International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.841

(08/2020)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

E-health multimedia systems, services and applications – Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)

Conformance of ITU-T H.810 personal health system: Personal Health Devices interface Part 1: Optimized Exchange Protocol: Personal Health Device

Recommendation ITU-T H.841

ITU-T

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Telepresence, immersive environments, virtual and extended reality	H.420–H.439
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
VEHICULAR GATEWAYS AND INTELLIGENT TRANSPORTATION SYSTEMS (ITS)	
Architecture for vehicular gateways	H.550–H.559
Vehicular gateway interfaces	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
Ubiquitous sensor network applications and Internet of Things	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	H.760–H.769
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789
E-HEALTH MULTIMEDIA SYSTEMS, SERVICES AND APPLICATIONS	
Personal health systems	H.810–H.819
Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)	**H.820–H.859**
Multimedia e-health data exchange services	H.860–H.869
Safe listening	H.870–H.879

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.841

Conformance of ITU-T H.810 personal health system: Personal Health Devices interface Part 1: Optimized Exchange Protocol: Personal Health Device

Summary

Recommendation ITU-T H.841 provides a test suite structure (TSS) and the test purposes (TP) for personal health devices using the IEEE 11073-20601 optimized exchange protocol in the Personal Health Devices (PHD) interface, based on the requirements defined in the Recommendations of the ITU-T H.810 sub-series, of which Recommendation ITU-T H.810 (2017) is the base Recommendation. The objective of this test specification is to provide a high probability of interoperability at this interface.

Recommendation ITU-T H.841 is a transposition of Continua Test Tool DG2016, Test Suite Structure & Test Purposes, Personal Health Devices Interface; Part 1: Optimized Exchange Protocol. Personal Health Device (Version 1.11, 2017-03-14), that was developed by the Personal Connected Health Alliance. A number of versions of this specification existed before transposition.

This Recommendation includes an electronic attachment with the protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T H.841	2015-01-13	16	11.1002/1000/12258
2.0	ITU-T H.841	2016-07-14	16	11.1002/1000/12934
3.0	ITU-T H.841	2017-04-13	16	11.1002/1000/13215
4.0	ITU-T H.841	2018-08-29	16	11.1002/1000/13678
5.0	ITU-T H.841	2020-08-13	16	11.1002/1000/14344

Keywords

Conformance testing, Continua Design Guidelines, e-health, ITU-T H.810, IEEE 11073-20601 optimized exchange protocol, Personal Health Devices Interface, personal area network, personal connected health devices, Personal Health Devices interface, touch area network.

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Electronic attachment: This Recommendation includes an electronic attachment with the protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A.

Introduction

This Recommendation is a transposition of Continua Test Tool DG2016, Test Suite Structure & Test Purposes, Personal Health Devices Interface; Part 1: Optimized Exchange Protocol. Personal Health Device (Version 1.11, 2017-03-14), that was developed by the Personal Connected Health Alliance. The table below shows the revision history of this test specification; it may contain versions that existed before transposition.

Version	Date	Revision history
1.4	2012-10-05	Initial release for Test Tool DG2011. This uses "TSS&TP_1.5_PAN-LAN_PART_1_v1.3.doc" as a baseline and adds new features included in [b-CDG 2011]: • Person ID and Errata
1.5	2013-05-24	Initial release for Test Tool DG2012. This uses "TSS&TP_DG2011_PAN-LAN_PART_1_v1.4.doc" as a baseline and adds new features included in [b-CDG 2012]: • Adds glucose meter • Adds body composition analyser device specialization • Adds basic electrocardiograph device specialization
1.6	2014-01-24	Initial release for Test Tool DG2013. This uses "TSS&TP_DG2012_PAN-LAN_PART_1_v1.5.doc" as a baseline and adds new features included in [b-ITU-T H.810 (2013)]/[b-CDG 2013]: • Adds glucose meter BLE • Adds BLE SSP support • Adds NFC new transport • Adds INR device specialization
1.7	2014-04-24	TM Lite & Doc Enhancements (Test Tool v4.0 Maintenance Release 1). It uses "TSS&TP_DG2013_PLT_PART_1_v1.6.doc" as a baseline and adds new features included in Documentation Enhancements: • "Other PICS" row added
1.8	2015-07-01	Initial release for Test Tool DG2015. It uses "TSS&TP_DG2013_PLT_PART_1_v1.7.doc" as a baseline and adds new features included in [b-ITU-T H.810 (2015)]/[b-CDG 2015]
1.9	2016-01-26	First maintenance release for Test Tool DG2015. It uses "TSS&TP_DG2015_LP-PAN_PART_1_v1.8.doc" as a baseline and adds some updates according to the Maintenance 2015 activity.
1.10	2016-09-20	Initial release for Test Tool DG2016. It uses "TSS&TP_DG2015_PLT_PART_1_v1.9.doc" as a baseline and adds new features included in [b-ITU-T H.810 (2016)]/[b-CDG 2016]
1.11	2017-03-14	Updates the baseline in "TSS&TP_DG2016_PHD_PART_1_ v1.10.doc" as part of the Maintenance 2016 activity: • Updates the applicability for TP/PLT/PHD/OXP/COM/BV-054_A
1.12	2018-02-27	Updates related to the inclusion of the power status monitor of Personal Health Devices device specialization (ISO/IEEE 11073-10427:2018) Updates related to the changes included in the glucose meter device specialization (ISO/IEEE 11073-10417:2017)
1.13	2020-06-02	Updates related to the inclusion of the value "7946" according to H.812.1 in [b-CDG 2017], corresponding to "MDC_TIME_SYNC_EBWW", and according to Bugzilla 1164 and 1165.

Recommendation ITU-T H.841

Conformance of ITU-T H.810 personal health system: Personal Health Devices interface Part 1: Optimized Exchange Protocol: Personal Health Device

1 Scope

The scope of this Recommendation[1] is to provide a test suite structure (TSS) and the test purposes (TP) for the Personal Health Devices interface based on the requirements defined in the Continua Design Guidelines (CDG) [ITU-T H.810 (2017)]. The objective of this test specification is to provide a high probability of interoperability at this interface.

The TSS and TP for the Personal Health Devices interface have been divided into the parts specified below. This Recommendation covers Part 1.

– **Part 1**: **Optimized exchange protocol Personal Health Device**

– Part 2: Optimized exchange protocol Personal Health Gateway

– Part 3: Continua design guidelines. Personal Health Device

– Part 4: Continua design guidelines. Personal Health Gateway

– Part 5: Device specializations. Personal Health Device. This document is divided into the following subparts:

  - Part 5A: Weighing scales
  - Part 5B: Glucose meter
  - Part 5C: Pulse oximeter
  - Part 5D: Blood pressure monitor
  - Part 5E: Thermometer
  - Part 5F: Cardiovascular fitness and activity monitor
  - Part 5G: Strength fitness equipment
  - Part 5H: Independent living activity hub
  - Part 5I: Adherence monitor
  - Part 5J: Insulin pump
  - Part 5K: Peak expiratory flow monitor
  - Part 5L: Body composition analyser
  - Part 5M: Basic electrocardiograph
  - Part 5N: International normalized ratio monitor
  - Part 5O: Sleep apnoea breathing therapy equipment (SABTE)
  - Part 5P: Continuous glucose monitor (CGM)
  - Part 5Q: Power status monitor (PSM)

– Part 6: Device specializations. Personal Health Gateway

– Part 7: Continua Design Guidelines. Personal Health Device BLE

[1] This Recommendation includes an electronic attachment with the protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation Annex A.

– Part 8: Continua Design Guidelines. Personal Health Gateway BLE

– Part 9: Personal Health Devices Transcoding Whitepaper. Personal Health Devices

– Part 10: Personal Health Devices Transcoding Whitepaper. Personal Health Gateway

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T H.810 (2017)] Recommendation ITU-T H.810 (2017), *Interoperability design guidelines for personal health systems*.

[ITU-T H.811] Recommendation ITU-T H.811 (2017), *Personal Health Devices Interface design guidelines*.

[ISO/IEEE 11073-104xx] ISO/IEEE 11073-104xx (in force), *Health informatics – Personal health device communication – Device specialization*.
NOTE – This is shorthand used to refer to the collection of device specialization standards that utilize [ISO/IEEE 11073-20601-2015A], where xx can be any number from 01 to 99, inclusive.

[ISO/IEEE 11073-20601-2016C] ISO/IEEE 11073-20601:2016, *Health informatics – Personal health device communication – Part 20601: Application profile – Optimized exchange protocol,* including ISO/IEEE 11073-20601:2016/Cor.1:2016.
https://www.iso.org/standard/66717.html with https://www.iso.org/standard/71886.html

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 agent [ISO/IEEE 11073-20601-2016C]: A node that collects and transmits personal health data to an associated manager.

3.1.2 manager [ISO/IEEE 11073-20601-2016C]: A node receiving data from one or more agent systems. Some examples of managers include a cellular phone, health appliance, set top box, or a computer system.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ATS	Abstract Test Suite
CDG	Continua Design Guidelines

CGM	Continuous Glucose Monitor
DUT	Device Under Test
GUI	Graphical User Interface
INR	International Normalized Ratio
IP	Insulin Pump
IUT	Implementation Under Test
MDS	Medical Device System
NFC	Near Field Communication
PAN	Personal Area Network
PCO	Point of Control and Observation
PCT	Protocol Conformance Testing
PHD	Personal Health Device
PHDC	Personal Healthcare Device Class
PHG	Personal Health Gateway
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation extra Information for Testing
PSM	Power Status Monitor
SABTE	Sleep Apnoea Breathing Therapy Equipment
SCR	Static Conformance Review
SDP	Service Discovery Protocol
SOAP	Simple Object Access Protocol
TCRL	Test Case Reference List
TCWG	Test and Certification Working Group
TP	Test Purpose
TSS	Test Suite Structure
USB	Universal Serial Bus
WDM	Windows Driver Model

5 Conventions

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY", "MAY NOT" in this Recommendation are to be interpreted as in [b-ETSI SR 001 262].

– SHALL is equivalent to 'must' or 'it is required to'.

– SHALL NOT is equivalent to 'must not' or 'it is not allowed'.

– SHOULD is equivalent to 'it is recommended to'.

– SHOULD NOT is equivalent to 'it is not recommended to'.

– MAY is equivalent to 'is permitted'.

– MAY NOT is equivalent to 'it is not required that'.

NOTE – The above-mentioned key words are capitalized for illustrative purposes only and they do not appear capitalized within this Recommendation.

Reference is made in the ITU-T H.800-series of Recommendations to different versions of the Continua Design Guidelines (CDG) by a specific designation. The list of terms that may be used in this Recommendation is provided in Table 1.

Table 1 – List of designations associated with the various versions of the CDG

CDG release	Transposed as	Version	Description	Designation
2017	–	7.0	Release 2017 of the CDG including maintenance updates of the CDG 2016 and additional guidelines that cover new functionalities.	–
2016 plus errata	[ITU-T H.810 (2016)]	6.1	Release 2016 plus errata noting all ratified bugs [b-CDG 2016].	–
2016	–	6.0	Release 2016 of the CDG including maintenance updates of the CDG 2015 and additional guidelines that cover new functionalities.	Iris
2015 plus errata	[b-ITU-T H.810 (2015)]	5.1	Release 2015 plus errata noting all ratified bugs [b-CDG 2015]. The 2013 edition of H.810 is split into eight parts in the H.810-series.	–
2015	–	5.0	Release 2015 of the CDG including maintenance updates of the CDG 2013 and additional guidelines that cover new functionalities.	Genome
2013 plus errata	[b-ITU-T H.810 (2013)]	4.1	Release 2013 plus errata noting all ratified bugs [b-CDG 2013].	–
2013	–	4.0	Release 2013 of the CDG including maintenance updates of the CDG 2012 and additional guidelines that cover new functionalities.	Endorphin
2012 plus errata	–	3.1	Release 2012 plus errata noting all ratified bugs [b-CDG 2012].	–
2012	–	3.0	Release 2012 of the CDG including maintenance updates of the CDG 2011 and additional guidelines that cover new functionalities.	Catalyst
2011 plus errata	–	2.1	CDG 2011 integrated with identified errata.	–
2011	–	2.0	Release 2011 of the CDG including maintenance updates of the CDG 2010 and additional guidelines that cover new functionalities [b-CDG 2011].	Adrenaline
2010 plus errata	–	1.6	CDG 2010 integrated with identified errata	–
2010	–	1.5	Release 2010 of the CDG with maintenance updates of the CDG Version 1 and additional guidelines that cover new functionalities [b-CDG 2010].	1.5

Table 1 – List of designations associated with the various versions of the CDG

CDG release	Transposed as	Version	Description	Designation
1.0	–	1.0	First released version of the CDG [b-CDG 1.0].	–

6 Test suite structure (TSS)

The test purposes (TPs) for the Personal Health Devices interface have been divided into the main subgroups specified below. Annex A describes the TPs for subgroups 1.2.1, 1.2.2 and 1.2.3 (shown in bold).

- Group 1: Personal Health Device (PHD)
  - Group 1.1: Transport (TR)
    - Subgroup 1.1.1: Design guidelines: Common (DGC)
    - Subgroup 1.1.2: USB design guidelines (UDG)
    - Subgroup 1.1.3: Bluetooth design guidelines (BDG)
    - Subgroup 1.1.4: Pulse oximeter design guidelines (PODG)
    - Subgroup 1.1.5: Cardiovascular design guidelines (CVDG)
    - Subgroup 1.1.6: Activity hub design guidelines (HUBDG)
    - Subgroup 1.1.7: ZigBee design guidelines (ZDG)
    - Subgroup 1.1.8: Glucose meter design guidelines (GLDG)
    - Subgroup 1.1.9: Bluetooth low energy design guidelines (BLEDG)
    - Subgroup 1.1.10: Basic electrocardiograph design guidelines (ECGDG)
    - Subgroup 1.1.11: NFC design guidelines (NDG)
  - Group 1.2: IEEE 20601 Optimized exchange protocol (OXP)
    - **Subgroup 1.2.1**: **PHD domain information model (DIM)**
    - **Subgroup 1.2.2**: **PHD service model (SER)**
    - **Subgroup 1.2.3**: **PHD communication model (COM)**
  - Group 1.3: Devices class specializations (CLASS)
    - Subgroup 1.3.1: Weighing scales (WEG)
    - Subgroup 1.3.2: Glucose meter (GL)
    - Subgroup 1.3.3: Pulse oximeter (PO)
    - Subgroup 1.3.4: Blood pressure monitor (BPM)
    - Subgroup 1.3.5: Thermometer (TH)
    - Subgroup 1.3.6: Cardiovascular (CV)
    - Subgroup 1.3.7: Strength (ST)
    - Subgroup 1.3.8: Activity hub (HUB)
    - Subgroup 1.3.9: Adherence monitor (AM)
    - Subgroup 1.3.10: Insulin pump (IP)
    - Subgroup 1.3.11: Peak flow (PF)
    - Subgroup 1.3.12: Body composition analyser (BCA)
    - Subgroup 1.3.13: Basic electrocardiograph (ECG)

    - Subgroup 1.3.14: International normalized ratio (INR)
    - Subgroup 1.3.15: Sleep apnoea breathing therapy equipment (SABTE)
    - Subgroup 1.3.16: Continuous glucose monitor (CGM)
    - Subgroup 1.3.17: Power status monitor (PSM)
  - Group 1.4: Personal health device transcoding whitepaper (PHDTW)
    - Subgroup 1.4.1: Whitepaper general requirements (GEN)
    - Subgroup 1.4.2: Whitepaper thermometer requirements (TH)
    - Subgroup 1.4.3: Whitepaper blood pressure requirements (BPM)
    - Subgroup 1.4.4: Whitepaper heart rate requirements (HR)
    - Subgroup 1.4.5: Whitepaper glucose meter requirements (GL)
    - Subgroup 1.4.6: Whitepaper weight scale requirements (WS)
    - Subgroup 1.4.7: Whitepaper pulse oximeter requirements (PLX)
    - Subgroup 1.4.8: Whitepaper continuous glucose monitoring requirements (CGM)
- Group 2: Personal Health Gateway (PHG)
  - Group 2.1: Transport (TR)
    - Subgroup 2.1.1: Design guidelines: Common (DGC)
    - Subgroup 2.1.2: USB design guidelines (UDG)
    - Subgroup 2.1.3: Bluetooth design guidelines (BDG)
    - Subgroup 2.1.4: Cardiovascular design guidelines (CVDG)
    - Subgroup 2.1.5: Activity hub design guidelines (HUBDG)
    - Subgroup 2.1.6: ZigBee design guidelines (ZDG)
    - Subgroup 2.1.7: Bluetooth low energy design guidelines (BLEDG)
    - Subgroup 2.1.8: NFC design guidelines (NDG)
  - Group 2.2: IEEE 20601 Optimized exchange protocol (OXP)
    - Subgroup 2.2.1: General (GEN)
    - Subgroup 2.2.2: PHD domain information model (DIM)
    - Subgroup 2.2.3: PHD service model (SER)
    - Subgroup 2.2.4: PHD communication model (COM)
  - Group 2.3: Devices class specializations (CLASS)
    - Subgroup 2.3.1: Weighing scales (WEG)
    - Subgroup 2.3.2: Glucose meter (GL)
    - Subgroup 2.3.3: Pulse oximeter (PO)
    - Subgroup 2.3.4: Blood pressure monitor (BPM)
    - Subgroup 2.3.5: Thermometer (TH)
    - Subgroup 2.3.6: Cardiovascular (CV)
    - Subgroup 2.3.7: Strength (ST)
    - Subgroup 2.3.8: Activity hub (HUB)
    - Subgroup 2.3.9: Adherence monitor (AM)
    - Subgroup 2.3.10: Insulin pump (IP)
    - Subgroup 2.3.11: Peak flow (PF)
    - Subgroup 2.3.12: Body composition analyser (BCA)

  - Subgroup 2.3.13: Basic electrocardiograph (ECG)
  - Subgroup 2.3.14: International normalized ratio (INR)
  - Subgroup 2.3.15: Sleep apnoea breathing therapy equipment (SABTE)
  - Subgroup 2.3.16: Continuous glucose monitor (CGM)
  - Subgroup 2.3.17: Power status monitor (PSM)
- Group 2.4: Personal health device transcoding whitepaper (PHDTW)
  - Subgroup 2.4.1: Whitepaper general requirements (GEN)
  - Subgroup 2.4.2: Whitepaper thermometer requirements (TH)
  - Subgroup 2.4.3: Whitepaper blood pressure measurement requirements (BPM)
  - Subgroup 2.4.4: Whitepaper heart rate requirements (HR)
  - Subgroup 2.4.5: Whitepaper glucose meter requirements (GL)
  - Subgroup 2.4.6: Whitepaper weight scale requirements (WS)
  - Subgroup 2.4.7: Whitepaper pulse oximeter requirements (PLX)
  - Subgroup 2.4.8: Whitepaper continuous glucose monitoring requirements (CGM)

7 Electronic attachment

The protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A can be downloaded from http://handle.itu.int/11.1002/2000/12067. See [b-PHD PICS & PIXIT], [b-PHG PICS & PIXIT] and [b-TI].

In the electronic attachment, letters "C" and "I" in the column labelled "Mandatory" are used to distinguish between "PICS" and "PIXIT" respectively during testing. If the cell is empty, the corresponding PICS is "independent". If the field contains a "C", the corresponding PICS is dependent on other PICS, and the logical expression is detailed in the "SCR_Expression" field. The static conformance review (SCR) is used in the test tool to assert whether the PICS selection is consistent.

Annex A

Test purposes

(This annex forms an integral part of this Recommendation.)

A.1 TP definition conventions

The test purposes (TPs) are defined according to the following rules:

- **TP Id**: This is a unique identifier (TP/<TT>/<DUT>/<GR>/<SGR>/<XX> – <NNN>). It is specified according to the naming convention defined below:
  - Each test purpose identifier is introduced by the prefix "TP".
  - <TT>: This is the test tool that will be used in the test case:
    - PAN: Personal area network (Bluetooth or USB)
    - LAN: Local area network (ZigBee)
    - PAN-LAN: Personal area network (Bluetooth or USB) - Local area network (ZigBee)
    - LP-PAN: Low power personal area network (Bluetooth Low Energy)
    - TAN: Touch area network (NFC)
    - PLT: Personal area network (Bluetooth or USB) – Local area network (ZigBee) – Touch area network (NFC)
  - <DUT>: This is the device under test:
    - PHD: Personal Health Device
    - PHG: Personal Health Gateway
  - <GR>: This identifies a group of test cases.
  - <SGR>: This identifies a subgroup of test cases.
  - <XX>: This identifies the type of testing:
    - BV: Valid behaviour test
    - BI: Invalid behaviour test
  - <NNN>: This is a sequential number that identifies the test purpose.
- **TP label**: This is the TP's title.
- **Coverage**: This contains the specification reference and clause to be checked by the TP:
  - Spec: This indicates the earliest version of the specification from which the testable items to be checked by the TP were included.
  - Testable item: This contains the testable items to be checked by the TP.
- **Test purpose**: This is a description of the requirements to be tested.
- **Applicability**: This contains the PICS items that define if the test case is applicable or not for a specific device. When a TP contains an "ALL" in this field it means that it applies to the device under test within that scope of the test (specialization, transport used, etc.).
- **Other PICS**: This contains additional PICS items (apart from the PICS specified in the Applicability row) which are used within the test case implementation and can modify the final verdict. When this row is empty, it means that only the PICS specified in the Applicability row are used within the test case implementation.

- **Initial condition**: This indicates the state to which the DUT needs to be moved at the beginning of TC execution.
- **Test procedure**: This describes the steps to be followed in order to execute the test case.
- **Pass/Fail criteria**: This provides criteria to decide whether the DUT passes or fails the test case.

A.2 Subgroup 1.2.1 – PHD domain information model (DIM)

TP Id		TP/PLT/PHD/OXP/DIM/BV-000		
TP label		MDS Object: Mandatory, Conditional and Optional Attributes		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MDSclass 1; M	MDSclass 2; M	MDSClassAttr 2; M
		MDSClassAttr 3; C	MDSClassAttr 4; M	MDSClassAttr 5; M
		MDSClassAttr 6; M	MDSClassAttr 7; C	MDSClassAttr 8; O
		MDSClassAttr 9; C	MDSClassAttr 10; C	MDSClassAttr 11; C
		MDSClassAttr 12; C	MDSClassAttr 13; C	MDSClassAttr 14; O
		MDSClassAttr 15; O	MDSClassAttr 16; O	MDSClassAttr 17; O
		MDSClassAttr 18; C	MDSClassAttr 19; O	MDSService 1; M
		MDSService 4; M	OperNormProc 2; M	OperNormProc 3; M
		ConfNormalProc 21; M	CommonCharac 3; M	ConfNormalProc 1; M
		OperNormProc 6; M	MDSMethod 7; O	ConfEventRep 28; M
		ConfEventRep 33; O	ConfEventRep 34; M	MDSMethod 6; M
		BaseTimOffset3; M		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Regulatory 4; M	Regulatory 5; O	
		multi_funct_LAN 1; M		
		General 2; M		
Test purpose		Check that: The Personal Health Device (PHD) supports a Get command that requests all attributes [AND] The PHD reports its MDS object attributes to the Personal Health Gateway (PHG) using a Data message with the "Remote Operation Response | Get" response. [AND] MDS object contains all mandatory attributes, conditional attributes as required by their conditions and it may contain optional attributes [AND] The total size of the response does not exceed the maximum APDU size established by the specialization [AND] MDS object attributes are static /dynamic or observational. [AND] The handle is entered in the obj-handle field and it is not included in the attribute ID list of the request or in the attribute list of the response. [AND] The MDS object is not considered part of the configuration. [AND]		

	Changes to any non-static attributes values on PM-stores or the MDS may be reported to the PHG in event reports at the discretion of the PHD [AND] A profile is expected to be identified by a name and a nomenclature value [AND] The PHD shall not include the Base Offset Time in any Continua configurations except for basic electrocardiograph (ECG), insulin pump (IP), continuous glucose monitor (CGM) and power status monitor (PSM) device specializations. [AND] If the base time (seconds field) is aligned with UTC (with an accuracy appropriate to the application), then this shall be designated by setting the mds-time-bo-time-utc-aligned bit in the Mds-Time-Info attribute.
Applicability	C_AG_OXP_000
Other PICS	C_AG_OXP_006, C_AG_OXP_007, C_AG_OXP_008, C_AG_OXP_009, C_AG_OXP_010, C_AG_OXP_011, C_AG_OXP_014, C_AG_OXP_015, C_AG_OXP_041, C_AG_OXP_071, C_AG_OXP_120, C_AG_OXP_188
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. Check that the MDS object and its attributes are not present in the Configuration Event Report. 5. The simulated PHG issues "roiv-cmip-get" command with the handle set to 0 (to request an MDS object) and an empty attribute-id-list to indicate all attributes. Record the invoke-id of the message sent. 6. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: Verify that the invoke-id is mirrored from the Get request. Verify that the DataApdu contains the SEQUENCE GetResultSimple (0x02 0x03) a. Verify that the GetResultSimple.obj-handle = 0x00 0x00 b. The GetResultSimple contains an AttributeList: □ AttributeList.count = number of implemented attributes that are included in the GET response. (must be greater than 3) □ AttributeList.length = the length of the remainder of the APDU Mandatory Attributes a. Mandatory attribute HANDLE shall not be present. b. Mandatory attribute System-Model □ attribute-id = MDC_ATTR_ID_MODEL (0x09 0x28) □ attribute-value.length = 0xXX 0xXX □ attribute-value = SystemModel □ SystemModel.manufacturer = OCTET STRING □ length = 0xXX 0xXX, where X is even and equal to the length of the value □ value = <Check with PIXITS> □ SystemModel.model-number = OCTET STRING □ length = 0xXX 0xXX, where X is even and equal to the length of the value □ value = <Check with PIXITS> c. Mandatory attribute System-Id □ attribute-id = MDC_ATTR_SYS_ID (0x09 0x84)

	□ attribute-type = OCTET STRING □ attribute-value.length = 10 bytes □ attribute-value = OCTET STRING(Size(8)) □ size = 0x00 0x08 □ value = <Check with PIXITS> d. Mandatory attribute Dev-Configuration-Id □ attribute-id = MDC_ATTR_DEV_CONFIG_ID (0x0A 0x44) □ attribute-type = ConfigId □ attribute-value.length = 2 bytes □ attribute-value = <between 0x00 0x01 and 0x7F 0xFF> Conditional and Optional Attributes e. One and only one of [System-Type-Spec_List] or [System-Type] shall be present. f. IF System-Type attribute is present: □ attribute-id = MDC_ATTR_SYS_TYPE (0x09 0x86) □ attribute-type = TYPE □ attribute-value.length = 4 bytes □ attribute-value = One of the supported specializations: ▪ MDC_DEV_SPEC_PROFILE_PULSE_OXIM (0x10 0x04) ▪ MDC_DEV_SPEC_PROFILE_BP (0x10 0x06) ▪ MDC_DEV_SPEC_PROFILE_TEMP (0x10 0x07) ▪ MDC_DEV_SPEC_PROFILE_SCALE (0x10 0x0F) ▪ MDC_DEV_SPEC_PROFILE_GLUCOSE (0x10 0x11) ▪ MDC_DEV_SPEC_PROFILE_HF_CARDIO (0x10 0x29) ▪ MDC_DEV_SPEC_PROFILE_HF_STRENGTH (0x10 0x30) ▪ MDC_DEV_SPEC_PROFILE_AI_ACTIVITY_HUB (0x10 0x47) ▪ MDC_DEV_SPEC_PROFILE_AI_MED_MINDER (0x10 0x48) ▪ MDC_DEV_SPEC_PROFILE_PEAK_FLOW (0x10 0x15) ▪ MDC_DEV_SPEC_PROFILE_BCA (0x10 0x14) ▪ MDC_DEV_SPEC_PROFILE_ ECG (0x10 0x06) ▪ MDC_DEV_SPEC_PROFILE_COAG (0x10 0x12) ▪ MDC_DEV_SPEC_PROFILE_SABTE(0x10 0x19) ▪ MDC_DEV_SPEC_PROFILE_INSULIN_PUMP (0x10 0x13) ▪ MDC_DEV_SPEC_PROFILE_CGM (0x10 0x1A) ▪ MDC_DEV_SPEC_PROFILE_PSM (0x10 0x1C) g. IF System-Type-Spec-List attribute is present □ attribute-id = MDC_ATTR_SYS_TYPE_SPEC_LIST (0x0A 0x5A) □ attribute-type = TypeVerList □ attribute-value.count = N (record for next step) □ attribute-value.length = N*4 bytes □ attribute-value = N of the following supported specializations: ▪ MDC_DEV_SPEC_PROFILE_PULSE_OXIM (0x10 0x04) ▪ MDC_DEV_SPEC_PROFILE_BP (0x10 0x06) ▪ MDC_DEV_SPEC_PROFILE_TEMP (0x10 0x07) ▪ MDC_DEV_SPEC_PROFILE_SCALE (0x10 0x0F)

	▪ MDC_DEV_SPEC_PROFILE_GLUCOSE (0x10 0x11) ▪ MDC_DEV_SPEC_PROFILE_HF_CARDIO (0x10 0x29) ▪ MDC_DEV_SPEC_PROFILE_HF_STRENGTH (0x10 0x30) ▪ MDC_DEV_SPEC_PROFILE_AI_ACTIVITY_HUB (0x10 0x47) ▪ MDC_DEV_SPEC_PROFILE_AI_MED_MINDER (0x10 0x48) ▪ MDC_DEV_SPEC_PROFILE_PEAK_FLOW (0x10 0x15) ▪ MDC_DEV_SPEC_PROFILE_BCA (0x10 0x14) ▪ MDC_DEV_SPEC_PROFILE_ECG (0x10 0x06) ▪ MDC_DEV_SPEC_PROFILE_COAG (0x10 0x12) ▪ MDC_DEV_SPEC_PROFILE_SABTE(0x10 0x19) ▪ MDC_DEV_SPEC_PROFILE_INSULIN_PUMP (0x10 0x13) ▪ MDC_DEV_SPEC_PROFILE_CGM (0x10 0x1A) ▪ MDC_DEV_SPEC_PROFILE_PSM (0x10 0x1C) ▪ Profiles for cardiovascular fitness and monitor specialization: If the PHD supports Step Counter profile THEN the PHD shall support Cardiovascular specialization • MDC_DEV_SUB_SPEC_PROFILE_STEP_COUNTER (0x10 0x68) ▪ Profiles for activity hub specialization: If the PHD supports any of the profiles defined for Activity Hub THEN the PHD shall support Activity Hub specialization: • MDC_DEV_SUB_SPEC_PROFILE_FALL_SENSOR (0x10 0x75) • MDC_DEV_SUB_SPEC_PROFILE_PERS_SENSOR (0x10 0x76) • MDC_DEV_SUB_SPEC_PROFILE_SMOKE_SENSOR (0x10 0x77) • MDC_DEV_SUB_SPEC_PROFILE_CO_SENSOR (0x10 0x78) • MDC_DEV_SUB_SPEC_PROFILE_WATER_SENSOR (0x10 0x79) • MDC_DEV_SUB_SPEC_PROFILE_GAS_SENSOR (0x10 0x7A) • MDC_DEV_SUB_SPEC_PROFILE_MOTION_SENSOR (0x10 0x7B) • MDC_DEV_SUB_SPEC_PROFILE_PROPEXIT_SENSOR (0x10 0x7C) • MDC_DEV_SUB_SPEC_PROFILE_ENURESIS_SENSOR (0x10 0x7D) • MDC_DEV_SUB_SPEC_PROFILE_CONTACTCLOSURE_SENSOR (0x10 0x7E) • MDC_DEV_SUB_SPEC_PROFILE_USAGE_SENSOR (0x10 0x7F) • MDC_DEV_SUB_SPEC_PROFILE_SWITCH_SENSOR (0x10 0x80) • MDC_DEV_SUB_SPEC_PROFILE_DOSAGE_SENSOR (0x10 0x81) • MDC_DEV_SUB_SPEC_PROFILE_TEMP_SENSOR (0x10 0x82) ▪ Profiles for basic electrocardiograph specialization: If the PHD supports any of the profiles defined for basic electrocardiograph THEN the PHD shall support basic electrocardiograph specialization: • MDC_DEV_SUB_SPEC_PROFILE_ECG (0x10 0x8C) • MDC_DEV_SUB_SPEC_PROFILE_HR (0x10 0x8D) ▪ Profiles for SABTE specialization: If PHD supports any of the profiles defined for SABTE THEN PHD shall support SABTE specialization, • MDC_DEV_SUB_SPEC_PROFILE_CPAP (0x10 0x94) • MDC_DEV_SUB_SPEC_PROFILE_CPAP_AUTO (0x10 0x95) • MDC_DEV_SUB_SPEC_PROFILE_BPAP (0x10 0x96) • Profiles for power status monitor specialization: If the PHD supports any of the

	profiles defined for Power Status THEN the PHD shall support power status monitor specialization: • MDC_DEV_SUB_SPEC_PROFILE_EIGHT_OR_LESS_BATTERIES (0x10 0x99) • MDC_DEV_SUB_SPEC_PROFILE_MORE_THAN_EIGHT_BATTERIES (0x10 0x9A) h. IF Attribute-Value-Map is present □ attribute-id = MDC_ATTR_ATRIBUTE_VAL_MAP (0X0A 0X55) □ attribute-type = AttrValMap □ attribute-value.count = M □ attribute-value.length = M*4 bytes □ attribute-value = <check that M attributes are defined here> i. IF MDS TimeInfo is present □ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) □ attribute-type = MdsTimeInfo □ attribute-value.length = 16 bytes □ attribute-value = ▪ mds-time-cap-state: • IF (C_AG_OXP_006 = TRUE) THEN mds-time-capab-real-time-clock = 1 ELSE mds-time-capab-real-time-clock = 0 • IF (C_AG_OXP_008 = TRUE) THEN mds-time-capab-set-clock = 1 ELSE mds-time-capab-set-clock = 0 • IF (C_AG_OXP_010 = TRUE) THEN mds-time-capab-relative-time = 1 ELSE mds-time-capab-relative-time = 0 • IF (C_AG_OXP_011 = TRUE) THEN mds-time-capab-high-res-relative-time = 1 ELSE mds-time-capab-high-res-relative-time = 0 • IF (C_AG_OXP_014 = TRUE) THEN mds-time-capab-bo-time = 1 ELSE mds-time-capab-bo-time = 0 • IF (C_AG_OXP_015 = TRUE) THEN mds-time-bo-time-utc-aligned(14) = 1 ELSE mds-time-bo-time-utc-aligned(14) = 0 • IF (C_AG_OXP_007 = TRUE AND C_AG_OXP_009 = TRUE) THEN mds-time-capab-sync-abs-time = 1 or 0 ELSE mds-time-capab-sync-abs-time = 0 • IF (C_AG_OXP_007= TRUE AND C_AG_OXP_010 = TRUE) THEN mds-time-capab-sync-rel-time = 1 or 0 ELSE mds-time-capab-sync-rel-time = 0 • IF (C_AG_OXP_007 = TRUE AND C_AG_OXP_011 = TRUE) THEN mds-time-capab-sync-hi-res-relative-time = 1 or 0 ELSE mds-time-capab-sync-hi-res-relative-time = 0. • IF (C_AG_OXP_007 = TRUE AND C_AG_OXP_014 = TRUE) THEN mds-time-capab-sync-bo-time = 1 or 0 ELSE mds-time-capab-sync-bo-time = 0 • IF (C_AG_OXP_007 = TRUE) THEN mds-time-capab-sync-abs-time = 1 OR mds-time-capab-sync-res-time = 1 OR mds-time-capab-sync-hi-res-relative-time = 1 OR mds-time-capab-sync-bo-time = 1 • Only one of mds-time-capab-real-time-clock and mds-time-capab-bo-time bits shall be set to 1. • Only one of mds-time-capab-sync-abs-time and mds-time-capab-sync-bo-time bits shall be set to 1. • Only one of mds-time-state-abs-time-synced and mds-time-state-bo-time-synced shall be set to 1 ▪ Time-sync-protocol:

	• IF (C_AG_OXP_007 = FALSE) THEN time-sync-protocol = MDC_TIME_SYNC_NONE ELSE time-sync-protocol = (MDC_TIME_SYNC_EBWW or MDC_TIME_SYNC_NTPV3 or MDC_TIME_SYNC_NTPV4 or MDC_TIME_SYNC_SNTPV4 or MDC_TIME_SYNC_SNTPV4330 or MDC_TIME_SYNC_BTV1) ▪ Time-sync-accuracy: • IF (C_AG_OXP_007= FALSE) THEN time-sync-accuracy = 0xFFFFFFFF ▪ Time-resolution-abs-time: • IF (C_AG_OXP_009 = FALSE AND C_AG_OXP_014 = FALSE) THEN time-resolution-abs-time = 0x0000 ▪ Time-resolution-rel-time: • IF (C_AG_OXP_010= FALSE) THEN time-resolution-rel-time = 0x0000 ▪ Time-resolution-high-res-time: • IF (C_AG_OXP_011 = FALSE) THEN time-resolution-high-res-time = 0x0000 j. IF attribute Date-and-Time is present ❑ attribute-id = MDC_ATTR_TIME_ABS (0x09 0x87) ❑ attribute-type = AbsoluteTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = ▪ century = ▪ year ≤ 99 ▪ month ≤ 12 ▪ day ≤ 31 ▪ hour ≤ 24 ▪ minute ≤ 60 ▪ second ≤ 60 ▪ sec-fractions ≤ 100 ❑ If Date-and-Time is present THEN Base-Offset-Time shall not be present. k. IF (C_AG_OXP_014 = TRUE) THEN Base-Offset-Time attribute shall be present ELSE Base-Offset-Time attribute shall not be present ❑ attribute-id = MDC_ATTR_TIME_BO (0x0A 0x81) ❑ attribute-type = BaseOffsetTime ❑ attribute-value.length = 8 bytes ❑ If Base-Offset-Time is present THEN Date-and-Time shall not be present. l. IF Relative-Time attribute is present ❑ attribute-id = MDC_ATTR_TIME_REL (0x09 0x8F) ❑ attribute-type = RelativeTime ❑ attribute-value.length = 4 bytes ❑ Verify that C_AG_OXP_010 is set to True m. IF HiRes-Relative-Time attribute is present ❑ attribute-id = MDC_ATTR_TIME_REL_HI_RES (0x09 0xE9) ❑ attribute-type = HighResRelativeTime ❑ attribute-value.length = 8 bytes n. IF Date-and-Time-Adjustment attribute is present ❑ attribute-id = MDC_ATTR_TIME_ABS_ADJUST (0x0A 0X62)_

	□ attribute-type = AbsoluteTimeAdjust □ attribute-value.length = 6 bytes □ attribute-value = 0 □ Note: If queried with Get MDS command, this attribute shall be not present or 0. o. IF Production-Specification attribute is present □ attribute-id = MDC_ATTR_ID_PROD_SPECN (0X09 0X2D) □ attribute-type = ProductionSpec □ attribute-value.length = <even> □ attribute-value = <Vendor specific> p. IF Power-Status attribute is present □ attribute-id = MDC_ATTR_POWER_STAT (0X09 0X55) □ attribute-type = PowerStatus □ attribute-value.length = 2 bytes □ attribute-value = <not relevant for this test> q. IF Battery-Level attribute is present □ attribute-id = MDC_ATTR_VAL_BATT_CHARGE (0X09 0X9C) □ attribute-type = INT-U16 □ attribute-value.length = 2 bytes □ attribute-value = <value between 0 and 100> If value >100, the meaning of the value is "undefined" r. IF Remain-Battery-Time attribute is present □ attribute-id = MDC_ATTR_TIME_BATT_REMAIN (0X09 0X88) □ attribute-type = BatMeasure □ attribute-value.length = 6 bytes □ attribute-value = <4 bytes to define the value. 2 remaining bytes to define the units, which shall be set to one of: MDC_DIM_MIN (0x08 0xA0), MDC_DIM_HR (0x08 0xC0) or MDC_DIM_DAY (0x08 0xE0) > s. IF attribute Reg Cert Data List is present □ attribute-id = MDC_ATTR_REG_CERT_DATA_LIST (0X0A 0X4B) □ attribute-type = RegCertDataList □ attribute-value.length = < Variable to be checked> □ attribute-value = <Depends on the Authorization body, checked on the Design Guidelines> t. IF Confirm Timeout attribute is present: □ attribute-id = MDC_ATTR_CONFIRM-TIMEOUT (0x09 0x14) □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes Furthermore, if MDS Scan Event Reports are sent by the PHD to report data for MDS object: 7. Wait for a Scan Event Report from the PHD. a. If the PHD sends Fixed Format Event Report for an MDS object, an Attribute-Value-Map has to be received in the GET response. b. If the PHD sends Variable Format Event Report for an MDS object, the attributes whose values can be reported will be the attributes defined as dynamic\observational: Attribute-Value-Map, Mds-Time-Info, Date-and-Time or Base-Offset-Time, Relative-Time, HiRes-Relative-Time, Date-and-Time-Adjustment, Power-Status, Battery-Level, Remaining-Battery-Time.Dynamic attribute Confirm-Timeout is recommended not to be present. If Static attributes are present, the value shall remain unchanged. Handle, System-Type, System-Model, System-Id, Dev-

	Configuration-Id, Production-Specification, Reg-Cert-Data-List, System-Type-Spec-List.
Pass/Fail criteria	• All checked values are as specified in the test procedure. • The total size of the response cannot exceed the sum of the APDU sizes of the supported specializations (limited to an absolute limit of 64512 octets): o Pulse oximeter → 9216 octets o Weighing scales → 896 octets o Glucose meter → 5120 octets or 64512 octets if the PHD supports PM-Store o Blood pressure → 896 octets o Thermometer → 896 octets o Independent activity hub → 5120 octets o Cardiovascular → 64512 octets or 6624 octets if it supports Step Counter Profile o Strength → 64512 octets o Adherence monitor → 1024 octets o Peak flow → 2030 octets o Body composition analyser → 7730 octets o Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store o Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store o International normalized ratio → 896 octets or 64512 if the PHD supports PM-Store o Insulin pump → 7168 octets or 5120 if PHD supports PM-Store o Sleep apnoea breathing therapy equipment → 64512 octets o Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store o Power status monitor → 940 octets for Simple PSM profile or 1660 if PHD supports Advanced PSM profile
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-000_B		
TP label		MDS Object: Dev-Configuration-Id and System-Id semantic feature		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MDSClassAttr 6; M	ConfNormalProc 2; M	ConfNormalProc 7 ; M
Test purpose		Check that: The Dev-Configuration-Id is consistent between the Configuring state and the Operating state. [AND] The PHD uses a "Remote Operation Invoke | Confirmed Event Report" data message with an event-type of MDC_NOTI_CONFIG to send its configuration to the PHG		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_291, C_AG_OXP_292, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test with a dev-config-id and a system-id. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG.		

	4. Record the field config-report-id. 5. If the config-report-id is that of the configuration being tested, the simulated PHG responds with a rors-cmip-confirmed-event-report with result "accepted", else the PHG responds with result "unsupported-config". 6. Repeat step 5 until the config-report-id is set to the configuration being tested. 7. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. d. Operating state is reached. 8. ELSE IF C_AG_OXP_291 OR C_AG_OXP_292 THEN: a. Once in Operating state, simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a “rors-cmip-get” service message in which the attribute-list contains a list of all implemented attributes of the MDS object. 9. Disconnect the device and connect it again. The simulated PHG receives an association request from the PHD. Check the system-id.
Pass/Fail criteria	• Dev-Config-Id must be the same in step 1 and step 4 • Dev-Config-Id must be the same in step 6 and in step 9 • System-Id must be the same in steps 1, 7 or 8, and 9
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-000_C		
TP label		MDS Object: Confirm-Timeout attribute, semantic feature		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MDSClassAttr 19; M	OperErrorCond 5; M	OperErrorCond 6; M
		TimeOutVar 1; C		
Test purpose		Check that: If the attribute Confirm-Timeout is supported, then its value matches with the actual timeout value that the PHD uses for the Confirmed Event Report generated from the MDS object. [AND] If the attribute is not present, the PHD shall use the value 3 s		
Applicability		(C_AG_OXP_182 OR C_AG_OXP_183 OR C_AG_OXP_184 OR C_AG_OXP_189) AND C_AG_OXP_053 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Record the Confirm-Timeout value from the Get MDS operation. If the attribute is not present in the MDS its value shall be $TO_{cer\text{-}mds}$ (3s). 2. Take a measurement with the PHD under test that will provoke an MDS event report to		

	be sent. 3. The PHD sends a "Remote Operation Invoke | Confirmed Event Report". 4. The simulated PHG does not respond for at least the time specified in the field Confirm-Timeout or 3 s if Confirm-Timeout is not supported.
Pass/Fail criteria	The PHD must wait the specified time before unassociating.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-001_A		
TP label		Static Numeric attributes derived from Metrics class and Dynamic Numeric attributes		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MetricClassAttr 1; M	MetricClassAttr 2; M	MetricClassAttr 3; O
		MetricClassAttr 4; M	MetricClassAttr 5; O	MetricClassAttr 7; O
		MetricClassAttr 8; O	MetricClassAttr 9; C	MetricClassAttr 10; O
		MetricClassAttr 11; O	MetricClassAttr 12; C	MetricClassAttr 13; O
		MetricClassAttr 14; O	MetricClassAttr 15; O	MetricClassAttr 16; C
		MetricClassAttr 17; C	MetricClassAttr 18; C	MetricClassAttr 19; O
		NumClass 1; M	NumClass 2; M	NumClassAttr 1; M
		NumClassAttr 2; C	NumClassAttr 3; C	NumClassAttr 4; C
		NumClassAttr 5; C	NumClassAttr 6; C	NumClassAttr 7; C
		NumClassAttr 8; O	ConfNormalProc 1;M	ConfEventRep 29; M
		ConfEventRep 30; M	ConfEventRep 31; C	ConfEventRep 33; O
		ConfEventRep 33; O		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M	General 2; M	
Test purpose		Check that: Numeric class is derived from the Metric base class. It inherits all mandatory attributes and conditional attributes as required by their conditions and it may import optional attributes. [AND] The nomenclature code to identify the Numeric class is MDC_MOC_VMO_METRIC_NU [AND] Static, dynamic and observational attributes. [AND] Changes to any attribute values of metric and scanner objects shall be reported to the PHG in scan event reports prior to sending event reports that depend on those values (e.g. scan-handle-attr-val-map and a group format event report or unit-code and the observed value). [AND] Continua PAN service components shall not include the Base Offset Time in any Continua configurations except for Basic electrocardiograph (ECG) device specialization.		
Applicability		C_AG_OXP_040 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009,C_AG_OXP_014, C_AG_OXP_041, C_AG_OXP_046, C_AG_OXP_047, C_AG_OXP_182, C_AG_OXP_183, C_AG_OXP_184, C_AG_OXP_189. C_AG_OXP_190, C_AG_OXP_192, C_AG_OXP_193, C_AG_OXP_194, C_AG_OXP_195, C_AG_OXP_196, C_AG_OXP_197, C_AG_OXP_198, C_AG_OXP_199, C_AG_OXP_200, C_AG_OXP_201, C_AG_OXP_202, C_AG_OXP_203, C_AG_OXP_230, C_AG_OXP_231, C_AG_OXP_232, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The PHD under test sends and the simulated PHG receives an association request from		

	the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. Numeric class attributes must be (ConfigReport -> ConfigObject-> AttributeList): a. Mandatory attribute Handle shall be present. □ attribute-type = HANDLE □ attribute-length = 2 bytes □ attribute-value = must be unique and non-zero. Actual value may be specified by the Device Specialization. b. Mandatory attribute Type shall be present in ConfigReport □ attribute-id = MDC_ATTR_ID_TYPE (0X09 0X2F) □ attribute-type = TYPE □ attribute-value = SEQUENCE OF (SIZE 6) □ attribute-value = <Not relevant in this test> c. Mandatory attribute Metric-Spec-Small should be present □ attribute-id = MDC_ATTR_METRIC_SPEC_SMALL □ attribute-type = MetricSpecSmall □ attribute-value.length = 2 bytes □ attribute-value = ▪ IF C_AG_OXP_201=TRUE and C_AG_OXP_041=FALSE THEN mss-avail-stored-data(1)=1(There is at least one object that has mss-avail.stored-data(1)=1) ▪ IF C_AG_OXP_201=TRUE and C_AG_OXP_041=TRUE THEN mss-avail-stored-data(1)=1 or 0 d. Only one attribute of Metric-Id and Metric-Id-List shall be present. e. If Metric-Id-List attribute is supported, it should be present in ConfigReport □ attribute-id = MDC_ATTR_ID_PHYSIO_LIST □ attribute-type = MetricIdList □ attribute-value.length = SEQUENCE OF (SIZE 2) □ attribute-value = □ The [Metric-Id-List] attribute shall be used if a compound observed value is used, which does not incorporate the Metric-Id directly. The order of the Metric-Id-List shall correspond to the order of the elements in the compound observed value. □ IF the PHD supports Metric-Id-List at least for a Numeric object (C_AG_OXP_190 =TRUE) THEN this attribute shall be present at least for one object, ELSE this attribute is not present. f. IF Metric-Id attribute is supported, it should be present in ConfigReport □ attribute-id = MDC_ATTR_ID_PHYSIO □ attribute-type = OID-Type □ attribute-value.length = 2 bytes g. IF Attribute-Value-Map is supported, it should be present in ConfigReport □ attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP (0X0A 0X5A) □ attribute-type = AttrValMap □ attribute-count = n (record for next attribute field) □ attribute-value.length = n*4 bytes □ attribute-value = <n attributes will be declared, each one will have 4 bytes

	composed of: ▪ attribute-id = 2 bytes (MDC_ATTR_*). ▪ Attribute-length = 0x00 0x02: (2 bytes to declare the length of the attribute, but the contents of the attribute in the event report is not these 2 bytes length) h. IF Supplemental-Types attribute is supported, it shall be present in ConfigReport: □ attribute-id = MDC_ATTR_SUPPLEMENTAL_TYPES □ attribute-type = SupplementalTypeList □ attribute-value.length = SEQUENCE OF (SIZE (4)) □ attribute-value = <Not relevant in this test> i. IF attribute Metric-Structure-Small is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_METRIC_STRUCTURE_SMALL □ attribute-type = MetricStructureSmall □ attribute-value.length = 2 bytes □ attribute-value = ▪ ms-struct = one of the following: • ms-struct-simple (0x00) • ms-struct-compound (0x01) • ms-struct-reserved (0x02) • ms-struct-compound-simple (0x03) ▪ ms-compound-no = one of the following: • IF ms-struct = ms-struct-simple THEN = 0 • ELSE = maximum number of components in a compound value j. IF attribute Metric-Id-Partition is supported, it should be present in ConfigReport □ attribute-id = MDC_ATTR_METRIC_ID_PART □ attribute-type = NomPartition □ attribute-value.length = 2 bytes □ attribute-value = one of the next ▪ nom-part-unspec (0x00 0x00) ▪ nom-part-obj (0x00 0x01) ▪ nom-part-metric (0x00 0x02) ▪ nom-part-alert (0x00 0x03) ▪ nom-part-dim (0x00 0x04) ▪ nom-part-vattr (0x00 0x05) ▪ nom-part-pgrp (0x00 0x06) ▪ nom-part-sites (0x00 0x07) ▪ nom-part-infrastruc (0x00 0x08) ▪ nom-part-fef (0x00 0x09) ▪ nom-part-ecg-extn (0x00 0x0A) ▪ nom-part-phd-dm (0x00 0x80) ▪ nom-part-phd-hf (0x00 0x81) ▪ nom-part-phd-ai (0x00 0x82) ▪ nom-part-ret-code(0x00 0xFF) ▪ nom-part-ext-nom (0x01 0x00) ▪ nom-part-priv (0x04 0x00)

	k. IF attribute Unit-Code is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_UNIT_CODE □ attribute-type = OID-Type □ attribute-value.length = 2 bytes □ attribute-value = One of MDC_PART_DIM (may be defined in the specialization) l. IF attribute Source-Handle-Reference is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_SOURCE_HANDLE_REF □ attribute-type = HANDLE □ attribute-value.length = 2 bytes □ attribute-value = < The value of an existing object´s handle > m. IF attribute Label-String is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_LABEL_STRING □ attribute-type = OCTET STRING □ attribute-value.length = □ attribute-value = <Textual representation of attribute Type, Printable ASCII> n. IF attribute Unit-Label-String is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_UNIT_LABEL_STRING □ attribute-type = OCTET STRING □ attribute-value.length = □ attribute-value = <Textual representation of attribute Unit-Code, Printable ASCII> o. IF attribute Accuracy is supported, it shall be present in ConfigReport: □ attribute-id = MDC_ATTR_NU_ACCUR_MSMT □ attribute-type = FLOAT-Type □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> p. IF attribute Measure-Active-Period is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_TIME_PD_MSMT_ACTIVE □ attribute-type = FLOAT-Type □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> q. Time-stamp attributes (Absolute-Time-Stamp,Base-Offset-Time,Relative-Time-Stamp and HiRes-Time-Stamp), observed values attributes (Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, Compound-Basic-Nu-Observed-Value, Compound-Nu-Observed-Value) and Measurement-Status shall not be present if mss-cat-setting and/or mss-cat-manual bit of the Metric-Spec-Small attribute is set to 0. If any of these bits is set to 1, observational attributes may be present in ConfigReport. Furthermore if MDS event reports are sent by the PHD, (the PHD sends fixed format value messages to report dynamic data for Numeric Objects or uses variable format event report, C_AG_OXP_182 = TRUE or C_AG_OXP_189 = TRUE): 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object.

	c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Take a measurement with the PHD 7. Wait for an event report from the PHD a. One of {Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compound-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, Compound-Basic-Nu-Observed-Value} shall be present b. IF attribute Simple-Nu-Observed-Value is present □ attribute-id = MDC_ATTR_NU_VAL_OBS_SIMP □ attribute-type = SimpleNuObsValue □ attribute-value.length = 4 bytes □ attribute-value =<Not relevant in this test> c. IF attribute Basic-Nu-Observed-Value is present □ attribute-id = MDC_ATTR_NU_VAL_OBS_BASIC □ attribute-type = BasicNuObsValue □ attribute-value.length = 2 bytes □ attribute-value = <Not relevant in this test> d. IF attribute Nu-Observed-Value is present □ attribute-id = MDC_ATTR_NU_VAL_OBS □ attribute-type = NuObsValue □ attribute-value.length = 10 bytes □ attribute-value = <Not relevant in this test> e. IF attribute Compound-Simple-Nu-Observed-Value is present □ attribute-id = MDC_ATTR_NU_CMPD_VAL_OBS_SIMP □ attribute-type = SimpleNuObsValueCmp □ attribute-value.length = SEQUENCE OF (SIZE (4)) □ attribute-value = <Not relevant in this test> f. IF attribute Compound-Basic-Nu-Observed-Value is present □ attribute-id = MDC_ATTR_NU_CMPD_VAL_OBS_BASIC □ attribute-type = BasicNuObsValueCmp □ attribute-value.length = SEQUENCE OF (SIZE(4)) □ attribute-value = <Not relevant in this test> g. IF attribute Compound-Nu-Observed-Value is present □ attribute-id = MDC_ATTR_NU_CMPD_VAL_OBS □ attribute-type = NuObsValueCmp □ attribute-value.length = SEQUENCE OF (SIZE (10)) □ attribute-value =<Not relevant in this test> h. IF attribute Measure-Active-Period is present □ attribute-id = MDC_ATTR_TIME_PD_MSMT_ACTIVE □ attribute-type = FLOAT-Type

	□ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> i. IF attribute Absolute-Time-Stamp is present □ attribute-id = MDC_ATTR_TIME_STAMP_ABS □ attribute-type = AbsoluteTime □ attribute-value.length = 8 bytes □ attribute-value = <Not relevant in this test> ▪ century = ▪ year ≤ 99 ▪ month ≤ 12 ▪ day ≤ 31 ▪ hour ≤ 24 ▪ minute ≤ 60 ▪ second ≤ 60 ▪ sec-fractions ≤ 100 □ If an PHD stores data, it shall associate a time stamp with the data. □ If an PHD supports Absolute-Time-Stamp, Base-Offset–Time-Stamp attribute shall not be supported. j. IF (C_AG_ OXP_014 = TRUE) THEN Attribute Base-Offset-Time-Stamp may be present ELSE it shall not be present □ attribute-id = MDC_ATTR_TIME_STAMP_BO □ attribute-type = BaseOffsetTime □ attribute-value.length = 8 bytes □ attribute-value = <Not relevant in this test> □ If an PHD stores data, it shall associate a time stamp with the data. □ If an PHD supports Base-Offset-Time-Stamp, Absolute-Time-Stamp attribute shall not be supported. k. IF attribute Relative-Time-Stamp is present □ attribute-id = MDC_ATTR_TIME_STAMP_REL □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> □ If a PHD stores data, it shall associate a time stamp with the data l. IF attribute HiRes-Time-Stamp is present □ attribute-id = MDC_ATTR_TIME_STAMP_REL_HI_RES □ attribute-type = HighResRelativeTime □ attribute-value.length = 8 bytes □ attribute-value =<Not relevant in this test> m. IF attribute Measurement Status is present □ attribute-id = MDC_ATTR_MSMT_STAT □ attribute-type = MeasurementStatus □ attribute-value.length = 2 bytes □ attribute-value = <not relevant for this test> n. Check dynamic attributes that can be present: Metric-Structure-Small, Metric-Id, Metric-Id-List, Metric-Id-Partition, Unit-Code,Source-Handle-Reference, Label-String, Unit-Label-String, Measure-Active-Period

Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	Checking that "IF C_AG_OXP_201=FALSE THEN mss-avail-stored data(1)=0" for MetricSpecSmall attribute has been removed because [ISO/IEEE 11073-20601-2015A] has specified (subsection A.11.3) that "The setting of bits 0 to 5 is primarily informational and shall be set if the condition is true but- a PHG cannot assume that if they are set the behaviour will be observed". According to Device Specialization spec, standard configurations (most of them) state the value for bit mss-avail-stored-data to 1, even if the PHD cannot store measurements. So, if the test procedure checks that if C_AG_OXP_201=FALSE then mss-avail-stored-data = 0 the test case will give a FAIL verdict and it would not be correct according to what [ISO/IEEE 11073-20601-2015A] states in clause A.11.3.

TP Id		TP/PLT/PHD/OXP/DIM/BV-001_B		
TP label		Static Enumerated attributes derived from Metrics class and Dynamic Enumerated attributes		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MetricClassAttr 1; M	MetricClassAttr 2; M	MetricClassAttr 3; O
		MetricClassAttr 4; M	MetricClassAttr 5; O	MetricClassAttr 7; O
		MetricClassAttr 8; O	MetricClassAttr 9; C	MetricClassAttr 10; O
		MetricClassAttr 11; O	MetricClassAttr 12; C	MetricClassAttr 13; O
		MetricClassAttr 14; O	MetricClassAttr 15; O	MetricClassAttr 16; C
		MetricClassAttr 17; C	MetricClassAttr 18; C	MetricClassAttr 19; O
		EnumClass 2; M	EnumClass 3; M	EnumClassAttr 1; M
		EnumClassAttr 2; C	EnumClassAttr 3; C	EnumClassAttr 4; C
		EnumClassAttr 5; C	EnumClassAttr 6; C	EnumClassAttr 7; O
		ConfNormalProc 1; M	ConfEventRep 29; M	ConfEventRep 30; M
		ConfEventRep 31; C	ConfEventRep 33; O	
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M	General 2; M	
Test purpose		Check that: Enumerated class is derived from the Metric base class. It inherits all mandatory attributes and conditional attributes as required by their conditions and it may import optional attributes. [AND] The nomenclature code to identify the Enumerated class is MDC_MOC_VMO_METRIC_ENUM. [AND] Static, dynamic and observational attributes. [AND] Changes to any attribute values of metric and scanner objects shall be reported to the PHG in scan event reports prior to sending event reports that depend on those values (e.g. scan-handle-attr-val-map and a group format event report or unit-code and the observed value). [AND] Continua PAN service components shall not include the Base Offset Time in any Continua configurations except for Basic electrocardiograph (ECG) device specialization.		
Applicability		C_AG_OXP_043 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009,C_AG_OXP_014, C_AG_OXP_041, C_AG_OXP_046, C_AG_OXP_047, C_AG_OXP_182, C_AG_OXP_183, C_AG_OXP_184, C_AG_OXP_189. C_AG_OXP_190, C_AG_OXP_192, C_AG_OXP_193, C_AG_OXP_194, C_AG_OXP_195, C_AG_OXP_196, C_AG_OXP_197, C_AG_OXP_198, C_AG_OXP_199, C_AG_OXP_200, C_AG_OXP_201, C_AG_OXP_202, C_AG_OXP_203, C_AG_OXP_230, C_AG_OXP_231, C_AG_OXP_232,		

	C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. Enumeration object attributes must be (ConfigReport -> ConfigObject-> AttributeList): a. Mandatory attribute Handle shall not be present □ attribute-type = HANDLE □ attribute-value = 2 bytes □ attribute-value = must be unique and non-zero. Actual value may be specified by the Device Specialization. b. Mandatory attribute Type shall be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_TYPE (0X09 0X2F) □ attribute-type = TYPE □ attribute-value = SEQUENCE OF (SIZE 6) □ attribute-value = <Not relevant in this test> c. Mandatory attribute Metric-Spec-Small should be present □ attribute-id = MDC_ATTR_METRIC_SPEC_SMALL □ attribute-type = MetricSpecSmall □ attribute-value.length = 2 bytes □ attribute-value = <Checked in the specializations > ▪ IF C_AG_OXP_202=TRUE and C_AG_OXP_041=FALSE THEN mss-avail-stored-data(1)=1 (There is at least one object that has mss-avail-stored-data(1) = 1) ▪ IF C_AG_OXP_202=TRUE and C_AG_OXP_041=TRUE THEN mss-avail-stored-data(1)=1 or 0 d. Only one attribute of Metric-Id and Metric-Id-List shall be present. e. IF Metric-Id-List attribute is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_PHYSIO_LIST □ attribute-type = MetricIdList □ attribute-value.length = SEQUENCE OF (SIZE 2) □ attribute-value = □ The [Metric-Id-List] attribute shall be used if a compound observed value is used, which does not incorporate the Metric-Id directly. The order of the Metric-Id-List shall correspond to the order of the elements in the compound observed value. f. IF Metric-Id attribute is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_PHYSIO □ attribute-type = OID-Type □ attribute-value.length = 2 bytes g. IF Attribute-Value-Map is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP (0X0A 0X55) □ attribute-type = AttrValMap □ attribute-count = n (record for next attribute field) □ attribute-value.length = n*4 bytes □ attribute-value = <n attributes will be declared, each one will have 4 bytes composed of:

	▪ attribute-id = 2 bytes (MDC_ATTR_*)
	▪ Attribute-length = 0x00 0x02: (2 bytes to declare the length of the attribute, but the contents of the attribute in the event report is not these 2 bytes length
	h. IF Supplemental-Types attribute is supported, it shall be present in ConfigReport:
	□ attribute-id = MDC_ATTR_SUPPLEMENTAL_TYPES
	□ attribute-type = SupplementalTypeList
	□ attribute-value.length = SEQUENCE OF (SIZE (4))
	□ attribute-value = <Not relevant in this test>
	i. IF attribute Metric-Structure-Small is supported, it should be present in ConfigReport:
	□ attribute-id = MDC_ATTR_METRIC_STRUCTURE_SMALL
	□ attribute-type = MetricStructureSmall
	□ attribute-value.length = 2 bytes
	□ attribute-value =
	▪ ms-struct = one of the following:
	• ms-struct-simple (0x01)
	• ms-struct-compound (0x02)
	• ms-struct-reserved (0x03)
	• ms-struct-compound-simple (0x04)
	▪ ms-compound-no = one of the following:
	• IF ms-struct = ms-struct-simple THEN = 0
	• ELSE = maximum number of components in a compound value
	j. IF attribute Metric-Id-Partition is supported, it should be present in ConfigReport:
	□ attribute-id = MDC_ATTR_METRIC_ID_PART
	□ attribute-type = NomPartition
	□ attribute-value.length = 2 bytes
	□ attribute-value = one of the next
	▪ nom-part-unspec (0x00 0x00)
	▪ nom-part-obj (0x00 0x01)
	▪ nom-part-metric (0x00 0x02)
	▪ nom-part-alert (0x00 0x03)
	▪ nom-part-dim (0x00 0x04)
	▪ nom-part-vattr (0x00 0x05)
	▪ nom-part-pgrp (0x00 0x06)
	▪ nom-part-sites (0x00 0x07)
	▪ nom-part-infrastruc (0x00 0x08)
	▪ nom-part-fef (0x00 0x09)
	▪ nom-part-ecg-extn (0x00 0x0A)
	▪ nom-part-phd-dm (0x00 0x80)
	▪ nom-part-phd-hf (0x00 0x81)
	▪ nom-part-phd-ai (0x00 0x82)
	▪ nom-part-ret-code(0x00 0xFF)
	▪ nom-part-ext-nom (0x01 0x00)
	▪ nom-part-priv (0x04 0x00)
	k. IF attribute Unit-Code is supported, it should be present in ConfigReport:

	❑ attribute-id = MDC_ATTR_UNIT_CODE ❑ attribute-type = OID-Type ❑ attribute-value.length = 2 bytes ❑ attribute-value = One of MDC_PART_DIM (may be defined in the specialization) l. IF attribute Source-Handle-Reference is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_SOURCE_HANDLE_REF ❑ attribute-type = HANDLE ❑ attribute-value.length = 2 bytes ❑ attribute-value = < The value of an existing object´s handle > m. IF attribute Label-String is supported, it should be present in ConfigReport ❑ attribute-id = MDC_ATTR_ID_LABEL_STRING ❑ attribute-type = OCTET STRING ❑ attribute-value.length = ❑ attribute-value = <Textual representation of attribute Type, Printable ASCII> n. IF attribute Unit-Label-String is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_UNIT_LABEL_STRING ❑ attribute-type = OCTET STRING ❑ attribute-value.length = ❑ attribute-value = <Textual representation of attribute Unit-Code, Printable ASCII> o. IF attribute Measure-Active-Period is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_TIME_PD_MSMT_ACTIVE ❑ attribute-type = FLOAT-Type ❑ attribute-value.length = 4 bytes ❑ attribute-value =<Not relevant for this test> p. IF attribute Enum-Observed-Value-Partition is supported it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_ENUM_OBS_VAL_PART ❑ attribute-type = NomPartition ❑ attribute-value.length = 2 bytes ❑ attribute-value = one of the next ▪ nom-part-unspec (0x00 0x00) ▪ nom-part-obj (0x00 0x01) ▪ nom-part-metric (0x00 0x02) ▪ nom-part-alert (0x00 0x03) ▪ nom-part-dim (0x00 0x04) ▪ nom-part-vattr (0x00 0x05) ▪ nom-part-pgrp (0x00 0x06) ▪ nom-part-sites (0x00 0x07) ▪ nom-part-infrastruc (0x00 0x08) ▪ nom-part-fef (0x00 0x09) ▪ nom-part-ecg-extn (0x00 0x0A) ▪ nom-part-phd-dm (0x00 0x80)

	▪ nom-part-phd-hf (0x00 0x81) ▪ nom-part-phd-ai (0x00 0x82) ▪ nom-part-ret-code(0x00 0xFF) ▪ nom-part-ext-nom (0x01 0x00) ▪ nom-part-priv (0x04 0x00) q. Time-stamp attributes (Absolute-Time-Stamp, Base-Offset-Time, Relative-Time-Stamp and HiRes-Time-Stamp), observed values attributes (Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Simple-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value) and Measurement-Status shall not be present if mss-cat-setting and/or mss-cat-manual bit of the Metric-Spec-Small attribute is set to 0. If any of these bits is set to 1, observational attributes may be present in ConfigReport. Furthermore if MDS event reports are sent by the PHD (the PHD sends fixed format value messages to report dynamic data for Enumeration Objects or uses variable format event report, C_AG_OXP_183=TRUE or C_AG_OXP_189=TRUE): 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Take a measurement with the PHD. 7. Wait for an event report from the PHD: a. One of Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Simple-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value shall be present. b. IF attribute Measure-Active-Period is present ❑ attribute-id = MDC_ATTR_TIME_PD_MSMT_ACTIVE ❑ attribute-type = FLOAT-Type ❑ attribute-value.length = 4 bytes ❑ attribute-value = <Not relevant for this test> c. IF attribute Enum-Observed-Value-Simple-OID is present ❑ attribute-id = MDC_ATTR_ENUM_OBS_VAL_SIMP_OID ❑ attribute-type = OID-Type ❑ attribute-value.length = 2 bytes ❑ attribute-value = <Not relevant in this test> d. IF attribute Enum-Observed-Value-Simple-Bit-Str is present ❑ attribute-id = MDC_ATTR_ENUM_OBS_VAL_SIMP_BIT_STR ❑ attribute-type = BITS-32 ❑ attribute-value.length = 4 bytes ❑ attribute-value = <Not relevant in this test> e. IF attribute Enum-Observed-Value-Basic-Bit-Str is present

	□ attribute-id = MDC_ATTR_ENUM_OBS_VAL_BASIC_BIT_STR □ attribute-type = BITS-16 □ attribute-value.length = 2 bytes □ attribute-value = <Not relevant in this test> f. IF attribute Enum-Observed-Value-Simple-Str is present □ attribute-id = MDC_ATTR_ENUM_OBS_VAL_SIMP_STR □ attribute-type = EnumPrintableString □ attribute-value.length = <Variable> □ attribute-value = <ASCII printable> g. IF attribute Enum-Observed-Value is present □ attribute-id = MDC_ATTR_VAL_ENUM_OBS □ attribute-type = EnumObsValue □ attribute-value.length = <Variable> □ attribute-value = <Checked in each specialization> h. IF attribute Enum-Observed-Value-Partition is present □ attribute-id = MDC_ATTR_ENUM_OBS_VAL_PART □ attribute-type = NomPartition □ attribute-value.length = 2 bytes □ attribute-value = one of the next ▪ nom-part-unspec (0x00 0x00) ▪ nom-part-obj (0x00 0x01) ▪ nom-part-metric (0x00 0x02) ▪ nom-part-alert (0x00 0x03) ▪ nom-part-dim (0x00 0x04) ▪ nom-part-vattr (0x00 0x05) ▪ nom-part-pgrp (0x00 0x06) ▪ nom-part-sites (0x00 0x07) ▪ nom-part-infrastruc (0x00 0x08) ▪ nom-part-fef (0x00 0x09) ▪ nom-part-ecg-extn (0x00 0x0A) ▪ nom-part-phd-dm (0x00 0x80) ▪ nom-part-phd-hf (0x00 0x81) ▪ nom-part-phd-ai (0x00 0x82) ▪ nom-part-ret-code (0x00 0xFF) ▪ nom-part-ext-nom (0x01 0x00) ▪ nom-part-priv (0x04 0x00) i. IF attribute Absolute-Time-Stamp is present □ attribute-id = MDC_ATTR_TIME_STAMP_ABS □ attribute-type = AbsoluteTime □ attribute-value.length = 8 bytes □ attribute-value = ▪ century = ▪ year ≤ 99 ▪ month ≤ 12

	▪ day ≤ 31 ▪ hour ≤ 24 ▪ minute ≤ 60 ▪ second ≤ 60 ▪ sec-fractions ≤ 100 ❑ If an PHD stores data, it shall associate a time stamp with the data (This case will be tested in TP/PLT/PHD/SER/BV-007) ❑ If an PHD supports Absolute-Time-Stamp, Base-Offset–Time-Stamp attribute shall not be supported. j. IF (C_AG_ OXP_014 = TRUE) THEN Attribute Base-Offset-Time-Stamp may be present ELSE it shall not be present ❑ attribute-id = MDC_ATTR_TIME_STAMP_BO ❑ attribute-type = BaseOffsetTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = <Not relevant in this test> ❑ If a PHD stores data, it shall associate a time stamp with the data. ❑ If a PHD supports Base-Offset-Time-Stamp, Absolute-Time-Stamp attribute shall not be supported. k. IF attribute Relative-Time-Stamp is present ❑ attribute-id = MDC_ATTR_TIME_STAMP_REL ❑ attribute-type = RelativeTime ❑ attribute-value.length = 4 bytes ❑ attribute-value = <Not relevant in this test> l. If a PHD stores data, it shall associate a time stamp with the data m. IF attribute HiResRelative-Time-Stamp is present ❑ attribute-id = MDC_ATTR_TIME_STAMP_REL_HI_RES ❑ attribute-type = HighResRelativeTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = <Not relevant in this test> n. IF attribute Measurement Status is present ❑ attribute-id = MDC_ATTR_MSMT_STAT ❑ attribute-type = MeasurementStatus ❑ attribute-value.length = 2 bytes ❑ attribute-value = <not relevant for this test> o. Check dynamic attributes that can be present: Metric-Structure-Small, Metric-Id, Metric-Id-List, Metric-Id-Partition, Unit-Code,Source-Handle-Reference, Label-String, Unit-Label-String, Measure-Active-Period
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	Checking that "IF C_AG_OXP_202=FALSE THEN mss-avail-stored data(1)=0" for MetricSpecSmall attribute has been removed because [ISO/IEEE 11073-20601-2015A] has specified (clause A.11.3) that "The setting of bits 0 to 5 is primarily informational and shall be set if the condition is true but-- a PHG cannot assume that if they are set the behaviour will be observed". According to the Device Specialization spec, standard configurations (most of them) state the value for bit mss-avail-stored-data to 1, even if the PHD cannot store measurements. So if test procedure checks that if C_AG_OXP_202=FALSE then mss-avail-stored-data =0 the test case will give a FAIL verdict, and it would not be correct according to what [ISO/IEEE 11073-20601-2015A] states (clause A.11.3).

TP Id		TP/PLT/PHD/OXP/DIM/BV-001_C		
TP label		Static RealTime-SA attributes derived from Metrics class and Dynamic RealTime-SA attributes.		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MetricClassAttr 1; M	MetricClassAttr 2; M	MetricClassAttr 3; O
		MetricClassAttr 4; M	MetricClassAttr 5; O	MetricClassAttr 7; O
		MetricClassAttr 8; O	MetricClassAttr 9; C	MetricClassAttr 10; O
		MetricClassAttr 11; O	MetricClassAttr 12; C	MetricClassAttr 13; O
		MetricClassAttr 14; O	MetricClassAttr 15; O	MetricClassAttr 16; C
		MetricClassAttr 17; C	MetricClassAttr 18; C	MetricClassAttr 19; O
		ArrayClass 1; M	ArrayClass 2; M	ArrayClassAttr 1; M
		ArrayClassAttr 2; M	ArrayClassAttr 3; M	ArrayClassAttr 4; M
		ConfNormalProc 1; M	ConfEventRep 29; M	ConfEventRep 30; M
		ConfEventRep 31; C	ConfEventRep 33; O	
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M	General 2; M	
Test purpose		Check that: RT-SA class is derived from the Metric base class. It inherits all mandatory attributes and conditional attributes as required by their conditions and it may import optional attributes. [AND] The nomenclature code to identify the RT-SA class is MDC_MOC_VMO_METRIC_SA_RT. [AND] Static, dynamic and observational attributes. [AND] Changes to any attribute values of metric and scanner objects shall be reported to the PHG in scan event reports prior to sending event reports that depend on those values (e.g. scan-handle-attr-val-map and a group format event report or unit-code and the observed value). [AND] Continua PAN service components shall not include the Base Offset Time in any Continua configurations except for Basic electrocardiograph (ECG) device specialization.		
Applicability		C_AG_OXP_042 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009,C_AG_OXP_014, C_AG_OXP_041, C_AG_OXP_046, C_AG_OXP_047, C_AG_OXP_182, C_AG_OXP_183, C_AG_OXP_184, C_AG_OXP_189. C_AG_OXP_190, C_AG_OXP_192, C_AG_OXP_193, C_AG_OXP_194, C_AG_OXP_195, C_AG_OXP_196, C_AG_OXP_197, C_AG_OXP_198, C_AG_OXP_199, C_AG_OXP_200, C_AG_OXP_201, C_AG_OXP_202, C_AG_OXP_203, C_AG_OXP_230, C_AG_OXP_231, C_AG_OXP_232, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. RealTime-SA object attributes must be(ConfigReport -> ConfigObject-> AttributeList): a. Mandatory attribute Handle shall not be present □ attribute-type = HANDLE □ attribute-length = 2 bytes □ attribute-value = must be unique and non-zero. Actual value may be specified		

	by the Device Specialization. b. Mandatory attribute Type shall be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_TYPE (0X09 0X2F) □ attribute-type = TYPE □ attribute-value = SEQUENCE OF (SIZE 6) □ attribute-value = <Not relevant in this test> c. Mandatory attribute Metric-Spec-Small should be present □ attribute-id = MDC_ATTR_METRIC_SPEC_SMALL □ attribute-type = MetricSpecSmall □ attribute-value.length = 2 bytes □ attribute-value = <Checked in the specializations> ▪ Bit mss-avail-store-data must be set to 0 d. Only one attribute of Metric-Id and Metric-Id-List shall be present. e. If Metric-Id-List attribute is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_PHYSIO_LIST □ attribute-type = MetricIdList □ attribute-value.length = SEQUENCE OF (SIZE 2) □ attribute-value = □ The [Metric-Id-List] attribute shall be used if a compound observed value is used, which does not incorporate the Metric-Id directly. The order of the Metric-Id-List shall correspond to the order of the elements in the compound observed value. f. IF Metric-Id attribute is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ID_PHYSIO □ attribute-type = OID-Type □ attribute-value.length = 2 bytes g. IF Attribute-Value-Map is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP (0X0A 0X55) □ attribute-type = AttrValMap □ attribute-count = n (record for next attribute field) □ attribute-value.length = n*4 bytes □ attribute-value = <n attributes will be declared, each one will have 4 bytes composed of: ▪ attribute-id = 2 bytes (MDC_ATTR_*) ▪ Attribute-length = 0x00 0x02: (2 bytes to declare the length of the attribute, but the contents of the attribute in the event report is not these 2 bytes length h. IF Supplemental-Types attribute is supported, it shall be present in ConfigReport: □ attribute-id = MDC_ATTR_SUPPLEMENTAL_TYPES □ attribute-type = SupplementalTypeList □ attribute-value.length = SEQUENCE OF (SIZE (4)) □ attribute-value = <Not relevant in this test> i. IF attribute Metric-Structure-Small is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_METRIC_STRUCTURE_SMALL □ attribute-type = MetricStructureSmall □ attribute-value.length = 2 bytes □ attribute-value =

	▪ ms-struct = one of the following: • ms-struct-simple (0x01) • ms-struct-compound (0x02) • ms-struct-reserved (0x03) • ms-struct-compound-simple (0x04) ▪ ms-compound-no = one of the following: • IF ms-struct = ms-struct-simple THEN = 0 • ELSE = maximum number of components in a compound value j. IF attribute Metric-Id-Partition is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_METRIC_ID_PART ❑ attribute-type = NomPartition ❑ attribute-value.length = 2 bytes ❑ attribute-value = one of the next ▪ nom-part-unspec (0x00 0x00) ▪ nom-part-obj (0x00 0x01) ▪ nom-part-metric (0x00 0x02) ▪ nom-part-alert (0x00 0x03) ▪ nom-part-dim (0x00 0x04) ▪ nom-part-vattr (0x00 0x05) ▪ nom-part-pgrp (0x00 0x06) ▪ nom-part-sites (0x00 0x07) ▪ nom-part-infrastruc (0x00 0x08) ▪ nom-part-fef (0x00 0x09) ▪ nom-part-ecg-extn (0x00 0x0A) ▪ nom-part-phd-dm (0x00 0x80) ▪ nom-part-phd-hf (0x00 0x81) ▪ nom-part-phd-ai (0x00 0x82) ▪ nom-part-ret-code(0x00 0xFF) ▪ nom-part-ext-nom (0x01 0x00) ▪ nom-part-priv (0x04 0x00) k. IF attribute Unit-Code is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_UNIT_CODE ❑ attribute-type = OID-Type ❑ attribute-value.length = 2 bytes ❑ attribute-value = One of MDC_PART_DIM (may be defined in the specialization) l. IF attribute Source-Handle-Reference is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_SOURCE_HANDLE_REF ❑ attribute-type = HANDLE ❑ attribute-value.length = 2 bytes ❑ attribute-value = < The value of an existing object´s handle > m. IF attribute Label-String is supported, it should be present in ConfigReport: ❑ attribute-id = MDC_ATTR_ID_LABEL_STRING ❑ attribute-type = OCTET STRING

   - ❑ attribute-value.length =
   - ❑ attribute-value = <Textual representation of attribute Type, Printable ASCII>

n. IF attribute Unit-Label-String is supported, it should be present in ConfigReport:

   - ❑ attribute-id = MDC_ATTR_UNIT_LABEL_STRING
   - ❑ attribute-type = OCTET STRING
   - ❑ attribute-value.length =
   - ❑ attribute-value = <Textual representation of attribute Unit-Code, Printable ASCII>

o. IF attribute Measure-Active-Period is supported, it should be present in ConfigReport:

   - ❑ attribute-id = MDC_ATTR_TIME_PD_MSMT_ACTIVE
   - ❑ attribute-type = FLOAT-Type
   - ❑ attribute-value.length = 4 bytes
   - ❑ attribute-value = <Not relevant in this test>

p. Mandatory attribute Sample-Period shall be present in ConfigReport:

   - ❑ attribute-id = MDC_ATTR_TIME_PD_SAMP
   - ❑ attribute-type = RelativeTime
   - ❑ attribute-value.length = 4 bytes
   - ❑ attribute-value = <Not relevant in this test>

q. Mandatory attribute Scale-and-Range-Specification shall be present in ConfigReport:

   - ❑ attribute-id = MDC_ATTR_SCALE_SPECN_I8; MDC_ATTR_SCALE_SPECN_I16; MDC_ATTR_SCALE_SPECN_I32
   - ❑ attribute-type = ScaleRangeSpec8 OR ScaleRangeSpec16 OR ScaleRangeSpec32
   - ❑ attribute-value.length = 1, 2 OR 4 bytes, depending of the type
   - ❑ attribute-value = <Not relevant in this test>

r. Mandatory attribute Sa-Specification shall be present in ConfigReport:

   - ❑ attribute-id = MDC_ATTR_SA_SPECN
   - ❑ attribute-type = SaSpec
   - ❑ attribute-value.length = 6 bytes
   - ❑ attribute-value = <Not relevant in this test>

s. Time-stamp attributes (Absolute-Time-Stamp, Base-Offset-Time, Relative-Time-Stamp and HiRes-Time-Stamp), observed values attributes (Simple-Sa-Observed-Value) and Measurement-Status shall not be present if mss-cat-setting and/or mss-cat-manual bit of the Metric-Spec-Small attribute is set to 0. If any of these bits is set to 1, observational attributes may be present in ConfigReport.

Furthermore, if MDS event reports are sent by the PHD (the PHD sends fixed format value messages to report dynamic data for RT-SA Objects or uses variable format event report, C_AG_OXP_184=TRUE or C_AG_OXP_189=TRUE):

5. IF C_AG_OXP_293 THEN:

   a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes.

   b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object.

   c. IF the mds-time-mgr-set-time bit is set:

      - ❑ The PHG moves to Configuring/Sending Set Time substate and:
        - IF C_AG_OXP_009 THEN it issues the Set-Time action command.

	• IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Take a measurement with the PHD 7. Wait for an event report from the PHD a. IF attribute Absolute-Time-Stamp is present ❑ attribute-id = MDC_ATTR_TIME_STAMP_ABS ❑ attribute-type = AbsoluteTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = <Not relevant in this test> ▪ century = ▪ year ≤ 99 ▪ month ≤ 12 ▪ day ≤ 31 ▪ hour ≤ 24 ▪ minute ≤ 60 ▪ second ≤ 60 ▪ sec-fractions ≤ 100 ❑ If an PHD stores data, it shall associate a time stamp with the data. ❑ If an PHD supports Absolute-Time-Stamp, Base-Offset–Time-Stamp attribute shall not be supported. b. IF (C_AG_ OXP_014 = TRUE) THEN Attribute Base-Offset-Time-Stamp may be present ELSE it shall not be present ❑ attribute-id = MDC_ATTR_TIME_STAMP_BO ❑ attribute-type = BaseOffsetTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = <Not relevant in this test> ❑ If a PHD stores data, it shall associate a time stamp with the data. ❑ If a PHD supports Base-Offset-Time-Stamp, Absolute-Time-Stamp attribute shall not be supported. c. IF attribute Measure-Active-Period is present ❑ attribute-id = MDC_ATTR_TIME_PD_MSMT_ACTIVE ❑ attribute-type = FLOAT-Type ❑ attribute-value.length = 4 bytes ❑ attribute-value = <Not relevant in this test> d. IF attribute Relative-Time-Stamp is present ❑ attribute-id = MDC_ATTR_TIME_STAMP_REL ❑ attribute-type = RelativeTime ❑ attribute-value.length = 4 bytes ❑ attribute-value = <Not relevant in this test> e. IF a PHD stores data, it shall associate a time stamp with the data f. IF attribute HiResRelative-Time-Stamp is present ❑ attribute-id = MDC_ATTR_TIME_STAMP_REL_HI_RES ❑ attribute-type = HighResRelativeTime ❑ attribute-value.length = 8 bytes

	❑ attribute-value = <Not relevant in this test> g. IF attribute Measurement Status is present ❑ attribute-id = MDC_ATTR_MSMT_STAT ❑ attribute-type = MeasurementStatus ❑ attribute-value.length = 2 bytes ❑ attribute-value = <not relevant for this test> h. Mandatory attribute Simple-Sa-Observed-Value ❑ attribute-id = MDC_ATTR_SIMP_SA_OBS_VAL ❑ attribute-type = OCTET STRING ❑ attribute-value.length = The length shall be even with padding bytes at the end. ❑ attribute-value = <Not relevant in this test> i. Check dynamic attributes that may be present: Metric-Structure-Small, Metric-Id, Metric-Id-List, Metric-Id-Partition, Unit-Code,Source-Handle-Reference, Label-String, Unit-Label-String, Measure-Active-Period
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-001_D		
TP label		RealTime-SA: Sa-Specification semantic		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ArrayClassAttr 4; M		
Test purpose		Check that: The Sa-Specification attribute is of type SaSpec		
Applicability		C_AG_OXP_042 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_046, C_AG_OXP_047, C_AG_OXP_180		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Take a measurement with the PHD of a value stored in a RT-SA object. 2. Wait until the PHG receives an event report, the attributes of interest are: a. Scale-and-Range-Specification: ❑ Attribute-value = See below ❑ lower-*-value = <Record for comparison> ❑ upper-*-value = <Record for comparison> b. Simple-Sa-Observed-Value ❑ Attribute-value = <Record for comparison>		
Pass/Fail criteria		• Upper values must be ≥ than the lower values • Measurement Value must be in the allowed range.		
Notes				

TP Id		TP/PLT/PHD/OXP/DIM/BV-001_E		
TP label		Numeric attributes: Metric-Id-List Semantic		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MetricClassAttr 9; C		
Test purpose		The order of the Metric-Id-List shall correspond to the order of the elements in the compound		

	observed value.
Applicability	C_AG_OXP_190 AND C_AG_OXP_000
Other PICS	C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_071, C_AG_OXP_188, C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG 5. Make the PHD send measurements for objects that support Metric-Id-List. 6. Wait for event reports. 7. The Compound Simple and Basic Numeric observed value will be taken from the Event Report received in step 4 and this information and the Metric-Id-List will be showed to the operator using a pop-up.
Pass/Fail criteria	The operator checks in step 7 that the order of the Metric-Id-List corresponds to the order of the elements in the compound observed value.
Notes	If the PHD does not send the Metric-Id-List value in the ConfigReport, the PHD will send it in an MDS-Event Report before sending an observation (compound value). SE is allowed for the Metric-Id-List.

TP Id		TP/PLT/PHD/OXP/DIM/BV-002_A		
TP label		PM-Store Object: Mandatory, Conditional and Optional Attributes 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	StoreClass 1; M	StoreClassAttr 1; M	StoreClassAttr 2; M
		StoreClassAttr 3; M	StoreClassAttr 4; O	StoreClassAttr 5; O
		StoreClassAttr 6; M	StoreClassAttr 7; O	StoreClassAttr 8; C
		StoreClassAttr 9; M	StoreClassAttr 11; M	PM-StoreService 1; M
		PersStoreMtrDatTransf 25; M		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M		
	Spec	[ISO/IEEE 11073-10472]		
	Testable items	MM_PMStoreAttr6; C		
	Spec	[IEEE 11073-10406]		
	Testable items	PerPMStoreAtt2; M	AperPMStoreAtt2; M	PMStoreServ1; M
		PMStoreServ2; M		

Test purpose	Check that: PM-Store objects contain all mandatory attributes, conditional attributes as required by their conditions and it may contain optional attributes [AND] The nomenclature code to identify the PM-Store class is MDC_MOC_VMO_PMSTORE [AND] The handle value is placed in the obj-handle field of the message and is not present in the attribute-id list of the request or the attribute-list of the response.
Applicability	C_AG_OXP_041 AND C_AG_OXP_000
Other PICS	C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_071, C_AG_OXP_187, C_AG_OXP_188, C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. Record the handle for the PM-Store object. 5. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 6. The PHD issues a GET response with the PM-Store attributes it supports: Verify the invoke-id is mirrored from the Get request. a. Mandatory attribute Handle shall not be present □ attribute-type = HANDLE □ attribute-value.length = 2 bytes □ attribute-value = <Not relevant in this test case> b. Mandatory attribute PM-Store-Capab □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value.length = 2 bytes □ attribute-value = one or more of the following bits may be set: ▪ pmsc-var-no-of-segm (bit 0) ▪ IF C_AG_OXP_187 then bit 4 (pmsc-epi-seg-entries) must be set ▪ IF C_AG_OXP_188 then bit 5 (pmsc-peri-seg-entries) must be set ▪ pmsc-abs-time-select (bit 6) ▪ pmsc-clear-segm-by-list-sup (bit 7) ▪ pmsc-clear-segm-by-time-sup (bit 8) ▪ pmsc-clear-segm-remove (bit 9) ▪ pmsc-multi-person (bit 12) ▪ All other bits shall be set to zero c. Mandatory attribute Store-Sample-Algorithm □ attribute-id = MDC_ATTR_METRIC_STORE_SAMPLE_ALG □ attribute-type = StoSampleAlg □ attribute-value.length = 2 bytes □ attribute-value = One of the next ▪ st-alg-nos(0x00 0x00) ▪ st-alg-moving-average(0x00 0x01)

	▪ st-alg-recursive(0x00 0x02) ▪ st-alg-min-pick(0x00 0x03) ▪ st-alg-max-pick(0x00 0x04) ▪ st-alg-median(0x00 0x05) ▪ st-alg-trended(0x02 0x00) ▪ st-alg-no-downsampling(0x04 0x00) ▪ st-alg-manuf-specific-start(0xF0 0x0061440) ▪ st-alg-manuf-specific-end(0xFF 0xFF) d. Mandatory attribute Operational-State □ attribute-id = MDC_ATTR_OP_STAT □ attribute-type = OperationalState □ attribute-value.length = 2 bytes □ attribute-value = One of the next ▪ disabled (0x00 0x00) ▪ enabled (0x00 0x01) ▪ notAvailable (0x00 0x02) e. Mandatory attribute Number-Of-Segments □ attribute-id = MDC_ATTR_NUM_SEG □ attribute-type = INT-U16 □ attribute-value.length = 2 bytes □ attribute-value = <not relevant for this test> f. IF attribute Clear-Timeout □ attribute-id = MDC_ATTR_CLEAR_TIMEOUT □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> □ IF the PHD supports the clear segment action, Clear-Timeout attribute is mandatory. g. IF attribute Sample-Period is present □ attribute-id = MDC_ATTR_TIME_PD_SAMP □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> h. IF Storage-Capacity-Count is present □ attribute-id = MDC_ATTR_METRIC_STORE_CAPAC_CNT □ attribute-type = INT-U32 □ attribute-value.length = 4 bytes □ attribute-value = See relation with next attribute i. IF Storage-Usage-Count is present □ attribute-id = MDC_ATTR_METRIC_STORE_USAGE_CNT □ attribute-type = INT-U32 □ attribute-value.length = 4 bytes □ attribute-value = always ≤ than Storage-Capacity-Count j. IF attribute PM-Store-Label

	□ attribute-id = MDC_ATTR_PM_STORE_LABEL_STRING □ attribute-type = OCTET STRING □ attribute-value.length = □ attribute-value = Printable ASCII 7. IF C_AG_OXP_293 THEN a. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 8. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 9. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports in the SegmentInfoList structure.
Pass/Fail criteria	• All checked values are as specified in the test procedure. • IF in step 6.f the Sample-Period was not present it must be present in each PM-Segment.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-002_B		
TP label		PM-Store Object: Mandatory, Conditional and Optional Attributes 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	StoreClassAttr 9; M	StoreClassAttr 2; M	StoreClassAttr 5; M
	Spec	[ISO/IEEE 11073-10472]		
	Testable items	MM_PMStoreAttr5; C		
Test purpose		Check that: PM-Store object includes the Number-Of-Segments attribute [AND] The Number-Of-Segments attribute is of type INT-U16 [AND] A PM-Store object may include the Store-Usage-Count attribute The Store-Usage-Count attribute shall be of type INT-U32		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Make sure there are no measurements being taken. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes. The attributes of interest are: a. Mandatory attribute PM-Store-Capab □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value.length = 2 bytes		

	❑ attribute-value = ▪ pmsc-var-no-of-segm. Record state for later comparison b. Mandatory attribute Number-Of-Segments ❑ attribute-id = MDC_ATTR_NUM_SEG ❑ attribute-type = INT-U16 ❑ attribute-value.length = 2 bytes ❑ attribute-value = <record for later comparison> 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_GET_INFO ❑ SegmSelection = all-segments 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports in the SegmentInfoList structure: a. Verify the invoke-id is mirrored from the Get request a. b. Data APDU ❑ Type = Response | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_GET_INFO ❑ SegmentInfoList = <Attributes of the segments> 6. Record the number of existing Segments. 7. If the PHD can record measurements in PM-Store while it is connected then take measurements whose values are stored in a PM-Segment. 8. Repeat steps 2 through 5.
Pass/Fail criteria	▪ In step 2.a, if bit pmsc-var-no-of-segm is not set, the number of segments stated in step 2.b and checked in step 5.b must remain unchanged. ▪ The PM-Store attribute Number-Of-Segments value must contain the exact number of segments recorded in step 6.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-002_C		
TP label		PM-Store Object: Clear-Timeout Semantics		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	StoreClassAttr 11; M		
Test purpose		Check that: The value of Clear-Timeout attribute matches with the actual timeout value that the PHD uses to wait for a response to the Clear-Segments action		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000 AND C_AG_OXP_071		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Take a measurement that would be placed in the PM-Store. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes.		

	4. Record the value of the Clear-Timeout Attribute (time in seconds = attribute-value*125/1e6). 5. The simulated PHG sends a Segment Clear to one of the PM-Segments: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR ❑ SegmSelection = all-segments 6. The PHD under test operation response: Verify the invoke-id is mirrored from the Get request. a. Data APDU ❑ Type = Response | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR
Pass/Fail criteria	The PHG has to receive the confirmation in less than the value specified in the Clear-Timeout attribute.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-002_D		
TP label		PM-Store Object: Episodic Semantics		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	StoreClassAttr 2; M		
Test purpose		Check that: If PM-Store has some or all PM-Segments than contain episodic entries then it has to contain explicit time stamp information		
Applicability		C_AG_OXP_041 AND C_AG_OXP_187 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. Make sure there are no measurements being taken. 2. The simulated PHG receives an association request from the PHD under test. 3. The simulated PHG responds with a result = accepted-unknown-config. 4. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message: a. Event-type=MDC_NOTI_CONFIG 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the		

	PHG. 6. Check that the PM-Store-Capab attribute has the pmsc-epi-seg-entries bit set. 7. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 8. The PHD shall respond to the Get-Segment-Info, indicating the attributes of the PM-Segment. 9. Check the PM-Segment-Entry-Map to make sure that a Time-Stamp is associated with the measurement data. 10. Take measurements with the PHD under test. 11. The simulated PHG sends a request for the PM-Segment Data to one of the PM-Segments that contains data (sends the Action MDC_ACT_SEG_TRIG_XFER). 12. The PHD issues an action response. 13. The PHD under test starts Data transfer: a. Data APDU ❑ Invoke | CfmEventReport ❑ Action = MDC_NOTI_SEGMENT_DATA ❑ SegmentDataEvent 14. The simulated PHG responds to transferred data APDU's.
Pass/Fail criteria	The PM-Segment-Entry-Map contains a Time-Stamp associated with measurement data and it has the correct format in the SegmentDataEvent received.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-002_E		
TP label		PM-Store Object: Mandatory, Conditional and Optional Attributes 3 Configuration		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	StoreClass 1; M	StoreClassAttr 1; M	StoreClassAttr 2; M
		StoreClassAttr 3; M	StoreClassAttr 4; O	StoreClassAttr 5; O
		StoreClassAttr 6; M	StoreClassAttr 7; O	StoreClassAttr 8; C
		StoreClassAttr 9; M	StoreClassAttr 11; M	ConfEventRep 29; M
		ConfEventRep 30; M	ConfEventRep 31; C	ConfEventRep 33; O
		PM-StoreService 3; O		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M		
Test purpose		Check that: PM-Store objects contain all mandatory attributes, conditional attributes as required by their conditions and it may contain optional attributes [AND] The nomenclature code to identify the PM-Store class is MDC_MOC_VMO_PMSTORE [AND] Static, dynamic and observational attributes. [AND] A PHD may also send scan event reports providing the PHG with updates of the current attribute values, but this is not a mandated PHD behaviour. [AND] Changes to any non-static attributes values on PM-stores or the MDS may be reported to the PHG in event reports at the discretion of the PHD		

Applicability	C_AG_OXP_041 AND C_AG_OXP_000
Other PICS	C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_071, C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The PM-Store object attributes must be(ConfigReport -> ConfigObject-> AttributeList): a. Mandatory attribute Handle shall not be present. □ attribute-type = HANDLE □ attribute-value.length = 2 bytes □ attribute-value = must be unique and non-zero. Actual value may be specified by the Device Specialization. b. Mandatory attribute PM-Store-Capab shall be present in ConfigReport: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value.length = 2 bytes □ attribute-value = <Not relevant in this test> c. Mandatory attribute Store-Sample-Algorithm shall be present in ConfigReport: □ attribute-id = MDC_ATTR_METRIC_STORE_SAMPLE_ALG □ attribute-type = StoSampleAlg □ attribute-value.length = 2 bytes □ attribute-value = <Not relevant in this test> d. IF Storage-Capacity-Count is supported, it shall be present in ConfigReport: □ attribute-id = MDC_ATTR_METRIC_STORE_CAPAC_CNT □ attribute-type = INT-U32 □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> e. IF Storage-Usage-Count is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_METRIC_STORE_USAGE_CNT □ attribute-type = INT-U32 □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> f. Mandatory attribute Operational-State should be present in ConfigReport: □ attribute-id = MDC_ATTR_OP_STAT □ attribute-type = OperationalState □ attribute-value.length = 2 bytes □ attribute-value = <Not relevant in this test> g. IF attribute PM-Store-Label is supported, it shall be present in ConfigReport: □ attribute-id = MDC_ATTR_PM_STORE_LABEL_STRING □ attribute-type = OCTET STRING □ attribute-value.length = □ attribute-value = <Not relevant in this test> h. IF attribute Sample-Period is supported, it shall be present in ConfigReport: □ attribute-id = MDC_ATTR_TIME_PD_SAMP

	□ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> i. Mandatory attribute Number-Of-Segments should be present in ConfigReport: □ attribute-id = MDC_ATTR_NUM_SEG □ attribute-type = INT-U16 □ attribute-value.length = 2 bytes □ attribute-value = <not relevant for this test> j. If attribute Clear-Timeout is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_CLEAR_TIMEOUT □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> Furthermore if MDS event reports are sent by the PHD: 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Take a measurement with the PHD. 7. Wait for a variable format event report from the PHD, check that dynamic attributes for PM-Store may be reported (Store-Usage-Count, Operational-State, Number-Of-Segments, Clear-Timeout).
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-003_A		
TP label		PM-Segment Object: Mandatory, Conditional and Optional Attributes		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-SegmAttr 1; M	PM-SegmAttr 2; M	PM-SegmAttr 3; C
		PM-SegmAttr 5; C	PM-SegmAttr 6; O	PM-SegmAttr 7; O
		PM-SegmAttr 8; O	PM-SegmAttr 9; O	PM-SegmAttr 10; O
		PM-SegmAttr 11; O	PM-StoreMeth 10; M	PM-StoreMeth 11; M
		PM-StoreMeth 1; M	PM-SegmAttr 19; C	PM-SegmAttr 20; C
	Spec	[IEEE 11073-10406]		
	Testable items	PerPMStoreAtt12; C	PMStoreObjMeth2; M	PerPMSegObj1; M
		PerPMSegObj2; M	PerPMSegObj3; C	PerPMSegObj4; M
		PerPMSegObj5; C	PerPMSegObj6; O	PerPMSegObj7; C
		PerPMSegObj8; C	PerPMSegObj9; C	PerPMSegObj10; C
		PerPMSegObj12; M	PerPMSegObj13; O	PerPMSegObj14; M
		PerPMSegObj15; O	PerPMSegObj16; M	PerPMSegObj17; C
		AperPMSegObj1; M	AperPMSegObj2; M	AperPMSegObj3; C
		AperPMSegObj4; M	AperPMSegObj5; O	AperPMSegObj6; O
		AperPMSegObj7; M	AperPMSegObj8; M	AperPMSegObj9;C
		AperPMSegObj10; C	AperPMSegObj11; C	AperPMSegObj12; M
		AperPMSegObj13; O	AperPMSegObj14; M	AperPMSegObj15; O
		AperPMSegObj16; M		
Coverage	**Spec**	[b-ITU-T H.810 (2015)]		
	Testable items	General 2; M		
Test purpose		Check that: If a PHD supports the PM-store class, the support of the Get-Segment-Info is mandatory [AND] Its PM-Segment objects contain all mandatory and conditional attributes as required by their conditions which may also contain optional attributes		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_188		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The PHD issues a GET response with the PM-Store attributes. Check for the existence of: a. attribute Sample-Period is present □ attribute-id = MDC_ATTR_TIME_PD_SAMP □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant in this test> 3. The simulated PHG shall send a Get-Segment-Info object action for the PM-Segment object with SegmSelection = all-segments to indicate the PM-Segment attributes of all		

	available PM-Segments. 4. The PHD issues a "rors-cmip-confirmed-action" response with the PM-Segment attributes it supports: Verify the invoke-id is mirrored from the Get request. a. Mandatory attribute Instance-Number □ attribute-id = MDC_ATTR_ID_INSTNO □ attribute-type = InstNumber □ attribute-length = 2 bytes □ attribute-value = unique in its PM-Store (This is why we ask for all the attributes of all the PM-Segment) b. Mandatory attribute PM-Segment-Entry-Map □ attribute-id = MDC_ATTR_PM_SEG_MAP □ attribute-type = PmSegmentEntryMap □ attribute-value = SEQUENCE, it must match the entries c. Mandatory attribute Operational-State □ attribute-id = MDC_ATTR_OP_STAT □ attribute-type = OperationalState □ attribute-value.length = 2 bytes □ attribute-value = {disabled(0), enabled(1), notAvailable(2)} d. Mandatory attribute Transfer-Timeout □ attribute-id = MDC_ATTR_TRANSFER_TIMEOUT □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes e. IF attribute PM-Seg-Person-Id is present □ attribute-id = MDC_ATTR_PM_SEG_PERSON_ID □ attribute-type = PersonId □ attribute-value.length = 2 bytes □ attribute-value = □ If the PM-Store is able to store data for multiple persons it shall set the pmsc-multi-person bit in the PM-Store-Capab attribute. If this bit is set, all PM-Segment instances contained in the PM-Store shall support the PM-Seg-Person-Id attribute, check with the attributes obtained in step 2. f. IF attribute Sample-Period is present □ attribute-id = MDC_ATTR_TIME_PD_SAMP □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = □ IF in step 2 Sample-Period was not present and values are sampled periodically this attribute must be present either in the PM-Store or alternatively in each PM-Segment. If values are sampled, then the pmsc-peri-seg-entries bit in the PM-Store-Capab attribute shall be set, check with attributes obtained in step 2 The [Sample-Period] attribute must be present in either the PM-Store or alternatively in each of the PM-Segments if values are sampled periodically - so the time difference for 2 entries in the Fixed-Segment-Data is constant (the pmsc-peri-seg-entries bit is the Pm-Store-Capab attribute is set). g. IF attribute Date-and-Time-Adjustment is present □ attribute-id = MDC_ATTR_TIME_ABS_ADJUST (0x0A 0x62) □ attribute-type = AbsoluteTimeAdjust □ attribute-value.length = 6 bytes

	❑ attribute-value = ❑ If the PHD ever adjusts the Date-and-Time, this attribute reports the time adjustment. h. IF attribute Segment-Label is present ❑ attribute-id = MDC_ATTR_PM_SEG_LABEL_STRING ❑ attribute-type = OCTET STRING ❑ attribute-value.length = consistent with value ❑ attribute-value = <printable ASCII> i. IF(C_AG_OXP_009 = TRUE) THEN attribute Segment-Start-Abs-Time may be present ELSE it shall not be present ❑ attribute-id = MDC_ATTR_TIME_START_SEG ❑ attribute-type = AbsoluteTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = ▪ century = ▪ year ≤ 99 ▪ month ≤ 12 ▪ day ≤ 31 ▪ hour ≤ 24 ▪ minute ≤ 60 ▪ second ≤ 60 ▪ sec-fractions ≤ 100 ❑ Note: This attribute is required if the PHD supports actions on the segment by time (i.e., the pmsc-abs-time-select and/or the pmsc-clear-segm-bytime- sup bits are set) this attribute shall be present. ❑ If this attribute is used, the Segment-Start-BO-Time shall not be used. j. IF (C_AG_OXP_009 = TRUE) THAN attribute Segment-End-Abs-Time may be present ELSE it hall not be present ❑ attribute-id = MDC_ATTR_TIME_END_SEG ❑ attribute-type = AbsoluteTime ❑ attribute-value.length = 8 bytes ❑ attribute-value = ▪ century = ▪ year ≤ 99 ▪ month ≤ 12 ▪ day ≤ 31 ▪ hour ≤ 24 ▪ minute ≤ 60 ▪ second ≤ 60 ▪ sec-fractions ≤ 100 ❑ Note: This attribute is required if the PHD supports actions on the segment by time (i.e., the pmsc-abs-time-select and/or the pmsc-clear-segm-bytime- sup bits are set) this attribute shall be present. ❑ If this attribute is used, the Segment-End-BO-Time shall not be used. k. IF (C_AG_ OXP_014 = TRUE) THEN attribute Segment-Start-BO-Time may be present ELSE it shall not be present ❑ attribute-id = MDC_ATTR_ TIME_START_SEG_BO

	□ attribute-type = BaseOffsetTime □ attribute-value.length = 8 bytes □ attribute-value = <Not relevant in this test> □ If this attribute is used, the Segment-Start-Abs-Time shall not be used. l. IF (C_AG_ OXP_014 = TRUE) THEN attribute Segment-End-BO-Time may be present ELSE it shall not be present □ attribute-id = MDC_ATTR_ TIME_START_SEG_BO □ attribute-type = BaseOffsetTime □ attribute-value.length = 8 bytes □ attribute-value = <Not relevant in this test> □ If this attribute is used, the Segment-End-Abs-Time shall not be used m. IF attribute Segment-Usage-Count is present □ attribute-id = MDC_ATTR_SEG_USAGE_CNT □ attribute-type = INT-U32 □ attribute-value.length = 4 bytes □ attribute-value = check that it has the number of stored entries n. IF attribute Segment-Statistics □ attribute-id = MDC_ATTR_SEG_STATS □ attribute-type = SegmentStatistics □ attribute-value.length = must be consistent with EntryMap □ attribute-value = o. IF attribute Confirm-Timeout is present □ attribute-id = MDC_ATTR_CONFIRM_TIMEOUT □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = 5. Repeat steps 3 and 4 for every Segment.
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-003_B		
TP label		PM-Segment Object: Semantic of Segment Statistic attribute		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-SegmAttr 11; O		
Test purpose		Check that: Segment-Statistics attribute values matches with the min/max/mean of the segment data content.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_074 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC_ACT_SEG_GET_INFO, with SegmSelection (all-segments). 2. The PHD responds with a "rors-cmip-confirmed-action", action-type MDC_ACT_SEG_GET_INFO, giving information about the attributes of every PM-Segment. For every segment, the Segment-Statistic attribute is recorded if it is supported		

	by the PHD. 3. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC_ACT_SEG_TRIG_XFER for a PM-Segment that supports the Segment-Statistic attribute. 4. The PHD sends a "rors-cmip-confirmed-action", action-type MDC_ACT_SEG_TRIG_XFER, with TrigSegmDataXferRsp "tsxr-successful". 5. The PHD sends a "roiv-cmip-confirmed-event-report", action-type MDC_NOTI_SEGMENT_DATA. 6. Record the segment-data-event-entries in step 5, calculate the min, max or mean for every entry and compare it with the Segment-Statistic attribute value.
Pass/Fail criteria	• The maximum of every entry recorded in step 5 is not higher than the max defined in the SegmentStatistic attribute recorded in step 2 if SegStatType is "segm-stat-type-maximum". • The minimum of every entry recorded in step 5 is not lower than the min defined in the SegmentStatistic attribute recorded in step 2 if SegStatType is "segm-stat-type-minimum". • The mean of every entry recorded in step 5 matches the mean defined in the SegmentStatistic attribute recorded in step 2 if SegStatType is "segm-stat-type-average".
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-003_C		
TP label		PM-Segment Object: Semantic of PM-Seg-Person-Id attribute		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-SegmAttr 3; O		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	General 6; C		
Test purpose		Check that: If the PM-Store is able to store data for multiple persons, it sets the pmsc-multi-person bit in the PM-Store-Capab attribute. [AND] If this bit is set, all PM-Segment instances contained in the PM-Store supports the PM-Seg-Person-Id attribute [AND] Continua service components designed to store and utilize data from multiple users simultaneously in one or more PM-Stores shall identify users and support the PM-Seg-Person-Id PM-Segment object attribute and set the pmsc-multi-person bit in the PM-Store-Capab PM-Store object attribute		
Applicability		C_AG_OXP_041 AND C_AG_OXP_035 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The PHD issues a GET response with the PM-Store attributes. 3. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments to indicate all PM-Segments attributes. 4. The PHD issues a response with the PM-Segment attributes it supports. 5. The simulated PHG sends a request for the PM-Segment Data. 6. The PHD issues an action response (action: MDC_ACT_SEG_TRIG_XFER, action-info-args: TrigSegmDataXferRsp).		

	7. The PHD under test sends a Segment-Data-Event message.
Pass/Fail criteria	The pmsc-multi-person bit in the PM-Store-Capab attribute must be set and all PM-Segment instances contained in the PM-Store must contain the PM-Seg-Person-Id attribute. In step 7, measurements stored in the PM-Stored have to be assigned correctly to every person.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-003_D		
TP label		PM-Segment Object: semantic of data-and-Time Adjustment attribute		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-SegmAttr 9; O	AbsTime 15; C	
Test purpose		Check that: If the PHD adjusts the Date-and-Time, then this attribute report the time adjustment		
Applicability		C_AG_OXP_041 AND C_AG_OXP_012 AND C_AG_OXP_000 AND C_AG_OXP_016		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Take a measurement with PHD. 2. Make a noticeable change in change in the Date or Time of the PHD. 3. Take a new measurement. 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Segments attributes. 5. The PHD issues a response with the PM-Segment attributes it supports, the attribute of interest: □ attribute-id = MDC_ATTR_TIME_ABS_ADJUST □ attribute-type = absolute-time-adjust □ attribute-length = 6 bytes □ attribute-value = <Must contain the adjustment (+-44505 years)>		
Pass/Fail criteria		The PM-Segment attribute Date-and-Time-Adjustment must inform of the change.		
Notes				

TP Id		TP/PLT/PHD/OXP/DIM/BV-003_E		
TP label		PM-Segment Object: semantic of data-and-Time Adjustment attribute. Disconnected		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-SegmAttr 9; O	AbsTime 15; C	
Test purpose		Check that: If the PHD adjusts the Date-and-Time, then this attribute report the time adjustment		
Applicability		C_AG_OXP_041 AND C_AG_OXP_012 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. Take measurements with PHD that are stored in a segment. 2. Make a noticeable change in change in the Date or Time of the PHD. 3. Take a new measurement. 4. The simulated PHG receives an association request from the PHD under test.		

	5. The simulated PHG responds with a result = accepted-unknown-config. 6. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 7. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 8. Once in the Operating state, the simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Segment attributes. 9. The PHD issues a response with the PM-Segment attributes it supports, the attribute of interest: □ attribute-id = MDC_ATTR_TIME_ABS_ADJUST □ attribute-type = absolute-time-adjust □ attribute-length = 6 bytes □ attribute-value = <Must contain the adjustment (+-44505 years)>
Pass/Fail criteria	The PM-Segment attribute Date-and-Time-Adjustment must inform of the change.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-005		
TP label		PM-Segment Object. Confirm Timeout		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-SegmAttr 13; O	OperErrorCond 5; M	OperErrorCond 6; M
		TimeOutVar 2; C		
Test purpose		Check that: If Confirm-Timeout attribute is supported, then its value matches with the actual timeout value that the PHD uses for the Confirmed Event Report generated from the PM-Store Object [AND] TOcer-pms: If the attribute is not present, the PHD shall use the value 3 s.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get-Segment-Info object action for the PM-Segment object with SegmSelection set to all-segments to indicate all PM-Segments attributes. 2. Record the Confirm-Timeout value from the Get PM-Segment operation. If the attribute is not present its value shall be 3 s(TO$_{\text{cer-pms}}$). 3. The simulated PHG sends a request for the PM-Segment Data with SegmSelection = Segment-id-list. 4. The PHD issues a response with the PM-Segments attributes (action: MDC_ACT_SEG_GET_INFO).		

	5. The simulated PHG sends a request for the PM-Segment Data. 6. The PHD issues an action response (action: MDC_ACT_SEG_TRIG_XFER, action-info-args: TrigSegmDataXferRsp). 7. The PHD under test sends a Segment-Data-Event message. 8. The simulated PHG does not respond for at least the time specified in the field Confirm-Timeout. 9. The PHD waits the Confirm-Timeout time and then it must send an abort message to the PHG and move to the Unassociated state.
Pass/Fail criteria	The PHD waits TO$_{\text{cer-pms}}$ time and then it must send an abort message to the PHG and changes to the Unassociated state.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-008		
TP label		EpiCfgScanner Object: Mandatory, Conditional and Optional Attributes		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ScanClassAttr 1; M	ScanClassAttr 2; M	ScanClassAttr 4; C
		ScanClassAttr 6; C	CfgScanAttr 1; M	CfgScanAttr 2; C
		CfgScanAttr 5; O	EpiCfgScanAttr 1; O	EpiCfgScanClass 3; M
		ConfNormalProc 1; M	ConfEventRep 29; M	ConfEventRep 30; M
		ConfEventRep 31; C	ConfEventRep 33; O	
	Spec	[IEEE 11073-10406]		
	Testable items	EpiScanObjAttr1; M	EpiScanObjAttr2; M	EpiScanObjAttr3; C
		EpiScanObjAttr4; C	EpiScanObjAttr5; M	EpiScanObjAttr6; O
		EpiScanObjAttr7; O		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M		
Test purpose		Check that: Scanner objects contain all mandatory attributes, conditional attributes as required by their conditions and it may contain optional attributes [AND] The nomenclature code to identify the Episodic Configurable Scanner class is MDC_MOC_SCAN_CFG_EPI [AND] Two consecutive event reports shall not have a time interval less than Min-Reporting-Interval [AND] Episodic Scanner Object attributes are static, dynamic or observational. [AND] Changes to any attribute values of metric and scanner objects shall be reported to the PHG in scan event reports prior to sending event reports that depend on those values (e.g. scan-handle-attr-val-map and a group format event report or unit-code and the observed value).		
Applicability		C_AG_OXP_047 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_144, C_AG_OXP_180		
Initial condition		The simulated PHG and PHD under test have been associated, but the PHD configuration is unknown for simulated PHG, so the PHD and the simulated PHG will be in the Configuring state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config.		

	3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The Configurable Episodic Scanner object (ConfigReport -> ConfigObject-> AttributeList) must have: a. Mandatory attribute Handle shall not be present □ attribute-type = HANDLE □ attribute-value = 2 bytes □ attribute-value = <Must be unique but not relevant in this test> b. Mandatory attribute Operational-State should be present un ConfigReport: □ attribute-id = MDC_ATTR_OP_STAT □ attribute-type = OperationalState □ attribute-value.length = 2 bytes □ attribute-value = 0 at start c. IF attribute Scan-Handle-List is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_SCAN_HANDLE_LIST □ attribute-type = HANDLEList □ attribute-value.length = □ attribute-value = <Not Relevant for this test> d. IF attribute Scan-Handle-Attr-Val-Map is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP □ attribute-type = HANDLEAttrValMap □ attribute-value.count = N □ attribute-value.length = <Variable> □ attribute-value = <Not Relevant for this test> e. Mandatory attribute Confirm-Mode should be present in ConfigReport: □ attribute-id = MDC_ATTR_CONFIRM_MODE □ attribute-type = ConfirmMode □ attribute-value = One of: ▪ unconfirmed (0x00 0x00) ▪ confirmed (0x00 0x01) f. Optional Confirm-Timeout should be present in ConfigReport: □ attribute-id = MDC_ATTR_CONFIRM_TIMEOUT □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant for this test> g. IF attribute Transmit-Window is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_TX_WIND □ attribute-type = INT-U16 □ attribute-value.length = 2 bytes □ attribute-value = 1 h. Optional attribute Min-Reporting-Interval should be present in ConfigReport: □ attribute-id = MDC_ATTR_SCAN_REP_PD_MIN □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes

	□ attribute-value = <defined by vendor> IF the PHD supports Min-Reporting-Interval attribute (C_AG_OXP_144) THEN at least there is a scanner object that supports this attribute, ELSE, no scanner objects support this attribute. 5. Furthermore check if Variable MDS Scan Event Reports are sent by the PHD for the Scanner object: a. Wait for a Scan Event Report from the PHD. • Attributes whose values may be reported will be the attributes defined as dynamic: Operational-State, Scan-Handle-List, Scan-Handle-Attr-Val-Map, Confirm-Mode, Confirm-Timeout, Transmit-Window and Min-Reporting-Interval. b. Set to enable the Operational-State for the Episodic Scanner object to make the Scanner object send event reports: • If the PHD sends Unbuf-Scan-Report-Fixed or Variable, Scan-Handle-List attributes shall be received previously. • If the PHD sends Unbuf-Scan-Report-Grouped, Scan-Handle-Attr-Val-Map attributes shall be received previously. c. Set to disable the Operation-State for the Episodic Scanner object and repeat step 5b for every episodic scanner object.
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-009		
TP label		PeriCfgScanner Object: Mandatory, Conditional and Optional Attributes		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ScanClassAttr 1; M	ScanClassAttr 2; M	ScanClassAttr 4; C
		ScanClassAttr 6; C	CfgScanAttr 1; M	CfgScanAttr 2; C
		CfgScanAttr 5; O	PeriCfgScanClass 2; M	PeriCfgScanAttr 1; M
		ConfNormalProc 1; M	ConfEventRep 29; M	ConfEventRep 30; M
		ConfEventRep 31; C	ConfEventRep 33; O	
	Spec	[IEEE 11073-10406]		
	Testable items	PerScanObjAttr1; M	PerScanObjAttr2; M	PerScanObjAttr3; C
		PerScanObjAttr4;	PerScanObjAttr5; M	PerScanObjAttr6; O
		PerScanObjAttr7; O	PerScanObjAttr8; M	
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 6; M		
Test purpose		Check that: Scanner objects contain all mandatory attributes, conditional attributes as required by their conditions and it may contain optional attributes [AND] The nomenclature code to identify the Periodic Configurable Scanner class is MDC_MOC_SCAN_CFG_PERI [AND] Periodic Scanner Object attributes are static, dynamic or observational. [AND] Changes to any attribute values of metric and scanner objects shall be reported to the PHG in scan event reports prior to sending event reports that depend on those values (e.g. scan-handle-attr-val-map and a group format event report or unit-code and the observed value).		
Applicability		C_AG_OXP_046 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_144, C_AG_OXP_180		

Initial condition	The simulated PHG and PHD under test have been associated, but the PHD configuration is unknown for the simulated PHG, so the PHD and the simulated PHG will be in the Configuring state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The Configurable Periodic Scanner object (ConfigReport -> ConfigObject-> AttributeList) must have: a. Mandatory attribute Handle shall not be present □ attribute-type = HANDLE □ attribute-value = 2 bytes □ attribute-value = must be unique<Not relevant in this test> b. Mandatory attribute Operational-State should be present in ConfigReport: □ attribute-id = MDC_ATTR_OP_STAT □ attribute-type = OperationalState □ attribute-value.length = 2 bytes □ attribute-value = 0 at start c. IF Attribute Scan-Handle-List is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_SCAN_HANDLE_LIST □ attribute-type = HANDLEList □ attribute-value.length = □ attribute-value = <Not relevant for this test> d. IF attribute Scan-Handle-Attr-Val-Map is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP □ attribute-type = HANDLEAttrValMap □ attribute-value.count = N □ attribute-value.length = <Variable> □ attribute-value = N metric-derived object must be specified here, verify the correct format of the object and that the handle points to the object. e. Mandatory attribute Confirm-Mode should be present in ConfigReport: □ attribute-id = MDC_ATTR_CONFIRM_MODE □ attribute-type = ConfirmMode □ attribute-value = One of: ▪ unconfirmed (0x00 0x00) ▪ confirmed (0x00 0x01) f. Optional Confirm-Timeout should be present in ConfigReport: □ attribute-id = MDC_ATTR_CONFIRM_TIMEOUT □ attribute-type = RelativeTime □ attribute-value.length = 4 bytes □ attribute-value = <Not relevant for this test>. g. IF attribute Transmit-Window is supported, it should be present in ConfigReport: □ attribute-id = MDC_ATTR_TX_WIND □ attribute-type = INT-U16 □ attribute-value.length = 2 bytes

	□ attribute-value = 1 h. Mandatory attribute Reporting-Interval should be present in ConfigReport: □ attribute-id = MDC_ATTR_SCAN_REP_PD □ attribute-type = RelativeTime □ attribute-length = 4 bytes □ attribute-value = <Not relevant for this test> 5. Furthermore check if the Variable MDS Scan Event Reports are sent by the PHD for the Scanner object: a. Wait for a Scan Event Report from the PHD. • Attributes whose values may be reported will be the attributes defined as dynamic: Operational-State, Scan-Handle-List, Scan-Handle-Attr-Val-Map, Confirm-Mode, Confirm-Timeout, Transmit-Window and Reporting-Interval. b. Set to enable Operational–State for the Periodic Scanner object to make the Scanner object send event reports: • If the PHD sends a Buf-Scan-Report-Fixed or Variable, Scan-Handle-List attributes shall be received previously. • If the PHD sends a Buf-Scan-Report-Grouped, Scan-Handle-Attr-Val-Map attributes shall be received previously. c. Set to disable Operation–State for Periodic Scanner object and repeat step 5b for every periodic scanner object.
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-010		
TP label		MDS objects methods and events. PHD data transmission		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MDSEvent 2; C	MDSEvent 3; C	MDSEvent 4; C
		MDSEvent 5; C	CommonCharac 3; M	
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	General 7; C		
Test purpose		Check that: If the PHD uses Variable Format Event Reporting and reports on a single patient, then it uses the MDS-Dynamic-Data-Update-Var Event to report dynamic data and the type of the Data APDU is ScanReportInfoVar [OR] If the PHD uses Fixed Format Event Reporting and reports on a single patient, then it uses the MDS-Dynamic-Data-Update-Fixed Event to report dynamic data and the type of the Data APDU is ScanReportInfoFixed [OR] If the PHD uses Variable Format Event Reporting and reports on multiple patients, then it uses the MDS-Dynamic-Data-Update-MP-Var Event to report dynamic data and the type of the Data APDU is ScanReportInfoMPVar [OR] If the PHD uses Fixed Format Event Reporting and reports on multiple patients, then it uses the MDS-Dynamic-Data-Update-MP-Fixed Event to report dynamic data and the type of the Data APDU is ScanReportInfoMPFixed [OR] The total size of the response does not exceed the maximum APDU size established by the specialization		

	[AND] Continua PAN service components designed to store and utilize data from multiple users simultaneously and that use agent-initiated measurement data transmission shall identify users and set the person-id field in the corresponding ScanReportPer* structure
Applicability	C_AG_OXP_000 AND (C_AG_OXP_182 OR C_AG_OXP_183 OR C_AG_OXP_184 OR C_AG_OXP_189)
Other PICS	C_AG_OXP_010, C_AG_OXP_031, C_AG_OXP_041, C_AG_OXP_053
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. Take some measurements with the PHD under test. 2. Wait until the PHD sends event reports with the data: a. APDU Type = 0xE7 0x00 b. Invoke-Id □ Length = 2 bytes □ Value = <Not relevant for this Test Case> c. CHOICE □ Length = 2 bytes □ Value = 0x01 0x00 (Unconfirmed) OR 0x01 0x01 (Confirmed) d. Obj-Handle □ Length = 2 bytes □ Value = 0 (MDS object) e. Event-Time □ Length = 4 bytes □ Value = <0xFF 0xFF 0xFF 0xFF> If NOT C_AG_OXP_010 f. IF the data is from one person and uses a variable format event reporting, it must be: □ Event-type = MDC_NOTI_SCAN_REPORT_VAR □ Event-info parameter = ScanReportInfoVar g. IF the data is from one person and uses a fixed format event reporting, it must be: □ Event-type = MDC_NOTI_SCAN_REPORT_FIXED □ Event-info parameter = ScanReportInfoFixed h. IF the data is from multiple persons and uses a variable format event reporting, it must be: □ Event-type = MDC_NOTI_SCAN_REPORT_MP_VAR □ Event-info parameter = ScanReportInfoMPVar i. IF the data is from multiple persons and uses a fixed format event reporting, it must be: □ Event-type = MDC_NOTI_SCAN_REPORT_MP_FIXED □ Event-info parameter = ScanReportInfoMPFixed
Pass/Fail criteria	• All checked values are as specified in the test procedure. • The total size of the event report cannot exceed the maximum APDU size established by the specialization. • If the PHD does not support confirmed event reports (C_AG_OXP_053= FALSE), the PHD cannot send confirmed event reports. • If the PHD does not use variable event report (C_AG_OXP_189= FALSE), the PHD cannot send variable event reports. • If the PHD supports multi-person event reports for one or more metric object (ScanReportPer*) (C_AG_OXP_031= TRUE), the PHD has to send multi-person event reports.

	• If C_AG_OXP_031= TRUE and MP event reports have been received, a pop-up will show the received measurements to make the operator identify if measurements have been correctly assigned to every person.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-011		
TP label		MDS objects methods. PHD real-time clock (RTC). Absolute-Time		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MDSMethod 4; M	AbsTime 1;C	AbsTime 2;C
		MDSMethod 6; M	BaseTimOffset2 ; M	
Test purpose		Check that: The PHD replies to a Set-Time method with a rors-cmip-confirmed-action response. If the PHD supports Set-Time, it shall respond with a rors-cmip-confirmed-action, but the action-info-args is empty in this response. [AND] The PHD when responding to a Set-Base-Offset-Time method shall do so using a rors-cmip-confirmed-action response. The PHD indicates whether the Set-Base-Offset-Time command is valid by using the mds-time-capab-set-clock bit in the Mds-Time-Info attribute. [AND] If the PHD supports Set-Base-Offset-Time, it shall respond with a rors-cmip-confirmed-action, but the action-info-args is empty in this response. [AND] If the base time seconds field and base time fraction of a second field are set to 0x0 in the arguments of the Set-Base-Offset-Time action (these values being undefined in NTP), then only the offset to local time shall be set. If the base time (seconds field) is aligned with UTC (with an accuracy appropriate to the application), then this shall be designated by setting the mds-time-bo-time-utc-aligned bit in the Mds-Time-Info attribute. [AND] The base time should be set with respect to some reference time, and shall be set so that the offset to any local time can be accommodated by the maximum value of the offset field		
Applicability		(C_AG_OXP_007 OR C_AG_OXP_008) AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends a Get request for the MDS object with an attribute-id-list set to 0 read all the attributes. 2. Check the Value of the MdsTimeCapab bits in the MDS-Time-Info-Attribute □ IF C_AG_OXP_009 = TRUE THEN check that mds-time-capab-real-time-clock(0) is set to TRUE indicating support of an RTC, ELSE this bit is set to FALSE. □ IF C_AG_OXP_008 = TRUE THEN check that mds-time-capab-set-clock(1) is set to TRUE indicating support of the Set Time Action ELSE this bit is set to FALSE. □ IF C_AG_OXP_014 = TRUE THEN check that mds-time-capab-bo-time(7) is set to TRUE indicating support of Base-Offset-Time and record the value of Base-Offset-Time attribute (MDC_ATTR_TIME_BO) ELSE this bit is set to FALSE indicating PHD does not support of a Base-Offset-Time. 3. IF Set Time Action is supported and C_AG_OXP_009 = TRUE: a. The simulated PHG sends a SET action: □ CHOICE = SetTimeInvoke □ action-type = MDC_ACT_SET_TIME □ the action-info-args are SetTimeInvoke • date-time = century ≤ 99, year ≤ 99, month ≤ 12, day ≤ 31, hour ≤ 24, minute ≤ 60, second ≤ 60, sec-fractions ≤ 100		

	- accuracy = 0 b. The PHD under test response must be a rors-cmip-confirmed-action but the action-info-args shall be empty in this response. c. The simulated PHG sends a Get request for the MDS object with an attribute-id-list set to 0 read all the attributes. The Date-and-Time attribute value matches with the Absolute Time set in step 3.a. 4. IF Set Time Action is supported and C_AG_OXP_014 = TRUE: a. The simulated PHG sends a SET action: □ CHOICE = SetBOTimeInvoke □ action-type = MDC_ACT_SET_BO_TIME □ the action-info-args are SetBOTimeInvoke - date-time = bo-seconds = 0x00 0x00 0x00 0x00, bo-fractions = 0x00 0x00, bo-time-offset =<original bo-time-offset> + 60 b. The PHD under test response must be a rors-cmip-confirmed-action but the action-info-args shall be empty in this response. c. The simulated PHG sends a Get request for the MDS object with an attribute-id-list set to 0 read all the attributes. The Base-Offset-Time attribute value matches with Base-Offset-Time set in step 4.a.
Pass/Fail criteria	- All checked values are as specified in the test procedure and check that the time has been set correctly in step 3.c or 4.c if that action was possible. - If Set Time Action and Base-Offset-Time is supported, then the value of bo-seconds and bo-fractions in step 2 and step 4.c shall be the same and only bo-time-offset is set.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-012		
TP label		MDS object events. PHD configuration event		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MDSEvent 1; M	MetricClassAttr 1; M	StoreClassAttr 1; M
		ScanClassAttr 1; M		
	Spec	[ISO/IEEE 11073-10419]		
	Testable items	SchStoreObjIP 1; M		
Test purpose		Check that: MDS object sends the MDS-Configuration-Event with an Event-Info parameter of type ConfigReport. Only confirmed mode. [AND] Each object shall have a unique identifier assigned by the PHD		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_010, C_AG_OXP_040, C_AG_OXP_041, C_AG_OXP_042, C_AG_OXP_043, C_AG_OXP_046, C_AG_OXP_047		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" (roiv-cmip-confirmed-event-report) message: a. APDU Type □ field-length = 2 bytes □ field-value = 0xE7 0x00		

	❑ This value is for association request "prst" (PrstApdu). b. invoke-id ❑ field- type = InvokeIDType ❑ field-length = 2 bytes ❑ field- value = This value identifies the message; the confirmed response that will be sent by the simulated PHG shall have the same invoke-id. c. obj-handle (EventReportArgumentSimple) ❑ field- type = HANDLE ❑ field-length = 2 bytes ❑ field- value = 0x00 0x00 ❑ This obj-handle represents MDS-Object. d. event-time (EventReportArgumentSimple) ❑ field- type = Relative Time ❑ field-length = 4 bytes ❑ If the PHD does not support relative time clock: ❑ field-value = ▪ IF NOT C_AG_OXP_010 THEN = 0xFF 0xFF 0xFF 0xFF e. event-type (EventReportArgumentSimple) ❑ field- type = OID-Type ❑ field-length =2 bytes ❑ field- value = 0x0D 0x1C (MDC_NOTI_CONFIG) f. config-report-id (ConfigReport) ❑ field- type = ConfigId ❑ field-length = 2 bytes ❑ field- value = <between 0x00 0x01 and 0x7F 0xFF> g. obj-class (ConfigReport → ConfigObjectList (ConfigObject)) ❑ field- type = OID-Type ❑ field-length = 2 bytes ▪ field- value = 0x00 0x06 (MDC_MOC_VMO_METRIC_NU) or 0x00 0x09 (MDC_MOC_VMO_METRIC_SA-RT) or 0x00 0x05 (MDC_MOC_VMO_METRIC_ENUM) or 0x00 0x12 (MDC_MOC_SCAN_CFG_EPI) 0x00 0x13 (MDC_MOC_SCAN_CFG_PERI) or 0x00 0x3D (MDC_MOC_VMO_PMSTORE) or 0x00 0x51 (MDC_MOC_VMO_SCHEDSTORE) or a value between 0xF000 and 0xFBFF ▪ IF the PHD supports at least one numeric object (C_AG_OXP_040=TRUE) then MDC_MOC_VMO_METRIC_NU shall be present, ELSE no numeric object is present. ▪ IF the PHD supports at least one PM-Store object (C_AG_OXP_041=TRUE) then MDC_MOC_VMO_PMSTORE shall be present, ELSE no PM-Store object is present. ▪ IF the PHD supports at least one RT-SA object (C_AG_OXP_042=TRUE) then MDC_MOC_VMO_METRIC_SA-RT shall be present, ELSE no RT-SA object is present. ▪ IF the PHD supports at least one enumerated object (C_AG_OXP_043=TRUE) then MDC_MOC_VMO_METRIC_ENUM shall be present, ELSE no enumerated object is present. ▪ IF the PHD supports at least one periodic scanner object (C_AG_OXP_046=TRUE) then MDC_MOC_SCAN_CFG_PERI shall be present, ELSE no periodic scanner object is present.

	▪ IF the PHD supports at least one episodic scanner object (C_AG_OXP_047=TRUE) then MDC_MOC_SCAN_CFG_EPI shall be present, ELSE no episodic scanner object is present. ▪ IF PHD supports at least one schedule store object (C_AG_IP_012=TRUE OR C_AG_IP_013=TRUE OR C_AG_IP_014=TRUE) then MDC_MOC_VMO_SCHEDSTORE shall be present, ELSE no episodic scanner object is present. h. obj-handle (ConfigReport → ConfigObjectList (ConfigObject)) ❑ field- type = HANDLE ❑ field-length = 2 bytes ❑ field-value = <Check that each object have a unique identifier and non -zero> i. attribute-id (ConfigReport → ConfigObjectList (ConfigObject)→Attribute List) ❑ field- type = OID-Type ❑ field-length = 2 bytes ❑ field-value = <between 0x0913 (2323) and 0x0A77 (2679)> or <between 0xF000(61440) and 0xFBFF(64511)>
Pass/Fail criteria	• All checked values are as specified in the test procedure. • The total size of the response cannot exceed the sum of the APDU sizes of the supported specializations (limited to an absolute limit of 64512 octets): ○ Pulse oximeter → 9216 octets ○ Weighing scales → 896 octets ○ Glucose meter → 5210 octets or 64512 octets if the PHD supports PM-Store ○ Blood pressure → 896 octets ○ Thermometer → 896 octets ○ Independent activity hub → 5120 octets ○ Cardiovascular → 64512 octets or 6624 octets if the PHD supports Step Counter Profile ○ Strength → 64512 octets ○ Adherence monitor → 1024 octets ○ Peak flow → 2030 octets ○ Body composition analyser → 7730 octets ○ Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store ○ Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store ○ International normalized ratio → 896 octets or 64512 if the PHD supports PM-Store ○ Insulin pump → 7168 octets or 5120 if PHD supports PM-Store ○ Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store. ○ Power status monitor → 940 octets for Simple PSM profile or 1660 for Advanced PSM profile
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-013		
TP label		PM-Store object methods. Clear-Segments method		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable	PM-StoreMeth 1; O	PM-StoreMeth 2; C	

	items	PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 6; M
		PM-StoreMeth 20; C	PM-StoreMeth 21; C	PM-StoreMeth 7; M
		PM-StoreMeth 30; O	PM-StoreMeth 31; M	
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	Communication 1; M		
Test purpose		Check that: The PHD may support PM-segment clearing. If the PHD supports this function (indicated by the pmsc-clear-segm-all-sup, pmsc-clear-segm-by-list-sup, and pmsc-clear-segm-by-time-sup flags in the PM-Store-Capab attribute being set), then it shall support clearing all segments (pmsc-clear-segm-all-sup) [AND] PHD supports the Clear-Segment (all segments) method and it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] If any of the selected segments are cleared, success (rors) shall be reported. However, success does not necessarily mean that all targeted segments were actually cleared (and potentially removed) since there maybe a subset that were protected or enabled. [AND] Otherwise, the return code shall be MDC_RET_CODE_UNKNOWN which indicates that only PHD protected segments where encountered during the operation		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.		
Test procedure		1. Make sure the PHD under test is not taking measurements which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD under test issues a GET response with the PM-Store attributes. Record the values of the PM-Store-Capab attribute. a. PM-Store-Capab: ❑ attribute-id = MDC_ATTR_PM_STORE_CAPAB ❑ attribute-type = PmStoreCapab ❑ attribute-value = Record the value of bit 10 (Indicates that PM-Segments in the SegmSelection data type can be cleared by segment selection –all segments). This bit shall be set to 1. 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports. 6. The simulated PHG sends a Clear-Segment: a. Data APDU		

	□ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR □ SegmSelection = all-segments 7. If the PHD does not protect all segments, the PHD under test operation response will be: a. Data APDU □ Type = Response | Confirmed Action □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR □ Check the invoke-id of the response is mirrored from the request. 8. If the PHD does protect all segments, the PHD under test operation response will be: a. Data APDU □ Type = Roer □ ErrorResult = no-allowed-by-object (24) and return code shall be MDC_RET_CODE_UNKNOWN. □ Check the invoke-id of the response is mirrored from the request 9. Delay 10. If the PHD has sent the confirmation in step 7, the simulated PHG sends a request for the PM-Segment Data to obtain all the segments: a. Data APDU □ Type = Invoke | Confirmed Action □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ SegmSelection = <Instance number of the selected PM-Segment that contained data before the clear-segment action> 11. The PHD under test issues an action response with the Data: a. Data APDU □ Type = Response | Confirmed Action □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ TrigSegmXferRsp = • IF pmsc-clear-segm-remove of the PM-Store-Capab attribute is NOT set then □ TrigSegmXferRsp = tsxr-fail-segm-empty • ELSE then □ TrigSegmXferRsp = tsxr-fail-no-such-segment
Pass/Fail criteria	• In step 7, the PHD must send a confirmation if the PHD does not protect any segments, otherwise the PHD shall send a roer message (step 8). • If the PHD sends the confirmation in step 7, the PHD shall send the response specified in step 11 at least for a segment. • After APDU received by the simulated PHG in step 11, the PHD does not send any message of type "Segment-data-event" with data stored.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_A		
TP label		PM-Store object methods. Clear-Segments List method		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 22; C
		PersStoreMtrDatTransf 24; M		
Test purpose		Check that: If the PHD supports the Clear-Segment (list of segments) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] The PHD may support PM-segment clearing. If the PHD supports this function(indicated by the pmsc-clear-segm-all-sup, pmsc-clear-segm-by-list-sup, and pmsc-clear-segm-by-time-sup flags in the PM-Store-Capab attribute being set) then it may support clearing a particular list of segments (pmsc-clear-segm-by-list-sup) [AND] If the PHD supports the segm-id-list choice in the SegmSelection action-info-args of the Clear-Segments method, the PHD shall set the pmsc-clear-segm-by-list-sup flag in the PM-Store-Capab attribute. [AND] If the PHG invokes the Clear-Segments method but the PHD does not support the particular action (list of segments or range of segments), then the PHD shall respond with a roer DataApdu with an RoerErrorValue of "not-allowed-by-object".		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least two PM-Segments with data stored.		
Test procedure		1. Make sure the PHD is not taking measures which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes, check the values of the PM-Store-Capab attribute: a. PM-Store-Capab: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value = Record the value of bit 7 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining a list) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports. IF bit 7 of PmStoreCapab was set:		

	6. The simulated PHG sends a Clear-Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR □ SegmSelection = segm-id-list (list of integers containing 2 of the instance numbers obtained in step 5) 7. The PHD under test operation response: a. Data APDU □ Type = Response | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR 8. Delay. 9. The simulated PHG sends a request for the PM-Segment Data of one of the cleared PM-Segments: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ SegmSelection = <Instance number of the selected PM-Segment that contained data before the clear-segment action> 10. The PHD issues an action response with the Data a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ TrigSegmXferRsp = ▪ IF pmsc-clear-segm-remove is NOT set then o TrigSegmXferRsp = tsxr-fail-segm-empty ▪ ELSE then o TrigSegmXferRsp = tsxr-fail-no-such-segment IF bit 7 of PMStoreCapab was NOT set 11. The simulated PHG sends a Clear-Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR □ SegmSelection = segm-id-list (list of integers containing 2 of the instance numbers obtained in step 5) 12. The PHD under test operation response: a. Data APDU □ Type = Roer □ ErrorResult = not-allowed-by-object (24)
Pass/Fail criteria	• In step 7, the PHD must send a confirmation • The last APDU received by the simulated PHG has no data

Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_B		
TP label		PM-Store object methods. Clear-Segments Time Range method 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 19; M
Test purpose		Check that: If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] For PM-Segments cleared using the by time method, only PM-Segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_072 AND C_AG_OXP_009 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD supports at least one PM-Segment with data stored.		
Test procedure		1. Make sure the PHD under test is not taking measurements which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD under test issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining an AbsTimeRange) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-Abs-Time" and "Segment-End-Abs-Time" of every PM-Segment. IF bit 8 of PMStoreCapab was set: 6. The simulated PHG sends a Clear-Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR		

	□ SegmSelection = abs-time-range, selecting a range with its boundaries set to an earlier date of any of the existing segments. 7. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) ELSE a. Data APDU □ Type = Response | Confirmed Action □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR
Pass/Fail criteria	In step 7 the PHD must send the specified error.
Notes	Error code was not clearly defined in the spec. In the new edition of [ISO/IEEE 11073-20601-2015A], the clear-segment using time range has been clarified. "For PM-segments cleared using the by time method, only PM-segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.", but if the PHG sends a Clear-Segment but the segment has not a Segment-Start-Abs-Time and Segment-End-Abs-Time within the specified time-period, the PHD will send a Roer message. At this point, it would be up to the PHD what error code (Roer message) to send (No-such-action, not-allowed-by-object, or both). If one wants to clear the segment due to all the internal timestamps that were saved in the segment as falling inside the given time period, then the PHD could do that. Returning an error is also possible.

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_C		
TP label		PM-Store object methods. Clear-Segments Time Range method 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 19; M
Test purpose		Check that: If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] For PM-Segments cleared using the by time method, only PM-Segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_072 AND C_AG_OXP_009 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.		

Test procedure	1. Make sure the PHD is not taking measures which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab: ❑ attribute-id = MDC_ATTR_PM_STORE_CAPAB ❑ attribute-type = PmStoreCapab ❑ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining an AbsTimeRange) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-Abs-Time" and "Segment-End-Abs-Time" of every PM-Segment. IF bit 8 oft PMStoreCapab was set: 6. The simulated PHG sends a Clear-Segment: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR ❑ SegmSelection = abs-time-range, selecting a range with its boundaries set to a later date of any of the existing segments. 7. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU ❑ Type = Roer ❑ ErrorResult = no-such-action (9) ELSE a. Data APDU ❑ Type = Response | Confirmed Action ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR
Pass/Fail criteria	In step 7 the PHD must send the specified error.
Notes	See Note for test case TP/PLT/PHD/OXP/DIM/BV-013_B.

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_D		
TP label		PM-Store object methods. Clear-Segments Time Range method 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 19; M
Test purpose		Check that: If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action		

	[AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] For PM-Segments cleared using the by time method, only PM-Segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.
Applicability	C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_072 AND C_AG_OXP_009 AND C_AG_OXP_000
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.
Test procedure	1. Make sure the PHD is not taking measures which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining an AbsTimeRange) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-Abs-Time" and "Segment-End-Abs-Time" of every PM-Segment. IF bit 8 oft PMStoreCapab was set: 6. The simulated PHG sends a Clear-Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR □ SegmSelection = abs-time-range, selecting a range with one of its boundaries set to an earlier date of any of the existing segments and the other set to date contained between Segment-Start-Abs-Time and Segment-End-Abs-Time of one of the PM-Segments 7. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) Else b. Data APDU □ Type = Response | Confirmed Action c. HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR

Pass/Fail criteria	In step 7 the PHD must send the specified error.
Notes	See Note for test case TP/PLT/PHD/OXP/DIM/BV-013_B.

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_E		
TP label		PM-Store object methods. Clear-Segments Time Range method 4		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 19; M
Test purpose		Check that: If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] For PM-Segments cleared using the by time method, only PM-Segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_072 AND C_AG_OXP_009 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.		
Test procedure		1. Make sure the PHD is not taking measures which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining an AbsTimeRange) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-Abs-Time" and "Segment-End-Abs-Time" of every PM-Segment. IF bit 8 of PMStoreCapab was set: 6. The simulated PHG sends a Clear-Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR		

	□ SegmSelection = abs-time-range, selecting a range with one of its boundaries set between Segment-Start-Abs-Time and Segment-End-Abs-Time of one of the PM-Segments and the other set to a later date of any of the existing segments 7. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) ELSE b. Data APDU □ Type = Response | Confirmed Action c. HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR
Pass/Fail criteria	In step 7 the PHD must send the specified error.
Notes	See Note for test case TP/PLT/PHD/OXP/DIM/BV-013_B.

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_F		
TP label		PM-Store object methods. Clear-Segments Time Range method 5		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 19; M
Test purpose		Check that: If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] For PM-Segments cleared using the by time method, only PM-Segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_072 AND C_AG_OXP_009 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.		
Test procedure		1. Make sure the PHD is not taking measures which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab:		

	❑ attribute-id = MDC_ATTR_PM_STORE_CAPAB ❑ attribute-type = PmStoreCapab ❑ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining an AbsTimeRange) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-Abs-Time" and "Segment-End-Abs-Time" of every PM-Segment. IF bit 8 oft PMStoreCapab was set: 6. The simulated PHG sends a Clear-Segment: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR ❑ SegmSelection = abs-time-range, selecting a range with its boundaries set to Segment-Start-Abs-Time and Segment-End-Abs-Time of one of the PM-Segments 7. The PHD under test operation response: a. Data APDU ❑ Type = Response | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR 8. Delay. 9. The simulated PHG sends a request for the PM-Segment Data of one of the cleared PM-Segments: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ SegmSelection = <Instance number of the selected PM-Segment that contained data before the clear-segment action in step 6> 10. The PHD issues an action response with the Data: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ TrigSegmXferRsp = ▪ IF pmsc-clear-segm-remove is NOT set then o TrigSegmXferRsp = tsxr-fail-segm-empty ▪ ELSE then o TrigSegmXferRsp = tsxr-fail-no-such-segment
Pass/Fail criteria	• In step 7 the PHD must send a confirmation • In step 10 the TrigSemgXferRsp must be the specified
Notes	See Note for test case TP/PLT/PHD/OXP/DIM/BV-013_B.

TP Id		TP/PLT/PHD/OXP/DIM/BV-013_G		
TP label		PM-Store object methods. Clear-Segments Time Range method 6		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 19; M
		PM-StoreMeth 23; C	PersStoreMtrDatTransf 24; M	
Test purpose		Check that: If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] The PHD may support PM-segment clearing. If the PHD supports this function(indicated by the pmsc-clear-segm-all-sup, pmsc-clear-segm-by-list-sup, and pmsc-clear-segm-by-time-sup flags in the PM-Store-Capab attribute being set) then it may support the time range selection criteria (pmsc-clear-segm-by-time-sup) [AND] For PM-Segments cleared using the by time method, only PM-Segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared. [AND] If the PHD supports the abs-time-range choice in the SegmSelection action-info-args of the Clear-Segments method, the PHD shall set the pmsc-clear-segm-by-time-sup flag in the PM-Store-Capab attribute. [AND] If the PHG invokes the Clear-Segments method but the PHD does not support the particular action (list of segments or range of segments), then the PHD shall respond with a roer DataApdu with a RoerErrorValue of “not-allowed-by-object”.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_009 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.		
Test procedure		1. Make sure the PHD is not taking measures which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining an AbsTimeRange) IF bit 8 oft PMStoreCapab was set:		

	4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-Abs-Time" and "Segment-End-Abs-Time" of every PM-Segment. 6. The simulated PHG sends a Clear-Segment: a. Data APDU ☐ Type = Invoke | Confirmed Action ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_CLEAR ☐ SegmSelection = abs-time-range, selecting a range with its boundaries set to include inside from Segment-Start-Abs-Time to Segment-End-Abs-Time of one of the PM-Segments 7. The PHD under test operation response: a. Data APDU ☐ Type = Response | Confirmed Action ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_CLEAR 8. Delay. 9. The simulated PHG sends a request for the PM-Segment Data of one of the cleared PM-Segments: a. Data APDU ☐ Type = Invoke | Confirmed Action ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_TRIG_XFER ☐ SegmSelection = <Instance number of the selected PM-Segment that contained data before the clear-segment action in step 6> 10. The PHD issues an action response with the Data a. Data APDU ☐ Type = Invoke | Confirmed Action ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_TRIG_XFER ☐ TrigSegmXferRsp = ▪ IF pmsc-clear-segm-remove is NOT set then o TrigSegmXferRsp = tsxr-fail-segm-empty ▪ ELSE then o TrigSegmXferRsp = tsxr-fail-no-such-segment IF bit 8 of PMStoreCapab was NOT set 11. Simulated PHG sends a Clear-Segment: a. Data APDU ☐ Type = Invoke | Confirmed Action, ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_CLEAR ☐ SegmSelection = abs-time-range, selecting a range with its boundaries set to the absolute minimum of Absolute-Time type and to the absolute maximum of the Absolute-Time type 12. The PHD under test operation response:

	IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) Else b. Data APDU □ Type = Response | Confirmed Action c. HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR
Pass/Fail criteria	• In step 7 the PHD must send a confirmation • In step 10 the TrigSemgXferRsp must be the specified • If the PHD does not support Clear-Segment by time, the PHD must send a roer (not-allowed-by -object) message.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-014		
TP label		PM-Store object methods. Clear-Segments method 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 7; M	StoreClassAttr 6; M	PM-StoreMeth 29; M
Test purpose		Check that: If all of the selected segments fail to clear (reason being protected or in Enabled state, the PHD shall reply with a not-allowed-by-object error (roer).The return code shall be set to MDC_RET_CODE_OBJ_BUSY if any of the segments failed due to being in Enabled state. [AND] Deletion of all selected PM-segments is not guaranteed by this method. While a PM-segment has the Operational-State attribute set to enabled it will not perform the requested deletion.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000 AND C_AG_OXP_071 AND C_AG_OXP_018		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Take measurements with the PHD of a value that is stored on a PM-Segment. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes. 4. The simulated PHG shall send a Get-Segment-Info object action with segmSelection set to all-segments to check what Segments are in use. 5. The simulated PHG sends a Clear-Segment to all segments: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_CLEAR □ SegmSelection = all-segments 6. The PHD under test operation response: a. Data APDU □ Type = roer □ value = not-allowed-by-object		

	☐ value-returncode = MDC_RET_CODE_OBJ_BUSY
Pass/Fail criteria	The PHD must respond with the specified error.
Notes	The purpose of this test is to check that it is not possible to clear a segment that is in use, i.e. operational-state set to 1 by the PHD.

TP Id		TP/PLT/PHD/OXP/DIM/BV-015		
TP label		PM-Store Object. Get-Segment-Id-List method		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 35		
Test purpose		Check that: The PHD may support the Get-Segment-Id-List method [AND] Values in the PM-Store-Capab attribute represent that support [AND] PHD response is as expected		
Applicability		C_AG_OXP_000 AND C_AG_OXP_041 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The PHD under test issues a GET response with the PM-Store attributes it supports, check the values of the PM-Store-Capab attribute: a. Data APDU ☐ attribute-id = MDC_ATTR_PM_STORE_CAPAB ☐ attribute-type = PMStoreCapab ☐ attribute-value = one or more of the following bits may be set: ▪ pmsc-var-no-of-segm (0) ▪ pmsc-segm-id-list-select(3) ▪ pmsc-epi-seg-entries(4) ▪ pmsc-peri-seg-entries(5) ▪ pmsc-abs-time-select(6) ▪ pmsc-clear-segm-by-list-sup(7) ▪ pmsc-clear-segm-by-time-sup(8) ▪ pmsc-clear-segm-remove(9) ▪ pmsc-clear-segm-all-sup(10) ▪ pmsc -multi-person(12) ▪ pmsc-get-segm-id-list-sup(14) (record for later use) IF pmsc-get-segm-id-list-sup(14) is NOT set 3. The simulated PHG sends a request for the PM-Store to retrieve a list of the instance numbers of all the PM-segments it contains a. Data APDU ☐ Type = Invoke | Confirmed Action, ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_GET_ID_LIST		

	□ <empty> 4. The PHD under test issues a response: a. Data APDU □ Type = Roer □ ErrorResult = not-allowed-by-object (24) IF pmsc -get-segm-id-list-sup(14) is set 5. The simulated PHG sends a request for the PM-Store to retrieve a list of the instance numbers of all the schedule segments it contains a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_ID_LIST □ <empty> 6. The PHD under test issues a response with the PM-Segments instance numbers a. Data APDU □ Type = Response | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_ID_LIST □ SegmentIdList
Pass/Fail criteria	The PHD properly sends the required list of PM-segment ids in supported cases or the specified error otherwise.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-016		
TP label		PM-Store object methods. Trig-Segment-Data-Xfer method 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 13; M	PM-StoreMeth 14; M	PM-StoreEvent 1; M
		PM-StoreEvent 2; M	CommonCharac 3; M	PM-StoreMeth 1; M
	Spec	[IEEE 11073-10406]		
	Testable items	PMStoreObjMeth3; M	PMStoreObjEvent1; M	PMStoreObjEvent2; M
Test purpose		Check that: The PHD supports the Trig-Segment-Data-Xfer method [AND] If PHD receives the Trig-Segment-Data-Xfer request method, then it responds with an operation type of rors-cmip-confirmed-action [AND] If PHD receives the Trig-Segment-Data-Xfer request method, then it responds with an action-info-args type TrigSegmDataXferRsp [AND] Once the data transfer is triggered via a Trig-Segment-Data-Xfer method, the PHD sends Segment-Data-Event messages until the complete Fixed-Segment-Data is transferred or the transfer is aborted by the PHG or PHD [AND] When sending a Segment-Data-Event event, the event type is MDC_NOTI_SEGMENT_DATA [AND]		

	When sending a [Segment-Data-Event] event the event-info parameter is SegmentDataEvent. [AND] The total size of the response does not exceed the maximum APDU size established by the specialization [AND] If an PHD supports the PM-store class, the support of the Get-Segment-Info and Trig-Segment-Data-Xfer methods is mandatory
Applicability	C_AG_OXP_041 AND C_AG_OXP_000
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with more data loaded that the maximum allowed by specialization.
Test procedure	1. The simulated PHG issues a GET for the PM-Store object. 2. The PHD under test responds with the attributes of the PM-Store. 3. The simulated PHG issues a Get-Segment-Info with SegmSelection set to all-segments. 4. The simulated PHG sends a request for the PM-Segment Data to one of the PM-Segments that contains data: a. Data APDU ☐ Type = Invoke | Confirmed Action, ☐ HANDLE = obj-handle ☐ Action = MDC_ACT_SEG_TRIG_XFER ☐ TrigSegmDataXferReq = <Instance number of the selected PM-Segment that contains the data> 5. The PHD issues an action response: a. Data APDU ☐ Type = Invoke | Confirmed Action, ☐ HANDLE = obj-handle b. Action = MDC_ACT_SEG_TRIG_XFER ☐ TrigSegmDataXferRsp = <Same Instance number> | tsxr-successful (0x00 0x00) 6. The PHD under test starts Data transfer: a. Data APDU ☐ Invoke | CfmEventReport ☐ Action = MDC_NOTI_SEGMENT_DATA ☐ SegmentDataEvent 7. The simulated PHG response to transferred data APDU's: a. Data APDU ☐ Type = Invoke | Confirmed Action ☐ HANDLE = obj-handle ☐ Action = MDC_NOTI_SEGMENT_DATA ☐ SegmentDataResult 8. The PHD under test repeats steps 6 and 7 until all the data is transferred.
Pass/Fail criteria	• All checked values are as specified in the test procedure • Data is transferred • The total size of the response cannot exceed the sum of the APDU sizes of the supported specializations (limited to an absolute limit of 64512 octets): ○ Pulse oximeter → 9216 octets

	o Weighing scales → 896 octets o Glucose meter → 5120 octets or 64512 octets if the PHD supports PM-Store o Blood pressure → 896 octets o Thermometer → 896 octets o Independent activity hub → 5120 octets o Cardiovascular → 64512 octets or 6624 octets if it supports Step Counter Profile o Strength → 64512 octets o Adherence monitor → 1024 octets o Peak flow → 2030 octets o Body composition analyser → 7730 octets o Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store o Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store o International normalized ratio → 896 octets or 64512 octets if the PHD supports PM-Store o Insulin pump → 7168 octets or 5120 if PHD supports PM-Store o Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store o Power status monitor → 940 octets for Simple PSM profile or 1660 for Advanced PSM profile
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-017		
TP label		PM-Store object methods. Trig-Segment-Data-Xfer method 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 16; M	PM-SegmAttr 4; M	
Test purpose		Check that: PM-Segment object includes the Operational-State attribute and The [Operational-State] attribute shall be of type [OperationalState] If PM-Segment is having data activity added to it, then Operational-State attribute is set to 'enabled', otherwise, it is set to 'disabled'. [AND] If Trig-Segment-Data-Xfer method is invoked on a PM-Segment that has an Operational-State of "enabled", then the PHD shall replies with a not-allowed-by-object error (roer) with a return code of MDC_RET_CODE_OBJ_BUSY		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000 AND C_AG_OXP_018		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment.		
Test procedure		1. The simulated PHG issues a GET for the PM-Store object. 2. The PHD under test responds with the attributes of the PM-Store. 3. The simulated PHG issues a Get-Segment-Info with SegmSelection set to all-segments. 4. The simulated PHG sends a request for the PM-Segment Data to one of the PM-Segments that is being used (OperationalState bit enabled): a. Data APDU ❑ Type = Invoke | Confirmed Action,		

	❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ TrigSegmDataXferReq = <Instance number of the selected PM-Segment that contains the data> 5. The PHD issues a "roer" message with reason = not-allowed-by-object (24) and return code = MDC_RET_CODE_OBJ_BUSY.
Pass/Fail criteria	The PHD must respond with the specified error.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-020		
TP label		Scanner object services. SET Operational-State service		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ScanClassServ 1; M		
	Spec	[IEEE 11073-10406]		
	Testable items	PerScanObjAttr9; M	EpiScanObjAttr9; M	
Test purpose		Check that: A PHD that has scanner derived objects supports the SET service for the Operational-State attribute of the scanner objects.		
Applicability		(C_AG_OXP_046 OR C_AG_OXP_047) AND C_AG_OXP_000		
Other PICS		C_AG_OXP_180		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends the scanner a SET Operational-State attribute: a. If C_AG_OXP_180, THEN APDU ❑ Type = Remote Operation Invoke | Confirmed Event Report ❑ roiv-cmip-confirmed-set ❑ attribute = OperationalState ❑ value = 0 b. If not C_AG_OXP_180, THEN APDU ❑ Type = Remote Operation Invoke | Event Report ❑ roiv-cmip-set ❑ attribute = OperationalState ❑ value = 0 2. If C_AG_OXP_180, the PHD under test must respond with a confirmation ELSE no response for roiv-cmip-set will be received. If C_AG_OXP_180, verify the invoke-id is mirrored from the Set request: a. APDU ❑ Type = Invoke | Confirmed Action ❑ result = accepted		
Pass/Fail criteria		The procedure is executed without errors.		
Notes		The semantics of the Operational-State Attribute are tested in TP/PLT/PHD/OXP/COM/BV-056.		

TP Id	TP/PLT/PHD/OXP/DIM/BV-021
TP label	CfgScanner object Attributes. Confirm-Timeout operation

Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CfgScanAttr 3; C	CfgScanAttr 4; C	OperErrorCond 5; M
		OperErrorCond 6; M	TimeOutVar 3; C	
Test purpose		Check that: If a Configurable Scanner is operating in confirmed mode then the value of attribute Confirm-Timeout matches with the actual timeout value that the PHD uses for the Confirmed Event Report generated from the Scanner object. [AND] TOcer-scan: If the attribute is not present, the PHD shall use the value 3 s.		
Applicability		(C_AG_OXP_046 OR C_AG_OXP_047) AND C_AG_OXP_053 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_180, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with a MDC_NOTI_CONFIG event to send its configuration to the PHG, record the Scanner attribute Confirm-Timeout, if it is not present the simulated PHG will use 3s as a default value. 4. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ☐ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ☐ Once its internal time setting operation is completed, the PHD responds to the PHG. 5. Wait for the PHD under test and the simulated PHG to reach the Operating state. 6. Take some measurements in the PHD. 7. The simulated PHG sets the Operational state of the scanner to 1: a. If C_AG_OXP_180 THEN APDU ☐ Type = Remote Operation Invoke | Confirmed Event Report ☐ roiv-cmip-confirmed-set ☐ attribute = OperationalState ☐ value = 1 b. If not C_AG_OXP_180 THEN APDU ☐ Type = Remote Operation Invoke | Event Report ☐ roiv-cmip-set ☐ attribute = OperationalState ☐ value = 1 8. If C_AG_OXP_180 the PHD under test must respond with a confirmation a. APDU ☐ Type = Invoke | Confirmed Action		

	□ result = accepted 9. Wait until the PHD under test starts to send data. 10. The PHG must not respond for at least the Confirm-Timeout time.
Pass/Fail criteria	The PHD must wait for a Confirmed Event Report Response message for a Confirm-TimeOut period. If the time expires, the PHD must send an abort to the PHG.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-023		
TP label		EpiCfgScanner object. Reports		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	EpiCFgScanClass 1; M	EpiCFgScanClass 2; M	EpiCfgScanEvent 1; C
		ScanClassConcep8; C	ScanClassConcep9; C	EpiCfgScanEvent 28; M
Test purpose		Check that: The PHD sends a report of an episodic scanner whenever one of the observed attributes changes its value [AND] The PHD supports at least one of the events identified in Table 16 of the spec (Unbuf-Scan-Report-Var; Unbuf-Scan-Report-Fixed; Unbuf-Scan-Report-Grouped; Unbuf-Scan-Report-MP-Var; Unbuf-Scan-Report-MP-Fixed; Unbuf-Scan-Report-MP-Grouped). [AND] Episodic scanners using the group, variable or fixed format shall create scan event reports where: If the scanner is episodic and no AttributeChangeSets are collected, the scan event report shall not be sent.		
Applicability		C_AG_OXP_047 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_010, C_AG_OXP_014, C_AG_OXP_180, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The Configurable Episodic Scanner object attribute of interest for this test is: a. Attribute Min-Interval-Reporting □ attribute-id = MDC_ATTR_SCAN_REP_PD_MIN □ attribute-type = RelativeTime □ attribute-length = 4 bytes □ attribute-value = <Record for later comparison> 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command.		

	❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. The simulated PHG sets the Operational state of the scanner to 1. 7. Take a measurement with the PHD under test. 8. Check that the simulated PHG receives the Event sent by the PHD with the changed value and reports it with a grouped type event: a. PrstApdu ❑ Remote Operation Invoke | Confirmed Event Report ❑ Type = MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED (0x0D 0x24) or MDC_NOTI_UNBUF_SCAN_REPORT_VAR (0x0D 0x22) or MDC_NOTI_UNBUF_SCAN_REPORT_FIXED (0x0D 0x23) or MDC_NOTI_UNBUF_SCAN_REPORT_MP_GROUPED (0x0D 0x27) or MDC_NOTI_UNBUF_SCAN_REPORT_MP_VAR (0x0D 0x25) or MDC_NOTI_UNBUF_SCAN_REPORT_MP_FIXED (0x0D 0x26) 9. Take measurements faster than the Reporting Interval recorded in step 4. 10. Wait for the next event report. 11. If it is possible, force the PHD not to change the values that are collected by the scanner object. 12. Wait for the next event report. 13. Check that no scanner event report is sent.
Pass/Fail criteria	• The PHD sends an event report when the attribute changes • The received events are of grouped, variable or fixed type • The event reports are not sent at a rate faster than the minimum reporting interval
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-027		
TP label		EpiCfgScanner object events. Unbuf-Scan-Report		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	EpiCfgScanEvent 10; C	EpiCfgScanEvent 11; C	EpiCfgScanEvent 13; C
		EpiCfgScanEvent 30; C	EpiCfgScanEvent 22; C	EpiCfgScanEvent 23; C
		EpiCfgScanEvent 25; C	EpiCfgScanEvent 33; C	EpiCfgScanEvent 2; C
		EpiCfgScanEvent 3; C	EpiCfgScanEvent 5; C	EpiCfgScanEvent 6; C
		EpiCfgScanEvent 7; C	EpiCfgScanEvent 9; C	EpiCfgScanEvent 14; C
		EpiCfgScanEvent 15; C	EpiCfgScanEvent 17; C	EpiCfgScanEvent 18; C
		EpiCfgScanEvent 19; C	EpiCfgScanEvent 21; C	EpiCfgScanEvent 28; C
		EpiCfgScanEvent 29; C	EpiCfgScanEvent 31; C	EpiCfgScanEvent 32; C
		ScannerGeneral1; O		
	Spec	[IEEE 11073-10406]		
	Testable items	EpiScanObjEv1; M	EpiScanObjEv3; M	EpiScanObjEv4; M
		EpiScanObjEv5; M	EpiScanObjEv6; M	EpiScanObjEv6; M
		EpiScanObjEv7; M	EpiScanObjEv8; M	ObjAccServ2; O
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	General 7; C		
Test purpose		Check that: If an Episodic Scanner uses Unbuf-Scan-Report-Grouped Events to report updated data, then it uses the ScanReportInfoGrouped Event-info parameter.		

	[AND] If an Episodic Scanner uses Unbuf-Scan-Report-MP-Grouped events to report updated data, then it uses the ScanReportInfoMPGrouped Event-info parameter. [AND] If an Episodic Scanner uses Unbuf-Scan-Report-Var events to report updated data, then it uses the ScanReportInfoVar Event-info parameter. [AND] If an Episodic Scanner uses Unbuf-Scan-Report-MP-Var events to report updated data, then it uses the ScanReportInfoMPVar Event-info parameter. [AND] If an Episodic Scanner uses Unbuf-Scan-Report-Fixed events to report updated data, then it uses the ScanReportInfoFixed Event-info parameter. [AND] If an Episodic Scanner uses Unbuf-Scan-Report-MP-Fixed events to report updated data, then it uses the ScanReportInfoMPFixed Event-info parameter. [AND] The event is triggered whenever data values change. [AND] If it reports data in confirmed mode (Confirmed-Mode attribute value is 1), then the PHD uses a roiv-cmip-confirmed-event-report operation. [AND] If it reports data in unconfirmed mode (Confirmed-Mode attribute value is 0), then the PHD uses a roiv-cmip-event-report operation. [AND] Continua PAN service components designed to store and utilize data from multiple users simultaneously and that use agent-initiated measurement data transmission shall identify users and set the person-id field in the corresponding ScanReportPer* structure.
Applicability	C_AG_OXP_047 AND C_AG_OXP_000
Other PICS	C_AG_OXP_033, C_AG_OXP_180
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. Take some measurements with the PHD under test. 2. The simulated PHG sets the Operational state of the scanner to 1. 3. Wait until the PHD under test starts to send its data. 4. Check that the PHD uses the ScanReportInfGrouped Event-info parameter, whenever data values change: a. PrstApdu □ Remote Operation Invoke | Confirmed Event Report OR | Event Report □ Event-Type = MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED (0x0D 0x24) □ scanReportInfoGrouped:SEQUENCE of: ▪ data-req-id = <Not relevant for this test> ▪ scan-report-no = <counter for detection of missing scan reports> ▪ obs-scan-grouped = SEQUENCE OF octet strings □ Or MDC_NOTI_UNBUF_SCAN_REPORT_MP_GROUPED (0x0D 0x27) □ ScanReportInfoMPGrouped.scan-per-grouped = SEQUENCE of: ▪ person-id.value = <record for comparison> ▪ obs-scan-grouped = <Not relevant for this Test> □ Or MDC_NOTI_UNBUF_SCAN_REPORT_VAR (0x0D 0x22)

	❑ ScanReportInfoVar= SEQUENCE of: ▪ data-req-id = <Not relevant for this test> ▪ scan-report-no = <counter for detection of missing scan reports> ▪ obs-scan-var = SEQUENCE OF ObservationScan ❑ Or MDC_NOTI_UNBUF_SCAN_REPORT_MP_VAR (0x0D 0x25) ❑ ScanReportInfoMPVar.scan-per-var = SEQUENCE of: ▪ person-id.value = <record for comparison> ▪ obs-scan-var = <Not relevant for this Test> ❑ Or MDC_NOTI_UNBUF_SCAN_REPORT_FIXED (0x0D 0x23) ❑ ScanReportInfoFixed= SEQUENCE of: ▪ data-req-id = <Not relevant for this test> ▪ scan-report-no = <counter for detection of missing scan reports> ▪ obs-scan-fixed = SEQUENCE OF ObservationScanFixed ❑ Or MDC_NOTI_UNBUF_SCAN_REPORT_MP_FIXED (0x0D 0x26) ❑ ScanReportInfoMPFixed.scan-per-fixed = SEQUENCE of: ▪ person-id.value = <record for comparison> ▪ obs-scan-fixed = <Not relevant for this Test>
Pass/Fail criteria	• The PHD sends data using grouped, variable or fixed event reports. • If the PHD supports multi-person event reports for one or more episodic scanner object (C_AG_OXP_033= TRUE) THEN the PHD uses MP Unbuf Event report, and check that every person-id is different from each other or "unknown-person-id" (65535). • If C_AG_OXP_033= TRUE and MP event reports have been received, a pop-up will show the received measurements to make the operator identify if measurements have been correctly assigned to every person.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-032_A		
TP label		PeriCfgScanner object Attribute. Reporting interval attribute 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PeriCfgScanClass 1; M	PeriCfgScanAttr 2; M	PeriCfgScanEvent 26; M
		PeriCfgScanClass 3; R	ScanClassConcep8; C	ScanClassConcep9; C
Test purpose		Check that: A periodic scanner in the active Operating state sends an event report at a rate of one per reporting interval, where the reporting interval is the value of the Reporting-Interval attribute. [AND] The same objects and attributes are included in each report regardless of whether their values have changed. [AND] When a period configurable scanner is enabled by a PHG, scan reports should be sent within a reasonable time and synchronized to the reporting interval of the scanner. The time between the scanner being enabled and the sending of the first scan report should be within the reporting interval plus 15 seconds. [AND] Periodic scanners using the group, variable or fixed format shall create scan event reports where: If the scanner is periodic and no AttributeChangeSets are collected, an empty scan event report shall be sent when the period expires.		
Applicability		C_AG_OXP_046 AND C_AG_OXP_000		

Other PICS	C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_180, C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test have been associated, but the PHD configuration is unknown for the simulated PHG, so the PHD and the simulated PHG will be in the Configuring state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The Configurable Periodic Scanner object attribute of interest for this test is: a. Mandatory attribute Reporting-Interval □ attribute-id = MDC_ATTR_SCAN_REP_PD □ attribute-type = RelativeTime □ attribute-length = 4 bytes □ attribute-value = < Record for later comparison > 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. The simulated PHG sets the Operational state of the scanner to 1. 7. Take several measurements. 8. Wait until the PHD under test starts to send its data. 9. Wait for the next event report. 10. If it is possible, force the PHD not to change the values that are collected by scanner object. 11. Wait for the next event report. 12. Check that an empty event report is sent.
Pass/Fail criteria	The Event reports must arrive periodically with a period the same as the time defined in Reporting-Interval. The time between the scanner being enabled (step 6) and the sending of the first scan report (step 7) should not exceed the reporting interval plus 15 seconds. An empty Scan Event Report is sent by the PHD under test when the value has not changed (step 12).
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-032_B		
TP label		PeriCfgScanner object Attribute. Reporting interval attribute 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PeriCfgScanClass 1; M	PeriCfgScanAttr 2; M	PeriCfgScanEvent 26; M

Test purpose	Check that: Event Reports include measurements that are acquired faster than reporting interval [AND] A periodic scanner in the active Operating state sends an event report at a rate of one per reporting interval, where the reporting interval is the value of the Reporting-Interval attribute. [AND] The same objects and attributes are included in each report regardless of whether their values have changed
Applicability	C_AG_OXP_046 AND C_AG_OXP_000
Other PICS	C_AG_OXP_180
Initial condition	The simulated PHG and PHD under test have been associated, but the PHD configuration is unknown for the simulated PHG, so the PHD and the simulated PHG will be in the Configuring state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The Configurable Periodic Scanner object attribute of interest for this test is: a. Mandatory attribute Reporting-Interval □ attribute-id = MDC_ATTR_SCAN_REP_PD □ attribute-type = RelativeTime □ attribute-length = 4 bytes □ attribute-value = < Record for later comparison > 5. The simulated PHG sets the Operational state of the scanner to 1. 6. Wait until the PHD under test sends two event reports. 7. Take measurements faster than the Reporting Interval recorded in step 4. 8. Wait for the next event report.
Pass/Fail criteria	• In step 6 verify that the received observed value is the same for the two events (same objects and attributes, but not attribute value). • Verify that in step 8 the received event contains a number of measurements higher than the number of measurements received in step 6.
Notes	In last paragraph of clause 6.3.9.5.1 there is an example where states that it must send ALL the measurements, not only the last change. It has to send all the registered observations. Example: A Periodic Configurable Scanner is set up to 'scan' two Metric objects with a Reporting-Interval of 1 sec. The two objects update their corresponding observed value periodically with an interval of 1 sec and ½ sec respectively. The Periodic Configurable Scanner then issues Event Reports every second containing one observation scan of Metric object #1 and two observation scans of Metric object #2.

TP Id		TP/PLT/PHD/OXP/DIM/BV-036		
TP label		PeriCfgScanner object events. Buf-Scan-Report		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PeriCfgScanEvent 10; C	PeriCfgScanEvent 11; C	PeriCfgScanEvent 13; C
		PeriCfgScanEvent 1; C	PeriCfgScanEvent 22; C	PeriCfgScanEvent 23; C
		PeriCfgScanEvent 25; C	PeriCfgScanEvent 2; C	PeriCfgScanEvent 3; C
		PeriCfgScanEvent 5; C	PeriCfgScanEvent 6; C	PeriCfgScanEvent 7; C
		PeriCfgScanEvent 9; C	PeriCfgScanEvent 14; C	PeriCfgScanEvent 15; C
		PeriCfgScanEvent 17; C	PeriCfgScanEvent 18; C	PeriCfgScanEvent 19; C

		PeriCfgScanEvent 21; C	ScannerGeneral1; O	
	Spec	[IEEE 11073-10406]		
	Testable items	PerScanObjEv1; M	PerScanObjEv3; M	PerScanObjEv4; M
		PerScanObjEv5; M	PerScanObjEv6; M	PerScanObjEv7; M
		PerScanObjEv8; M	ObjAccServ2; O	
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	General 7; C		
Test purpose		Check that: If a Periodic Configurable Scanner uses Buf-Scan-Report-Grouped Events to report updated data, then it uses the ScanReportInfoGrouped Event-Info parameter. [AND] If a Periodic Configurable Scanner uses Buf-Scan-Report-MP-Grouped Events to report updated data, then it uses the ScanReportInfoMPGrouped Event-Info parameter. [AND] If a Periodic Configurable Scanner uses Buf-Scan-Report-Var Events to report updated data, then it uses the ScanReportInfoVar Event-Info parameter. [AND] If a Periodic Configurable Scanner uses Buf-Scan-Report-MP-Var Events to report updated data, then it uses the ScanReportInfoMPVar Event-Info parameter. [AND] If a Periodic Configurable Scanner uses Buf-Scan-Report-Fixed Events to report updated data, then it uses the ScanReportInfoFixed Event-Info parameter. [AND] If a Periodic Configurable Scanner uses Buf-Scan-Report-MP-Fixed Events to report updated data, then it uses the ScanReportInfoMPFixed Event-Info parameter. [AND] If it reports data in confirmed mode (Confirmed-Mode attribute value is 1), then the PHD uses a roiv-cmip-confirmed-event-report operation. [AND] If it reports data in unconfirmed mode (Confirmed-Mode attribute value is 0), then the PHD uses a roiv-cmip-event-report operation. [AND] The PHD supports at least one of the events identified in Table 18 of the spec (Buf-Scan-Report-Var; Buf -Scan-Report-Fixed; Buf -Scan-Report-Grouped; Buf -Scan-Report-MP-Var; Buf -Scan-Report-MP-Fixed; Buf -Scan-Report-MP-Grouped). [AND] Continua PAN service components designed to store and utilize data from multiple users simultaneously and that use agent-initiated measurement data transmission shall identify users and set the person-id field in the corresponding ScanReportPer* structure		
Applicability		C_AG_OXP_046 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_034, C_AG_OXP_180		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Make a change to one of the observed values by the PHD under test. 2. The simulated PHG sets the Operational state of the scanner to 1. 3. Wait until the PHD under test starts to send its data. 4. Check that the simulated PHG receives the Event send by the PHD with the changed value and reports it with a grouped type event: a. DataApdu		

	❑ Remote Operation Invoke | Confirmed Event Report or | Event Report ❑ Event-Type = MDC_NOTI_BUF_SCAN_REPORT_GROUPED (0x0D 0x2A) ❑ ScanReportInfoGrouped:SEQUENCE of: ▪ data-req-id = <Not relevant for this test> ▪ scan-report-no = <counter for detection of missing scan reports> ▪ obs-scan-grouped = SEQUENCE OF octet strings ❑ Or MDC_NOTI_BUF_SCAN_REPORT_MP_GROUPED (0x0D 0x2D) ❑ ScanReportInfoMPGrouped.scan-per-grouped = SEQUENCE of: ▪ person-id.value = <record for comparison> ▪ obs-scan-grouped = <Not relevant for this Test> ❑ Or MDC_NOTI_BUF_SCAN_REPORT_VAR (0x0D 0x28) ❑ ScanReportInfoVar= SEQUENCE of: ▪ data-req-id = <Not relevant for this test> ▪ scan-report-no = <counter for detection of missing scan reports> ▪ obs-scan-var = SEQUENCE OF ObservationScan ❑ Or MDC_NOTI_BUF_SCAN_REPORT_MP_VAR (0x0D 0x2B) ❑ ScanReportInfoMPVar.scan-per-var = SEQUENCE of: ▪ person-id.value = <record for comparison> ▪ obs-scan-var = <Not relevant for this Test> ❑ Or MDC_NOTI_BUF_SCAN_REPORT_FIXED (0x0D 0x29) ❑ ScanReportInfoFixed= SEQUENCE of: ▪ data-req-id = <Not relevant for this test> ▪ scan-report-no = <counter for detection of missing scan reports> ▪ obs-scan-fixed = SEQUENCE OF ObservationScanFixed ❑ Or MDC_NOTI_BUF_SCAN_REPORT_MP_FIXED (0x0D 0x2C) ❑ ScanReportInfoMPFixed.scan-per-fixed = SEQUENCE of: ▪ person-id.value = <record for comparison> ▪ obs-scan-fixed = <Not relevant for this Test>
Pass/Fail criteria	• The PHD sends grouped, variable or fixed format event reports. • If the PHD supports multi-person event reports for one or more periodic scanner object (C_AG_OXP_034= TRUE) THEN the PHD uses MP Buf Event report, check that every person-id is different from each other or "unknown-person-id" (65535). • If C_AG_OXP_034 = TRUE and MP event reports have been received, a pop-up will show the received measurements to make the operator identify if the measurements have been correctly assigned to every person.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-042		
TP label		CfgScanner object. Confirm-Mode attribute		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CfgScanAttr 1;M		
Test purpose		Check that: IF Confirmed-Mode attribute value is 1, THEN the PHD uses a roiv-cmip-confirmed-event-report operation. IF Confirmed-Mode attribute value is 0, THEN the PHD uses a roiv-cmip-event-report		

	operation
Applicability	(C_AG_OXP_046 OR C_AG_OXP_047) AND C_AG_OXP_000
Other PICS	C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_180, C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG, record the Scanner attribute Confirm-Mode. 4. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 5. Wait for the PHD under test and the simulated PHG to reach the Operating state. 6. The simulated PHG sets the Operational state of the scanner to 1. 7. Wait until the PHD under test starts to send data. 8. Check that the simulated PHG receives the Event sent by the PHD: IF the Confirm-Mode recorded in step 3, is "confirmed": a. PrstApdu ❑ Remote Operation Invoke | Confirmed Event Report IF the Confirm-Mode recorded in step 3, is "unconfirmed": a. PrstApdu ❑ Remote Operation Invoke | Event Report
Pass/Fail criteria	IF the Confirm-Mode value is confirmed THEN the PHD sends a Confirmed Event Report. IF the Confirm-mode value is unconfirmed THEN the PHD sends an Unconfirmed Event Report.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-043		
TP label		PM-Store object. Change Unit Code attribute		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	StoreClassGen 1; M		
Test purpose		Check that: If an attribute value in a PM-segment depends on another attribute value not stored in the PM-segment, then that dependent attribute shall not change value during the lifetime of the PM-segment. Otherwise, the PHD shall store the dependent attribute value in the PM-segment.		
Applicability		(C_AG_OXP_073) AND C_AG_OXP_000 AND C_AG_DGC_018		

Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. Make a change to the contextual attribute Unit-Code for an object that is stored in the PM-Store. 2. The simulated PHG sends a request (Get-Segment-Info) for the PM-Segment attributes with SegmSelection = 1 to obtain all the segments for the PM-Store: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = all-segments 3. The PHD issues a response with the PM-Segments attributes: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmentInfoList: Record value for PM-Segment-Entry-Map attribute 4. Repeat steps 1 and 2 for every PM-Store.
Pass/Fail criteria	In step 2, there is at least one segment that stores Unit-code attribute (PM-Segment-Entry-Map).
Notes	

<table>
<tr><td colspan="2"><b>TP Id</b></td><td colspan="3">TP/PLT/PHD/OXP/DIM/BV-044</td></tr>
<tr><td colspan="2"><b>TP label</b></td><td colspan="3">PeriCfgScanner object. Reporting interval and FIFO</td></tr>
<tr><td rowspan="2"><b>Coverage</b></td><td><b>Spec</b></td><td colspan="3">[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]</td></tr>
<tr><td><b>Testable items</b></td><td>ScanClassConcep4; M</td><td>ScanClassConcep7; M</td><td></td></tr>
<tr><td colspan="2"><b>Test purpose</b></td><td colspan="3">Check that:
The periodic scanner also requires that the rate of generation of all collected AttributeChangeSets shall have a fixed timing relationship with each other and with the period of the periodic scanner.
[AND]
The periodic scanner shall insert AttributeChangeSets from the same object in the scan event report in FIFO order.</td></tr>
<tr><td colspan="2"><b>Applicability</b></td><td colspan="3">C_AG_OXP_046 AND C_AG_OXP_000</td></tr>
<tr><td colspan="2"><b>Other PICS</b></td><td colspan="3">C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_180, C_AG_OXP_293</td></tr>
<tr><td colspan="2"><b>Initial condition</b></td><td colspan="3">The simulated PHG and PHD under test are in the Unassociated state.</td></tr>
<tr><td colspan="2"><b>Test procedure</b></td><td colspan="3">1. The simulated PHG receives an association request from the PHD under test.
2. The simulated PHG responds with a result = accepted-unknown-config.
3. The PHD under test responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG.
4. Record attribute for Periodic Scanner Object:
a. Mandatory attribute Reporting-Interval
□ attribute-id = MDC_ATTR_SCAN_REP_PD
□ attribute-type = RelativeTime
□ attribute-length = 4 bytes</td></tr>
</table>

	❑ attribute-value = < Record for later comparison > 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Wait for 4 * Reporting Interval (the reasonable time for the scanner being enabled and the sending of the first scan report) or 4*15 seconds, whichever is greater. 7. Set the Operational State to 1 for the Periodic Scanner object. 8. Wait until the PHD under test starts to send its data and record it. 9. Set the Operational State to 0 for the Periodic Scanner object. 10. Wait for 4 * Reporting Interval or 4*15 seconds, whichever is greater. 11. Set the Operational State to 1 for the Periodic Scanner object. 12. Wait until the PHD under test starts to send its data and record it.
Pass/Fail criteria	In steps 8 and 12 the same number of observations must be received from the PHD. In step 12, check that measurements have been received following a FIFO sequence.
Notes	

TP Id		TP/PLT/PHD/OXP/DIM/BV-045		
TP label		PM-Store object methods. Clear-Segments Base-Offset-Time Range method 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M
		PM-StoreMeth 8; O	PersStoreMtrDatTransf 20; M	PersStoreMtrDatTransf 21; M
		PersStoreMtrDatTransf 22; O	PersStoreMtrDatTransf 23; M	PM-StoreMeth 32; M
		PM-StoreMeth 33; M		
Test purpose		Check that: If the PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action [AND] According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely [AND] The Instance-Number of all other PM-Segments is unaffected by clearing a segment [AND] For PM-segments cleared using the by time method and using base time with offset, only PM-Segments having Segment-Start-BO-Time and Segment-End-BO-Time fields entirely within the specified time period are cleared. In using Segment-Start-BO-Time and Segment-End-BO-Time, the base time shall have a valid time (i.e., a non-zero value).		

Applicability	C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_072 AND C_AG_OXP_014 AND C_AG_OXP_000
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state and the PHD supports at least one PM-Segment with data stored.
Test procedure	1. Make sure the PHD under test is not taking measurements which are stored in PM-Segments. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD under test issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute: a. PM-Store-Capab: ❑ attribute-id = MDC_ATTR_PM_STORE_CAPAB ❑ attribute-type = PmStoreCapab ❑ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining a time range) 4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-BO-Time" and "Segment-End-BO-Time" of every PM-Segment. IF bit 8 of PMStoreCapab was set: 6. The simulated PHG sends a Clear-Segment: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR ❑ SegmSelection = bo-time-range, selecting a range with its boundaries that are not within the Segment-Start-BO-Time and Segment-End-BO-Time 7. The PHD under test operation response: a. Data APDU ❑ Type = Roer ❑ ErrorResult = no-such-action (9)
Pass/Fail criteria	In step 7 the PHD must send the specified error.
Notes	Error code was not clearly defined in the spec. In the new edition of [ISO/IEEE 11073-20601-2015A], the clear-segment using time range has been clarified. "For PM-segments cleared using the by time method, only PM-segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.", but if the PHG sends a Clear-Segment but the segment has not a Segment-Start-Abs-Time and Segment-End-Abs-Time within the specified time-period, the PHD will send a Roer message. At this point, it would be up to the PHD what error code (Roer message) to send (No-such-action, not-allowed-by-object, or both). If one wants to clear the segment due to all the internal timestamps that were saved in the segment as falling inside the given time period, then the PHD could do that. Returning an error is also possible.

TP Id		TP/PLT/PHD/OXP/DIM/BV-046		
TP label		PM-Store object methods. Clear-Segments Base-Offset-Time Range method 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable	PM-StoreMeth 1; O	PM-StoreMeth 2; C	PM-StoreMeth 6; M

<table>
<tr><td rowspan="3"></td><td rowspan="3"><b>items</b></td><td>PM-StoreMeth 8; O</td><td>PersStoreMtrDatTransf 20; M</td><td>PersStoreMtrDatTransf 21; M</td></tr>
<tr><td>PersStoreMtrDatTransf 22; O</td><td>PersStoreMtrDatTransf 23; M</td><td>PM-StoreMeth 32; M</td></tr>
<tr><td>PM-StoreMeth 33; M</td><td>PM-StoreMeth 23; C</td><td>PersStoreMtrDatTransf 24; M</td></tr>
<tr><td colspan="2"><b>Test purpose</b></td><td colspan="3">Check that:
If PHD supports the Clear-Segment (time range) method, then it responds to Clear-Segment requests with a Data APDU with an operation type rors-cmip-confirmed-action
[AND]
According to PM-Store-Capab attribute this method removes all entries from the specified PM-Segment, leaving it empty, or it removes the defined PM-Segment completely
[AND]
The Instance-Number of all other PM-Segments is unaffected by clearing a segment
[AND]
The PHD may support PM-segment clearing. If the PHD supports this function(indicated by the pmsc-clear-segm-all-sup, pmsc-clear-segm-by-list-sup, and pmsc-clear-segm-by-time-sup flags in the PM-Store-Capab attribute being set) then it may support the time range selection criteria (pmsc-clear-segm-by-time-sup)
[AND]
For PM-segments cleared using the by time method and using base time with offset, only PM-Segments having Segment-Start-BO-Time and Segment-End-BO-Time fields entirely within the specified time period are cleared. In using Segment-Start-BO-Time and Segment-End-BO-Time, the base time shall have a valid time (i.e., a non-zero value). If the offset field has a value 0x7FFF (32767), then only PM-segments having base time entirely within the specified base time period are cleared, otherwise for any other value of offset field only PM-segments having local time (base time with offset added) entirely within the specified time period are cleared.
[AND]
If the PHG invokes the Clear-Segments method but the PHD does not support the particular action (list of segments or range of segments), then the PHD shall respond with a roer DataApdu with a RoerErrorValue of "not-allowed-by-object".</td></tr>
<tr><td colspan="2"><b>Applicability</b></td><td colspan="3">C_AG_OXP_041 AND C_AG_OXP_071 AND C_AG_OXP_014 AND C_AG_OXP_000</td></tr>
<tr><td colspan="2"><b>Other PICS</b></td><td colspan="3"></td></tr>
<tr><td colspan="2"><b>Initial condition</b></td><td colspan="3">The simulated PHG and PHD under test are in the Operating state and the PHD has at least one PM-Segment with data stored.</td></tr>
<tr><td colspan="2"><b>Test procedure</b></td><td colspan="3">1. Make sure the PHD is not taking measures which are stored in PM-Segments.
2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes.
3. The PHD issues a GET response with the PM-Store attributes, record the values of the PM-Store-Capab attribute:
a. PM-Store-Capab:
❑ attribute-id = MDC_ATTR_PM_STORE_CAPAB
❑ attribute-type = PmStoreCapab
❑ attribute-value = Record the value of bit 8 (Indicates that PM-Segments in the SegmSelection data type can be cleared by defining a TimeRange)
IF bit 8 oft PMStoreCapab was set:
4. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments.
5. The PHD issues a response (rors-cmip-confirmed-action) with the PM-Segment attributes it supports, record the attributes "Segment-Start-BO-Time" and "Segment-BO-Abs-Time" of every PM-Segment.
6. The simulated PHG sends a Clear-Segment:</td></tr>
</table>

	a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR ❑ SegmSelection = bo-time-range, selecting a range with its boundaries set to Segment-Start-BO-Time and to Segment-End-BO-Time of one of the PM-Segments. 7. The PHD under test operation response: a. Data APDU ❑ Type = Response | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR 8. Delay. 9. The simulated PHG sends a request for the PM-Segment Data of one of the cleared PM-Segments: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ SegmSelection = <Instance number of the selected PM-Segment that contained data before the clear-segment action in step 6> 10. The PHD issues an action response with the Data a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ TrigSegmXferRsp = ▪ IF pmsc-clear-segm-remove(9) = 0 THEN TrigSegmXferRsp = tsxr-fail-segm-empty ELSE TrigSegmXferRsp = tsxr-fail-no-such-segment IF bit 8 of PMStoreCapab was NOT set 11. Simulated PHG sends a Clear-Segment: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_CLEAR ❑ SegmSelection = bo-time-range, selecting a range with its boundaries set to the a minimum of Base-Offset-Time type and to the absolute of the Base-Offset-Time type 12. The PHD under test operation response: a. Data APDU ❑ Type = Roer ❑ ErrorResult = no-allowed-by-object (24)
Pass/Fail criteria	• In step 7 the PHD must send a confirmation • In step 10 the TrigSemgXferRsp must be the specified • If the PHD does not support Clear-Segment by time, the PHD must send roer (not-allowed-by -object)
Notes	

A.3 Subgroup 1.2.2 – PHD service model (SER)

TP Id		TP/PLT/PHD/OXP/SER/BV-000		
TP label		Object Access Services: No-Such-Action Error		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ObjAccessServ 3; M	DataTrans 2; O	MDSMethod 4; M
		PersStoreMtrDatTransf 24; M	MDSMethod 6; M	PM-StoreMeth 34; C
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	General 2; M		
Test purpose		Check that: If a request for a confirmed action is received by a PHD that does not support the action, the PHD replies with an error (roer) value of no-such-action [AND] If the PHG invokes the Clear-Segments method but the PHD does not support this function at all then the PHD shall respond with a roer DataApdu with an RoerErrorValue of “no-such-action”. [AND] If the PHD supports Set-Time, it shall respond with a rors-cmip-confirmed-action. If the PHD does not support Set-Time, it shall respond with a no-such-action error (roer). [AND] If the PHD does not support Set-Base-Offset-Time, it shall respond with a no-such-action error (roer) [AND] PHD shall not include the Base Offset Time in any Continua configurations except for Basic electrocardiograph (ECG) and Insulin Pump (IP) device specializations.		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_008, C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_071		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends a roiv-cmip-confirmed-action with action-type = MDC_ACT_DATA_REQUEST. 2. The PHD under test shall reply with an error. The expected fields sent by the PHD are: a. Error-value □ field-type = INT-U16 □ field-length = 2 bytes □ field- value = no-such-action(9) 3. IF the PHD under test does not support Set Time (i.e., C_AG_OXP_008 = FALSE) OR the PHD under test supports Set Time for Base-Offset-Time (i.e., C_AG_OXP_008 = TRUE and C_AG_OXP_014 = TRUE) THEN the simulated PHG sends a roiv-cmip-confirmed-action with action-type = MDC_ACT_SET_TIME using Absolute-Time (SetTimeInvoke). 4. The PHD under test shall reply with an error. The expected fields sent by the PHD are: a. Error-value □ field-type = INT-U16 □ field-length = 2 bytes □ field- value = no-such-action(9) 5. IF the PHD under test does not support Set Time (i.e., C_AG_OXP_008 = FALSE) OR the PHD under test supports Set Time for Absolute-Time Time (i.e., C_AG_OXP_008 = TRUE and C_AG_OXP_009 = TRUE) THEN the PHG sends a roiv-cmip-confirmed-action with action-type = MDC_ACT_SET_BO_TIME using Base-Offset-Time		

	(SetBOTimeInvoke). 6. The PHD under test shall reply with an error. The expected fields sent by the PHD are: a. Error-value □ field-type = INT-U16 □ field-length = 2 bytes □ field- value = no-such-action(9) 7. IF C_AG_OXP_041 and does not support the Clear-Segment action then the simulated PHG sends a roiv-cmip-confirmed-action with action-type = MDC_ACT_SEG_CLR. 8. The PHD under test shall reply with an error. The expected fields sent by the PHD are: a. Error-value □ field-type = INT-U16 □ field-length = 2 bytes □ field- value = no-such-action(9) 9. IF C_AG_OXP_041 and it does not support selection by time range THEN the simulated PHG sends a roiv-cmip-confirmed-action with action-type = MDC_ACT_SEG_GET_INFO and SegmSelection = abs-time-range (if the PHD supports Absolute Time) or SegmSelection = bo-time-range (if the PHD supports Base Offset Time) 10. The PHD under test shall reply with an error. The expected fields sent by the PHD are a. Error-value □ field-type = INT-U16 □ field-length = 2 bytes □ field- value = no-such-action(9)
Pass/Fail criteria	The PHD under test sends a No-Such-Action Error and the invoke-id is mirrored from the roiv-cmip-* messages.
Notes	

TP Id		TP/PLT/PHD/OXP/SER/BV-001		
TP label		Configuration event report: dev-configuration-id is locally unique		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ConfEventRep 2; C	ConfEventRep 6; M	ConfEventRep 19; M
		ConfEventRep 24; M	ConfEventRep 25; R	ConfNormalProc 13; C
		ConfNormalProc 14;M	ConfExitCond 4; C	ConfEventRep 27: M
		AgentStateMach 79; M	ConfNormalProc 25; R	ConfEventRep 37; R
Test purpose		Check that: If PHD has multiple device configurations, the assigned Dev-Configuration-Id values are locally unique [AND] The PHD transfers its configuration the PHG using a configuration event report [AND] The PHD consistently uses the Dev-Configuration-Id for subsequent associations. [AND] If the PHD receives an unsupported configuration message, the PHD sends a further configuration. This process is repeated until the PHD has attempted all configurations. When it sends an Association Release message with a reason code of no-more-configurations to indicate that it is unable to operate with the PHG the PHD moves to Disassociating state. [AND] The same Dev-Configuration-Id shall not be used by an PHD for subsequent associations to identify a different device configuration.		

	[AND] An PHD should use the same value for Dev-Configuration-Id in future Association Requests with the PHG to denote the same configuration of the device.
Applicability	C_AG_OXP_000
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The PHD under test sends an Association Request to the simulated PHG. The expected fields sent by the PHD are: a. dev-config-id ❑ field-type = ConfigId ❑ field-length = 2 bytes ❑ field- value = Record it for comparison b. Data-Req-Mode-Capab: ❑ field-length = 4 bytes ❑ field- value = 0xXX 0xXX 0x01 0xXX (Agent initiated) 2. The simulated PHG responds with an accepted-unknown-config. 3. The PHD sends a configuration event report, with the following fields: dev-config-id ❑ field-type = ConfigId ❑ field-length = 2 bytes ❑ field- value = <Record this Value> 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD sends a new configuration event report with a new configuration (if it has more). 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a ReleaseRequest with the reason "no-more-configurations" and the PHD moves to the Disassociating state. 7. The simulated PHG responds with a Release Response message. 8. Wait for the PHD under test to send an Association Request. 9. Repeat steps 2 to 5 until the PHD under test sends a Release Request with the reason "no-more-configurations".
Pass/Fail criteria	• Dev-config-id is the same for the two first messages and every other ConfigId must be different from all others before the Release Request of step 6 • The PHD shall send a Release-request (no-more-configurations) in step 6 • Verify that the PHD should use the same Dev-Config-Id in steps 2 – 5 and that in step 9 and in every received configuration in step 9 it is the same as one of the received in the configuration messages in step 3
Notes	

TP Id		TP/PLT/PHD/OXP/SER/BV-001_A		
TP label		Configuration event report: Maximum Size		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CommonCharac 3; M		
Test purpose		Check that: The total size of the response does not exceed the maximum APDU size established by the specialization		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_041		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responds with an "accepted-unknown". 3. The PHD under test sends its configuration with an event report. Record the size of the event report. 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD under test sends a new configuration event report with a new configuration (if it has more). Record the size of the event report. 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a ReleaseRequest with the reason "no-more-configurations".		
Pass/Fail criteria		• Total size of the response cannot exceed the sum of the APDU sizes of the supported specializations (limited to an absolute limit of 64512 octets): ○ Pulse oximeter → 9216 octets ○ Weighing scales → 896 octets ○ Glucose meter → 5120 octets or 64512 octets if the PHD supports PM-Store ○ Blood pressure → 896 octets ○ Thermometer → 896 octets ○ Independent activity hub → 5120 octets ○ Cardiovascular → 64512 octets or 6624 octets if the PHD supports Step Counter Profile ○ Strength → 64512 octets ○ Adherence monitor → 1024 octets ○ Peak flow → 2030 octets ○ Body composition analyser → 7730 octets ○ Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store ○ Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store ○ International normalized ratio → 896 octets or 64512 if the PHD supports PM-Store ○ Insulin pump → 7168 octets or 5120 if PHD supports PM-Store ○ Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store ○ Power status monitor → 940 octets for Simple PSM profile or 1660 for Advanced PSM profile		
Notes				

TP Id		TP/PLT/PHD/OXP/SER/BV-002		
TP label		Configuration event report: Change attributes values		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ConfEventRep 10; O	ConfEventRep 13; M	ConfEventRep 14; M
Test purpose		Check that: If the PHD adds new attributes to an object or changes attribute values during the association, then it does not send a new configuration. [AND] In subsequent associations, when a previously used Dev-Configuration-Id is specified, the configuration being referenced does not include any changes made during a prior association. [AND] The PHD only makes persistent changes to a configuration by re-associating and specifying a different Dev-Configuration-Id and the new configuration desired at configuration time		
Applicability		C_AG_OXP_098 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responds with an accepted-unknown-config. 3. The PHD responds with a roiv-cmip-confirmed-event report message with a MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. If ConfigId (ConfigReport) matches the tested configuration, the simulated PHG responds with "accepted-config" and records the ConfigReport received in step 3. 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 it issues the Set-Time action command. • IF C_AG_OXP_014 it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Wait for the PHD under test to reach Operating state. 7. Make a change to the attribute or add it. 8. Check that the event report informing about the attribute change or addition is received. 9. Send a release-request to the PHD under test with reason normal (0). 10. Make the PHD try to re-associate. 11. The simulated PHG responds with an accepted-unknown-config. 12. Check the attribute that has been changed or added is not present when the PHD sends the ConfigReport.		
Pass/Fail criteria		Changes made to the attribute must not be present in the second association.		
Notes		The attribute that is changed in step 7 must be an attribute whose "initial" value is defined in the ConfigReport.		

<table>
<tr><td colspan="2"><b>TP Id</b></td><td colspan="3">TP/PLT/PHD/OXP/SER/BV-004</td></tr>
<tr><td colspan="2"><b>TP label</b></td><td colspan="3">PHD transmits data in a fixed format Event Report</td></tr>
<tr><td rowspan="3"><b>Coverage</b></td><td><b>Spec</b></td><td colspan="3">[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]</td></tr>
<tr><td rowspan="2"><b>Testable items</b></td><td>FormatEventRep 1; M</td><td>FormatEventRep 4; O</td><td>PersonEventRep 2; O</td></tr>
<tr><td>FormatEventRep 6; M</td><td>FormatEventRep 7; M</td><td></td></tr>
<tr><td colspan="2"><b>Test purpose</b></td><td colspan="3">Check that:
If PHD transmits data in fixed format, then it reports the object handle and the attribute values are in the same order and size as specified in the Attribute-Value-Map
[AND]
These Attribute-Value-Map attribute shall be defined and transmitted to the PHG before fixed format event report transfer commences
[AND]
The order of these elements is defined by the order in which the attribute identifiers are listed in the Attribute-Value-Map. The PHD controls the order and communicates it to the PHG via the Attribute-Value-Map attribute</td></tr>
<tr><td colspan="2"><b>Applicability</b></td><td colspan="3">(C_AG_OXP_182 OR C_AG_OXP_183 OR C_AG_OXP_184) AND C_AG_OXP_000</td></tr>
<tr><td colspan="2"><b>Other PICS</b></td><td colspan="3">C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293</td></tr>
<tr><td colspan="2"><b>Initial condition</b></td><td colspan="3">The simulated PHG and PHD under test are in the Configuring state.</td></tr>
<tr><td colspan="2"><b>Test procedure</b></td><td colspan="3">1. The PHD under test sends its configuration to the simulated PHG. Save the number of measurement objects (config-obj-list.count=n).
2. Every measurement object has an obj-handle and one or more attributes. The objects of interest for this test cases are all the metric derived ones (Obj-class=MDC_MOC_VMO_METRIC_ENUM 0x00 0x05, Obj-class=MDC_MOC_VMO_METRIC_NU 0x00 0x06, or Obj-class=MDC_MOC_VMO_METRIC_SA_RT 0x00 0x09). For each of them, check:
a. Attribute Attribute-Val-Map
□ attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP (0x0A 0x55)
□ AttrValMap.count = N (number of objects for this measurement object)
□ AttrValMap.length = L
b. For each attribute (of the N present) check the ID and the length at which it will be transmitted
□ field-type=MDC_ATTR_*
□ field-value=0xXX 0xXX, where the length will be declared (K).
3. IF C_AG_OXP_293 THEN:
a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes.
b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object.
c. IF the mds-time-mgr-set-time bit is set:
□ The PHG moves to Configuring/Sending Set Time substate and:
• IF C_AG_OXP_009 THEN it issues the Set-Time action command.
• IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command.
□ Once its internal time setting operation is completed, the PHD responds to the PHG.
4. Wait for the PHD under test to reach the Operating state and take some measurements.
5. When the PHD under test sends an event report to the simulated PHG with measurement observations, check the following:
□ event-type= MDC_NOTI_SCAN_REPORT_FIXED (0X0D 0X1D)</td></tr>
</table>

	□ ObservationScanFixed.count= P (where P<=N, and it is the number of objects reported in this event report) □ obj-handle = <It has to be the same that obj-handle of the Measurement object sent in the PHD's configuration> □ obs-val-data.length= 0xXX 0xXX, where the value is the length "K". □ The actual observed measurement value will come in the next field, but this value is of no interest for this test case IF the Absolute-Time attribute is present in the Attribute-Val-Map THEN: 6. Record the value of the received Time Stamp 7. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0 (to request an MDS object) and an empty attribute-id-list to indicate all attributes. 8. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object, record the Date-and-Time value.
Pass/Fail criteria	• The Fixed Event report contains the same list of attributes (and sizes) in the same order that was declared in the configuration message • The Time Stamp and Date-and-Time values are coherent • The Attr-Val-Map for an object has to be received prior to the measurement (Config Report or MDS Event Report)
Notes	

TP Id		TP/PLT/PHD/OXP/SER/BV-004_A		
TP label		PHD transmits data in variable format Event Report		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	FormatEventRep 4; O	PersonEventRep 2; O	
Test purpose		Check that: If PHD transmits data in variable format, then the event report fits to specified format		
Applicability		C_AG_OXP_189 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG, record the attribute Attribute-Value-Map. 4. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG.		

	5. Once in the Operating state take a measurement with the PHD under test. 6. Wait for the event report from the PHD under test. 7. When the PHD under test sends an event report to the simulated PHG with measurement observations, check the following: a. If the data is from a single person □ Event-type = MDC_NOTI_SCAN_REPORT_VAR □ Event-info parameter = ScanReportInfoVar □ attribute.identification-field = <Variable> □ Value-length = <Variable> □ Value = <Not relevant for this Test> b. If the data is from multiple persons □ Event-type = MDC_NOTI_SCAN_REPORT_MP_VAR □ Event-info parameter = ScanReportInfoMPVar □ attribute.identification-field = <Variable> □ Value-length = <Variable> □ Value = <Not relevant for this Test>
Pass/Fail criteria	• Variable format event report contains the listed fields.
Notes	

TP Id		TP/PLT/PHD/OXP/SER/BV-005		
TP label		PHD transmits data in a grouped format Event Report (Scanner Objects)		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	FormatEventRep 2; M	PersonEventRep 2; O	FormatEventRep 8; M
		ConfNormalProc 4; M		
Test purpose		Check that: If PHD transmit data in a grouped format, then it reports the scanner object's handle along with the scanned objects' attribute values in the same order and size as specified in the Scan-Handle-Attr-Val-Map [AND] This attribute (Scan-Handle-Attr-Value-Map) shall be defined before grouped event report transfer commences.		
Applicability		(C_AG_OXP_046 OR C_AG_OXP_047) AND C_AG_OXP_048 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_041 , C_AG_OXP_180		
Initial condition		The simulated PHG and PHD under test are in the Configuring state.		
Test procedure		1. When the PHD under test sends its configuration to the simulated PHG, the number of the measurement object (config-obj-list.count=n) must be saved. 2. Every measurement object has an obj-handle and one or more attributes: a. Attribute Scan-Handle-Value-Map □ attribute-id = MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP (0x0A 0x53) □ HandleAttrValMap.count = N (number of object for this measurement object) □ HandleAttrValMap.length = L b. For each attribute (of the L present) its length is needed: □ HandleAttrValMap = <One of an actual metric derived object> □ HandleAttrValMap.count = K (number attributes of this object) □ HandleAttrValMap.length = M		

	3. The sum of the length values is the total length of the measurement data for this object, in this case, the sum of all the M´s 4. When the PHD under test sends an event report to the simulated PHG with a measurement observation, the format of this message is: a. PrstApdu (0xE7 0x00) □ obj-handle = <It has to be the same that obj-handle of the Measurement object sent in the PHD’s configuration> □ event-type = MDC_NOTI_BUF_SCAN_REPORT_GROUPED or MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED □ grouped-length = L <This length has to be the same as the total length of the measurement data for this object, it is the sum calculated when the PHD sent its configuration>
Pass/Fail criteria	• The metric derived objects must appear in the event report in the same order as were declared on the configuration report. The length of the event report must match the length indicated by the Handle-Attribute-Value-Map and cannot exceed the maximum APDU size established by the specialization: ○ Pulse oximeter → 9216 octets ○ Weighing scales → 896 octets ○ Glucose meter → 5120 octets or 64512 octets if the PHD supports PM-Store ○ Blood pressure → 896 octets ○ Thermometer → 896 octets ○ Independent activity hub → 5120 octets ○ Cardiovascular → 64512 octets or 6624 octets if it supports Step Counter Profile ○ Strength → 64512 octets ○ Adherence monitor → 1024 octets ○ Peak flow → 2030 octets ○ Body composition analyser → 7730 octets ○ Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store ○ Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store ○ International normalized ratio → 896 octets or 64512 if the PHD supports PM-Store ○ Insulin pump → 7168 octets or 5120 if PHD supports PM-Store ○ Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store ○ Power status monitor → 940 octets for Simple PSM profile or 1660 for Advanced PSM profile
Notes	

TP Id		TP/PLT/PHD/OXP/SER/BV-007		
TP label		Temporarily Stored Measurements		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	TempStored 1; O	TempStored 2; C	TempStored 3; C
		TempStored 7; R	TempStored 8; M	CommonCharac 3; M
Test purpose		Check that: Only Metric derived objects that are not real time arrays (Numeric and Enumeration objects) are supported as Temporarily Stored Measurements [AND] Temporarily Stored Measurements requires the use of time stamp attributes (Date-and-Time, Relative-Time, HiRes-Relative-Time)		

	[AND] The PHD ensures ownership of the measurements is successfully transferred to the PHG by using confirmed event reports [AND] the PHD does not provide more than 25 Temporarily Stored Measurements in order to limit the amount of data transported by this mechanism [AND] The total size of the response does not exceed the maximum APDU size established by the specialization
Applicability	C_AG_OXP_032 AND C_AG_OXP_000
Other PICS	C_AG_OXP_009, C_AG_OXP_014,C_AG_OXP_041, C_AG_OXP_293
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The PHD under test takes more than 25 measurements before connecting to the simulated PHG. 2. The PHD under test gets connected to the simulated PHG. The Metric-Spec-Small attribute sent in ConfigReport is recorded for numeric and enumeration objects. 3. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 4. Once in the Operating state, check that: a. No more than 25 Temporary Stored Measurements are sent in the same event report. b. That the event reports used to transmit the measurements should be confirmed. c. That every Temporary Stored Measurement sent has a Time Stamp attribute (Date-and-Time, Relative-Time, HIRes-Relative-Time or Base-Offset-Time) d. Data sent is Enumerated or Numeric. e. Metric-Spec-Small – mss-avail-stored-data bit is set.
Pass/Fail criteria	• The conditions in step 4 are met • The total size of the response cannot exceed the sum of the APDU sizes of the supported specializations (limited to an absolute limit of 64512 octets): o Pulse oximeter → 9216 octets o Weighing scales → 896 octets o Glucose meter → 5120 octets or 64512 octets if the PHD supports PM-Store o Blood pressure → 896 octets o Thermometer → 896 octets o Independent activity hub → 5120 octets o Cardiovascular → 64512 octets or 6624 octets if the PHD supports Step Counter Profile o Strength → 64512 octets

	○ Adherence monitor → 1024 octets ○ Peak flow → 2030 octets ○ Body composition analyser → 7730 octets ○ Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store ○ Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store ○ International normalized ratio → 896 octets or 64512 if the PHD supports PM-Store ○ Insulin pump → 7168 octets or 5120 if PHD supports PM-Store ○ Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store ○ Power status monitor → 940 octets for Simple PSM profile or 1660 for Advanced PSM profile
Notes	It is possible that the PHD supports Temporarily Stored Measurements (TSM) for a set of objects and that it does not support TSM for other set of objects. The test tool identifies the objects that support TSM using the Metric-Spec-Small attribute – mss-avail-stored-data(1) bit. When this bit is set to 1 in one object, the test tool assumes that this object stores TSM, although it is not totally sure because this bit is "informational". From [ISO/IEEE 11073-20601-2015A] clause A.11.3: • IF object stores TSM THEN mss-avail-stored-data(1) SHALL be set to 1 • IF object does not store TSM THEN mss-avail-stored-data(1) MAY be set to 1 or 0 If the PHD under test sets mss-avail-stored-data(1) bit to 1 for one object but this object does not store TSM and the PHD does not include the time stamp in event report, the test tool gives a FAIL verdict since the test tool identifies that this object stores TSM. When the vendor implements this behaviour in a device, a waiver may be required to complete the Certification. See Bugzilla #840 and contact Continua TOM for further details (http://continua.plugfests.com/show_bug.cgi?id=840).

A.4 Subgroup 1.2.3 – PHD communication model (COM)

TP Id		TP/PLT/PHD/OXP/COM/BV-003_A		
TP label		Communication Characteristics: Reliable virtual channel 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CommuCharac 2; M	CommuCharac 3; M	
Test purpose		Check that: The “reliable” virtual channel (i.e. a “reliable” transport service) of the Type 1 transport profiles is used for all messages related to the association procedure: aarq, rlre [AND] The “reliable” virtual channel (i.e. a “reliable” transport service) of the Type 1 transport profiles is used for all messages related to the Confirmed service mechanism (prst.roiv-cmip-confirmed-action, prst.roiv-cmip-confirmed-event-report, prst.roiv-cmip-get, prst.roiv-cmip-confirmed-set) (prst.rors-cmip-confirmed-action, prst.rors-cmip-confirmed-event-report, prst.rors-cmip-get, prst.rors-cmip-confirmed-set)		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG sends an Association Response with result = accepted-unknown-config. 3. The PHD under test responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. IF C_AG_OXP_293 THEN:		

	a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 5. Wait until the PHD under test reaches the Operating state. 6. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0 (to request an MDS object) and an empty attribute-id-list to indicate all attributes. 7. The PHD responds with a "rors-cmip-get". 8. IF C_AG_OXP_041 THEN a. The simulated PHG sends a Get request for the PM-Store with an attribute-id-list set to 0 to indicate all PM-Store attributes. b. The PHD under test issues a GET response. c. The simulated PHG shall send a Get-Segment-Info object action for the PM-Segment object with SegmSelection = all-segments to indicate the PM-Segments attributes of all available PM-Segments. d. The PHD under test issues a "rors-cmip-confirmed-action" response. 9. IF (C_AG_OXP_046 OR C_AG_OXP_047) THEN a. The simulated PHG sends a "roiv-cmip-confirmed-set" to set the OperationalState of the scanner object to 1. b. The PHD under test responds with a "rors-cmip-set". 10. The simulated PHG sends a Release Request to the PHD under test with reason = normal(0). 11. The PHD under test responds with a Release Response.
Pass/Fail criteria	The "reliable" virtual channel must be used in steps 1, 3, 7, 8.b, 8.d, 9.b and 11.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-003_B		
TP label		Communication Characteristics: Reliable virtual channel 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CommuCharac 2; M		
Test purpose		Check that: The “reliable” virtual channel (i.e. a “reliable” transport service) of the Type 1 transport profiles is used for all messages related to the association procedure: rlrq		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The PHD under test sends an Association Request from the PHD under test. 2. The simulated PHG sends an Association Response with result = accepted-unknown-config. 3. The PHD under test responds with a "Remote Operation Invoke | Confirmed Event		

	Report" message with an MDC_NOTI_CONFIG event to send its configuration to the simulated PHG. 4. The simulated PHG responds with a "unsupported-config", waits for a new configuration and keeps responding with "unsupported-config" to every new configuration. 5. The PHD under test sends a Release Request.
Pass/Fail criteria	The "reliable" virtual channel must be used for the Release Request.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-003_C		
TP label		Communication Characteristics: Reliable virtual channel 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CommuCharac 2; M		
Test purpose		Check that: The “reliable” virtual channel (i.e. a “reliable” transport service) of the Type 1 transport profiles is used for all messages related to the association procedure: abrt		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends a Release Response to the PHD under test. 2. The PHD responds with an Abort message.		
Pass/Fail criteria		The "reliable" virtual channel must be used for the Abort message.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-003_D		
TP label		Communication Characteristics: Reliable virtual channel 4		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	CommuCharac 4; M		
Test purpose		Check that: The “reliable” virtual channel (i.e. a “reliable” transport service) of the Type 1 transport profiles is used for all messages related to fault or abnormal conditions: roer		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends a roiv-cmip-confirmed-action with action-type = MDC_ACT_DATA_REQUEST. 2. The PHD under test shall reply with a "roer" with reason = no-such-action (9).		
Pass/Fail criteria		The "reliable" virtual channel must be used for the "roer" message.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-006_A		
TP label		Agent State machine. Accepted known configuration		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 21; M		

Test purpose	Check that: If aare(accepted) is received while in the Associating state, then the PHD under test moves to the Operating state.
Applicability	C_AG_OXP_000 AND (C_AG_OXP_291 OR C_AG_OXP_292)
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. The simulated PHG receives an association request from the PHD under test (the PHD passes to the Associating state). 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD under test responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the simulated PHG. The configuration report is reported. 4. The simulated PHG sends an abort message. 5. The simulated PHG and the PHD move to the Unassociated state. 6. The simulated PHG receives an association request from the PHD under test. 7. The simulated PHG responds with a result = accepted. 8. The PHD must change to the Operating state. • IF the PHD supports the Scanner object: The simulated PHG sends a Set command for the Scanner object and the PHD shall reply. • ELSE IF the PHD under test supports PM-Store, the simulated PHG sends a Get Segment Info action and the PHD shall reply. • ELSE the simulated PHG waits to receive measurements from the PHD.
Pass/Fail criteria	The PHD under test has passed to the Operating state after the last step.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-006_B		
TP label		Agent State machine		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 64; M	ObjAccessServ 4; C	
Test purpose		Check that: If roiv-* is received while in the Operating state, then the PHD transmits a (rors-*, roer-*, or rorj-*) and remains in the same state. [AND] If an error occurs in executing a confirmed action, then the error shall be indicated by returning an error (roer) with an appropriate error value and, where appropriate, additional information on the error may be included in the parameter field using one of the return codes from the return codes partition.		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_071, C_AG_OXP_180		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends a "roiv-cmip-get" to the PHD, to get all the attributes for an MDS object. 2. The PHD responds with a "rors-cmip-get" message. 3. IF C_AG_OXP_180 THEN the simulated PHG sends a "roiv-cmip-confirmed-set", setting to default value the attribute Operational State for a scanner object. IF NOT C_AG_OXP_180 THEN the simulated PHG sends a roiv-cmip-set, setting to default value the attribute Operational State for a scanner object. 4. IF C_AG_OXP_180 THEN the PHD responds with a rors-cmip-confirmed-set if it		

	supports a scanner object, otherwise it responds with a roer-* or rorj-*. IF NOT C_AG_OXP_180 the PHD does not respond if it supports a scanner object, otherwise it responds with a roer-* or rorj-*. 5. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC-ACT_DATA_REQUEST. 6. The PHD responds with a "rors-cmip-confirmed-action", "roer-*" or "rorj-*". 7. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC_ACT_SET_TIME. 8. The PHD responds with a "rors-cmip-confirmed-action", "roer-*" or "rorj-*". 9. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC_ACT_SEG_GET_INFO. 10. The PHD responds with a "rors-cmip-confirmed-action" if the PM-Store object is supported by the PHD, otherwise it responds with a "roer-*" or "rorj-*". 11. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC_ACT_SEG_TRIG_XFER. 12. The PHD responds with a "rors-cmip-confirmed-action" if the PM-Store object is supported by the PHD, otherwise, a "roer-*" or "rorj-*". 13. The simulated PHG sends a "roiv-cmip-confirmed-action", action-type MDC_ACT_SEG_CLR (all-segments). If the PHD supports Clear-Segment action, THEN the test tool performs a GET request to read the Clear-Timeout attribute for Pm-Store. 14. The PHD responds with a "rors-cmip-confirmed-action" if the PM-Store object is supported by the PHD and it supports Clear-Segment action, otherwise, a "roer-*" or "rorj-*".
Pass/Fail criteria	• The PHD replies with messages specified in steps 2, 4, 6, 8, 10, 12 and 14 of the test procedure. • If the PHD sends a roer message, check that the error value is correct and that a parameter may be included.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-007		
TP label		Agent State machine. Accepted unknown configuration		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 22; M	AgentStateMach 53; M	AssocResp 6; M
		ConfProc 1; M	ConfExitCond 3; M	
Test purpose		Check that: If aare(accepted-unknown-config) is received while in the Associating state, then the PHD moves to "Sending Config" state. The PHG has accepted the association but it does not have a configuration. [AND] When a PHD receives a response that the configuration is unknown, it moves to the Configuring state and follow the procedures specified to transfer its configuration [AND] If a rors-cmip-confirmed-event-report (accepted-config) is received while in the Waiting Approval state, then the PHD moves to the Operating state.		
Applicability		C_AG_OXP_000 AND (C_AG_OXP_291 OR C_AG_OXP_292)		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state		
Test procedure		1. The simulated PHG receives an association request from the PHD under test (the PHD passes to the Associating state). 2. The simulated PHG responds with a result = accepted-unknown-config.		

	3. The PHD under test shall go to the "Sending Config" substate, and responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG (the PHD shall go to the "waiting-approval" substate). 4. The simulated PHG sends a "Remote Operation Response | Confirmed Event Report" with result "accepted-config". 5. The PHD under test shall move to the Operating state. • IF the PHD under test supports the Scanner object: The simulated PHG sends a Set command for the Scanner object and the PHD under test shall reply. • ELSE IF the PHD under test supports PM-Store the simulated PHG sends a Get Segment Info action and the PHD shall reply. • ELSE the simulated PHG waits for receiving measurements from the PHD under test.
Pass/Fail criteria	The PHD under test has passed to the Operating state after the last step.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-009		
TP label		Agent State machine. Leaving the Operating State 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 59; M		
Test purpose		Check that: If aarq is received while in the Operating state, then the PHD transmits an abrt(Abort-reason undefined) and moves to Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends an Association Request to the PHD under test. 2. The PHD under test responds with an Abort message abrt(Abort-reason undefined).		
Pass/Fail criteria		• The PHD transmits correctly the Abort message (abrt) with reason undefined and changes to the Unassociated state. • The simulated PHG must not receive any message other than an Association Request after step 2.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-010		
TP label		Agent State machine. Leaving the Operating State 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 60; M		
Test purpose		Check that: If aare is received while in Operating state, then the PHD transmits an abrt(Abort-reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends an Association Response. 2. The PHD responds with an Abort message abrt(Abort-reason undefined).		

Pass/Fail criteria	- The PHD transmits correctly the Abort message (abrt) with reason undefined and changes to the Unassociated state. - The simulated PHG must not receive any message other than an Association Request after step 2.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-011		
TP label		Agent State machine. Leaving the Operating State 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 61; M		
Test purpose		Check that: If rlrq is received while in the Operating state, then the PHD transmits an rlre (normal) and moves to the Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends an Association Release Request (Rlrq) message to the PHD under test, with reason =0 (normal). 2. The PHD under test shall respond with an Association Release Response (Rlre) message with reason =0 (normal) and shall go to the Unassociated state.		
Pass/Fail criteria		The PHD under test transmits correctly the Rlre message.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-012		
TP label		Agent State machine. Leaving the Operating State 4		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 62; M		
Test purpose		Check that: If rlre is received while in the Operating state, then the PHD transmits an abrt(Abort-reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG sends an Association Release Response to the PHD under test, with reason =0 (normal) 2. The PHD under test responds with an Abort message abrt(Abort-reason undefined).		
Pass/Fail criteria		- The PHD under test transmits correctly the Abort message abrt(Abort-reason undefined) and changes to the Unassociated state. - The simulated PHG must not receive any message other than an Association Request after step 2.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-013
TP label		Agent State machine. Association timeout
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]

	Testable items	AgentStateMach 2; M	AgentStateMach 16; M	AgentStateMach 17; M
		AssocErrorCond 1; M	AssocErrorCond 2; M	AssocErrorCond 3; M
Test purpose		Check that: If timeout and maximum retry limit are not reached while in the Associating state, then the PHD continues transmitting aarq [AND] In the case of timeout, the PHD attempts to associate up to the maximum retry count is reached or association is successful. [AND] If timeout and maximum retry limit are reached when sending aarq, then the PHD transmits an abort message abrt(Abort-reason response-timeout) and moves to the Unassociated state. [AND] If the TOassoc period expires, the PHD shall re-transmit the Association Request message with a new TOassoc period. This process shall be repeated until an Association Response is received or RCassoc (retry count: association procedure) attempts have been made after the first timeout, whichever comes first. This results in a maximum of RCassoc + 1 Association Requests		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test (and PHD under test passes to the Associating state). 2. The simulated PHG does NOT respond with any message. 3. The PHD under test shall wait for the TO$_{assoc}$ timer to expire (10 seconds) and retransmit a new association request. 4. Steps 2-3 shall be repeated until the Retry Count has reached (=3). 5. As the PHG has not answered to any of the 4 messages, the PHD under test shall send an abort message abrt(Abort-reason response-timeout) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		The TOassoc timer and the RC values are properly implemented and in the last step the PHD under test shall transmit an Abort message (abrt) with reason response-timeout.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-020		
TP label		Agent State machine. Connected Associating 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 20; M		
Test purpose		Check that: If aarq is received while in Associating state, then the PHD transmits an aare(rejected permanent) and moves to the Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected Associating state.		
Test procedure		1. The simulated PHG issues an Association Request 2. The PHD under test sends an Association Response message to the PHG: • reason = rejected-permanent(1), • data-proto-id=data-proto-id-empty • data-proto-info=omit		

Pass/Fail criteria	• The PHD sends the detailed AARE message and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-021		
TP label		Agent State machine. Connected Associating 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 24; M		
Test purpose		Check that: If rlrq is received during Associating state, then the PHD transmits an abort message abrt(Abort-reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected Associating state.		
Test procedure		1. The simulated PHG issues a Release Request. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-022		
TP label		Agent State machine. Connected Associating 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 25; M		
Test purpose		Check that: If rlre is received during association state, then the PHD transmits an abrt(Abort-reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected Associating state.		
Test procedure		1. The simulated PHG issues a Release Response with reason = normal(0). 2. The PHD under test sends an abort message (Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (Abort-reason undefined) and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id	TP/PLT/PHD/OXP/COM/BV-023
TP label	Agent State machine. Connected Associating 4

Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 27; M		
Test purpose		Check that: If prst (any APDU not covered in 3.* (corrupt, unknown, unexpected, etc.) is received during Associating state, then the PHD transmits an abort message abrt (Abort-reason undefined) and moves to the Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected Associating state.		
Test procedure		1. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0 (to request the MDS object) and the attribute-idlist set to "all-attributes". 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-023_A		
TP label		Agent State machine. Connected Associating 5		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 27; M		
Test purpose		Check that: If prst (any APDU not covered in 3.* (corrupt, unknown, unexpected, etc.) is received during Associating state, then the PHD transmits an abort message abrt (Abort-reason undefined) and moves to the Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected Associating state.		
Test procedure		1. The simulated PHG sends a badly formatted message. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-030		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 45; M		
Test purpose		Check that: If aarq is received while in the Waiting Approval state, then the PHD transmits an abrt (reason undefined) and moves to Unassociated state		

Applicability	C_AG_OXP_000
Other PICS	
Initial condition	The PHD under test is in the Waiting Approval state.
Test procedure	1. The simulated PHG issues an Association Request. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.
Pass/Fail criteria	• The PHD sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-031		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 46; M		
Test purpose		Check that: If aare is received while in Waiting Approval state, then the PHD transmits an abrt (reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the Waiting Approval state.		
Test procedure		1. The simulated PHG issues an Association Response with reason = accepted(0). 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-032		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 47; M		
Test purpose		Check that: If rlrq is received while in Waiting Approval state, then the PHD transmits a rlre and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the Waiting Approval state.		
Test procedure		1. The simulated PHG issues a Release Request. 2. The PHD under test sends a Release Response to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the Release Response message and changes to the		

	Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-033		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 4		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 48; M		
Test purpose		If rlre is received while in Waiting Config state, then the PHD transmits an abrt (reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the Waiting Approval state.		
Test procedure		1. The simulated PHG issues a Release Response with reason = normal(0). 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • Simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-034		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 5		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 50; M	OperNormProc 5; R	
Test purpose		Check that: If roiv-cmip-get, handle=0 is received while in Waiting Approval state, then the PHD transmits an rors-cmip-get with the MDS attributes or roer not-allowed-by-object if request is not for all attributes and PHD does not support the request and PHD remains in Waiting Approval state. [AND] If the PHG requests specific MDS object attributes, as indicated by the elements in attribute-id-list, and if this capability is not implemented, then the PHD shall respond with an error (roer) message with an error-value of not-allowed-by-object,		
Applicability		C_AG_OXP_000 AND (C_AG_OXP_291 OR C_AG_OXP_292)		
Other PICS		C_AG_OXP_100		
Initial condition		The PHD under test is in the connected associated configuring Waiting Approval state.		
Test procedure		1. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0 (to request the MDS object) and the attribute-id list set to MDC_ATTR_SYS_ID. 2. The PHD under test responds with a "rors-cmip-get" message or a roer message (not-allowed-by-object). 3. The PHD under test remains in Waiting Approval state. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result "unsupported-config". 5. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG or Rlrq (no-more-		

	configurations).
Pass/Fail criteria	The process detailed above must be successfully completed.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-034_A		
TP label		Agent State machine. Get Request Sending Configuring		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 37; M		
Test purpose		Check that: If roiv-cmip-get, handle=0 is received while in Sending Config state, then the PHD transmits an rors-cmip-get with the MDS attributes		
Applicability		C_AG_OXP_000 AND (C_AG_OXP_291 OR C_AG_OXP_292)		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responses with an accepted-unknown-config. 3. The simulated PHG issues "roiv-cmip-get" command with the handle set to 0 (to request the MDS object) and the attribute-idlist set to "all-attributes". 4. The PHD under test responds with a "rors-cmip-get" message in which the attribute=list contains a list of all implemented attributes of the MDS object. 5. The PHD under test must send its configuration.		
Pass/Fail criteria		The process detailed above must be completed.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-035		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 6		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 51; M		
Test purpose		Check that: If roiv-* but not (roiv-cmip-get, handle=0) is received while in Waiting Approval state, then the PHD transmits a roer (no-such-object-instance) and remains in Waiting Approval state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected associated configuring Waiting Approval state.		
Test procedure		1. The simulated PHG issues a GET with handle = 1. 2. The PHD under test must send a "roer" with reason = no-such-object-instance(1). 3. The PHD under test remains in the Waiting Approval state. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result "unsupported-config". 5. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG or Rlrq (no-more-configurations).		
Pass/Fail criteria		The process detailed above must be successfully completed.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-036_A		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 7		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 54; M		
Test purpose		Check that: If rors-*, roer-*, or rorj-*,but not rors-cmip-confirmed-event-report is received while PHD is in "Waiting Approval" state, an PHD shall transmit an abrt(reason undefined) and move to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected associated configuring Waiting Approval state.		
Test procedure		1. The simulated PHG issues a Prst message, rors-cmip- get. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-036_B		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 8		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 54; M		
Test purpose		Check that: If rors-*, roer-*, or rorj-*,but not rors-cmip-confirmed-event-report is received while PHD is in "Waiting Approval" state, an PHD shall transmit an abrt(reason undefined) and move to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the connected associated configuring Waiting Approval state.		
Test procedure		1. The simulated PHG issues a roer message. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-036_C		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 9		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 54; M		
Test purpose		Check that:		

	If rors-*, roer-*, or rorj-*, but not rors-cmip-confirmed-event-report is received while the PHD is in "Waiting Approval" state, a PHD shall transmit an abrt(reason undefined) and move to the Unassociated state.
Applicability	C_AG_OXP_000
Other PICS	
Initial condition	The PHD under test is in the connected associated configuring Waiting Approval state.
Test procedure	1. The simulated PHG issues a rorj message. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.
Pass/Fail criteria	• The PHD under test sends the Abort message (abrt) with reason undefined and changes to Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-037		
TP label		Agent State machine. Connected Disassociation 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 70; M		
Test purpose		Check that: If aarq is received while in the Disassociating state, then the PHD transmits an abrt (Abort-reason undefined) and moves to Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD is in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responds with an accepted-unknown-config. 3. The PHD sends a configuration event report. 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD sends a new configuration event report with a new configuration (if it has more). 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a ReleaseRequest with reason "no-more-configurations". The PHD moves to the Disassociating state. 7. The simulated PHG sends an AARQ message. 8. The PHD responds with an Abort message (abrt) with reason undefined. 9. The PHD and the PHG move to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 9		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-038		
TP label		Agent State machine. Connected Disassociation 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 71; M		
Test purpose		Check that: If aare is received while in Disassociating state, the PHD shall transmit an abrt (reason undefined) and move to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD is in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responds with an accepted-unknown-config. 3. The PHD sends a configuration event report. 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD sends a new configuration event report with a new configuration (if it has more). 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a ReleaseRequest with reason "no-more-configurations". The PHD moves to the Disassociating state. 7. The simulated PHG sends an AARE message. 8. The PHD responses with an Abort message (abrt) with reason undefined. 9. The PHD and the PHG move to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (abrt) with reason undefined and changes to Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 9		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-039		
TP label		Agent State machine. Connected Disassociation 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 72; M		
Test purpose		Check that: If rlrq is received while in Disassociating state, the PHD shall transmit a rlre (normal) and remain in the same state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD is in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responses with an accepted-unknown-config. 3. The PHD sends a configuration event report. 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD sends a new configuration event report with a new configuration (if it has more). 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a		

	ReleaseRequest with reason "no-more-configurations". The PHD moves to the Disassociating state. 7. The simulated PHG sends a Rlrq message (reason=normal). 8. The PHD responses with a Rlre message. 9. The PHD and PHG remain in the same state.
Pass/Fail criteria	• The PHD sends the Rlre message and remain in the same state • The simulated PHG must not receive any message other than an Association Request after step 9
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-040_A		
TP label		Agent State machine. Connected Disassociation 4		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 76; M		
Test purpose		Check that: If roer is received while in the Disassociating state, then the PHD transmits an abrt (Abort-reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD is in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responses with an accepted-unknown-config. 3. The PHD sends a configuration event report. 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD sends a new configuration event report with a new configuration (if it has more). 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a ReleaseRequest with reason "no-more-configurations". The PHD moves to the Disassociating state. 7. The simulated PHG sends a Roer message. 8. The PHD responds with an Abort message (reason undefined). 9. The PHD and the PHG move to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort (reason undefined) message and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 9		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-040_B		
TP label		Agent State machine. Connected Disassociation 5		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 76; M		
Test purpose		Check that: If rorj is received while in the Disassociating state, then the PHD transmits a abrt (reason undefined).and moves to the Unassociated state.		
Applicability		C_AG_OXP_000		

Other PICS	
Initial condition	The PHD is in the Disassociating state.
Test procedure	1. The simulated PHG sends an Rorj message. 2. The PHD responses with an Abort message (reason undefined). 3. The PHD and the PHG move to the Unassociated state.
Pass/Fail criteria	• The PHD sends the Abort message (reason undefined) and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 9
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-042		
TP label		Association request format		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AssocRequest 3; C	AssocRequest 4; C	MessageEncod 1;M
		AssocRequest 6; M		
	Spec	[b-ITU-T H.810 (2015)]		
	Testable items	General 3; M	Dev.Information 9; M	
Test purpose		Check that: If a PHD sets the data-proto-id to data-proto-id-20601, then it adheres to the abstract syntax definitions specified for data types and message exchange. [AND] The data-proto-info field is filled in with a PhdAssociationInformation structure which defines the following information: - The version of the data exchange protocol. - The specific DataApdu encoding rule(s) supported by the PHD. The PHD sets one or more of the encoding-rules bits. - The PHD always supports MDER. That is, the mder bit of encoding-rules is set by the PHD. - The PHD may offer other encoding rules, besides MDER, to the PHG by setting other bits in the encoding-rules. - The version of the nomenclature used a field indicating all functional units and optional features supported by the PHD. - The system type (PHD in this case). - A unique System-Id of the PHD. The PHD uses EUI-64 to identify itself. - A dev-config-id, which identifies the current configuration of the PHD. - A data-req-mode-capab, which defines the data request modes supported by the PHD. - An option-list that contains a list of additional attributes the PHD wishes to communicate. [AND] The PHD shall place at most one data-proto element containing the field data-proto-id set to data-proto-id-20601 in the data-proto-list. [AND] Continua service components supporting a device specialization other than the basic electrocardiograph (ECG) shall set only the version 1 bit in the protocol version field of the PHDAssociationInformation structure in the AARQ.		
Applicability		C_AG_OXP_000		

Other PICS	C_AG_OXP_002
Initial condition	The simulated PHG and PHD under test are in the Unassociated state.
Test procedure	1. The PHD under test sends an AARQ message to the simulated PHG. The expected fields sent by the PHD are: a. APDU Type ❑ field-length =2 bytes ❑ field-value =0xE2 0x00 (AareApdu) b. The following two bytes indicate the length of the message. c. assoc-version ❑ field-type = AssociationVersion ❑ field-length =BITS-32 ❑ field-value = ▪ Only one bit can be set d. The following four bytes indicate: ❑ data-proto-list.count (two bytes) = At most there is a data-proto-id set to data-proto-id-20601 ❑ Length of the message (two bytes) e. data-proto-id ❑ field-type = DataProtoId ❑ field-length =INT-U16 ❑ field-value = 0x50 0x79 (20601) ❑ data-proto-id=20601 indicates exchange protocol follows this standard, and data-proto-info shall contain PhdAssociationInformation. f. The DataProto.Info field must contain two bytes and indicates the data-proto-info.length g. protocol-version ❑ field-type = Protocol Version ❑ field-length =BITS-32 ❑ IF the PHD supports insulin pump (IP) (C_AG_OXP_158 = TRUE) THEN ▪ field-value = At least bit protocol-version3(2) is set to 1 (0x20 0x00 0x00 0x00 OR 0xA0 0x00 0x00 0x00 OR 0x60 0x00 0x00 0x00 OR 0xE0 0x00 0x00 0x00) ▪ This value shows that version 3 of the data exchange protocol is supported (protocol-version3(2)=1). ❑ IF the PHD supports power status monitor (PSM) with Simple PSM or Advanced PSM profile (C_AG_OXP_270 = TRUE OR C_AG_OXP_271 = TRUE) THEN ▪ field-value = At least bit protocol-version3(2) and bit protocol-version2(1) are set to 1 (0x60 0x00 0x00 0x00 OR 0xE0 0x00 0x00 0x00) ▪ This value shows that version 2 and 3 of the data exchange protocol are supported (protocol-version3(2)=1 protocol-version2(1)=1). ❑ IF PHD supports basic electrocardiograph (ECG) or international normalized ratio (INR) or continuous glucose monitor (CGM) device specialization (C_AG_OXP_165 = TRUE OR C_AG_OXP_164 = TRUE OR C_AG_OXP_163 = TRUE OR C_AG_OXP_157 = TRUE) THEN ▪ field-value = At least bit protocol-version2(1) is set to 1 (0x40 0x00 0x00 0x00 OR 0xC0 0x00 0x00 0x00) ▪ This value shows that version 2 of the data exchange protocol is supported (protocol-version2(1)=1). ELSE

	▪ field-value = 0x80 0x00 0x00 0x00 ▪ This value shows that version 1 of the data exchange protocol is supported (protocol-version1(0)=1). h. encoding rules □ field-type = EncodingRules □ field-length = BITS-16 □ field-value= depends on the encoding rules supported/selected. ▪ Bit 0 (mder) must always be set ▪ and xer(1) or/and per(2) may be set (optional). i. nomenclature version □ field-type = NomenclatureVersion □ field-length =BITS-32 □ field-value = 0x80 0x00 0x00 0x00 □ This value indicates version 1 is supported (nom-version1(0) is set). j. functional–units □ field-type = FunctionalUnits □ field-length = BITS-32 □ filed-value = ▪ Bit 0 must be 0 ▪ Bits 1 and 2 may be set ▪ The rest of the bits must not be set k. system type □ field-type = SystemType □ field-length = BITS-32 □ field- value = 0x00 0x80 0x00 0x00 (sys-type-agent) l. system-id □ field-type = OCTET STRING □ field-length = 0x00 0x0A □ field-value = 0x00 0x0X 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX (octet string length = 8 | EUI-64 manufacturer and device) □ This value will be System Id attribute of MDS Object. m. dev-config-id □ field-type = ConfigId □ field-length = INT-U16 □ field-value = <Not relevant for this Test> n. Data-Req-Mode-Capab: □ field-type = DataReqModeCapab □ field-length = INT-U16 □ field-value = SEQUENCE { ▪ data-req-mode-flags DataReqModeFlags, ▪ data-req-init-agent-count INT-U8, -- maximum number of parallel agent-initiated data requests ▪ data-req-init-manager-count INT-U8, -- maximum number of parallel manager initiated data requests o. option-list:

		❑ field-type: AttributeList		
Pass/Fail criteria		The structure and values of the association request message is correct.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-049		
TP label		Configuring Procedure 4		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ConfErrorCond 1; M	ConfErrorCond 2; M	
Test purpose		Check that: In the Configuring state, the PHD waits for the “Remote Operation Response | Confirmed Event Report | MDC_NOTI_CONFIG” message for an TO$_{config}$ period. [AND] If the TO$_{config}$ period expires, then the PHD sends an Association Abort message to the PHG and transition back to the Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test (the PHD passes to the Associating state). 2. The simulated PHG responds with an Association Response with result = "accepted-unknown-config". 3. The PHD under test sends a configuration event report. 4. The simulated PHG does not respond to the configuration event report for more than TO$_{config}$ time.		
Pass/Fail criteria		The PHD must wait for a TO$_{config}$. If the time expires, the PHD must send an abort to the PHG.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-051		
TP label		Non-standard configuration, Dev-config-id		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ConfNormalProc 20; C	ConfEventRep 21; M	
Test purpose		Check that: A PHD that has a non-standard configuration assigns a unique identifier to its configuration by generating a value for dev-config-id in the range between extended-config-start and extended-config-end, inclusive.		
Applicability		C_AG_OXP_181 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test with a dev-config-id and a system-id. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result "accepted" if the Config-Id received in step 3 is the configuration that is being tested, in		

	this case ConfigReport is recorded. Otherwise, the PHG responds with a rors-cmip-confirmed-event-report with result "unsupported-config" and step 3 is repeated. 5. Check the config-report-id in the ConfigReport recorded in step 4.
Pass/Fail criteria	The config-report-id value checked in step 5 is in the range between the extended-config-start (16384) and the extended-config-end (32767), inclusive.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-052_B		
TP label		Operating procedures. Specific Attributes request		
Coverage	**Spec**	[b-ITU-T H.810 (2015)]		
	Testable items	OperNormProc 4; O	OperNormProc 5; R	
Test purpose		Check that The PHD under test supports retrieval of a specific list of attributes [AND] If the PHG requests specific MDS object attributes, indicated by the elements in attribute-id-list, and the PHD supports this capability, then the PHD shall respond with a rors-cmip-get message in which the attribute-list contains a list of the requested attributes of the MDS object that are implemented. It is not required for an PHD to support this capability. If this capability is not implemented then the PHD shall respond with an error (roer) message with an error-value of not-allowed-by-object,		
Applicability		C_AG_OXP_000		
Other PICS		C_AG_OXP_100		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG issues a "Remote Operation Invoke | Get" command with: a. Obj-handle set to 0 (to request an MDS object) b. attribute-id-list.count=1 and a single AVA_Type MDC_ATTR_DEV_CONFIG_ID (0X0A 0X44) to retrieve the mandatory "Dev-Configuration-Id" attribute 2. The PHD under test responds with: • IF C_AG_OXP_100 THEN: with a "rors-cmip-get" service message which contains the "Dev-Configuration-Id" • ELSE: with a "roer" service message with error-value set to not-allowed-by-object (24) 3. The simulated PHG issues a "Remote Operation Invoke | Get" command with: a. Obj-handle set to 0 (to request an MDS object) b. attribute-id-list empty to request all the attributes of MDS 4. The PHD responds with a "rors-cmip-get" service message which contains all the supported attributes of the MDS. 5. The simulated PHG issues a "Remote Operation Invoke | Get" command with a. Obj-handle set to 0 b. attribute-id-list set to an attribute NOT supported by the PHD 6. The PHD responds with a "rors-cmip-get" service message: • IF C_AG_OXP_100 THEN: attribute-list must be empty • ELSE: with a "roer" service message with error-value set to not-allowed-by-object (24) 7. The simulated PHG issues a "Remote Operation Invoke | Get" command with a. Obj-handle set to 0 b. attribute-id-list contains one supported attribute and one unsupported attribute 8. The PHD responds with a "rors-cmip-get" service message:		

	• IF C_AG_OXP_100 THEN: attribute-list must contain the supported attribute • ELSE: with a "roer" service message with error-value set to not-allowed-by-object (24)
Pass/Fail criteria	• In step 2 the PHD properly sends the requested attribute or the error (not-allowed-by-object) • In steps 6 and 8 the received attribute list must be empty if NOT C_AG_OXP_100 or the roer if the action is not supported
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-053		
TP label		Operating procedures. Agent-initiated transmission 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MeasureDataTransf 4; C	MeasureDataTransf 5; C	MeasureDataTransf 6; C
Test purpose		Check that: The PHD indicates that support Agent-Initiated measurements via the DataReqModeCapab structure or the PHD has one or more instances of a Scanner object in the PHD's configuration [AND] The PHD uses the Event Report Service to send a spontaneous measurement to the PHG without being requested by the PHG first. [AND] The PHD uses for this purpose a DataApdu message in a "Remote Operation Invoke | Event Report" command and one of the MDC_NOTI_SCAN_REPORT_* event-types		
Applicability		C_AG_OXP_000 AND (C_AG_OXP_182 OR C_AG_OXP_183 OR C_AG_OXP_184 OR C_AG_OXP_189)		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the disconnected Operating state.		
Test procedure		1. The PHD under test must send an Association Request to the simulated PHG which contains the DataReqModeFlags field (of the DataReqModeCapab attribute). 2. Check the value of the bit 15 (data-req-supp-init-agent). Check that if the bit is not set, there is at least one Scanner object in the PHD under test. 3. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: ❑ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. ❑ Once its internal time setting operation is completed, the PHD responds to the PHG. 4. Once the device is in the Operating state take a measurement and check that, if the bit was set, the PHD under test sends the measurement value to the simulated PHG without the PHG requesting it using a "Remote Operation Invoke | Confirmed Event Report" or a "Remote Operation Invoke | Event Report" message with one of the MDC_NOTI_SCAN_REPORT_* event-types. Record the scan-report-no for later comparison and check data-req-id.		

	5. Take another measurement, record the scan-report-no of the event and check data-req-id.
Pass/Fail criteria	The PHD is able to send agent-initiated measurement reports, uses a correct event-type for doing so and the scan-report-no of the second event has increased once unit Data-req-id is set to data-req-id-agent-initiated (61440).
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-053_A		
TP label		Operating procedures. Invoke-id		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 50;M		
Test purpose		Check that: In the remote operation invoke messages (roiv-*), invoke-id is an opaque handle that allows the sender of the message to identify the associated response message (if any). Since the handle is opaque the receiver can make no other assumptions about invoke-id.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0, an empty attribute-id-list to indicate all attributes and invoke-id =20. Record the invoke-id of the message sent. 2. The PHD responds with a "rors-cmip-get" service and the invoke id is 20. 3. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0, an empty attribute-id-list to indicate all attributes and invoke-id =15. Record the invoke-id of the message sent. 4. The PHD responds with a "rors-cmip-get" service and the invoke id is 15. 5. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0, an empty attribute-id-list to indicate all attributes and invoke-id =30. Record the invoke-id of the message sent. 6. The PHD responds with a "rors-cmip-get" service and the invoke id is 30. 7. The simulated PHG issues a "roiv-cmip-get" command with the handle set to 0, an empty attribute-id-list to indicate all attributes and invoke-id =20. Record the invoke-id of the message sent. 8. The PHD responds with a "rors-cmip-get" service and the invoke id is 20.		
Pass/Fail criteria		In steps 2, 4, 6 and 8, the invoke–id has the correct value.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-054		
TP label		Agent-initiated transmission.Scan-report-no		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MeasureDataTransf 46; M		
Test purpose		Check that: An agent-initiated transfer from the MDS or scanner objects, by way of contrast, establishes a flow that terminates only when the association is broken. Thus for the agent-initiated transfer, the scan-report-no starts at 0,		
Applicability		C_AG_OXP_000 AND (C_AG_OXP_182 OR C_AG_OXP_183 OR C_AG_OXP_184 OR C_AG_OXP_189 OR C_AG_OXP_046 OR C_AG_OXP_047)		
Other PICS		C_AG_OXP_180		

Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. If the PHD supports the scanner object, the simulated PHG sends a Set action to set the Operational-State of the scanner to 1 (enabled). 2. Check that the first MDS-Event-Report, for the Metric object or Scanner object, scan-report-no starts at 0.
Pass/Fail criteria	In step 2, the scan-report-no shall be 0.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-054_A		
TP label		Agent-initiated transmission.Separate scan-report-no		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2016C]		
	Testable items	MeasureDataTransf 52; M		
Test purpose		Check that: There will be a separate scan-report-no for confirmed (data-req-id 0xF000) and unconfirmed (data-req-id 0xF001) scan event reports.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_053 AND C_AG_OXP_293 AND NOT C_AG_OXP_268 AND NOT C_AG_OXP_269		
Other PICS				
Initial condition		The simulated PHG and the PHD under test are in the Operating state.		
Test procedure		1. PHD sends some measurements using unconfirmed event reports 2. PHD sends some measurements using confirmed event reports		
Pass/Fail criteria		In step 2, verify that scan-report-no sequences from unconfirmed and confirmed event reports are independent.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-056		
TP label		Operating procedures. Agent-initiated transmission. Scanner objects		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	MeasureDataTransf 9; M	MeasureDataTransf 10; C	MeasureDataTransf 47; M
		ScanClassAttr 3; M		
Test purpose		Check that: Scanner objects begin with Operational-State disabled on PHD with bi-directional communication until the PHG enables it [AND] The data-req-id field in the Scan Report is set to data-req-id-agent-initiated. [AND] If the Scanner's Operational-State attribute is set to disabled halts transmission of Event Reports, then the scan-report-no will continue counting where it was halted before [AND] This attribute (Operational-State) indicates if the scanner is sending event reports or not. If the scanner is sending event reports, the attribute value shall be set to enabled; otherwise, it shall be set to disabled.		
Applicability		(C_AG_OXP_046 OR C_AG_OXP_047) AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014,C_AG_OXP_180, C_AG_OXP_293		
Initial condition		The simulated PHG is in the Waiting Config state and the PHD under test is in the Sending Config state.		

Test procedure	1. The PHD under test must send its configuration to the PHG. The scanner object must have the Operational-State set to 0. 2. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 3. The simulated PHG sends a Set action to set the Operational-State of the scanner to 1 (enabled): a. APDU □ Type = Remote Operation Invoke | Confirmed Event Report □ roiv-cmip-confirmed-set □ attribute = OperationalState □ value = 1 4. Several measurements are taken with the PHD under test. The Data-req-id field will be checked 5. Once the PHD under test starts to transmit its data, the PHG sets the Operational-State to 0: a. APDU □ Type = Remote Operation Invoke | Confirmed set □ roiv-cmip-confirmed-set □ attribute = OperationalState □ value = 0 6. The PHD must stop sending its data. Record the last scan-report-no. 7. The simulated PHG resumes the PHD data transmission by setting the Operational-State back to 1: a. APDU □ Type = Remote Operation Invoke | Confirmed set □ roiv-cmip-confirmed-set □ attribute = OperationalState □ value = 1 8. Several measurements are taken with the PHD under test. Check that the scan-report-no starts counting where it halted before (step 4).
Pass/Fail criteria	• In step 4, the PHD has to start to transmit data and the data-req-id field is set to data-req-id-agent-initiated • In step 6, the PHD has to stop to transmit data • In step 8, the PHD has to start again to transmit data and scan-report-no has to start counting where it was halted in step 4
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-072		
TP label		Operating procedures. PM-Store		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PersStoreMtrDatTransf 3; M	PersStoreMtrDatTransf 4; M	PM-StoreMeth 26; M
		PM-StoreMeth 27; O		
Test purpose		Check that: The PHD supports a get-segment-info request to all segments and a particular segment [AND] The PHD may support a get-segment-info request for a time range selection criteria according to pmsc-abs-time-select in the PM-Store-Capab attribute [AND] The PHD shall support the all-segments choice in the SegmSelection action-info-args of the Get- Segment-Info method. [AND] The PHD may support the segm-id-list and/ or abs-time-range choice in the SegmSelection action-info-args of the Get-Segment-Info method. In this case the PHD shall set the pmsc-segm-id-list-select and/ or pmsc-abs-time-select flag in the PM-Store-Capab attribute.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_009, C_AG_OXP_014		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The PHD under test issues a GET response with the PM-Store attributes it supports, check the values of the PM-Store-Capab attribute: a. PM-Store-Capab: □ attribute-id = MDC_ATTR_PM_STORE_CAPAB □ attribute-type = PmStoreCapab □ attribute-value = Record the value of bit 6 (Indicates that PM-Segments in the SegmSelection data type can be selected by defining an abs-time-range) and bit 3 (indicates that PM-Segments in the SegmSelection data type can be selected by defining a list of segment identifiers) 3. The simulated PHG sends a request for the PM-Segment Data with SegmSelection = 1 to obtain all the segments: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = all-segments 4. The PHD under test issues a response with the PM-Segments attributes a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmentInfoList IF bit 3 of PmStoreCapab was set: 5. The simulated PHG sends a request for the PM-Segment Data with SegmSelection = segm-id-list which is known because in the previous phase the information of all the segments was retrieved:		

a. Data APDU

□ Type = Invoke | Confirmed Action,

□ HANDLE = obj-handle

□ Action = MDC_ACT_SEG_GET_INFO

□ SegmSelection = segm-id-list (List of integers with the instance numbers of the selected Segments)

6. The PHD under test issues a response with the required PM-Segments attributes:

a. Data APDU

□ Type = Invoke | Confirmed Action,

□ HANDLE = obj-handle

□ Action = MDC_ACT_SEG_GET_INFO

□ SegmentInfoList

IF Protocol Version 3

7. Simulated PHG sends a Get-Segment-Info:

a. Data APDU

□ Type = Invoke | Confirmed Action,

□ HANDLE = obj-handle

□ Action = MDC_ACT_SEG_GET_INFO

□ SegmSelection = segm-id-list (empty list)

8. The PHD under test issues a response with

a. Data APDU

□ Type = Invoke | Confirmed Action,

□ HANDLE = obj-handle

□ Action = MDC_ACT_SEG_GET_INFO

□ SegmentInfoList = (empty list)

IF bit 3 of PMStoreCapab was NOT set:

9. The simulated PHG sends a Get-Segment-Info:

a. Data APDU

□ Type = Invoke | Confirmed Action,

□ HANDLE = obj-handle

□ Action = MDC_ACT_SEG_GET_INFO

□ SegmSelection = segm-id-list (List of integers with the instance numbers of the selected Segments)

10. The PHD under test operation response:

a. Data APDU

□ Type = Roer

□ ErrorResult = no-such-action (9) or not-allowed-by-object (24)

IF bit 6 of PmStoreCapab was set AND the PHD reports absolute-time:

11. The simulated PHG sends a Get-Segment-Info:

a. Data APDU

□ Type = Invoke | Confirmed Action,

□ HANDLE = obj-handle

□ Action = MDC_ACT_SEG_GET_INFO

□ SegmSelection = abs-time-range, selecting a range with its boundaries set to an earlier date of any of the existing segments.

	12. The PHD under test operation response: a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) 13. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = abs-time-range, selecting a range with its boundaries set to a later date than any of the existing segments 14. The PHD under test operation response: a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) 15. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = abs-time-range, selecting a range with one of its boundaries set to an earlier date than any of the existing segments and the other set to a date contained between Segment-Start-Abs-Time and Segment-End-Abs-Time of one of the PM-Segments 16. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) ELSE b. Data APDU □ Type = Response | Confirmed Action □ HANDLE = obj-handle c. Action = _ACT_SEG_GET_INFO □ SegmentInfoList = {empty} 17. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = abs-time-range, selecting a range with one of its boundaries set to a date contained between Segment-Start-Abs-Time and Segment-End-Abs-Time of one of the PM-Segments and the other set to a date later than any of the existing segments 18. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU

❑ Type = Roer

❑ ErrorResult = no-such-action (9)

ELSE

b. Data APDU

❑ Type = Response | Confirmed Action

❑ HANDLE = obj-handle

c. Action = _ACT_SEG_GET_INFO❑ SegmentInfoList = {empty}

19. The simulated PHG sends a Get-Segment-Info:

a. Data APDU

❑ Type = Invoke | Confirmed Action,

❑ HANDLE = obj-handle

❑ Action = MDC_ACT_SEG_GET_INFO

❑ SegmSelection = abs-time-range, selecting a range with its boundaries set to Segment-Start-Abs-Time and Segment-End-Abs-Time of one of the PM-Segments

20. The PHD under test operation response:

a. Data APDU

❑ Type = Invoke | Confirmed Action,

❑ HANDLE = obj-handle

❑ Action = MDC_ACT_SEG_GET_INFO

❑ SegmentInfoList = Containing the attributes of the selected Segments

21. The simulated PHG sends a Get-Segment-Info:

a. Data APDU

❑ Type = Invoke | Confirmed Action,

❑ HANDLE = obj-handle

❑ Action = MDC_ACT_SEG_GET_INFO

❑ SegmSelection = abs-time-range, selecting a range with its boundaries set to include inside from Segment-Start-Abs-Time to Segment-End-Abs-Time one of the PM-Segments

22. The PHD under test operation response:

a. Data APDU

❑ Type = Invoke | Confirmed Action,

❑ HANDLE = obj-handle

❑ Action = MDC_ACT_SEG_GET_INFO

❑ SegmentInfoList = Containing the attributes of the selected Segments

IF bit 6 of PmStoreCapab was set AND the PHD reports the base-offset-time:

23. The simulated PHG sends a Get-Segment-Info:

a. Data APDU

❑ Type = Invoke | Confirmed Action,

❑ HANDLE = obj-handle

❑ Action = MDC_ACT_SEG_GET_INFO

❑ SegmSelection = bo-time-range, selecting a range with its boundaries set to an earlier date than any of the existing segments.

24. The PHD under test operation response:

IF NOT Protocol Version 3

a. Data APDU

  - □ Type = Roer
  - □ ErrorResult = no-such-action (9)

ELSE

- b. Data APDU
  - □ Type = Response | Confirmed Action
  - □ HANDLE = obj-handle
- c. Action = _ACT_SEG_GET_INFO
  - □ SegmentInfoList = {empty}
- 25. The simulated PHG sends a Get-Segment-Info:a.
  - □ Type = Invoke | Confirmed Action,
  - □ HANDLE = obj-handle
  - □ Action = MDC_ACT_SEG_GET_INFO
  - □ SegmSelection = bo-time-range, selecting a range with its boundaries set to a later date than any of the existing segments.

26. The PHD under test operation response:

IF NOT Protocol Version 3

- a. Data APDU
  - □ Type = Roer
  - □ ErrorResult = no-such-action (9)

ELSE

- b. Data APDU
  - □ Type = Response | Confirmed Action
  - □ HANDLE = obj-handle
- c. Action = _ACT_SEG_GET_INFO
  - □ SegmentInfoList = {empty}

27. The simulated PHG sends a Get-Segment-Info:

- a. Data APDU
  - □ Type = Invoke | Confirmed Action,
  - □ HANDLE = obj-handle
  - □ Action = MDC_ACT_SEG_GET_INFO
  - □ SegmSelection = bo-time-range, selecting a range with one of its boundaries set to an earlier date than any of the existing segments and the other set to a date contained between Segment-Start-Bo-Time and Segment-End-Bo-Time of one of the PM-Segments

28. The PHD under test operation response:

IF NOT Protocol Version 3

- a. Data APDU
  - □ Type = Roer
  - □ ErrorResult = no-such-action (9)

ELSE

  - b. Data APDU
    - □ Type = Response | Confirmed Action
    - □ HANDLE = obj-handle
  - c. Action = _ACT_SEG_GET_INFO
    - □ SegmentInfoList = {empty}

	29. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = bo-time-range, selecting a range with one of its boundaries set to a date contained between Segment-Start-Bo-Time and Segment-End-Bo-Time of one of the PM-Segments and the other set to a later date than any of the existing segments 30. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) ELSE b. Data APDU □ Type = Response | Confirmed Action □ HANDLE = obj-handle c. Action = _ACT_SEG_GET_INFO □ SegmentInfoList = {empty} 31. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = bo-time-range, selecting a range with its boundaries set to Segment-Start-Bo-Time and Segment-End-Bo-Time of one of the PM-Segments 32. The PHD under test operation response: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmentInfoList = Containing the attributes of the selected Segments 33. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = bo-time-range, selecting a range with its boundaries set to include inside from Segment-Start-Bo-Time to Segment-End-Bo-Time of one of the PM-Segments 34. The PHD under test operation response: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO

	□ SegmentInfoList = Containing the attributes of the selected Segments IF bit 6 of PMStoreCapab was NOT set: 35. The simulated PHG sends a Get-Segment-Info: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_GET_INFO □ SegmSelection = abs-time-range, selecting a range with its boundaries set to the absolute minimum of Absolut-Time type and to the absolute maximum of the Absolute-Time type 36. The PHD under test operation response: IF NOT Protocol Version 3 a. Data APDU □ Type = Roer □ ErrorResult = no-such-action (9) or not-allowed-by-object (24) ELSE b. Data APDU □ Type = Response | Confirmed Action □ HANDLE = obj-handle c. Action = _ACT_SEG_GET_INFO □ SegmentInfoList = {empty}
Pass/Fail criteria	The PHD properly sends the required PM-Segment attributes in all cases (all-segments, specific segments and time range selected segments) or the specified error.
Notes	For Clear-Segment, [ISO/IEEE 11073-20601-2015A] has defined the error code not-allowed-by-object when the PHD does not support the particular action (list of segments or range of segments), but for Get- Segment-Info any error code is defined. For this reason both error codes have been added to the test procedure because the error code used previously is not clearly defined in the change request.

TP Id		TP/PLT/PHD/OXP/COM/BV-073		
TP label		Operating procedures. Error Code		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PersStoreMtrDatTransf 6; M		
Test purpose		Check that: If there is an error when the PHG access to a segment of a PM-Store, then the PHD returns an appropriate error code in the response and ignores the transmit request		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The simulated PHG sends a request for the PM-Segment Data with SegmSelection = all-segments. 3. The simulated PHG sends a request for the PM-Store Data to a non-existent Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle		

	□ Action = MDC_ACT_SEG_TRIG_XFER □ TrigSegmDataXferReq 4. The PHD issues a response: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ TrigSegmDataXferRsp = tsxr-fail-no-such-segment(1)
Pass/Fail criteria	The response from PHD under test must be of type tsxr-fail-no-such-segment(1).
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-073_A		
TP label		Operating procedures. Transfer PM-Segment content		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2016C]		
	Testable items	PersStoreMtrDatTransf 6; M		
Test purpose		Check that: If the PHG accesses successfully to a segment of a PM-Store, then the PHD sends a tsxr-successful response code to indicate that it has received the request and it can be honoured		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The simulated PHG sends a request for the PM-Segment Data with SegmSelection = all-segments. 3. The simulated PHG sends a request for the PM-Store Data to a Segment: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ TrigSegmDataXferReq 4. The PHD issues a response: a. Data APDU □ Type = Invoke | Confirmed Action, □ HANDLE = obj-handle □ Action = MDC_ACT_SEG_TRIG_XFER □ TrigSegmDataXferRsp □ Check that the invoke-id of the response is mirrored from the request		
Pass/Fail criteria		TrigSegmDataXferRsp must be one of: • tsxr-successful(0) • tsxr-fail-clear-in-process(2) • tsxr-fail-segm-empty(3) • tsxr-fail-not-otherwise-specified(512)		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-074		
TP label		Operating procedures. Segment Data Event		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PersStoreMtrDatTransf 7; M	PersStoreMtrDatTransf 8; M	PersStoreMtrDatTransf 9; M
		PersStoreMtrDatTransf 10; M	PersStoreMtrDatTransf 12; M	CommonCharac 3; M
Test purpose		Check that: The PHD sends confirmed Segment-Data-Event event reports until all entries in the PM-Segment are sent to the PHG or the transfer is aborted by either the sevtsta-agent-abort or sevtsta-manager-abort bits [AND] The PHD fills in the SegmentDataEvent structure with information about the segment being sent. [AND] The PHD always sets any sevtsta-manager-* bits to 0. [AND] If the message contains the first entry and/or the last entry of the data entries, then the PHD sets the sevtsta-first-entry and/or sevtsta-last-entry bits, respectively [AND] When transferring a segment, the PHD uses the segm-data-event-entries field to send all the entries. [AND] The total size of the response does not exceed the maximum APDU size established by the specialization		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Take some measurements with the PHD under test. 2. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 3. The PHD issues a GET response with the PM-Store attributes it supports. 4. The simulated PHG sends a request for the PM-Segment info with SegmSelection = 1 to obtain all the segments: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_GET_INFO ❑ SegmSelection = all-segments 5. The PHD issues a response with the PM-Segments attributes: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_GET_INFO ❑ SegmentInfoList 6. The simulated PHG sends a request for a PM-Segment Data: a. Data APDU		

<table>
<tr><th></th><th></th></tr>
<tr><td></td><td>❑ Type = Invoke | Confirmed Action,
❑ HANDLE = obj-handle
❑ Action = MDC_ACT_SEG_TRIG_XFER
❑ TrigSegmDataXferReq
7. The PHD issues a response:
a. Data APDU
❑ Type = Invoke | Confirmed Action,
❑ HANDLE = obj-handle
❑ Action = MDC_ACT_SEG_TRIG_XFER
❑ TrigSegmDataXferRsp
8. The PHD under test starts a Data transfer:
a. Data APDU
❑ Invoke | CfmEventReport
❑ Action = MDC_NOTI_SEGMENT_DATA
❑ SegmentDataEvent
❑ Segm-data-event-entries = Data
9. The simulated PHG responds to transferred data APDU's:
a. Data APDU
❑ Type = Invoke | Confirmed Action
❑ HANDLE = obj-handle
❑ Action = MDC_NOTI_SEGMENT_DATA
❑ SegmentDataResult
10. Steps 8 and 9 are repeated until all the data has been sent.</td></tr>
<tr><td><b>Pass/Fail criteria</b></td><td>• The PHD replies to the Get request with the requested Data and sevtsta-manager-* bits to 0
• In the first Data event sent sevtsta-first-entry bit must be set by the PHD
• In the last data event sent the sevtsta-last-entry bit must be set by the PHD
• In step 7 the total size of the message cannot exceed the maximum APDU size established by the specialization:
○ Pulse oximeter → 9216 octets
○ Weighing scales → 896 octets
○ Glucose meter → 5120 octets or 64512 octets if the PHD supports PM-Store
○ Blood pressure → 896 octets
○ Thermometer → 896 octets
○ Independent activity hub → 5120 octets
○ Cardiovascular → 64512 octets or 6624 octets if the PHD supports Step Counter Profile
○ Strength → 64512 octets
○ Adherence monitor → 1024 octets
○ Peak flow → 2030 octets
○ Body composition analyser → 7730 octets
○ Basic ECG/Simple ECG → 7168 octets or 64512 octets if the PHD supports PM-Store
○ Basic ECG/Heart rate → 1280 octets or 64512 octets if the PHD supports PM-Store
○ International normalized ratio → 896 octets or 64512 if the PHD supports PM-Store</td></tr>
</table>

	○ Insulin pump → 7168 octets or 5102 if PHD supports PM-Store ○ Continuous glucose monitor → 896 octets or 5120 if PHD supports PM-Store ○ Power status monitor → 940 octets for Simple PSM profile and 1660 for Advanced PSM profile
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-076		
TP label		Operating procedures. PM-Segment structure		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PersStoreMtrDatTransf 15; M		
Test purpose		Check that: Each entry of the Segment Data is formatted according to the structure defined in the PM-Segment PM-Segment-Entry-Map.		
Applicability		C_AG_OXP_041 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG shall send a Get request for the PM-Store object with an attribute-id-list set to 0 to indicate all PM-Store attributes. 2. The PHD issues a GET response with the PM-Store attributes. 3. The simulated PHG issues a Get-Segment-Info action with SemgSelection set to all-segments, for this test we are interested in: a. Mandatory attribute PM-Segment-Entry-Map ❑ attribute-id = MDC_ATTR_PM_SEG_MAP ❑ attribute-type = PmSegmentEntryMap ❑ attribute-value = 4. The simulated PHG sends a request for the PM-Segment that contains data: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ TrigSegmDataXferReq 5. The PHD issues an action response: a. Data APDU ❑ Type = Invoke | Confirmed Action, ❑ HANDLE = obj-handle ❑ Action = MDC_ACT_SEG_TRIG_XFER ❑ TrigSegmDataXferRsp 6. The PHD under test starts Data transfer: a. Data APDU ❑ Invoke | CfmEventReport ❑ Action = MDC_NOTI_SEGMENT_DATA ❑ SegmentDataEvent: –sevtsta-first-entry(0)=1 –segm-data-event-entries=Data		

	7. The simulated PHG responds to transferred data APDU's with an abort transfer: b. Data APDU □ Type = Invoke | Confirmed Action □ HANDLE = obj-handle □ Action = MDC_NOTI_SEGMENT_DATA □ SegmentDataResult –sevtsta-manager-abort(12)=1
Pass/Fail criteria	The format of the data has to coincide with the format expressed in the PmSegmentEntryMap field and the PHD does not send any SegmentDataEvent after step 7.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-080		
TP label		Release Request. Outstanding invoke-id		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	DisassocProc 3;M	DisassocProc 4;M	DisassocProc 5;R
Test purpose		Check that: Should a PHD receive an Association Release Request when it has an outstanding invoke-id,it shall respond with an Association Release Respond and assume that it shall receive no response to its request. [AND] After the side that received the Association Release Request sends the Association Release Response, it shall transition to the Unassociated state [AND] When the peer receives the Association Release Response, it shall transition to the Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an Associating Request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a roiv-cmip-confirmed-event report message with a MDC_NOTI_CONFIG event to send its configuration to the PHG. TOconfig is started. 4. The simulated PHG sends a Release Request with Reason = 0 "normal". 5. The PHD under test responds with a Release Response and changes to the Unassociated state. 6. Wait for a time equal to TOconfig.		
Pass/Fail criteria		During the period of step 6 the PHD does not send any abort message. After that point, the only message that may be received by the PHG is a new Association request.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-081_A		
TP label		Disassociating procedure. Release Request Reason 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	DisassocProc 2; M		

Test purpose	Check that: The Association Release Request contains a ReleaseRequestReason with reason = normal to indicate the reason for releasing the association
Applicability	C_AG_OXP_186 AND C_AG_OXP_000
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. Force the PHD to send a Release Request. 2. Check that the PHG receives a Release Request with reason = normal (0).
Pass/Fail criteria	The Association Release Request contains a ReleaseRequestReason to indicate the reason for releasing the association. The Reason code shall be the one described in step 2.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-081_B		
TP label		Disassociating procedure. Release Request Reason 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	DisassocProc 2; M		
Test purpose		Check that: The Association Release Request contains a ReleaseRequestReason with reason = configuration-changed to indicate the reason for releasing the association		
Applicability		C_AG_OXP_191 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Change the configuration of the PHD under test (adding or removing objects from the DIM). 2. Check that the simulated PHG receives a Release Request with reason = configuration-changed (2).		
Pass/Fail criteria		The Association Release Request contains a ReleaseRequestReason to indicate the reason for releasing the association. The Reason code shall be the one described in step 2.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-081_C		
TP label		Disassociating procedure. Release Request Reason 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	DisassocProc 2; M		
Test purpose		Check that: The Association Release Request contains a ReleaseRequestReason with reason = no-more-configurations to indicate the reason for releasing the association		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. Disconnect and again connect the PHD under test to the simulated PHG. 2. The simulated PHG receives an association request from the PHD under test. 3. The simulated PHG responds with a result = accepted-unknown-config. 4. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG.		

	5. The simulated PHG responds with an "unsupported-config". 6. Repeat steps 4 and 5 until the PHD sends a Release Request.
Pass/Fail criteria	The Association Release Request contains a ReleaseRequestReason = no-more-configurations (1).
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-083		
TP label		Disassociating procedure. Association Release Response		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	DisassocProc 8; M	DisassocProc 9; M	
Test purpose		Check that: When the PHD sends an Association Release message and waits for an Association Release Response message for a $TO_{release}$ period without receives it, then the PHD sends an Association Abort message and moves to Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. IF the PHD under test can be forced by the user to send a release request, send it. If not, follow this procedure: a. Disconnect and again connect the PHD under test to the simulated PHG. b. The simulated PHG receives an association request from the PHD under test. c. The simulated PHG responds with a result = accepted-unknown-config. d. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. e. The simulated PHG responds with an "unsupported-config". f. Repeat steps d and e until the PHD sends a Release Request. 2. Once the PHD under test has sent a Release Request: a. The simulated PHG does not respond to the request for at least $TO_{Release}$ (3 seg.).		
Pass/Fail criteria		The PHD waits the $TO_{Release}$ time and then it must send an abort message to the PHG.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-084		
TP label		Absolute time 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	TimeCoord 1; C	AbsTime 1; C	AbsTime 2; C
		AbsTime 3; C	AbsTime 5; C	
Test purpose		Check that: All bits references in the subclauses are part of this attribute [Mds-time-Info] [AND] If the PHD has an internal real-time clock (RTC), then it indicates this capability by setting the mds-time-capab-real-time-clock bit [AND] If the PHD supports the Set-Time action, then it indicates this capability by setting the mds-time-capab-set-clock bit [AND]		

	The PHD indicates whether it synchronizes absolute time using the mds-time-capab-sync-abs-time bit [AND] The mds-time-state-abs-time-synced bit is set only when the PHD believes its wall clock time is synchronized with the external clock source.
Applicability	C_AG_OXP_009 AND C_AG_OXP_013 AND C_AG_OXP_000
Other PICS	C_AG_OXP_007
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. The simulated PHG issues "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. 2. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object. The attribute of interest of this test is MDSTimeInfo: a. Mds-Time-Info: □ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) □ attribute-type = MdsTimeInfo □ attribute-value.length = 2 bytes □ mds-time-capab-real-time-clock must be set b. IF the PHD can synchronize its absolute time then: □ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) □ attribute-type = MdsTimeInfo □ attribute-value.length = 2 bytes □ mds-time-capab-sync-abs-time must be set □ time-sync-protocol field must indicate what protocol is used for synchronization 3. IF mds-time-capab-sync-abs-time = 1 THEN: a. Ask the test operator to connect the external source that is going to be used to synchronize the PHD AbsoluteTime b. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. c. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: □ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) □ attribute-type = MdsTimeInfo □ attribute-value.length = 2 bytes □ mds-time-capab-sync-abs-time must be set □ mds-time-state -abs-time-synced must be set
Pass/Fail criteria	Check that the attribute mds-time-cap-state has correct values.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-085_A		
TP label		Absolute time 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AbsTime 11; C	MDSService 7; O	
Test purpose		Check that: If a PHD is associated with a PHG when Date-and-Time is adjusted, then it sends an event		

	report that contains the new Date-and-Time.
Applicability	C_AG_OXP_006 AND C_AG_OXP_009 AND C_AG_OXP_012 AND C_AG_OXP_000 AND C_AG_OXP_016
Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Operating state.
Test procedure	1. Take a measurement with the PHD. 2. Make a noticeable change in the Date or Time of the PHD. 3. Take a new measurement. 4. Wait for a roiv-cmip-event-report OR a roiv-cmip-confirmed-event-report from the PHD. 5. Verify that the device sends a variable format event report to update the Date-and-Time attribute on the MDS before it sends any measurement updates and that every measure taken before the time change is sent in the same event report.
Pass/Fail criteria	The PHD transmitted data comes from the same unbroken timeline which means that every measure taken before the time change has a date-and-time-adjustment.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-085_B		
TP label		Absolute time 2: PM-Store		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AbsTime 14; C		
Test purpose		Check that: If a PHD collects PM-Store measurements and the Date-and-Time is adjusted, then the PHD ensures that each PM-Segment includes only measurements from the same unbroken timeline		
Applicability		C_AG_OXP_012 AND C_AG_OXP_041 AND C_AG_OXP_000 AND C_AG_OXP_016		
Other PICS		C_AG_OXP_009, C_AG_OXP_014, C_AG_OXP_293		
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. While the PHD is disconnected, make it store measurements in PM-Segments of every PM-Store, after doing this, connect the PHD. 2. The simulated PHG receives an association request from the PHD under test. 3. The simulated PHG responds with a result = accepted-unknown-config. 4. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message: o Event-type=MDC_NOTI_CONFIG 5. IF C_AG_OXP_293 THEN: a. Once in Configuring/Sending GetMDS substate simulated PHG issues roiv-cmip-get command with handle set to 0 (to request for MDS object) and attribute-id-list set to 0 to indicate all attributes. b. The PHD responds with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. c. IF the mds-time-mgr-set-time bit is set: □ The PHG moves to Configuring/Sending Set Time substate and: • IF C_AG_OXP_009 THEN it issues the Set-Time action command. • IF C_AG_OXP_014 THEN it issues the Set-Base-Offset-Time action command. □ Once its internal time setting operation is completed, the PHD responds to the PHG. 6. Record the PM-Store handle, PM-Store-Capab and Number-Of-Segments of every PM-Store object. 7. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object		

	with SegmSelection set to all-segments. 8. The PHD shall respond to the Get-Segment-Info, indicating the attributes of the PM-Segment. 9. Make a noticeable change in the Date or Time of the PHD. 10. Take a new measurement. 11. The simulated PHG shall send a Get-command for every PM-Store. 12. The PHD shall respond to the Get command, indicating the attributes of the PM-Store. Record. 13. The simulated PHG shall send a Get-Segment-Info object action for the PM-Store object with SegmSelection set to all-segments. 14. The PHD shall respond to the Get-Segment-Info, indicating the attributes of the PM-Segment: ○ The Date-and-Time adjustment attribute is present ○ If the pmsc-var-no-of-segm is set to 1 (PM-Store-Capab) then verify that the Pm-Store has created a new segment
Pass/Fail criteria	• If the pmsc-var-no-of-segm is set to 1, the number of segments recorded in step 6 has increased in step 12 and the Date and Time Adjustment is present for every new added segment • If the pmsc-var-no-of-segm is set to 0, the Date and Time Adjustment is present at least for one segment
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-085_C		
TP label		Absolute time 2: Store and Forward		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AbsTime 12; C	AbsTime 13; C	
Test purpose		Check that: If a PHD collects temporary measurements and the Date-and-Time is adjusted, then the PHD ensures that all measurements included in an event report come from the same unbroken timeline [AND] The first reported data in the event report shall be the MDS attribute Date-and-Time-Adjustment that defines the number of 1/100th of seconds to add to align with the current clock (e.g. if the clock was advanced by 60 minutes, this would report 360000) followed by measurement data.		
Applicability		C_AG_OXP_009 AND C_AG_OXP_012 AND C_AG_OXP_032 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the disconnected state.		
Test procedure		1. Take some measurements with the PHD under test. 2. Make a change in the Date or Time of the PHD by advancing its clock 60 minutes. 3. Take new measurements. 4. Connect the PHD under test to the simulated PHG. 5. Once in the Operating state the PHG has to receive a variable event report containing the Date-and-Time-Adjustment attribute with the value = 360000.		
Pass/Fail criteria		The PHD recorded data before the Time change must be in a different segment than those recorded after the time change.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-086		
TP label		Relative time 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	RelativeTime 2; C	RelativeTime 5; C	RelativeTime 6; C
Test purpose		Check that: The PHD indicates that supports relative time by setting the mds-time-capab-relative-time bit in the Mds-Time-Info attribute [AND] The PHD indicates whether it synchronizes relative time using the mds-time-capab-sync-rel-time bit. [AND] If synchronization is supported, then the mds-time-state-rel-time-synced bit is set only when the PHD believes its relative clock is synchronized with the external source		
Applicability		C_AG_OXP_010 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_007, C_AG_OXP_008		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. 2. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: a. Mds-Time-Info shall be present: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-relative-time must be set b. IF the PHD can synchronize its relative timer then: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-sync-rel-time must be set ❑ time-sync-protocol field must indicate what protocol is used for synchronization 3. IF the mds-time-capab-sync-rel-time = 1 THEN: a. Ask the test operator to connect the external source that is going to be used to synchronize the PHD Relative-Time. b. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. c. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-sync-rel-time must be set ❑ mds-time-state -rel-time-synced must be set		
Pass/Fail criteria		All checked values are as specified in the test procedure.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-087		
TP label		High-resolution Relative time		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	Hi-resRelativeTime 1; C	Hi-resRelativeTime 4; C	Hi-resRelativeTime 5; C
		Hi-resRelativeTime 6; C		
Test purpose		Check that: The PHD indicates support for high resolution relative time by setting the mds-time-capab-high-res-relative-time bit in the Mds-Time-Info attribute [AND] If synchronization is supported, then the PHD sets mds-time-state-hi-res-relative-time-synced bit only when it believes its relative clock is synchronized with the external source. [AND] When the PHD disconnects from the clock synchronization source, it clears the synced bit as soon as it exceeds the accuracy of the clock synchronization parameters.		
Applicability		C_AG_OXP_011 AND C_AG_OXP_000		
Other PICS		C_AG_OXP_007, C_AG_OXP_008		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. 2. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: a. To support Hires-Relative Time: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-high-res-relative-time must be set b. IF the PHD can synchronize its High Resolution Relative timer then: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-sync-hi-res-relative-time must be set ❑ time-sync-protocol field must indicate what protocol is used for synchronization 3. IF the mds-time-capab-sync-hi-res-relative-time = 1 THEN: c. Ask the test operator to connect the external source that is going to be used to synchronize the PHD Hi-Resolution-Relative-Time. d. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. e. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-sync-hi-res-relative-time must be set ❑ mds-time-state-hi-resrelative-time-synced must be set ❑ time-sync-accuracy will be recorded.		

	f. Ask the test operator to disconnect the external source that has been used to synchronize the PHD Hi-Resolution-Relative-Time. g. Wait a time interval longer than the time specified in time-sync-accuracy; if it is undefined, the test operator has to wait a discretionary time that is enough to exceed the accuracy of the clock synchronization. h. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. i. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-state-hi-resrelative-time-synced must be clear
Pass/Fail criteria	All checked values are as specified in the test procedure.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-088		
TP label		Base-Offset-Time 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	BaseTimOffset1; M		
Test purpose		Check that: If the base time is changed, then the time adjustment shall be indicated using the same mechanisms as for absolute time.		
Applicability		C_AG_OXP_014 AND C_AG_OXP_013 AND C_OXP_000		
Other PICS		C_AG_OXP_007		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. 2. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object. The attribute of interest of this test is MDSTimeInfo: a. Mds-Time-Info: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-bo-time must be set b. IF the PHD can synchronize its base-offset-time (i.e., C_AG_OXP_007 = TRUE) THEN: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-sync-bo-time must be set ❑ time-sync-protocol = MDC_TIME_SYNC_EBWW or MDC_TIME_SYNC_NTPV3 or MDC_TIME_SYNC_NTPV4 or MDC_TIME_SYNC_SNTPV4 or MDC_TIME_SYNC_SNTPV4330 or MDC_TIME_SYNC_BTV1 3. IF mds-time-capab-sync-bo-time = 1 THEN:		

		a. Ask the test operator to connect the external source that is going to be used to synchronize the PHD Base-Offset time. b. The simulated PHG issues a "Remote Operation Invoke | Get" command with the handle set to 0 (to request an MDS object) and the attribute-id-list set to 0 to indicate all attributes. c. The PHD responds with a "rors-cmip-get" service message in which the attribute-list contains a list of all implemented attributes of the MDS object: ❑ attribute-id = MDC_ATTR_MDS_TIME_INFO (0X0A 0X45) ❑ attribute-type = MdsTimeInfo ❑ attribute-value.length = 2 bytes ❑ mds-time-capab-sync-bo-time must be set ❑ mds-time-state-bo-time-synced must be set		
Pass/Fail criteria		Check the attribute mds-time-cap-state has correct values.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-092		
TP label		Supported Standard Configuration		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	ConfNormalProc 16; M		
Test purpose		Check that: The PHD sends the supported standard configurations as a fall back if the extended configurations are unsupported.		
Applicability		C_AG_OXP_001 AND C_AG_OXP_002 AND C_AG_OXP_000		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Unassociated state.		
Test procedure		1. The simulated PHG receives an association request from the PHD under test. 2. The simulated PHG responds with a result = accepted-unknown-config. 3. The PHD responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the PHG. 4. The simulated PHG responds with a "unsupported-config". 5. Repeat steps 3 and 4 until the PHD sends a Release Request with reason = "no-more-configurations".		
Pass/Fail criteria		The PHD under test must send at least one configuration with dev-config-id = <between 0x00 0x01 and 0x3F 0xFF > (standard range values) as a fallback if the extended configurations are not supported by the simulated PHG.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-093		
TP label		Operating State. Abort message		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 63; M		
Test purpose		Check that: If abrt received, then the PHD moves to Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				

Initial condition	The simulated PHG and PHD are in the Operating state.
Test procedure	1. The simulated PHG sends an Abort message to the PHD under test. 2. Wait for an event report for the PHD.
Pass/Fail criteria	The simulated PHG must not receive any message other than an Association Request.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-096_A		
TP label		Agent State machine. Connected Associated Configuring Waiting Approval 11		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 78; M		
Test purpose		Check that: If prst (Any APDU not covered in 5.* (corrupt, unknown, unexpected, etc) is received while in the Waiting Approval state, then the PHD transmits an abort message abrt (reason undefined) and moves to Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the Waiting Approval state.		
Test procedure		1. The simulated PHG sends a badly formatted message. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-097		
TP label		Agent State machine. Leaving Operating State 5		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 80; M		
Test purpose		Check that: If prst (Any APDU not covered in 2.* (corrupt, unknown, unexpected, etc)) is received while in Operating state, then the PHD transmits an abrt (reason undefined) and moves to Unassociated state.		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the Operating state.		
Test procedure		1. The simulated PHG issues a Prst message, rors-cmip- get. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-097_A		
TP label		Agent State machine. Leaving Operating State 6		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 80; M		
Test purpose		Check that: If prst (Any APDU not covered in 8.* (corrupt, unknown, unexpected, etc)) is received while in Operating state, then the PHD transmits an abrt (reason undefined) and moves to Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD under test is in the Operating state.		
Test procedure		1. The simulated PHG sends a badly formatted message. 2. The PHD under test sends an abort message abrt(Abort-reason undefined) to the PHG and shall pass to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the Abort message (abrt) with reason undefined and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-098_A		
TP label		Agent State machine. Connected Disassociation 7		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 81; M		
Test purpose		Check that: If prst (Any APDU not covered in 9.* (corrupt, unknown, unexpected, etc) is received while in the Disassociating state, then the PHD transmits an abrt (Abort-reason undefined) and moves to Unassociated state		
Applicability		C_AG_OXP_000		
Other PICS				
Initial condition		The PHD is in the Unassociated state.		
Test procedure		1. The PHD under test sends an Association Request to the simulated PHG. 2. The simulated PHG responds with an accepted-unknown-config. 3. The PHD under test sends a configuration event report. 4. The simulated PHG responds with an unsupported-configuration. 5. The PHD sends a new configuration event report with a new configuration (if it has more). 6. Repeat the last two steps recording all the ConfigId-values until the PHD sends a Release Request with the reason "no-more-configurations". The PHD moves to the Disassociating state. 7. The simulated PHG sends a badly formatted message. 8. The PHD responds with an Abort message (abrt) with reason undefined. 9. The PHD and the PHG move to the Unassociated state.		
Pass/Fail criteria		• The PHD under test sends the abort message (abrt) with reason undefined and changes to the Unassociated state		

	• The simulated PHG must not receive any message other than an Association Request after step 9.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-099		
TP label		Operating procedures. PM-Store Specific Attributes request		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	PersStoreMtrDatTransf 2; C		
Test purpose		Check that: The attribute-id-list shall be left empty to query for all attributes of the PM-store object. Alternatively, specific attributes of an object may be queried by listing the desired Attribute IDs found in Table 9. It is not required for a PHD to support this capability. If this capability is not implemented then the PHD shall respond with an error (roer) message with an error-value of not-allowed-by-object		
Applicability		C_AG_OXP_000 AND C_AG_OXP_041		
Other PICS		C_AG_OXP_101		
Initial condition		The simulated PHG and PHD under test are in the Operating state.		
Test procedure		1. The simulated PHG issues a "Remote Operation Invoke | Get" command with: a. the Obj-handle set to PM-Store object handle (to request its attributes) b. the attribute-id-list.count=1 and a single AVA_Type MDC_ATTR_PM_STORE_CAPAB (0X0A 0X4D) to retrieve the mandatory "PM-Store-Capab" attribute 2. The PHD under test responds with: • IF C_AG_OXP_101 THEN: with a "rors-cmip-get" service message which contains the "PM-Store-Capab" • ELSE: with a "roer" service message with the error-value set to not-allowed-by-object (24) 3. The simulated PHG issues a "Remote Operation Invoke | Get" command with: a. the Obj-handle set to PM-Store object handle (to request its attributes) b. the attribute-id-list empty to request all the attributes of PM-Store object 4. The PHD responds with a "rors-cmip-get" service message which contains all the supported attributes of the PM-Store object. 5. The simulated PHG issues a "Remote Operation Invoke | Get" command with: a. the Obj-handle set to PM-Store object handle (to request its attributes) b. the attribute-id-list set to an attribute NOT supported by the PM-Store object 6. The PHD responds with a "rors-cmip-get" service message: • IF C_AG_OXP_101 THEN: the attribute-list must be empty • ELSE: with a "roer" service message with the error-value set to not-allowed-by-object (24) 7. The simulated PHG issues a "Remote Operation Invoke | Get" command with: a. the Obj-handle set to PM-Store object handle (to request its attributes) b. the attribute-id-list contains one supported attribute and one unsupported attribute 8. The PHD responds with a "rors-cmip-get" service message: • IF C_AG_OXP_101 THEN: the attribute-list must contain the supported attribute • ELSE: with a "roer" service message with the error-value set to not-allowed-by-object (24)		
Pass/Fail criteria		• In step 2 the PHD properly sends the requested attribute or the error		

	• In steps 6 and 8 the received attribute list must be empty if NOT C_AG_OXP_101 or roer if the action is not supported
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-100		
TP label		Agent State machine. Leaving the Configuring/Waiting GetMDS Substate 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 91; M		
Test purpose		Check that: If Rlrq is received while in the Configuring/Waiting GetMDS substate, a PHD shall transmit an Rlre (normal) and move to the Unassociated state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. The PHD is in the Configuring/Waiting GetMDS substate and the Simulated PHG is in the Configuring/Sending GetMDS substate.		
Test procedure		1. The simulated PHG sends an Association Release Request (Rlrq) message to the PHD under test with reason=0 (normal) 2. The PHD under test shall respond with an Association Release Response (Rlre) message with reason=0 (normal), and shall go to the Unassociated state.		
Pass/Fail criteria		The PHD under test transmits correctly the Rlre message		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-101		
TP label		Agent State machine. Leaving the Configuring/Waiting GetMDS Substate 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 92; M		
Test purpose		Check that: If Rlre is received while in the Configuring/Waiting GetMDS substate, then PHD transmits an Abrt (Abort-reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. PHD is in the Configuring/Waiting GetMDS substate and Simulated PHG is in the Configuring/Sending GetMDS substate.		
Test procedure		1. The simulated PHG sends Association Release Response to the PHD under test, with reason=0 (normal) 2. The PHD under test responds with an Abort message (Abrt) with reason undefined.		

Pass/Fail criteria	• The PHD under test transmits correctly the Abort message (Abrt) with reason undefined and changes to Unassociated state. • The simulated PHG must not receive any message other than an Association Request after step 2.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-102		
TP label		Configuring/Waiting GetMDS Substate. Abort message		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 93; M		
Test purpose		Check that: If abrt received, then the PHD moves to Unassociated state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. PHD is in the Configuring/Waiting GetMDS substate and Simulated PHG is in the Configuring/Sending GetMDS substate.		
Test procedure		1. The simulated PHG sends an Abort Message to the PHD under test 2. Wait for an event report for PHD.		
Pass/Fail criteria		The simulated PHG must not receive any message other than an Association Request.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-103		
TP label		Agent State machine. Leaving the Configuring/Waiting GetMDS Substate 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 94; M		
Test purpose		Check that: If prst (Any APDU not covered in 8.* (corrupt, unknown, unexpected, etc)) is received while in Configuring/Waiting GetMDS substate, then PHD transmits an abrt (reason undefined) and moves to Unassociated state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. PHD is in the Configuring/Waiting GetMDS substate and Simulated PHG is in the Configuring/Sending GetMDS substate.		
Test procedure		1. The simulated PHG issues a Prst message, rors-cmip- get. 2. The PHD under test sends an abort message abrt(reason undefined) to the PHG and shall pass to the Unassociated state		

Pass/Fail criteria	• The PHD under test sends the Abort message abrt(reason undefined) and changes to Unassociated • The simulated PHG must not receive any message other than an Association Request after step 2.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-104		
TP label		Agent State machine. Connected Associated Configuring/Waiting GetMDS 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 95; M		
Test purpose		Check that: If roiv-cmip-get (handle = 0) is received while in Waiting GetMDS substate, then PHD transmits a rors-cmip-get (MDS attributes) and moves to Waiting SetTime substate or Operating state.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS		C_AG_OXP_004		
Initial condition		The simulated PHG and PHD under test are in Configuring state. PHD is in the Configuring/Waiting GetMDS substate and Simulated PHG is in the Configuring/Sending GetMDS substate.		
Test procedure		1. The simulated PHG issues GET with (handle = 0) 2. The PHD under test must send a rors-cmip-get with MDS attributes. 3. IF C_AG_OXP_004, PHD under test moves in Waiting SetTime substate. ELSE, PHD under test moves to Operating state.		
Pass/Fail criteria		Process detailed above must be successfully completed		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-105		
TP label		Agent State machine. Connected Associated Configuring/Waiting GetMDS 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 96; M		
Test purpose		Check that: If roiv-* but not (roiv-cmip-get, handle=0) is received while in Waiting GetMDS substate, then PHD transmits a roer (no-such-object-instance) and remains in Waiting GetMDS substate.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. PHD is in the Configuring/Waiting GetMDS substate and Simulated PHG is in the Configuring/Sending GetMDS substate.		

Test procedure	1. The simulated PHG issues a roiv-cmip-get (handle = 1) 2. The PHD under test must send a "roer" with reason = no-such-object-instance(1) 3. The PHD under test remains in Waiting GetMDS substate. 4. The simulated PHG responds with a roiv-cmip-get (handle = 0). 5. The PHD responds with a rors-cmip-get (MDS Attributes)
Pass/Fail criteria	Process detailed above must be successfully completed.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-106		
TP label		Agent State machine. Leaving the Configuring/Waiting SetTime Substate 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 101; M		
Test purpose		Check that: If Rlrq is received while in the Configuring/Waiting SetTime substate, a PHD shall transmit an Rlre (normal) and move to the Unassociated state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293 AND C_AG_OXP_004		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. PHD is in the Configuring/Waiting SetTime substate and Simulated PHG is in the Configuring/Sending SetTime substate.		
Test procedure		1. The simulated PHG sends an Association Release Request (Rlrq) message to the PHD under test with reason=0 (normal) 2. The PHD under test shall respond with an Association Release Response (Rlre) message with reason=0 (normal), and shall go to the Unassociated state.		
Pass/Fail criteria		PHD under test transmits correctly the Rlre message		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-107		
TP label		Agent State machine. Leaving the Configuring/Waiting SetTime Substate 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 102; M		
Test purpose		Check that: If Rlre is received while in the Configuring/Waiting SetTime substate, then PHD transmits an Abrt (reason undefined) and moves to the Unassociated state.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293 AND C_AG_OXP_004		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Configuring state. The PHD is in the		

	Configuring/Waiting SetTime substate and the simulated PHG is in the Configuring/Sending SetTime substate.
Test procedure	1. The simulated PHG sends an Association Release Response to the PHD under test, with reason=0 (normal) 2. The PHD under test responds with an Abort message (Abrt) with reason undefined.
Pass/Fail criteria	• The PHD under test transmits correctly the Abort message (Abrt) with reason undefined and changes to the Unassociated state. • The simulated PHG must not receive any message other than an Association Request after step 2.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-108		
TP label		Configuring/Waiting SetTime Substate. Abort message		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 103; M		
Test purpose		Check that: If an abort message abrt is received, then the PHD moves to Unassociated state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293 AND C_AG_OXP_004		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting SetTime substate and the simulated PHG is in the Configuring/Sending SetTime substate.		
Test procedure		1. The simulated PHG sends an Abort Message to the PHD under test 2. Wait for an event report for PHD.		
Pass/Fail criteria		The simulated PHG must not receive any message other than an Association Request.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-109		
TP label		Agent State machine. Leaving the Configuring/Waiting SetTime Substate 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 104; M		
Test purpose		Check that: If prst (Any APDU not covered in 10.* (corrupt, unknown, unexpected, etc)) is received while in Configuring/Waiting SetTime substate, then the PHD transmits an abrt (reason undefined) and moves to the Unassociated state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293 AND C_AG_OXP_004		
Other PICS				

Initial condition	The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting SetTime substate and the simulated PHG is in the Configuring/Sending SetTime substate.
Test procedure	1. The simulated PHG issues a Prst message, rors-cmip- get. 2. The PHD under test sends an abort message abrt(reason undefined) to the PHG and shall pass to the Unassociated state
Pass/Fail criteria	• The PHD under test sends the Abort message abrt(reason undefined) and changes to the Unassociated state • The simulated PHG must not receive any message other than an Association Request after step 2.
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-110		
TP label		Agent State machine. Connected Associated Configuring/Waiting SetTime 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 105; M		
Test purpose		Check that: If roiv-cmip-confirmed-action (set time) is received while in Waiting SetTime substate, then PHD transmits a rors-cmip-confirmed-action (set time) and moves to Operating state.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293 AND C_AG_OXP_004		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting SetTime substate and the simulated PHG is in the Configuring/Sending SetTime substate.		
Test procedure		1. The simulated PHG issues a roiv-cmip-confirmed-action (set time) 2. The PHD under test must send a rors-cmip-confirmed-action (set time) 3. The PHD under test moves to the Operating state.		
Pass/Fail criteria		Process detailed above must be successfully completed		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-111		
TP label		Agent State machine. Connected Associated Configuring/Waiting SetTime 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 106; M		
Test purpose		Check that: If roiv-cmip-confirmed-action (but not set time) is received while in Waiting SetTime substate, then the PHD transmits a roer (no-such-object-instance) and remains in the Waiting SetTime substate.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293 AND C_AG_OXP_004		

Other PICS	
Initial condition	The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting SetTime substate and the simulated PHG is in the Configuring/Sending SetTime substate.
Test procedure	1. The simulated PHG issues a roiv-cmip-confirmed-action (but not set time) 2. The PHD under test must send a "roer" with reason = no-such-object-instance(1) 3. The PHD under test remains in the Waiting SetTime substate. 4. The simulated PHG issues a roiv-cmip-confirmed-action (set time) 5. The PHD under test must send a rors-cmip-confirmed-action (set time)
Pass/Fail criteria	Process detailed above must be successfully completed
Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-112		
TP label		Agent State machine. Accepted known configuration Protocol Version 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 82; M		
Test purpose		Check that: If aare(accepted) is received while in the Associating state, then PHD under test moves to the Operating state		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in Unassociated state		
Test procedure		1. Simulated PHG receives an association request from the PHD under test (PHD passes to Associating state). 2. Simulated PHG responds with a result = accepted-unknown-config 3. PHD under test responds with a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the simulated PHG. Configuration report is reported. 4. Simulated PHG sends an abrt message. 5. Simulated PHG and PHD moves to Unassociated state. 6. Simulated PHG receives an association request from the PHD under test 7. Simulated PHG responds with a result = accepted.		
Pass/Fail criteria		The PHD under test has passed to Configuring/Waiting GetMDS substate after last step.		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-113
TP label		Agent State machine. Connected Associated Configuring/Sending Config 1
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]

	Testable items	AgentStateMach 83; M		
Test purpose		Check that: If roiv-cmip-get (handle = 0) is received while in Sending Config substate, then PHD transmits a roer (no-such-object-instance) and remains in Sending Config substate.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Sending Config substate and the simulated PHG is in the Configuring/ Waiting for Config substate.		
Test procedure		1. The simulated PHG issues a roiv-cmip-get (handle = 0) 2. The PHD under test must send a "roer" with reason = no-such-object-instance(1) 3. The PHD under test remains in Sending Config substate. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result "unsupported-config". 5. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG or Rlrq (no-more-configurations)		
Pass/Fail criteria		Process detailed above must be successfully completed		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-114		
TP label		Agent State machine. Connected Associated Configuring/Waiting Approval 1		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 84; M		
Test purpose		Check that: If roiv-cmip-get (handle = 0) is received while in Waiting Approval substate, then PHD transmits a roer (no-such-object-instance) and remains in Waiting Approval substate.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting Approval substate and the simulated PHG is in the Configuring/Checking Config substate.		
Test procedure		1. The simulated PHG issues a roiv-cmip-get (handle = 0) 2. The PHD under test must send a "roer" with reason = no-such-object-instance(1) 3. The PHD under test remains in Waiting Approval substate. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result "unsupported-config". 5. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG or Rlrq (no-more-configurations)		
Pass/Fail criteria		Process detailed above must be successfully completed		

Notes	

TP Id		TP/PLT/PHD/OXP/COM/BV-115		
TP label		Agent State machine. Connected Associated Configuring/Waiting Approval 2		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 85; M		
Test purpose		Check that: If roiv-cmip-confirmed-action (set time) is received while in Waiting Approval substate, then PHD transmits a roer (no-such-object-instance) and remains in Waiting Approval substate.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting Approval substate and the simulated PHG is in the Configuring/Checking Config substate.		
Test procedure		1. The simulated PHG issues a roiv-cmip-confirmed-action (set time) 2. The PHD under test must send a “roer” with reason = no-such-object-instance(1) 3. The PHD under test remains in Waiting Approval substate. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result “unsupported-config”. 5. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG or Rlrq (no-more-configurations)		
Pass/Fail criteria		Process detailed above must be successfully completed		
Notes				

TP Id		TP/PLT/PHD/OXP/COM/BV-116		
TP label		Agent State machine. Connected Associated Configuring/Waiting Approval 3		
Coverage	**Spec**	[ISO/IEEE 11073-20601-2015A] and [ISO/IEEE 11073-20601-2016C]		
	Testable items	AgentStateMach 86; M		
Test purpose		Check that: If roiv-cmip-confirmed-action (but not set time) is received while in Waiting Approval substate, then the PHD transmits a roer (no-such-object-instance) and remains in the Waiting Approval substate.		
Applicability		C_AG_OXP_000 AND C_AG_OXP_293		
Other PICS				
Initial condition		The simulated PHG and PHD under test are in the Configuring state. The PHD is in the Configuring/Waiting Approval substate and the simulated PHG is in the Configuring/Checking Config substate.		
Test procedure		1. The simulated PHG issues a roiv-cmip-confirmed-action (but not set time)		

	2. The PHD under test must send a “roer” with reason = no-such-object-instance(1) 3. The PHD under test remains in the Waiting Approval substate. 4. The simulated PHG responds with a rors-cmip-confirmed-event-report with result “unsupported-config”. 5. The PHD responds with a roiv-cmip-confirmed-event report message with an MDC_NOTI_CONFIG event to send its configuration to the PHG or Rlrq (no-more-configurations)
Pass/Fail criteria	Process detailed above must be successfully completed
Notes	

Bibliography

[b-ITU-T H.810 (2013)] Recommendation ITU-T H.810 (2013), *Interoperability design guidelines for personal health systems.*

[b-ITU-T H.810 (2015)] Recommendation ITU-T H.810 (2015), *Interoperability design guidelines for personal health systems*.

[b-ITU-T H.810 (2016)] Recommendation ITU-T H.810 (2016), *Interoperability design guidelines for personal health systems*.

[b-ITU-T X.290] Recommendation ITU-T X.290 (1994), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts.*

[b-ITU-T X.296] Recommendation ITU-T X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements.*

[b-CDG 1.0] Continua Health Alliance, Continua Design Guidelines v1.0 (2008), *Continua Design Guidelines.*

[b-CDG 2010] Continua Health Alliance, Continua Design Guidelines v1.5 (2010), *Continua Design Guidelines.*

[b-CDG 2011] Continua Health Alliance, Continua Design Guidelines (2011), "Adrenaline", *Continua Design Guidelines.*

[b-CDG 2012] Continua Health Alliance, Continua Design Guidelines (2012), "Catalyst", *Continua Design Guidelines.*

[b-CDG 2013] Continua Health Alliance, Continua Design Guidelines (2013), "Endorphin", *Continua Design Guidelines*.

[b-CDG 2015] Continua Health Alliance, Continua Design Guidelines (2015), "Genome", *Continua Design Guidelines*.

[b-CDG 2016] Personal Connected Health Alliance, Continua Design Guidelines (2016), "Iris", *Continua Design Guidelines*.

[b-CDG 2017] Personal Connected Health Alliance, Continua Design Guidelines (2017), "Keratin", *Continua Design Guidelines*.

[b-ETSI 300 406] ETSI ETS 300 406 (1995), *Methods for Testing and Specifications (MTS); Protocol and profile conformance testing specifications; Standardization methodology.*
www.etsi.org/deliver/etsi_i_ets/300400_300499/300406/01_60/ets_300406e01p.pdf

[b-ETSI SR 001 262] ETSI SR 001 262 v1.8.1 (2003-12), *ETSI drafting rules*.
https://docbox.etsi.org/MTS/MTS/10-PromotionalMaterial/MBS-20111118/Referenced%20Documents/Drafting%20Rules.pdf

[b-PHD PICS & PIXIT] PHD PICS and PIXIT Test Tool v8.0.0.0 – Excel sheet v1.13.
http://handle.itu.int/11.1002/2000/12067

[b-PHG PICS & PIXIT] PHG PICS and PIXIT Test Tool v8.0.0.0 – Excel sheet v1.11.
http://handle.itu.int/11.1002/2000/12067

[b-TI] PHD Testable items. Test Tool v8.0.0.0 – Excel sheet v1.10.
http://handle.itu.int/11.1002/2000/12067

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020