory): specifies the MAC address of the logical device;
     • DLMS version number (optional): specifies the DLMS version number used by the server. Default value is 6;
     • Password (optional): specifies the password needed for the association.

See the PIXIT template in Annex B.

6.4 Conformance Test Executor

6.4.1 General Requirements

6.4.1.1 Hardware requirements

The CTT program runs on any Windows - Intel machine, Pentium or newer, with 200 MHz or faster, minimally 64 MB RAM.

The tool requires a free serial interface (RS232 port) which shall not be shared by any other applications during running time.
6.4.1.2 Operating system requirements

The operating systems supported are Windows NT Workstation 4.0, Windows 2000 Professional and Windows XP.

6.4.1.3 Language

The CTT test cases are written in the "Object Pascal" programming language. A description of this language is available at:


6.4.2 Handling the tool – the Graphical User Interface (GUI)

The CTT is equipped with a Graphical User Interface. The operation of the GUI is explained below.

Main menu

- **Run**
  - Start and run the tests.
- **Abort**
  - Abort the running test.
- **Exit**
  - Close CTT

View

- **PICS and PIXIT Editors**
  - Show the PICS and PIXIT files in their respective editors. Clicking right opens a context sensitive menu. Allows to load, save and edit the PICS/PIXIT files.
  
  A modified file is automatically saved before running the tests. The last loaded files are automatically reloaded at startup.

- **Report and Log**

  Clicking right in the Report's windows opens a contextual menu. Select Display source (enabled when text is selected) to open the Test sources window and display the test case whose name is selected. Select Save... to save the Report to a text file.

  The Log displays a listing of the actions executed by CTT. Fatal error messages are also displayed in the Log.

  Clicking right in the Log's windows opens a contextual menu. Select Display source (enabled when text is selected) to open the Test sources window and display the test case whose name is selected. Select Save... to save the Log to a text file.

- **Test Sources**
  - Show the test case source code in a floating window.

  The Test Sources window is divided in two panes. The left pane shows the hierarchy of the test cases. The right pane displays the source code of the test case selected in the left pane.

- **Line Traffic**
The Line Traffic window displays the bytes (in hex-string notation) sent (green) and received (red) over the serial line.

Clicking right in the window opens a contextual menu. Select Clear to erase all the data.

Options
Opens the Option dialog box. Enables to select the PC port used for connecting the IUT, to select the enabled test cases (All, None, All HDLC, All Appl, All COSEM). The item Miscellaneous allows to select the fonts and to enable logging HDLC frames and COSEM PDU-s.

About
Help
Provides extensive Help.
About
Provides version, licensing and support information about the CTT.

6.4.3 Preparation and execution of conformance tests

6.4.3.1 General
Test items are servers (e.g. meters). They are connected via the RS232 port of the PC.

6.4.3.2 Preparation of the IUT
The IUT shall be prepared for conformance testing with a sufficient amount of data such that all the tests can be performed sufficiently. If the amount of data is insufficient, the CTT will mark the test as “SKIPPED”. The criterion for skipping is part of the test specification.

6.4.3.3 Preparation and handling of the PICS document
The CTT supports the preparation of the PICS file.

- The PICS is a text file external to the tool.
- It can be edited with the CTT using the PICS template provided.
- The tool checks the syntax of the PICS and detects errors.
- Before each test, the PICS is automatically read by the tool.
- The contents of the PICS is included into the test report.

To prepare the PICS, Start CTT, select View/PICS and PIXIT Editors, in the upper Pane (PICS) click right and select Load... Then navigate to the file PICS_TEMPLATE.txt. Fill up all the PICS file according to the instructions. Then Click right, select Save... and give a file name to the PICS file. From now on, CTT will automatically load this file at start-up.

6.4.3.4 Preparation and handling of the PIXIT document
The CTT supports the preparation of the PIXIT file.

- The PIXIT is a text file external to the tool.
- It can be edited with the CTT using the PIXIT template provided.
- The tool checks the syntax of the PIXIT and detects errors.
- Before each test, the PIXIT is automatically read by the tool.
- The contents of the PIXIT is included into the test report.

To prepare the PIXIT, follow the same steps as for the PICS, starting from the PIXIT_TEMPLATE.txt.

6.4.3.5 Running the CT
Connect the device under test to a PC-serial port that is not reserved by another application. If you are using an optical probe, make sure to know if the probe is echoing or not.
• Select **Options**. On the **Communication** page select the port number to which the device under test is connected. On the **Enabled tests** page select the tests to be performed.
• Select **View/Line Traffic**, resize and move the window to a convenient position.
• Select **Run/Run**.

### 6.4.3.6 Test log

To be able to follow the progress of the test, a log feature is provided. The Log displays a listing of the actions executed by CTT. Fatal error messages are also displayed in the Log. The log file can be saved for further analysis.

### 6.4.3.7 Line traffic

To be able to follow the exchange of messages, a line traffic feature is provided. The Line Traffic window displays the bytes (in hex-string notation) sent (green) and received (red) over the serial line. The line traffic can be saved for further analysis.

### 6.4.3.8 Test Report

The CTT provides a test report after each run. The test report has the following characteristics:

1. It is an automatically generated text file.
2. The test report is generated for each physical device.
3. The contents of the Test Report is human readable.
4. The test report is authenticated by a digital signature.

The Report is divided in several sections.

• The Header section identifies the CTT version as well as the license owner and the date and time of the testing.
• The Identification section shows the Manufacturer and Device declarations taken from the PICS file.
• The Summary section displays a resume of the test results sorted by groups.
• The next 3 sections show the results of the HDLC tests, the APPL tests and the COSEM tests respectively.
• The next two sections show the PICS and PIXIT files.
• The last line is the authentication signature.

### 6.4.3.9 Authenticity of the PICS/PIXIT files and the Test Report

The possibility to validate the authenticity of the PICS/PIXIT files is essential to enable self/testing by the manufacturer or vendor of the implementation.

• The authenticity of the PICS/PIXIT is guaranteed by the fact that the PICS/PIXIT becomes part of the test report.
• The authenticity of the test report is guaranteed by a digital signature, which is added by the CTT. The signature can be validated by a built-in feature of the CTT: the test report provided by the testing laboratory can be loaded in the CTT and the digital signature can be checked.

### 6.4.4 Tool Validation

The validation of the CTT has been done in several steps:

1. The test plans for the various layer have been written by experts from different manufacturers – members of WG CT – based on the DLMS/COSEM specification.
2. The test routines have been systematically compared with the DLMS/COSEM standards and the test plans by the WG CT.
3. The CTT test routines have been tested with different server implementations by independent parties.
4. DLMS/COSEM features, which are not implemented in any server device, are not implemented in the CTT either. These will be added when at least one server implementation will exist.

6.4.5 Handling of CTT copyright aspects

The appropriate security measures are provided by EuroDCS.

6.4.6 Preparation of implementations for conformance testing

The possibility to select individual test groups or any combination of test groups to be run, the log and the line traffic features are also useful in preparing the IUT for conformance testing. However, the clear purpose of the CTT is to perform automated conformance testing.

6.4.7 PICS template and example

6.4.7.1 PICS/PIXIT syntax

The PICS and PIXIT files are text files that contain information needed by CTT for automatic testing.

PICS and PIXIT files contain entities’ declarations of the form:

**Entity_name = Entity_value**

*Entity_name* is a predefined identifier, *Entity_value* is either:

- an integer constant,
- a string (or hex-string) constant,
- a predefined constant,
- a set of predefined constants,
- a set of integers or
- a structure.

Examples:

```
InactivityTimeout = 120000  integer constant
PassWord = 'Hello'   string constant
UpperMacAddress = '12ff'  Hex-string constant,
ContextName = SHORT_NAME predefined constant
Conformance = [READ, WRITE, MULTIPLE_REFERENCES, PARAMETRIZED_ACCESS]
SelectiveAccessSelectors = [1,2] set of integer constants (in the range 0..255)
PhysicalLayer =
{  PortType = OPTICAL
  EchoingPort = true
  OpeningMode = DIRECT_HDLC
  HdlcBaud = 9600}
```
### 6.4.7.2 PICS template

// PICS TEMPLATE

// (MANDATORY) Manufacturer's info
Manufacturer = {
    Name = x // string
}

// (MANDATORY) Device information
// The device identification must be the same in both PICS and PIXIT files.
Device = {
    Ident = x // string
}

// (OPTIONAL) ClassExtraInfo
// As many ClassExtraInfo as required. The scope of these declarations (at this level) is
// global. These declarations will affect each object instance of the specified class/version
// encountered during the test
ClassExtraInfo = {
    ClassId = x // integer
    Version = x // integer

    // As many AttributeExtraInfo as needed
    AttributeExtraInfo = {
        AttributeId = x // mandatory, integer
        AccessRights = x // optional, one of WRITE_ONLY READ_ONLY READ_WRITE
        NO_ACCESS
        WriteTestData = x // optional, string
    }
}

// (OPTIONAL) InstanceExtraInfo
// As many InstanceExtraInfo as required. The scope of these declarations (at this level) is
// global. These declarations will affect each object instance of the specified logical name
// encountered during the test
InstanceExtraInfo = {
    LogicalName = x // hexstring
    Version = x // integer

    // As many AttributeExtraInfo as needed, see here above
    AttributeExtraInfo = {
    }
}

// (MANDATORY) As many LogicalDevices as needed.
LogicalDevice[0] = {

    // (OPTIONAL) Manufacturer of this logical device
    Manufacturer = {
        Name = x // string
    }

    // (OPTIONAL) Device identification of this logical device.
    Device = {
        Ident = x // string
    }

    // (OPTIONAL) as many ClassExtraInfo and/or InstanceExtraInfo as needed. These declarations override any corresponding declaration. These declarations only affect the classes and/or objects processed in this logical device.
    Association[0] = {
    }
}
ContextName = x  // one of SHORT_NAMES LONG_NAMES
MechanismName = NO_SECURITY  // Association [0] requires this
Conformance = [x]  // mandatory set of conformance bits
Enable = x  // optional, one of TRUE FALSE
OBISFieldIdentifiers = [x]  // mandatory set of ABSTRACT, ELECTRICITY

// (OPTIONAL) as many ClassExtraInfo and/or InstanceExtraInfo as needed.
These
// declarations override any corresponding declarations. These declarations
// only affect the classes and/or objects processed in this Association.

}

// (MANDATORY)
Association [1] = {
   ContextName = x  // one of SHORT_NAMES LONG_NAMES
   MechanismName = LOW_LEVEL_SECURITY  // Association [1] requires this
   Conformance = [x]  // mandatory set of conformance bits
   Enable = x  // optional, one of TRUE FALSE

   // (OPTIONAL) as many ClassExtraInfo and/or InstanceExtraInfo as needed.
   These
   // declarations override any corresponding declarations. These declarations
   // only affect the classes and/or objects processed in this Association
}

}  // END

6.4.7.3 PICS example
Manufacturer = {
   Name = ‘Micromin Inc.’
}
// The device identification must be the same in both PICS and PIXIT files.
Device = {
   Ident = ‘SuperMeter for DLMS UA’
}

LogicalDevice[0] = {
   Manufacturer = {
      Name = ‘Micromin Inc.’
   }
   Device = {
      Ident = ‘Any string’
   }
   Association [0] = {
      ContextName = SHORT_NAMES
      MechanismName = NO_SECURITY
      Conformance = [READ, WRITE, MULTIPLE REFERENCES, PARAMETRIZED_ACCESS]
      OBISFieldIdentifiers = [ABSTRACT, ELECTRICITY]
   }
   Association [1] = {
      ContextName = SHORT_NAMES
      MechanismName = LOW_LEVEL_SECURITY
      Conformance = [READ, WRITE, MULTIPLE REFERENCES, PARAMETRIZED_ACCESS]
      OBISFieldIdentifiers = [ABSTRACT]
   }
}
ClassExtraInfo = {
   ClassId = 1
   Version = 2
   AttributeExtraInfo = {
...
AttributeId = 1
   AccessRights = READ_ONLY
 }
AttributeExtraInfo = {
   AttributeId = 3
   AccessRights = READ_WRITE
 }

// END LogicalDevice[0]

6.4.8 PIXIT template and example

6.4.8.1 PIXIT template

// PIXIT TEMPLATE

// (MANDATORY) Manufacturer's info
Manufacturer = {
   Name = x // string
 }

// (MANDATORY) Device information
// The device identification must be the same in both PICS and PIXIT files.
Device = {
   Ident = x // string
 }

// (MANDATORY) Physical layer infos
PhysicalLayer = {
   PortType = x // mandatory, one of OPTICAL RS232
   EchoingPort = x // mandatory, one of TRUE FALSE
   OpeningMode = x // mandatory, one of DIRECT_HDLC MODE_E
   HdlcBaud = x // optional, integer 9600
   ModeEIdString = x // optional, string
 }

]*) HOW TO SPECIFY CLIENT AND SERVER ADDRESSES -------------------------------
1 byte addressing is specified as in the following example:
ServerLowerMacAddress = '10'

2 bytes addressing is specified as in the following example:
ServerLowerMaxAddress = '0010'

]*)// (MANDATORY) Data link layer infos
DataLinkLayer = {
   InactivityTimeout = x // mandatory integer
   InterFrameTimeout = x // mandatory integer
   ResponseTimeout = x // mandatory integer
   ServerLowerMacAddress = x // mandatory hexstring
   DISCToNDMTimeout = x // mandatory integer
   AddressingSchemes = [x] // optional set of ONE_BYTE_ADDRESSING,
   TWO_BYTES_ADDRESSING, FOUR_BYTES_ADDRESSING
   ServerMaxInfoLengthTransmit = x // optional integer
}
6.4.8.2 PIXIT example

// PIXIT
Manufacturer = {
  Name = "Micromin inc.'
}

Device = {
  Ident = 'Supermeter for DLMS UA' // must match the one in the PICS file
}

PhysicalLayer = {
  PortType = OPTICAL
  EchoingPort = TRUE
  OpeningMode = MODE_E
  HdlcBaud = 9600
  ModeEIdString = 'Hello'
}

(*--------------------------------------------------------------------------
 HOW TO SPECIFY CLIENT AND SERVER ADDRESSES
------------------------------------------
  1 byte addressing is specified as in the following example:
    ServerLowerMacAddress = '10'
  2 bytes addressing is specified as in the following example:
    ServerLowerMaxAddress = '0010'
--------------------------------------------------------------------------*)

DataLinkLayer = {
  InactivityTimeout = 120000
  InterFrameTimeout = 50
  ResponseTimeout = 2000
  ServerLowerMacAddress = '10'
  DISCToNDMTimeout = 2000
  AddressingSchemes = [ONE_BYTE_ADDRESSING, TWO_BYTES_ADDRESSING]
}

ApplicationLayer = {
  LogicalDevice [0] = {
    Association[0] = {
      ClientMacAddress = '10'
      ServerUpperMacAddress = '11'
      DLMSVersionNumber = 7
    }
  }
}

// (MANDATORY) Application layer info
ApplicationLayer = {
  ServerMaxReceivePduSize = x // mandatory integer
}

// (MANDATORY) As many LogicalDevices as needed.
LogicalDevice [0] = {
  // (MANDATORY) As many Associations as needed
  Association[0] = {
    ClientMacAddress = x // mandatory hexstring
    ServerUpperMacAddress = x // mandatory hexstring
    PassWord = x // mandatory when association > 0, string
    DLMSVersionNumber = x // optional integer, default = 6
  }
}

// END
Association[1] = {
    ClientMacAddress = '64'
    ServerUpperMacAddress = '01'
    PassWord = '12345678'
}

// END
7. DLMS/COSEM compliance certification process

This chapter describes the necessary step to obtain a DLMS/COSEM compliance certification from the DLMS UA for a server implementing the specification.

7.1 Licensing the CTT

The CTT can be licensed by any member of the DLMS UA - manufacturer of the server (or a third party tester) from Euro DCS http://www.eurodcs.de at the commercial conditions agreed by the DLMS UA.

The license fee consists of a purchasing fee and a yearly maintenance fee.

7.2 Performing the conformance test

The vendor (or a third party tester) performs the CT.

If the test is done by the vendor – which is the preferred method - the test laboratory shall be an identifiable part of the vendor’s organisation.

- The server is prepared so that it allows the CTT access to the objects or sections to be tested. All objects to be tested must be available under the associations declared in the PICS. These associations must be available under “no security” or under “low level security”. For low level security, the passwords are part of the PIXIT.

- It is the responsibility of the manufacturer to restrict access rights to attributes, which must not be modified during the test. This can be done by providing extra information in the PICS.

- The PICS and PIXIT information is provided (loaded or edited).

- The server is connected to the CTT through a serial connection.

- The CTT is started and it produces a test report (containing the verdicts).

7.3 Verification and acceptance of test report

The test report shall be sent to the DLMS UA for verification and registration. The DLMS UA verifies the authenticity of the report and checks the results.

If the test report is accepted, the DLMS UA informs the sender accordingly. Should problems arise, the DLMS UA contacts the contact person at the sender's company.

Upon acceptance of the test report, the DLMS UA recommends that the manufacture displays the “DLMS/COSEM Compliant” Mark on its product and product literature.

The test reports are filed by the DLMS UA, but they are not published. The users may obtain details from the vendor.
7.4 The “DLMS/COSEM compliant” mark

The “DLMS/COSEM compliant” mark is shown below:

![DLMS/COSEM compliant mark]

7.5 Testing different configurations

The CT is made on one or more different configurations, characteristic for the server implementation, identified by the type and revision. It is recommended that the IUT is configured for the tests in such a way that all associations are visible and at least one instance of each available interface classes are instantiated in each association.

It is up to the purchaser to request the vendor to perform a CT on the characteristic configurations as deemed necessary.

7.6 Disclaimer

The DLMS UA took all possible effort to ensure that the conformance test tool is in line with the DLMS/COSEM specification and provides a reasonable depth of testing.

The acceptance of the test report, the display of the “DLMS/COSEM compliant mark” and the listing of the equipment as compliant by the DLMS UA, does not mean however that an absolute proof of conformance is given.

7.7 Arbitration

Arbitration may be required if there is a disagreement between a vendor and a purchaser on test results of a certain implementation. Upon request from the vendor or the purchaser, the DLMS UA will repeat the conformance test, using the original metering equipment type, version and configuration and the original PICS and PIXIT files.

Another case is when certified product purchased by a customer exhibits deficiencies, which were undetected during the test.

The resolution of such cases is left to the provider and the purchaser in the first place.

The involvement of the DLMS UA is limited to verify that the tool works according to the standards.
8. List of certified equipment

A certification is more than an important marketing argument for the manufacturer; it stands for the quality of a product. Availability of certified products is thus an important information for purchasers.

To support both manufacturers and purchasers, the DLMS UA maintains an online list of certified equipment. The information provided is shown in the table below.

<table>
<thead>
<tr>
<th>Company</th>
<th>Product</th>
<th>Certification date</th>
<th>Registration No.</th>
<th>Test tool version</th>
<th>Test laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siemens Metering</td>
<td>Landis &amp; Gyr Dialog ZMD410CT44.2 407 B12</td>
<td>4 June 2002</td>
<td>1001</td>
<td>1.00</td>
<td>Siemens Metering</td>
</tr>
<tr>
<td>Actaris Metering Systems</td>
<td>SL7000 GeniEC4</td>
<td>4 June 2002</td>
<td>1002</td>
<td>1.00</td>
<td>Actaris Metering Systems</td>
</tr>
</tbody>
</table>

The list is kept up-to-date on the website of the DLMS UA:

http://www.dlms.com
9. The DLMS/COSEM conformance test maintenance process

The DLMS UA maintains the CTT to eliminate problems with the tool found during testing, to enhance tests and to accommodate changes in the standards.

The recommended procedure is the following:

1. A proposal, together with a justification is made to add or modify a test. This can be initiated by any member of the DLMS UA, implementing the specification or by the DLMS UA itself.
2. The request is investigated by the DLMS UA.
3. If the request is accepted, test plans are amended by the DLMS UA.
4. The new tests are implemented by the tool developer.
5. The new tests are validated by the DLMS UA.
6. The amended test plans 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.

9.1 Maintenance: use case 1 – introducing a new standard OBIS code

A vendor 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 will check if the proposal is in line with the standards. If approved, the OBIS table is amended and the new version is integrated in the tool. All licensees receive the updated version. The file name of the OBIS tables used for testing is included in the test report.

9.2 Maintenance: use case 2 – modification of an existing test

If it is found that despite of careful validation, any conformance test is not fully compliant with the standard, or it is necessary to enhance a test, the a proposal may be submitted to the DLMS UA by the vendor or the user and the process described above is followed.

9.3 Maintenance: use case 3 – adding a test for a new standard feature

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

The vendor submits the proposed test plan to the DLMS UA and the process described above is followed.
9.4 Maintenance: 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. Depending on the effort needed, this may require extra financial resources from the licensees.
10. Physical layer test plan

10.1 Scope

This document specifies the tests to be performed in order to verify the conformance on physical layer level. The tests are based on the physical layer specification of IEC 62056-42 and IEC 62056-21.

As most of the physical layer tests are difficult to perform automatically, and these tests are implicitly done during testing the higher layers, these tests are not implemented in Version 1.0 of the Conformance Test Tool.

more details, see complete Yellow Book ...
11. Data Link Layer test plan

11.1 Foreword

Copyright

© Copyright 1997-2002 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.

11.2 Scope

This document specifies the tests to be performed in order to verify the conformance on data link level. The tests are based on the data link layer specifications IEC 62056-46, ISO/IEC 13239 and ISO/IEC 8802-2.

Based on ISO/IEC 9646-1, recommendation X.290 Conformance testing a distinction is made between positive and negative test cases:

• In a positive test case, a syntactically correct message is sent from the test tool to the IUT in the right context/state.
• In a negative test case, a syntactically invalid message is sent or the message – syntactically correct or incorrect – is sent in an inopportune context/state.

A client, which is conformant to the specification will not send syntactically invalid messages or inopportune messages to the Server.

For the result, a distinction is made between:

• 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 with an observed outcome;
• syntactically invalid test event: a test event which syntactically is not allowed by the protocol recommendation;
• 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.

A PASS verdict is given when the observed outcome satisfies the test purpose and is valid with respect to the standard and with respect to the PICS.
A FAIL verdict is given when the observed outcome is syntactically invalid or inopportune with respect to the standards or the PICS.

Version 2.0 is in line with the CTT version 1.0.

more details, see complete Yellow Book ...
12. COSEM application layer test plan

12.1 Foreword

Copyright

© Copyright 1997-2002 DLMS User Association

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

12.2 Scope

This document specifies the tests to be performed in order to verify conformance on COSEM application layer level. The tests are based on the application layer specification as defined in IEC 62056-53.

Based on ISO/IEC 9646-1, recommendation X.290 Conformance testing a distinction is made between positive and negative test cases:

In a positive test case, a syntactically and semantically correct message is sent from the test tool to the IUT in the right context/state.
In a negative test case, a syntactically or semantically invalid message is sent or the message – correct or incorrect – is sent in an inopportune context/state.

A good client will not send incorrect messages or inopportune messages to the Server.

For the result, a distinction is made between:
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 with an observed outcome;
syntactically invalid test event: a test event which syntactically is not allowed by the protocol recommendation;
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.

A PASS verdict is given when the observed outcome satisfies the test purpose and is valid with respect to the standard and with respect to the PICS.
A FAIL verdict is given when the observed outcome is syntactically invalid or inopportune with respect to the standards or the PICS.

Version 2.0 is in line with the CTT version 1.0.

more details, see complete Yellow Book ...
13. Interface objects test plan

13.1 Foreword

Copyright


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.

13.2 Scope

This document specifies the tests to be performed in order to ensure conformance on COSEM Interface Object level between equipment of different manufacturers. The tests are based on the Interface classes IEC 62056-61 and OBIS Object identification system IEC 62056-62 specifications.

more details, see complete Yellow Book ...