INTERNATIONAL STANDARD

ISO/IEEE 11073-20601

First edition
2010-05-01

AMENDMENT 1
2015-03-01

Health informatics — Personal health device communication

Part 20601: Application profile — Optimized exchange protocol

AMENDMENT 1

Informatique de santé — Communication entre dispositifs de santé personnels

Partie 20601: Profil d'application — Protocole d'échange optimisé

AMENDEMENT 1

Reference number
ISO/IEEE 11073-20601:2010/Amd 1:2015(E)

ISO copyright office
Case postale 56
CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York
NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web www.ieee.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from ISO or the IEEE Standards Association.

Amendment 1 to ISO/IEEE 11073-20601 was prepared by the 11073 Committee of the Engineering in Medicine and Biology Society of the IEEE (as IEEE 11073-20601). It was adopted by Technical Committee ISO/TC 215, *Health informatics,* in parallel with its approval by the ISO member bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE. Both parties are responsible for the maintenance of this document.

(blank page)

IEEE STANDARDS ASSOCIATION

Health informatics—Personal health device communication

Part 20601: Application profile—Optimized Exchange Protocol

Amendment 1

IEEE Engineering in Medicine and Biology Society

Sponsored by the
IEEE 11073™ Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

24 January 2011

IEEE Std 11073-20601a™-2010
(Amendment to
IEEE Std 11073-20601™-2008)

IEEE Std 11073-20601a™-2010
(Amendment to
IEEE Std 11073-20601™-2008)

Health informatics—Personal health device communication

Part 20601: Application profile—Optimized Exchange Protocol

Amendment 1

Sponsor

IEEE 11073™ Standards Committee

of the

IEEE Engineering in Medicine and Biology Society

Approved 30 September 2010

IEEE-SA Standards Board

Approved 8 June 2011

American National Standards Institute

Abstract: Issues found within IEEE Std 11073-20601-2008 while implementing the standard and/or testing and certifying products are addressed in this amendment.

Keywords: 11073, medical device communication, personal health devices

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-7381-6462-5 STD97019
Print: ISBN 978-0-7381-6463-2 STDPD97019

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "AS IS."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Introduction

This introduction is not part of IEEE Std 11073-20601a-2010, Health informatics—Personal health device communication—Part 20601: Application profile—Optimized Exchange Protocol—Amendment 1.

This amendment addresses issues found while implementing the standard and/or testing and certifying products.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Personal Health Devices Working Group had the following membership:

Douglas P. Bogia, *Chair*

Manfred Aigner
Juan García-Prieto Cuesta
Karsten Aalders
Charles R. Abbruscato
Maher Abuzaid
Jon Adams
Murtaza Ali
Rolf Ambuehl
Lawrence Arne
Serafin Arroyo
Kevin Arruda
Muhammad Asim
Merat Bagha
Doug Baird
David Baker
Anindya Bakshi
Ananth Balasubramanian
Sunlee Bang
M Jonathan Barkley
David Bean
John Bell
Rudy Belliardi
Yazid Benazzouz
George A. Bertos
Ola Björsne
Thomas Blackadar
Marc Blanchet
Thomas Bluethner
Funieru Bogdan
Xavier Boniface
Shannon Boucousis
Julius Broma
Lyle G. Bullock, Jr.
Bernard Burg
Chris Burns
Anthony Butt
Jeremy Byford-Rew
Carole C. Carey
Santiago Carot-Nemesio
Randy W. Carroll
Rahul Chauhan
James Cheng
Maria Cherkaoui
Peggy Chien
Silviu Chiricescu
Jiho Choi
Chia-Chin Chong
Jinhan Chung
Malcolm Clarke
Rick A. Cnossen
Moshe Cohen
John T. Collins
Cory Condek
Todd H. Cooper
Nigel Cox
Tomio Crosley
Jeremy Cummin
Allen Curtis
Rupert Dance
Jesús Daniel Trigo
Sushil K. Deka
Pedro de-las-Heras-Quiros
Jim DelloStritto
Matthew d'Entremont
Kent Dicks
Hyoungho Do
Amy Droitcour
Brian Dubreuil
Jakob Ehrensvard
Fredrik Einberg
Roger M. Ellingson
Michihiro Enokida
Javier Escayola Calvo
Leonardo Estevez
Roger Feeley
Bosco T. Fernandes
Gear Fisher
Julie N. Fleischer
Roger Florkowski
Joseph W. Forler
Russell Foster
Peter Fredriksson XX
Eric Freudenthal
Matthias Frohner
Miguel Galarraga
Marcus Garbe
John Garguilo
Rick Geimer
Igor Gejdos
Ferenc Gerbovics
Nicolae Goga
Julian Goldman
Raul Gonzalez Gomez
Chris Gough
Channa Gowda
Niclas Granqvist
Charles Gropper
Amit Gupta
Jeff Guttmacher
Christian Habermann
Mikey Hagerty
Jerry Hahn
Robert Hall
Rickey L. Hampton
Sten Hanke
Kai Hassing
Marc Daniel Haunschild
Hiroshi Hayashi
Mike Hayes
Mike Hayes
Timothy L. Hirou
Allen Hobbs
Alex Holland
Arto Holopainen
Robert Hoy
Frank Hsu
Sen-Der Huang
David Hughes
Robert D. Hughes
Nick Hunn
Hugh Hunter
Hitoshi Ikeda
Yutaka Ikeda
Philip O. Isaacson
Atsushi Ito
Praduman Jain
Ho-In Jeon
Danny Jochelson
Chris Johnson
Phaneeth Junga
Akiyoshi Kabe
Steve Kahle
Tomio Kamioka
Kyung Hee Kang
Andy Kaschl
Junzo Kashihara
Kohichi Kashiwagi
Ralph Kent
Kurt M. Kermes
Ikuo Keshi
John Keys
Minho Kim
Taekon Kim
Tetsuya Kimura
Michael Kirwan
Alfred Kloos
Jeongmee Koh
Jean-Marc Koller
Thomas Kottler
Alexander Kraus
Ramesh Krishna
Falko Kuester
Pierre Landau
Rami Lee
Sungkee Lee
Woojae Lee

Yonghee Lee
Kathryn A. Lesh
Qiong Li
Patrick Lichter
Joon-Ho Lim
John Lin
Wei-Jung Lo
Charles Lowe
Jeremy Martin
Sandra Martinez
Miguel Martínez
de Espronceda Cámara
Dennis Mathews
Peter Mayhew
Jim McCain
Richard McPartland
László Meleg
Jinsei Miyazaki
Darr Moore
Joe Morrissey
Piotr Murawski
Soundharya Nagasubramanian
Jae-Wook Nah
Alex Neefus
Michael E. Nidd
Tetsu Nishimura
Jim Niswander
Hiroaki Niwamoto
Thomas Norgall
Yoshiteru Nozoe
Brett Olive
Charles Palmer
Bud Panjwani
Carl Pantiskas
Mikey Paradis
Jong-Tae Park
Soojun Park
Jayant Parthasarathy
Phillip E. Pash
TongBi Pei
Soren Petersen
Arnie Pittler
Carol Primdahl
Arif Rahman
Barry Reinhold
Brian Reinhold
Melvin I. Reynolds
Jeffrey S. Robbins
Scott Robertson
Timothy Robertson
David Rosales
Don Rosen
Paivi M. Ruuska
Bill Saltzstein
Stefan Sauermann
John Sawyer
Paul S. Schluter
Johannes Schmidt
Lars Schmitt
Mark G. Schnell
Richard A. Schrenker
Antonio Scorpiniti
Kwang Seok Seo
Riccardo Serafin
Frank Shen
Min Shih
Mazen Shihabi
Tommy Shing
Krishna Shingala
Robert Smith
Ivan Soh
Motoki Sone
Emily Sopensky
Andreas Staubert
Nick Steblay
Lars Steubesand
John Stivoric
Raymond Strickland
Hermanni Suominen
Lee Surprenant
Ravi Swami
Ray Sweidan
Kunihiro Takiuchi
Francis Tam
Haruyuyki Tatsumi
Randy Thomas
Brad Tipler
Jonas Tirén
James Tomcik
Gary Tschautscher
Masato Tsuchid
Ken Tubman
Yoshihiro Uchida
Sunil Unadkat
Philipp Urbauer
Laura Vanzago
Alpo Värri
Daniel von Büren
Rudi Voon
Isobel Walker
David Wang
Jerry P. Wang
Yao Wang
Steve Warren
Fujio Watanabe
Toru Watsuji
Jeff Webber
Eric White
Paul Williamson
Jan Wittenber
Jia-Rong Wu
Will Wykeham
Ariton Xhafa
Ricky Yang
Chi-Lu Yang (III)
Steven Yecies
Done-Sik Yoo
Jason Zhang
Thomas Zhao
Daidi Zhong
Szymon Zysko

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Vered Bar
H Stephen Berger
Christopher Biernacki
Douglas P. Bogia
Lyle Bullock
Randy W. Carroll
Keith Chow
Malcolm Clarke
Fredrik Einberg
Niclas Granqvist
Randall Groves
Kai Hassing
Werner Hoelzl
Atsushi Ito
Piotr Karocki
Kurt M. Kermes
Bruce Kraemer
G. Luri
Melvin I. Reynolds
Bartien Sayogo
Lars Schmitt
Gil Shultz
James Smith
Lars Steubesand
Jonas Tirén
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 30 September 2010, it had the following membership:

Robert M. Grow, *Chair*
Richard H. Hulett, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Karen Bartleson
Victor Berman
Ted Burse
Clint Chaplin
Andy Drozd
Alexander Gelman
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Ronald C. Petersen
Thomas Prevost
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, *NRC Representative*

Richard DeBlasio, *DOE Representative*

Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Kathryn M. Bennett
IEEE Standards Program Manager, Technical Program Development

Contents

Health informatics—Personal health device communication

Part 20601: Application profile—Optimized Exchange Protocol

Amendment 1

NOTE—The editing instructions contained in this corrigendum define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.[1]

1.3 Context

Insert the following text between the current sixth and seventh paragraphs:

Some device specializations describe broad categories of device types (e.g., the IEEE 11073-10441 model device types that promote cardiovascular activity such as step counters or exercise cycles). Other device

[1] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

specializations have a narrow focus on a single device type (e.g., IEEE 11073-10408 model thermometers). Specializations that address one or more device types may also define *profiles*. A profile further constrains the model defined in a specialization to increase interoperability (e.g., the step counter profile utilizes a limited portion of IEEE 11073-10441 modeling).

The ISO/IEEE P11073-00103 [B8][1] technical report describes the overall personal health space with further definition of the underlying use cases and usage models.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

Change the first paragraph and insert new footnote as shown. Delete [B6] from Annex K and renumber subsequent references in Annex K:

For the purposes of this standard, the following terms and definitions apply. ~~*The Authoritative Dictionary of IEEE Standards* [B6]~~*The IEEE Standards Dictionary: Glossary of Terms & Definitions* should be ~~referenced~~ consulted for terms not defined in this clause.[2]

Changes existing definitions and insert new definitions as shown:

~~**3.1.1 agent:**~~ A node that collects and transmits personal health data to an associated manager.

attribute: Data representing a property of an object. Attributes, together with actions, define an object.

AttributeChangeSet: The set of attribute value changes that represents an atomic update of an object. The MDS or scanner is notified when an AttributeChangeSet is completed. Collections of these AttributeChangeSets are mapped by the MDS or scanner into one of the ObservationScan structures in a scan event report that gets sent to the manager. The manager updates its object with the set of attribute value changes contained in the ObservationScan before deriving any semantic behavior.

~~**3.1.2 compute engine:**~~ *See:* **manager**.

~~**3.1.3 confirmed:**~~ An application-level, completion notification service mechanism. For EVENT REPORT services (i.e., the data plane), confirmation allows the agent to know when the manager has “accepted responsibility” for a piece of data so that the agent can delete that data. For the ACTION, GET, and SET services (i.e., the control plane), confirmation allows the manager to know when the agent has “completed” the requested transaction.

~~**3.1.4 device:**~~ A physical device implementing either an agent or manager role.

dynamic attribute: An attribute with a value that may change during an association. The attribute value should be sent at configuration time and shall be sent at or before the time when the value would be needed for interpreting a reported observation. The value may be updated later (e.g., in a scan or segment data event report). The attribute value remains in effect until updated by a later scan or segment data event report or the system exits the association.

~~**3.1.5 handle:**~~ An unsigned 16-bit number that is locally unique and identifies one of the object instances within an agent.

[1] The numbers in brackets correspond to the numbers of the bibliography in Annex K.
[2] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

~~3.1.6~~ **manager:** A node receiving data from one or more agent systems. Examples of managers include a cellular phone, health appliance, set top box, or computer system.

metric: A metric object models different forms of measurements.

observational attribute: An attribute value that may change during the life of an association. The value may be sent in a scan or segment data event report. When a set of observational attribute values are received, these values are combined with the available context information (i.e., all related dynamic and static attribute values) to represent the observation at the observation time. Unlike dynamic and static attribute values, the observational attribute values are combined only once with context information (i.e., the observational attribute values are not reused when any new attribute values are received in the future).

object: A unit that represents some functionality or item in a device whose properties are described by attributes. Metric objects represent measurements (such as blood pressure, weight, or temperature), the MDS represents the device, PM-store objects represent the permanent storage mechanisms in an agent, and Scanners represent a control and reporting mechanism.

~~3.1.7~~ **personal health device:** A device used in personal health applications.

~~3.1.8~~ **personal telehealth device:** *See:* personal health device.

static attribute: An attribute value that does not change (its value remains fixed) during the life of an association. The value is sent in the configuration event report. The value remains in effect until the system exits the associated state.~~An attribute value that does not change (its value remains fixed) during the life of an association.~~

NOTE—Do not confuse the use of static in this document with static as used in the C programming language.

3.2 Acronyms and abbreviations

Change as shown and insert the following new lines in alphabetical order:

ECG	electrocardiogram or electrocardiograph
FIFO	first-in, first-out
PM	persistent metric
UTC	coordinated universal time ~~coordinated~~

6. Personal health device DIM

6.2 Nomenclature usage

Change Table 1 as shown:

Table 1—Partitions in the nomenclature

Partition number	Nomenclature category
0	Unspecified
1	Object-oriented (OO)
2	Supervisory control and data acquisition (SCADA)
3	Events
4	Dimensions (units of measurement)
5	Virtual attributes
6	Parameter groups
7	[Body] sites
8	Infrastructure
9	File Exchange Format
10	ECG Extensions
11	IDCO Extensions
12~~9~~–127	Reserved
128	Personal health devices disease management
129	Personal health devices health and fitness
130	Personal health devices aging independently
131–25~~4~~	Reserved
255	Return codes
256	External nomenclature references
257–1023	Reserved
1024	Private
1025–65 535	Reserved

6.3 Personal health object class definitions

6.3.1 General

Replace Figure 4 with the following figure:

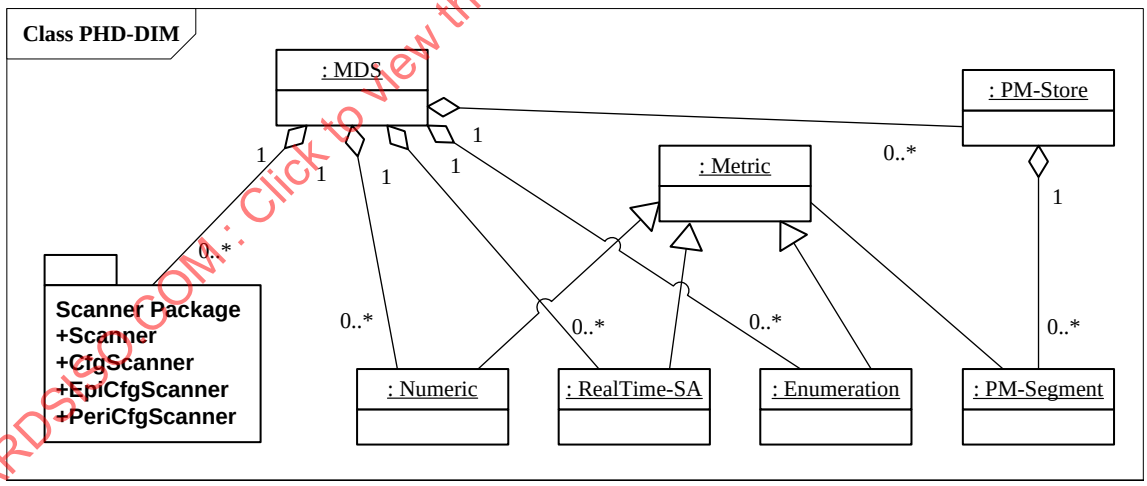


Figure 4—Personal health device—DIM

Change the following paragraphs as shown:

The attributes for each class are defined in tables that specify the name of the attribute, its nomenclature reference ID, its type, a description of the attribute, and its qualifiers. The qualifiers Conditional, Optional, and Mandatory identify when the attribute is to be implemented in the object. A conditional attribute is

implemented based~~The qualifiers mean O — Attribute is Optional, M — Attribute is Mandatory, and C — Attribute is Conditional and depends~~ on the condition(s) stated in the Remark column. Conditional attributes shall be implemented if the ~~condition~~ remark applies and may be implemented otherwise. Optional attributes may be implemented ~~on~~ by an agent. Mandatory attributes shall be implemented by an agent.

An attribute is further qualified as static, dynamic, or observational. Static attributes shall not change value during the life of an association. The value is sent in the configuration event report. The value remains in effect until the system exits the associated state. ~~The attribute shall have a fixed value from the time the operating state is reached until the system exits the associated state.~~Dynamic attributes have a value that may change during the life of an association. The attribute value should be sent at configuration time and shall be sent at or before the time when the value would be needed for interpreting a reported observation. The value may be updated later (e.g., in a scan or segment data event report). The attribute value remains in effect until updated by a later scan or segment data event report ~~by a later scan report (a state)~~or the system exits the association~~d state~~. ~~Attributes are further qualified based on the persistence of the attribute when reported to a manager. A persistent attribute has a value that remains valid until that value is updated (a state).~~ Observational attributes have a value that may change during the life of an association. The value may be sent in a scan or segment data event report. When a set of observational attribute values are received, these values are combined with the available context information (i.e., all related dynamic and static attribute values) to represent the observation at the observation time. Unlike dynamic and static attribute values, the observational attribute values are combined only once with context information (i.e., the observational attribute values are not reused when any new attribute values are received in the future). In the metric object tables, attributes that are labeled as observations may be addionally flagged as a (1) setting, (2) manual entry, or (3) calculation entirely from manual and/or setting entry attributes. In those three cases, the attribute is to be treated by the manager as dynamic. The agent sets these flags in the Metric-Spec-Small attribute. The attribute types are summarized in Table 1a.

Insert the following new table:

Table 1a—Attribute types

Attribute type	Can be reported by…	Attribute value remains in effect until…
dynamic attribute may change during the life of an association	• Should be reported in a configuration event report and shall be sent at or before the time when the value would be needed for interpreting a report observation • May be reported in a scan or segment data event report	• updated by a later scan or segment data event report • the system exits the association state
observational attribute may change during the life of an association	• Shall be reported in a scan or segment data event report	• these values are combined with the available context information (i.e., all related dynamic and static attribute values) to represent the observation at the observation time.
static attribute shall not change during the life of an association	• Shall be reported in the configuration event report	• the system exits the association state

6.3.2 MDS class

6.3.2.3 MDS class attributes

Change Table 2 as shown and insert the following text after the table:

Table 2—MDS attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Handle	MDC_ATTR_ID_ HANDLE	HANDLE	The Handle attribute represents a reference ID for this object. The value of the MDS Handle attribute shall be 0.	Mandatory Static
System-Type	MDC_ATTR_SYS_TYPE	TYPE	This attribute defines the type of the agent, as defined in nomenclature (e.g., weighing scale). The values shall come from ISO/IEEE 11073-10101 [B12], nom-part-object partition, and subpartition MD-Gen (Medical Device – Generic). ~~Either this attribute or~~One and only one of System-Type or System-Type-Spec-List shall be present. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Conditional Static
System-Model	MDC_ATTR_ID_MODEL	SystemModel	This attribute defines manufacturer and model number of the agent device. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static

Table 2—MDS attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
System-Id	MDC_ATTR_SYS_ID	OCTET STRING	This attribute is an IEEE EUI-64, which consists of a 24-bit unique organizationally unique identifier (OUI) followed by a 40-bit manufacturer-defined ID. The OUI shall be a value assigned by the IEEE Registration Authority (http://standards.ieee.org/regauth/index.html) and shall be used in accordance with IEEE Std 802-2001.[3] ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Dev-Configuration-Id	MDC_ATTR_DEV_ CONFIG_ID	ConfigId	This attribute defines the identification of the agent device configuration. This Dev-Configuration-Id is static during the lifetime of an association; it is normally exchanged during the association procedure. The manager can GET this attribute during operation. If this attribute is queried prior to when the agent and manager agree upon a configuration, the agent shall return the configuration ID that is being offered at that time. For more information on this attribute, see 8.8. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Attribute-Value-Map	MDC_ATTR_ ATTRIBUTE_VAL_MAP	AttrValMap	This attribute defines the attributes that are reported in the fixed format data update messages (see 7.4.5 for more information). ~~Usage of this attribute is mandatory i~~If the agent ~~device~~ uses fixed format value messages to report dynamic data for the object, this attribute shall be set prior to sending such a report. This attribute value may change ~~in~~ between sending such event reports and shall be sent in an event report if it changes while associated.	Conditional Dynamic
Production-Specification	MDC_ATTR_ID_PROD_ SPECN	ProductionSpec	This attribute defines component revisions, serial numbers, and so on in a manufacturer-specific format. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Optional Static
Mds-Time-Info	MDC_ATTR_MDS_TIME _ INFO	MdsTimeInfo	This attribute defines the time handling capabilities and the status of the MDS. Usage of this attribute is required if synchronization or settable time is supported.	Conditional Dynamic
Date-and-	MDC_ATTR_TIME_ABS	AbsoluteTime	This attribute defines the date and	Conditional

[3] For information on references, see Clause 2.

Table 2—MDS attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Time			time of an agent with resolution of 1/100 of a second, if available. For more information on this attribute, see 8.12. If the agent reports AbsoluteTime in any ~~other~~ message, it shall report its current value of AbsoluteTime in this attribute. If this attribute is used the Base-Offset-Time attribute shall not be used.	Observational
Base-Offset-Time	MDC_ATTR_TIME_BO	BaseOffsetTime	This attribute defines the date and time of an agent as a base time with the offset in minutes to local time. For more information on this attribute, see 8.12. If the agent reports BaseOffsetTime in any message, it shall report its current value of base time in this attribute. If this attribute is used the Date-and-Time attribute shall not be used.	Conditional Observational
Relative-Time	MDC_ATTR_TIME_REL	RelativeTime	If the agent reports RelativeTime in any ~~other~~ message, it shall report its current value of RelativeTime in this attribute.	Conditional Observational
HiRes-Relative-Time	MDC_ATTR_TIME_REL_HI_RES	HighResRelativeTime	If the agent reports HighResRelativeTime in any ~~other~~ message, it shall report its current value of HighResRelativeTime in this attribute.	Conditional Observational
Date-and-Time-Adjustment	MDC_ATTR_TIME_ABS_ADJUST	AbsoluteTimeAdjust	This attribute reports any date and time adjustments that occur either due to a person's changing the clock or events such as daylight savings time. This is used in event reports only. If queried with Get MDS Object command, this value shall be not present or 0. If the agent ever adjusts the date and time, this attribute is used in an event report to report such adjustment.	Conditional Observational
Power-Status	MDC_ATTR_POWER_STAT	PowerStatus	This attribute reports whether power is being drawn from battery or main power lines and the status of charging.	Optional Dynamic
Battery-Level	MDC_ATTR_VAL_BATT_CHARGE	INT-U16	This attribute reports the percentage of battery capacity remaining, which is undefined if value > 100.	Optional Observational
Remaining-Battery-Time	MDC_ATTR_TIME_BATT_REMAIN	BatMeasure	This attribute represents the predicted amount of operational time left on the batteries. The BatMeasure's unit shall be set to one of MDC_DIM_MIN, MDC_DIM_HR, or MDC_DIM_DAY for minutes, hours, or days, respectively.	Optional Observational
Reg-Cert-Data-List	MDC_ATTR_REG_CERT_DATA_LIST	RegCertDataList	This attribute lists various regulatory and/or certification compliance items to which the agent claims adherence as an informative statement. The	Optional Static

Table 2—MDS attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
			Implementation Conformace Statements (see Clause 9) take precedence over this attribute and are the legally binding claims. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	
System-Type-Spec-List	MDC_ATTR_SYS_TYPE_SPEC_LIST	TypeVerList	This attribute reports the type(s) of the agent, as defined in nomenclature (e.g., weighing scale). The values shall come from ISO/IEEE 11073-10101 [B12], nom-part-infrastruct partition, subpartition DEVspec, and reference ISO/IEEE 11073-104zz specializations. If an agent does not follow any specialization, the list shall be left blank. If an agent follows a profile within a specialization, it shall report both the specialization and the profile nomenclature values in this list. This list shall also contain the version of the specialization(s)/profile(s). One and only one of System-Type or System-Type-Spec-List shall be present. ~~Either this attribute or System-Type shall be present.~~ If the agent is multifunction, this attribute shall be present. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Conditional Static
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	This informational timeout attribute defines the minimum time that the agent shall wait for a Response message from the manager after issuing a Confirmed Event Report invoke message before timing out and transitioning to the Unassociated state. This is an informational attribute for the benefit of the manager. If this attribute is supplied, it shall match the actual timeout value that the agent uses for the Confirmed Event Report generated from the MDS object. This attribute is informational for the manager in the sense that the manager does not use this attribute in an actual implementation of the protocol (i.e., the manager does not time out on an agent-generated Confirmed Event Report). However, the manager might wish to use this information to prioritize its handling	Optional Dynamic

Table 2—MDS attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
			of a “short” timeout agent over that of a “long” timeout agent. Currently the attribute is Dynamic, but agent implementations should treat as Static since it is likely this will change to Static in the future. If this attribute value is changed, it shall be changed only when the scanner is disabled. The attribute value change is communicated to the manager using an agent-initiated event report from the MDS object.	

Note that the attribute System-Type is independent of the similarly named System-Type field in the PhdAssociationInformation structure of an association request.

The MDS attributes represent a device level view and are independent of the particular configuration that has been offered or accepted. For instance, the System-Type-Spec-List or Reg-Cert-Data-List provide all capabilities that the device may offer. The current configuration may or may not provide everything listed in the attribute.

6.3.2.4 MDS object methods

Insert a new row in Table 3 as shown:

Table 3—MDS object methods

Method/Action	Mode	Action-type	action-info-args	Resulting action-info-args
MDS-Data-Request	Confirmed	MDC_ACT_DATA_REQUEST	DataRequest	DataResponse
Set-Time	Confirmed	MDC_ACT_SET_TIME	SetTimeInvoke	None
Set-Base-Offset-Time	Confirmed	MDC_ACT_SET_BO_TIME	SetBOTimeInvoke	None

Insert the following text:

— **Set-Time:**
This method allows the manager system to set a real-time clock (RTC) with the absolute time. The agent indicates whether the Set-Time command is valid by using the mds-time-capab-set-clock bit in the Mds-Time-Info attribute (see Table 2). If the agent supports Set-Time, it shall respond with a rors-cmip-confirmed-action, but the action-info-args is empty in this response. If the agent does not support Set-Time, it shall respond with a no-such-action error (roer).

— **Set-Base-Offset-Time:**
This method allows the manager system to set a real-time clock (RTC) with a base time and the offset in minutes to local time. The agent 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 (see Table 2). If the agent supports Set-Base-Offset-Time, it shall respond with a rors-cmip-confirmed-action, but the action-info-args is empty in this response. If the agent does not support Set-Base-Offset-Time, it shall respond with a no-such-action error (roer). 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. An agent may support either the absolute time or the base time but not both. 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.

An agent may support either absolute time or base time but not both.MDS object events.

6.3.2.5 MDS object events

Change Table 4 as shown:

Table 4—MDS object events

Event	Mode	Event-type	Event-info parameter	Event-reply-info
MDS-Configuration-Event	Confirmed ~~or unconfirmed~~	MDC_NOTI_CONFIG	ConfigReport	ConfigReportRsp
MDS-Dynamic-Data-Update-Var	Confirmed or unconfirmed	MDC_NOTI_SCAN_REPORT_VAR	ScanReportInfoVar	-
MDS-Dynamic-Data-Update-Fixed	Confirmed or unconfirmed	MDC_NOTI_SCAN_REPORT_FIXED	ScanReportInfoFixed	-
MDS-Dynamic-Data-Update-MP-Var	Confirmed or unconfirmed	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	-
MDS-Dynamic-Data-Update-MP-Fixed	Confirmed or unconfirmed	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	-

6.3.2.6 Other MDS services

6.3.2.6.1 GET service

Change the subclause and insert new text as shown:

Any agent supporting two-way communication links shall support the GET service to retrieve the values of all implemented MDS object attributes. The GET service can be invoked as soon as the agent receives the Association Response and moves to the Associated state, including the Operating and Configuring substates. With the exception of the date-time-adjustment attribute, if a manager does not have the current value of a needed MDS attribute, then the GET service shall be used. An agent may also send scan event reports providing the manager with updates of the current attribute values, but this is not a mandated agent behavior with the exceptions described for attribute-value-map and date-and-time-adjustment in Table 2.

The manager may request the MDS object attributes of the agent in which case the manager shall send the "Remote Operation Invoke | Get" command (see roiv-cmip-get in A.10.2) with the reserved handle value of 0. The agent shall respond by reporting its MDS object attributes to the manager using the "Remote Operation Response | Get" response (see rors-cmip-get in A.10.2). In the response to a Get MDS Object command, only attributes implemented by the agent are returned. See 8.9.3.2 for a full explanation of the GET operation.

NOTE—Requirements, such as the need for an agent to request a manager to set its time, may necessitate that a manager request the MDS object from an agent immediately after the manager has entered the Associated state (see 8.12.2).

Due to the nature of the attributes reported in the MDS Object, the manager should send the Get MDS Object request right after sending the Association Response.

6.3.2.6.2 SET service

Change the following paragraph as shown:

There are currently no ~~settable attributes~~uses of the MDS SET service defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.

6.3.3 Metric class

6.3.3.3 Metric class attributes

Change Table 5 as shown:

Table 5—Metric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Handle	MDC_ATTR_ID_HANDLE	HANDLE	The Handle attribute represents a reference ID for this object. Each object shall have a unique ID assigned by the agent. The handle identifies the object in event reports sent to the manager. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Type	MDC_ATTR_ID_TYPE	TYPE	This attribute defines a specific static type of this object as defined in the nomenclature (e.g., pulse rate for a specific numeric object instance). The Type attribute contains the nomenclature partition and term code IDs for context-free, extensible identification. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Supplemental-Types	MDC_ATTR_ SUPPLEMENTAL_TYPES	SupplementalTypeList	This attribute may be used to convey supplemental information about the object beyond the Type and Metric-Id attributes. Supplemental information covers conditions like the location of the sensor or the rate at which the object reacts to changes. Device specializations define the expected usage of this attribute. For example, IEEE Std 11073-10471 [B5] defines location nomenclature for specifying the location of a sensor in a home and ISO/IEEE P11073-10404 [B9] defines three supplemental types for fast response, slow response, and spot checking of the pulse rate or blood oxygenation. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Optional Static

Table 5—Metric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Metric-Spec-Small	MDC_ATTR_METRIC_SPEC_SMALL	MetricSpecSmall	This attribute describes the characteristics of the measurements. This attribute is dynamic to allow an agent to alter the bit settings prior to sending the first observation (e.g., to adjust a standard configuration). Once an observation is sent the agent shall not update this attribute's value.	Mandatory Dynamic
Metric-Structure-Small	MDC_ATTR_METRIC_STRUCT_SMALL	MetricStructureSmall	This attribute describes the structure of the measurement. If not present, the manager shall assume MetricStructureSmall := {ms-struct-simple, 0}. This attribute is dynamic to allow an agent to alter the settings prior to sending the first observation (e.g., to adjust a standard configuration). Once an observation is sent the agent shall not update this attribute's value.	Optional Dynamic
Measurement-Status	MDC_ATTR_MSMT_STATUS	MeasurementStatus	This attribute indicates the validity of a particular value or set of samples. If a scanner reports an RTSA and there is a possibility that the scanner may need to report missing observations (see 6.3.9.2) this attribute shall be present.	~~O~~Conditional Observational

Table 5—Metric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Metric-Id	MDC_ATTR_ID_PHYSIO	OID-Type	This attribute may be used to hold an identification that is more specific than the generic ID in the Type attribute. If the Metric-Id-Partition attribute is valued, it defines the nomenclature partition for this attribute. Otherwise, the OID-Type is taken from the same nomenclature partition as defined in the partition field of the Type attribute. This attribute is needed only if identification changes during operation and the Type attribute does not contain full identification. For example, if the Type attribute contains a generic temperature code (MDC_TEMP), this attribute could report a specific, but changing, identification such as MDC_TEMP_ORAL or MDC_TEMP_RECT. Only one attribute of Metric-Id and Metric-Id-List shall be present.	Optional Dynamic
Metric-Id-List	MDC_ATTR_ID_PHYSIO_ LIST	MetricIdList	This attribute shall be used if a compound observed value is used and does not incorporate the Metric-Id directly (e.g., Compound-Simple-Nu-Observed-Value or Compound-Basic-Nu-Observed-Value) so that elements in the observed value list can be identified individually. The order of the Metric-Id-List shall correspond to the order of the elements in the compound observed value. Only one attribute of Metric-Id and Metric-Id-List shall be present.	Conditional Dynamic
Metric-Id-Partition	MDC_ATTR_METRIC_ID_ PART	NomPartition	This attribute may be used to define the partition from which the Metric-Id or Metric-Id-List nomenclature terms were taken. If not present, the partition is the same as the nomenclature partition defined in the partition field of the Type attribute.	Optional Dynamic
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	This attribute defines the nomenclature code for the units of measure from the nom-part-dim partition (e.g., MDC_DIM_KILO_G).	Optional Dynamic

Table 5—Metric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Attribute-Value-Map	MDC_ATTR_ATTRIBUTE_VAL_MAP	AttrValMap	This attribute defines the attributes that are reported in the fixed format data update messages. ~~Usage of this attribute is mandatory i~~If the agent uses fixed format value messages to report dynamic measurement data for the object, this attribute shall be set prior to sending such a report. This attribute value may change in between event reports.	Conditional Dynamic
Source-Handle-Reference	MDC_ATTR_SOURCE_HANDLE_REF	HANDLE	This attribute establishes a relation of this object instance to a source object (e.g., pulse references sourcing SpO2). This attribute is used whenever it is required to model an explicit relation between object instances to define dependencies. The usage of this attribute is defined by device specializations.	Optional Dynamic
Label-String	MDC_ATTR_ID_LABEL_STRING	OCTET STRING	This attribute defines the textual representation of Type attribute in printable ASCII. The value of this attribute is at the complete discretion of the agent's manufacturer. It could potentially be useful for a manager as a display string or as an aid in deciding how to behave when it does not understand the MDC_ATTR_ID_TYPE as reported by the agent.	Optional Dynamic
Unit-LabelString	MDC_ATTR_UNIT_LABEL_STRING	OCTET STRING	This attribute defines the textual representation of Unit-Code dimension in printable ASCII. The value of this attribute is at the complete discretion of the agent's manufacturer. It could potentially be useful for a manager as a display string or as an aid in deciding how to behave when it does not understand the MDC_ATTR_UNIT_CODE as reported by the agent.	Optional Dynamic

Table 5—Metric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Absolute-Time-Stamp	MDC_ATTR_TIME_STAMP_ABS	AbsoluteTime	This attribute defines the date and time of observation with resolution of 1/100 of a second, if available. For more information on this attribute, see 8.12. If an agent stores data (either in a PM-store object or as a "temporarily stored measurement"), it shall associate a time stamp (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp, or HiRes-Time-Stamp) with the data. If this attribute is used the Base-Offset-Time-Stamp attribute shall not be used.	Conditional Observational
Base-Offset-Time-Stamp	MDC_ATTR_TIME_STAMP_BO	BaseOffsetTime	This attribute defines the base date and time of observation and the offset in minutes to local time. For more information on this attribute, see 8.12. If an agent stores data (either in a PM-store object or as a "temporarily stored measurement"), it shall associate a time stamp (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp, or HiRes-Time-Stamp) with the data. If this attribute is used, the Absolute-Time-Stamp attribute shall not be used.	Conditional Observational
Relative-Time-Stamp	MDC_ATTR_TIME_STAMP_REL	RelativeTime	This attribute defines the time of observation (time stamp in a relative time format/number of clock ticks as defined by RelativeTime data type). If an agent stores data, it shall associate a time stamp (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp, or HiRes-Time-Stamp) with the data.	Conditional Observational
HiRes-Time-Stamp	MDC_ATTR_TIME_STAMP_REL_HI_RES	HighResRelative Time	This attribute defines the time of observation (time stamp in a high-resolution relative time format/number of clock ticks as defined by a HighResRelativeTime data type). If an agent stores data, it shall associate a time stamp (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp, or HiRes-Time-Stamp) with the data.	Conditional Observational
Measure-Active-Period	MDC_ATTR_TIME_PD_MSMT_ACTIVE	FLOAT-Type	This attribute defines the time duration of the observation period in seconds.	Optional Dynamic

6.3.3.4 Metric object methods

Change as shown:

There are currently no metric object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.

~~None~~

6.3.3.6 Other metric services

Change as shown:

There are currently no uses of the metric SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.

~~None~~

6.3.4 Numeric class

6.3.4.3 Numeric class attributes

Change Table 6 as shown:

Table 6—Numeric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Simple-Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS_SIMP	SimpleNuObsValue	This attribute defines the numerical observed value of the object, without any further embedded status information as found in Nu-Observed-Value. One and only one of Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, or Compound-Basic-Nu-Observed-Value shall be present.	Conditional Observational

Table 6—Numeric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Compound-Simple-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS_SIMP	SimpleNuObsValueCmp	This attribute represents an array of Simple-Nu-Observed-Values. One and only one of Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, or Compound-Basic-Nu-Observed-Value shall be present.	Conditional Observational
Basic-Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS_BASIC	BasicNuObsValue	This attribute defines the numerical observed value of the object, without any further embedded status information, but with a smaller numerical representation compared to Simple-Nu-Observed-Value. One and only one of Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, or Compound-Basic-Nu-Observed-Value shall be present.	Conditional Observational
Compound-Basic-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS_BASIC	BasicNuObsValueCmp	This attribute represents an array of Basic-Nu-Observed-Values. One and only one of Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, or Compound-Basic-Nu-Observed-Value shall be present.	Conditional Observational
Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS	NuObsValue	This attribute defines the numerical observed value of the object and combines it with measurement status and unit information. It is used when status/unit are dynamic and are always provided together with the value. One and only one of Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, or Compound-Basic-Nu-Observed-Value shall be present.	Conditional Observational

Table 6—Numeric attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Compound-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS	NuObsValueCmp	This attribute combines an array of value, status, and unit. This attribute is available for use only in variable format event reports. One and only one of Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value, or Compound-Basic-Nu-Observed-Value shall be present.	Conditional Observational
Accuracy	MDC_ATTR_NU_ACCUR_MSMT	FLOAT-Type	This attribute defines the maximum deviation of actual value from reported observed value (if it can be specified). ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Optional Static

6.3.4.4 Numeric object methods

Change as shown:

There are currently no numeric object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.4.5 Numeric object events

Change as shown:

There are currently no numeric object events defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.4.6 Other numeric services

Change as shown:

There are currently no uses of the numeric SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.5 RT-SA class

6.3.5.3 RT-SA class attributes

Change Table 7 as shown:

Table 7—RT-SA attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Sample-Period	MDC_ATTR_TIME_PD_SAMP	RelativeTime	This attributes defines time interval between successive samples given in 1/8 of a millisecond. Thus, 8000 = 1 s. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Simple-Sa-Observed-Value	MDC_ATTR_SIMP_SA_OBS_VAL	OCTET STRING	This byte array contains the samples that are reported by the agent in the format that is described by the Sa-Specification and Scale-and-Range-Specification. The length shall be even with padding bytes at the end. Sa-Specification defines the actual number of utilized bytes.	Mandatory Observational
Scale-and-Range-Specification	MDC_ATTR_SCALE_SPECN_I8 MDC_ATTR_SCALE_SPECN_I16 MDC_ATTR_SCALE_SPECN_I32	ScaleRangeSpec8 ScaleRangeSpec16 ScaleRangeSpec32	This attribute defines mapping between samples and actual values as well as measurement range. The type depends on sample resolution (sample-size field within sample-type field of Sa-Specification). Exactly one of the three specifications shall be included. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Sa-Specification	MDC_ATTR_SA_SPECN	SaSpec	This attribute describes the sample array and sample types. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static

6.3.5.4 RT-SA object methods

Change as shown:

There are currently no RT-SA object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.5.5 RT-SA object events

Change as shown:

There are currently no RT-SA object events defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.5.6 Other RT-SA services

Change as shown:

There are currently no uses of the RT-SA SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.6 Enumeration class

6.3.6.3 Enumeration class attributes

Change Table 8 as shown:

Table 8—Enumeration attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Enum-Observed-Value-Simple-OID	MDC_ATTR_ENUM_OBS_VAL_SIMP_OID	OID-Type	The value is reported as a nomenclature code. If the Enum-Observed-Value-Partition attribute is valued, it defines the nomenclature partition for this attribute. Otherwise, the OID-Type is taken from the same nomenclature partition as defined in the partition field of the Type attribute. One and only one of Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str , Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str, or Enum-Observed-Value shall be present.	Conditional Observational
Enum-Observed-Value-Simple-Bit-Str	MDC_ATTR_ENUM_OBS_VAL_SIMP_BIT_STR	BITS-32	The value is reported as a bit string of 32-bits. One and only one of Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str , Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str, or Enum-Observed-Value shall be present.	Conditional Observational
Enum-Observed-Value-Basic-Bit-Str	MDC_ATTR_ENUM_OBS_VAL_BASIC_BIT_STR	BITS-16	The value is reported as a bit string of 16-bits. One and only one of Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str, or Enum-Observed-Value shall be present.	Conditional Observational
Enum-Observed-Value-Simple-Str	MDC_ATTR_ENUM_OBS_VAL_SIMP_STR	EnumPrintableString	The value is reported as an ASCII printable string. One and only one of Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str , Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str, or Enum-Observed-Value shall be present.	Conditional Observational

Table 8—Enumeration attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Enum-Observed-Value	MDC_ATTR_VAL_ENUM_OBS	EnumObsValue	This attribute defines a structured observed value that permits additional flexibility about the data type of the reported value. One and only one of Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str , Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str, or Enum-Observed-Value shall be present.	Conditional Observational
Enum-Observed-Value-Partition	MDC_ATTR_ENUM_OBS_VAL_PART	NomPartition	This attribute may be used to define the partition from which the Enum-Observed-Value-Simple-OID or the Enum-Observed-Value's observation OID nomenclature term was taken. If not present, the partition is the same as the nomenclature partition defined in the partition field of the Type attribute.	Optional Dynamic

6.3.6.4 Enumeration object methods

Change as shown:

There are currently no enumeration object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.6.5 Enumeration object events

Change as shown:

There are currently no enumeration object events defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.6.6 Other enumeration services

Change as shown:

There are currently no uses of the enumeration SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.7 PM-store class

6.3.7.1 General

Change as shown and insert a new paragraph:

An instance of the Persistent Metric store (PM-store) class provides long-term storage capabilities for metric data. Data are stored in a variable number of PM-segment objects (see 6.3.8). The stored data of the PM-store object are requested from the agent by the manager using object access services (see 7.3). Anybody not familiar with the PM-store concept may wish to read Annex C for a conceptual overview prior to reading the following subclauses.

Attribute values stored in a PM-segment may require the use of additional attributes from that object to describe the measurement; a common example being the unit of measure. 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 agent shall store the dependent attribute value in the PM-segment.

6.3.7.3 PM-store class attributes

Change Table 9 as shown:

Table 9—PM-store attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual~~-~~ifiers
Handle	MDC_ATTR_ID_HANDLE	HANDLE	The Handle attribute represents a reference ID for this object. Each object shall have a unique ID assigned by the agent. The handle identifies the object in event reports sent to the manager and to address the object instance in messages invoking object methods. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
PM-Store-Capab	MDC_ATTR_PM_STORE_CAPAB	PmStoreCapab	This attribute defines basic capabilities of the PM-store object instance. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Store-Sample-Algorithm	MDC_ATTR_METRIC_STORE_SAMPLE_ALG	StoSampleAlg	This attribute describes how the sample values stored in the PM-segment have been processed. The StoSampleAlg structure describes the available sampling algorithms. If there is no specific sampling algorithm used (in other words the sample values are raw data), then this attribute shall have a value of st-alg-no-downsampling. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static

Table 9—PM-store attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Store-Capacity-Count	MDC_ATTR_METRIC_STORE_CAPAC_CNT	INT-U32	This attribute is the maximum number of stored PM-segment entries (entries in all contained PM-segments). ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Optional Static
Store-Usage-Count	MDC_ATTR_METRIC_STORE_USAGE_CNT	INT-U32	This attribute is the actual number of currently stored PM-segment entries (entries in all contained PM-segments).	Optional Dynamic
Operational-State	MDC_ATTR_OP_STAT	OperationalState	The attribute indicates if new entries are currently being inserted in any of the contained PM-segments. If any PM-segment contained by this PM-store is having data actively added to it, this attribute shall be set to enabled. Otherwise, it shall be set to disabled.	Mandatory Dynamic
PM-Store-Label	MDC_ATTR_PM_STORE_LABEL_STRING	OCTET STRING	This attribute is an application-dependent label for the PM-store in printable ASCII to indicate its intended use and may be used for display purposes. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Optional Static
Sample-Period	MDC_ATTR_TIME_PD_SAMP	RelativeTime	This attribute determines the frequency at which entries are added to the PM-segments. If values are sampled periodically, ~~T~~this attribute shall be present either in the PM-store (in which case it applies to all periodically storing PM-segments in the PM-store) or alternatively in each PM-segment~~, if values are sampled periodically~~, so the time difference for two entries in the Fixed-Segment-Data is constant (i.e., the pmsc-peri-seg-entries bit in the Pm-Store-Capab attribute is set). ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Conditional Static
Number-Of-Segments	MDC_ATTR_NUM_SEG	INT-U16	This attribute is the number of currently instantiated PM-segments contained in the PM-store. Note that the PM-segment attribute Instance-Number is NOT related to this number (i.e., does not need to be in the range from 0 to Number-Of-Segments), but shall be retrieved with the Get-Segment-Info method.	Mandatory Dynamic

Table 9—PM-store attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual-ifiers
Clear-Timeout	MDC_ATTR_CLEAR_TIMEOUT	RelativeTime	This timeout attribute defines the minimum time that the manager shall wait for the completion of a PM-store clear command. If, after the manager sends a Confirmed Action(Clear Segments) invoke command, the timeout expires before the manager receives the corresponding Confirmed Action response message, the manager shall transition to the Unassociated state as described in 8.9.5.6. This attribute is required if the agent supports the clear segment action.	Conditional Dynamic

6.3.7.4 PM-store object methods

Change the text immediately after Table 10 as shown:

If an agent supports the PM-store class, the support of the Get-Segment-Info and Trig-Segment-Data-Xfer methods is mandatory. Support for the Clear-Segments method is optional and is indicated in the PM-Store-Capab attribute.

If a manager supports the PM-store class, the support of sending the Get-Segment-Info and Trig-Segment-Data-Xfer methods is mandatory. Support for sending the Clear-Segments method is optional.

— **Clear-Segments:**
This method allows the manager to delete the data currently stored in one or more selected PM-segments. All entries in the selected PM-segments are deleted. If the agent supports a variable number of PM-segments, the agent may delete empty PM-segments. Additionally, the agent may clear PM-segments without direction from the manager (e.g., the user of the agent could choose to delete data stored on the agent); however, if doing so while in an Associated state, the Instance-Number shall remain ~~empty~~ valid referencing an empty PM-segment, for the duration of the association. The Instance-Number of all other PM-segments shall be unaffected by clearing a segment.

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. Also, the agent may decide to protect certain segments from deletion making them read only (e.g., the user of the agent chose to "lock" certain data). Both protected and enabled segments will remain unaffected by a clear-segments operation.

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.

If all of the selected segments fail to clear (reason being protected or in enabled state), ~~If this method is invoked on a PM-segment that has the Operational-State attribute set to enabled,~~ the agent shall reply with a not-allowed-by-object error (roer). The ~~with a~~ return code shall be set to~~of~~ MDC_RET_CODE_OBJ_BUSY if any of the segments failed due to being in enabled state. Otherwise, the return code shall be MDC_RET_CODE_UNKNOWN, which indicates that only agent protected segments were encountered during the operation.

For PM-segments cleared using the by time method and using absolute time, only PM-segments having Segment-Start-Abs-Time and Segment-End-Abs-Time fields entirely within the specified time period are cleared.

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 nonzero value). If the offset field has a value 0x7FFF (32767), then only PM-segments having a base time entirely within the specified base time period are cleared; otherwise, for any other value of the offset field, only PM-segments having local time (base time with offset added) entirely within the specified time period are cleared.

Note that the behavior of the Clear-Segments method is application specific. The method may remove all entries from the specified PM-segment, leaving it empty, or it may remove the defined PM-segment completely. This behavior is defined in the PM-Store-Capab attribute. For specific applications, recommendations are defined in corresponding device specializations, making use of the PM-store.

If the agent supports the Clear-Segment method, the agent shall support at least the all-segments choice in the SegmSelection action-info-args of the Clear-Segments method.

If the agent supports the all-segments choice in the SegmSelection action-info-args of the Clear-Segments method, the agent shall set the pmsc-clear-segm-all-sup flag in the PM-Store-Capab attribute. If the agent supports the segm-id-list choice in the SegmSelection action-info-args of the Clear-Segments method, the agent shall set the pmsc-clear-segm-by-list-sup flag in the PM-Store-Capab attribute. If the agent supports the abs-time-range or bo-time-range choice in the SegmSelection action-info-args of the Clear-Segments method, the agent shall set the pmsc-clear-segm-by-time-sup flag in the PM-Store-Capab attribute.

If the agent does not support a Clear-Segments action from the manager, the agent shall respond with a no-such-action error (roer).

If the manager supports sending the Clear-Segments method, the manager shall support at least the all-segments choice in the SegmSelection action-info-args of the Clear-Segments method. The manager may support additional choices.

— **Get-Segment-Info:**
This method allows the manager to retrieve PM-segment attributes of one or more PM-segments, with the exception of the Fixed-Segment-Data attribute which contains the actual stored data and is retrieved by using the Trig-Segment-Data-Xfer method. In particular, the Get-Segment-Info method allows the manager to retrieve the ~~Instance-Number~~ attributes and their data contents ~~of~~ from the PM-segment object instances identified by the SegmSelection parameter ~~and their data contents~~.

The agent shall support the all-segments choice in the SegmSelection action-info-args of the Get-Segment-Info method. The agent 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 agent shall set the pmsc-segm-id-list-select and/or pmsc-abs-time-select flag in the PM-Store-Capab attribute.

For PM-segment info returned by time, the segments are selected using the same mechanism as described under Clear-Segments.

If the manager supports sending the Get-Segment-Info method, the manager shall support at least the choice all-segments in the SegmSelection action-info-args of the Get-Segment-Info method. The manager may support additional choices.

6.3.7.6 Other PM-store services

6.3.7.6.1 GET service

Change the first paragraph as shown:

Support for the GET service shall be provided by any agent that supports one or more PM-store objects only while in the Operating state. The manager uses the GET service to retrieve the values of all PM-store object attributes. If a manager does not have the current value of a needed PM-store attribute, then the GET service shall be used. An agent may also send scan event reports providing the manager with updates of the current attribute values, but this is not a mandated agent behavior.

Insert the following sentence as the final sentence in the last paragraph:

See 8.9.3.4.2 for a full explanation of the GET operation.

Insert the following subclause:

6.3.7.6.2 SET service

There are currently no uses of the PM-store SET service defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.

6.3.8 PM-segment class

6.3.8.3 PM-segment class attributes

Change Table 12 as shown and insert the NOTE immediately after the table:

Table 12—PM-segment attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Instance-Number	MDC_ATTR_ID_INSTNO	InstNumber	The Instance-Number is the ID of a specific PM-segment object instance. Each instance shall have a unique number, within the context of a PM-store, assigned by the agent. It is used by the manager to address a PM-segment.	Mandatory
PM-Segment-Entry-Map	MDC_ATTR_PM_SEG_MAP	PmSegmentEntryMap	This attribute defines the format and contents of one stored entry. An entry has an optional header containing information applicable to all elements in the entry. The entry then contains one or more elements, defined by the class, metric ID, handle, and an attribute value map defining the object attributes for each element in the PM-segment.	Mandatory

Table 12—PM-segment attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
PM-Seg-Person-Id	MDC_ATTR_PM_SEG_PERSON_ID	PersonId	This standard supports devices that have simple support for data from multiple persons. A person ID is used to differentiate different persons. 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. Otherwise, this attribute is not defined.	Conditional
Operational-State	MDC_ATTR_OP_STAT	OperationalState	This attribute indicates if new entries are currently being inserted into this PM-segment. If this PM-segment is having data actively added to it, this attribute shall be set to enabled. Otherwise, it shall be set to disabled.	Mandatory
Sample-Period	MDC_ATTR_TIME_PD_SAMP	RelativeTime	This attribute defines the frequency at which entries are added to the PM-segment. If values are sampled periodically, t~~T~~his attribute shall be present either in the PM-store (in which case it applies to all periodically storing PM-segments in the PM-store) or alternatively in each PM-segment. If values are sampled periodically, then the pmsc-peri-seg-entries bit in the PM-Store-Capab attribute shall be set.	Conditional
Segment-Label	MDC_ATTR_PM_SEG_LABEL_STRING	OCTET STRING	This attribute is an application-dependent label in printable ASCII for the segment to indicate its intended use and may be used for display purposes.	Optional
Segment-Start-Abs-Time	MDC_ATTR_TIME_START_SEG	AbsoluteTime	This attribute defines the start time of segment. This attribute is required if the segment supports actions on the segment by time (i.e., the pmsc-abs-time-select and/or the pmsc-clear-segm-by-time-sup bits are set). If this attribute is used, the Segment-Start-BO-Time shall not be used.	~~O~~Conditional

Table 12—PM-segment attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Segment-End-Abs-Time	MDC_ATTR_TIME_END_SEG	AbsoluteTime	This attribute defines the end time of segment. This attribute is required if the segment supports actions on the segment by time (i.e., the pmsc-abs-time-select and/or the pmsc-clear-segm-by-time-sup bits are set). If this attribute is used, the Segment-End-BO-Time shall not be used.	~~O~~Conditional
Date-and-Time-Adjustment	MDC_ATTR_TIME_ABS_ADJUST	AbsoluteTimeAdjust	This attribute reports any date and time adjustments that occur either due to a person's changing the clock or events such as daylight savings time. If the agent ever adjusts the date and time, this attribute ~~is~~ reports such an adjustment.	Conditional
Segment-Start-BO-Time	MDC_ATTR_TIME_START_SEG_BO	BaseOffsetTime	This attribute defines the start time of the segment as base time and the offset in minutes to local time. This attribute is required if the segment supports actions on the segment by time (i.e., the pmsc-abs-time-select and/or the pmsc-clear-segm-by-time-sup bits are set). Base time with offset is recommend if time adjustments are expected. If this attribute is used, the Segment-Start-Abs-Time shall not be used.	Conditional
Segment-End- BO-Time	MDC_ATTR_TIME_END_SEG_BO	BaseOffsetTime	This attribute defines the end time of segment as base time and the offset in minutes to local time. This attribute is required if the segment supports actions on the segment by time (i.e., the pmsc-abs-time-select and/or the pmsc-clear-segm-by-time-sup bits are set). Base time with offset is recommend if time adjustments are expected. If this attribute is used, the Segment-End-Abs-Time shall not be used.	Conditional
Segment-Usage-Count	MDC_ATTR_SEG_USAGE_CNT	INT-U32	This attribute gives the actual (current) number of stored entries.	Optional
Segment-Statistics	MDC_ATTR_SEG_STATS	SegmentStatistics	This attribute defines the array for reporting minimum, mean, maximum statistics for each element to be tagged.	Optional

Table 12—PM-segment attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Fixed-Segment-Data	MDC_ATTR_SEG_FIXED_DATA	N/A The data is stored internal to the device and so this data type never occurs in any protocol definition directly.	This attribute defines the segment data transferred as an array of entries in a format as specified in the PM-Segment-Entry-Map attribute. This is defined here as an opaque data structure without a defined data type. Note that this attribute is not directly accessible; it is only retrievable by the manager using the PM-store ~~Trig-Segm-Data-Xfer~~ Trig-Segment-Data-Xfer method.	Mandatory
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	This informational timeout attribute defines the minimum time that the agent shall wait for a Response message from the manager after issuing a Confirmed Event Report invoke message before timing out and transitioning to the Unassociated state. This is an informational attribute for the benefit of the manager. If this attribute is supplied, it shall match the actual timeout value that the agent uses for the Confirmed Event Report generated from the PM-store object. This attribute is informational for the manager in the sense that the manager does not use this attribute in an actual implementation of the protocol (i.e., the manager does not time out on an agent-generated Confirmed Event Report). However, the manager might wish to use this information to prioritize its handling of a “short” timeout agent over that of a “long” timeout agent.	Optional
Transfer-Timeout	MDC_ATTR_TRANSFER_TIMEOUT	RelativeTime	This timeout attribute defines the minimum time that the manager shall wait for the complete transfer of PM-segment information. If the timeout expires prior to the reception of the complete PM-segment, the manager shall transition to the Unassociated state as described in 8.9.5.6.	Mandatory

NOTE—The attribute qualifiers “static,” “dynamic,” and “observational” are omitted from Table 12 since PM-segments are dynamic (the object itself may change during the lifetime of an association).

Change the final paragraph as shown:

The Fixed-Segment-Data attribute may hold very large amounts of data, depending on the agent capabilities and the application. An agent may choose to restrict the maximum size of the Fixed-Segment-Data attribute in a way that is aligned with the maximum transmission unit of the transport system. In order to support this type of behavior, a manager that supports PM-stores shall be able to support the transfer of Fixed-Segment-Data attributes in multiple application messages.

6.3.8.4 PM-segment object methods

Change as shown:

There are currently no PM-segment object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.8.5 PM-segment object events

Change as shown:

There are currently no PM-segment object events defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.8.6 Other PM-segment services

Change as shown:

There are currently no uses of the PM-segment SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9 Scanner classes

6.3.9.1 General

Change the first paragraph as shown:

A scanner serves two purposes: (1) It allows the manager to control the flow of data, and (2) it is an optimized packaging and reporting mechanism. It enables collected sets of attribute value changes (AttributeChangeSets) from one or more metric objects to be packaged together in a single event report in a more efficient manner than can be done by using MDS events. A scanner implementation is either episodic or periodic, and both types of scanners may use variable, fixed, or group formats (see 7.4.5) in scan event reports. ~~A scanner object is an observer and "summarizer" of object attribute values. It observes attributes of metric objects (e.g., numeric objects) and generates summaries in the form of notification event reports.~~ See Figure 5 for the class hierarchy of the scanner classes. Each class is described in 6.3.9.2 through 6.3.9.5, respectively.

Delete the final paragraph of the subclause:

~~The different scanner classes (periodic and episodic) as well as the different instances should be used to distribute different data types across one or many data flows represented by a scanner instance. A pulse oximetry application might choose to have a periodic configurable scanner for RT-SA objects with a Reporting Interval of 50 ms, a periodic configurable scanner with a Reporting Interval of 1 s for numeric~~

~~and enumeration objects, and an episodic configurable scanner for beat-to-beat metric objects (numeric or enumeration objects).~~

Insert the following subclause:

6.3.9.1a Conceptual model

A scanner does not scan objects (that is, it does not read the state of the object and report what is there whether or not any changes have occurred). Instead, a scanner collects AttributeChangeSets and maps them to ObservationScans in scan event reports: Episodic Scanners dispatch scan event reports when an episode completes. An episode is defined by the application, but in general, it is one or more AttributeChangeSets (where no two AttributeChangeSets come from the same object) that occur episodically (the time between episodes is unknown). Periodic Scanners dispatch scan event reports when the period reported in the Reporting-Interval attribute expires, and in this case, more than one AttributeChangeSet may come from the same object.

Error conditions with the sensor, or other conditions such as no data available when the scanner is first enabled, may result in AttributeChangeSets not being present when the scan event report is generated. The scan event report that is generated in these cases depends upon the scanner type and format used as follows.

Episodic and Periodic scanners using the group format shall create scan event reports where:

— Any AttributeChangeSet containing observational attribute values from the Numeric Object (Table 6) is set to NaN. If the AttributeChangeSet contains the Measurement-Status attribute, then this value shall indicate either invalid or not-available as appropriate.

— Any AttributeChangeSet containing the Simple-Sa-Observed-Value of an RT-SA object shall include the Measurement-Status attribute. The Measurement-Status attribute shall indicate either invalid or not-available as appropriate. The Simple-Sa-Observed-Value values in this case are undefined, and the manager shall attach no significance to the reported values.

— Any AttributeChangeSet containing observational attribute values from the enumeration object (Table 8) shall be set to a suitable enumeration, and where appropriate include the Measurement-Status attribute. The Measurement-Status attribute shall indicate either invalid or not-available as appropriate. If the Measurement-Status attribute is used to indicate invalid or not-available, the manager shall attach no significance to the reported value.

— If the scanner is epsiodic and no AttributeChangeSets are collected, the scan event report shall not be sent.

— If the scanner is periodic and no AttributeChangeSets are collected, an empty scan event report shall be sent.

Episodic and Periodic scanners using the variable or fixed format shall create scan event reports where:

— Only the collected AttributeChangeSets are mapped to the ObservationScans .

— If the scanner is episodic and no AttributeChangeSets are collected, the scan event report shall not be sent.

— If the scanner is periodic and no AttributeChangeSets are collected, an empty scan event report shall be sent.

The periodic scanner is distinct from the episodic scanner by being able to collect multiple AttributeChangeSets from the same object before dispatching. The periodic scanner also requires that the rate of generation of all collected AttributeChangeSets shall have a fixed timing relationship with the period of the periodic scanner. Where an AttributeChangeSet does not have an explicit timestamp, its timestamp shall be inferred from the timestamp of the scan event report. It follows that any

AttributeChangeSet that is collected at a time different from the time of the scan event report shall be reported with its own timestamp. The periodic scanner shall insert AttributeChangeSets from the same object in the scan event report in first-in, first-out (FIFO) order.

Different observation reporting requirements may be accomplished by using a set of periodic and episodic scanners, one to manage the characteristics of each observation stream. For example, a pulse oximeter might use a periodic configurable scanner with a Reporting-Interval of 50 ms for the RT-SA object representing the plethysmogram, a periodic configurable scanner with a Reporting-Interval of 1 s for the numeric representing the saturated oxygen level and any enumeration objects reporting on events associated with the value, and an episodic configurable scanner for beat-to-beat metric objects (numeric or enumeration).

6.3.9.2 Scanner class

6.3.9.2.3 Scanner class attributes

Change Table 13 as shown:

Table 13—Scanner attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Scanners are identified by handles. ~~This attribute shall remain unchanged after the configuration is agreed upon.~~	Mandatory Static
Operational-State	MDC_ATTR_OP_STAT	OperationalState	This attribute indicates whether 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. The manager shall use the SET action to request a change in the value of this attribute.~~This attribute defines if scanner is active enabled or disabled and can be set by the manager.~~	Mandatory Dynamic
Scan-Handle-List	MDC_ATTR_SCAN_HANDLE_LIST	HANDLEList	This attribute defines the metric-derived objects that might be reported in the Unbuf-Scan-Report-Var, Buf-Scan-Report-Var, Unbuf-Scan-Report-Fixed, Buf-Scan-Report-Fixed, or any of the four multiple-person equivalents. For episodic scanners, a particular object is included in an event report whenever AttributeChangeSet eminates from that object~~there are changes in one or more attribute values~~. For periodic scanners, the collected AttributeChangeSets from the ~~scanned~~ objects ~~and attribute values~~ are reported in each period. The manager shall not assume the order of the ~~objects~~ ObservationScans contained in the event reports is the same as the order of the Scan-Handle-List. This attribute shall be present if any of these eight reporting styles are used by the scanner. This attribute shall be set prior to sending such a report. This attribute value may change in between event reports as long as the scanner is disabled. The attribute value change is communicated to the manager using an agent-initiated event report from the MDS object.	Conditional Dynamic

Table 13—Scanner attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Scan-Handle-Attr-Val-Map	MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP	HandleAttrValMap	This attribute defines the metric-derived objects, the attributes, and the order in which objects and attribute values are reported in a Unbuf-Scan-Report-Grouped, Buf-Scan-Report-Grouped, Unbuf-Scan-Report-MP-Grouped, or Buf-Scan-Report-MP-Grouped. All values shall be present to maintain a consistent layout of message. ~~This attribute shall be present i~~If any of these four reporting styles are used, this attribute shall be set prior to sending such a report. This attribute value may change in between event reports as long as the scanner is disabled. The attribute value change is communicated to the manager using an agent-initiated event report from the MDS object.	Conditional Dynamic

6.3.9.2.4 Scanner object methods

Change as shown:

There are currently no scanner object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9.2.6 Other scanner services

Insert the following text as the first item in the list:

— **GET service:**
There is currently no GET service defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.

6.3.9.3 CfgScanner class

6.3.9.3.3 Configurable scanner class attributes

Change Table 14 as shown:

Table 14—Configurable scanner attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Confirm-Mode	MDC_ATTR_CONFIRM_MODE	ConfirmMode	This attribute defines whether event reports are sent confirmed or unconfirmed. Currently, the attribute is Dynamic, but agent implementations should treat as Static since it is likely this will change to Static in the future. If this attribute value is changed, it shall be changed only when the scanner is disabled. The attribute value change is communicated to the manager using an agent-initiated event report from the MDS object.	Mandatory Dynamic
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	This informational timeout attribute defines the minimum time that the agent shall wait for a Response message from the manager after issuing a Confirmed Event Report invoke message before timing out and transitioning to the Unassociated state. This is an informational attribute for the benefit of the manager. If this attribute is supplied, it shall match the actual timeout value that the agent uses for the Confirmed Event Report generated from the scanner object. This attribute is informational for the manager in the sense that the manager does not use this attribute in an actual implementation of the protocol (i.e., the manager does not time out on an agent-generated Confirmed Event Report). However, the manager might wish to use this information to prioritize its handling of a “short” timeout agent over that of a “long” timeout agent. Currently, the attribute is Dynamic, but agent implementations should treat as Static since it is likely this will change to Static in the future. If this attribute value is changed, it shall be changed only when the	~~C~~Optional Dynamic

Table 14—Configurable scanner attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
			scanner is disabled. The attribute value change is communicated to the manager using an agent-initiated event report from the MDS object.	
Transmit-Window	MDC_ATTR_TX_WIND	INT-U16	This attribute defines informative data provided by the agent that may help a manager optimize its configuration. The Transmit-Window represents the number of unacknowledged confirmed event reports that the agent will allow to be outstanding. For this version of this standard, the attribute shall have only a value of 1. Currently, the attribute is Dynamic, but agent implementations should treat as Static since it is likely this will change to Static in the future. If this attribute value is changed, it shall be changed only when the scanner is disabled. The attribute value change is communicated to the manager using an agent-initiated event report from the MDS object.	Optional Dynamic

6.3.9.3.4 Configurable scanner object methods

Change as shown:

There are currently no configurable scanner object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9.3.6 Other configurable scanner services

Change as shown:

There are currently no uses of the configurable scanner SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9.4 EpiCfgScanner class

6.3.9.4.3 Episodic configurable scanner class attributes

Change Table 15 as shown:

Table 15—Episodic configurable scanner attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers~~.~~
Min-Reporting-Interval	MDC_ATTR_SCAN_REP_PD_MIN	RelativeTime	This attribute provides an estimate of the expected minimum time between any two successive event reports.	Optional Dynamic

6.3.9.4.4 Episodic configurable scanner object methods

Change as shown:

There are currently no episodic configurable scanner object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9.4.5 Episodic configurable scanner object events

Insert the following sentence as the last sentence of the first paragraph:

If a manager supports episodic scanners, it shall support all the events identified in Table 16.

Delete NOTE 2 at the bottom of Table 16. Change NOTE 1 as shown:

NOTE—For variable and fixed format reports, if no ~~attribute of~~AttributeChangeSet is collected from an object ~~changes its value~~, then no ~~data of~~AttributeChangeSet from this object ~~are~~is included in the scan report. If no AttributeChangeSets are collected, no scan event report is sent.

Change the third item in the dashed list after Table 16 as shown:

— **Unbuf-Scan-Report-Grouped:**
This style is used when the scanner object is used to send the data in its most compact format. The Scan-Handle-Attr-Val-Map attribute describes the objects and attributes that are included and the format of the message.

6.3.9.4.6 Other episodic configurable scanner services

Change as shown:

There are currently no uses of the episodic configurable scanner SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9.5 PeriCfgScanner class

6.3.9.5.1 General

Change as shown and insert the following new text:

The PeriCfgScanner class represents a class that can be instantiated. PeriCfgScanner objects are used to send reports containing Periodic data~~, that is, data sampled during fixed periods. It buffers any data value changes to be sent as part of a periodic report~~. Event reports shall be sent with a time interval equal to the Reporting-Interval attribute value.

The number of observations for each metric object is dependent on the metric object's update interval and the scanner's Reporting-Interval.

When a periodic configurable scanner is enabled by a manager, 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 s.

NOTE—The 15 s is expected to provide an ample initialization time.

Example: A periodic configurable scanner is set up to ~~"scan"~~report on two metric objects with a Reporting-Interval of 1 s. The two objects update their corresponding observed value periodically with an interval of 1 s and ½ s, 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. The objects in a Scan-Handle-Attr-Val-Map would contain two entries for the object with a ½ s update interval.

6.3.9.5.3 Periodic configurable scanner object attributes

Change Table 17 as shown:

Table 17—Periodic configurable scanner object attributes

Attribute name	Attribute ID	Attribute type	Remark	Qual~~-~~ifiers
Reporting-Interval	MDC_ATTR_SCAN_REP_PD	RelativeTime	Reporting period of the event reports.	Mandatory Dynamic

6.3.9.5.4 Periodic configurable scanner object methods

Change as shown:

There are currently no periodic configurable scanner object methods defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.3.9.5.5 Periodic configurable scanner object events

Insert the following sentence as the last sentence of the first paragraph:

If a manager supports periodic scanners, it shall support all the events identified in Table 18.

Insert the following NOTE immediately after Table 18:

NOTE—For variable and fixed format reports, if no AttributeChangeSet is collected from an object, no AttributeChangeSets from that object are included in the scan event report. If no AttributeChangeSets are collected, then an empty scan event report is generated when the period expires.

6.3.9.5.6 Other periodic configurable scanner services

Change as shown:

There are currently no uses of the periodic configurable scanner SET or GET services defined in this standard; however, ISO/IEEE 11073-104zz device specializations or proprietary definitions may exist.~~None~~

6.4 Information model extensibility rules

Insert the following text between the second and third paragraphs:

Vendor-specified classes may be defined. Vendor-specified objects may be created from those classes or any classes defined in this series of standards.

7. Personal health device service model

7.3 Object access services

Change the following sentence in the last paragraph as shown:

If an error occurs in executing a confirmed action, then the error ~~may~~ 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.

7.4 Specific application of object access EVENT REPORT services for personal health devices

7.4.3 Configuration event report

7.4.3.2 Agent device configuration

Change the following paragraph as shown and insert a new second paragraph:

The set of non-MDS objects and attributes that exists in an agent denotes the agent device configuration and is associated with a Dev-Configuration-Id value (see Table 2). In case an agent owns multiple device configurations, the assigned Dev-Configuration-Id values shall be locally unique. During the lifetime of an association, the configuration of an agent shall remain fixed, that is, the set of objects shall remain fixed. However, the agent may add new attributes to an object or change attribute values as described in 7.4.3.2.

An agent that requires a different configuration shall release the association and establish a new association with the desired configuration.

The MDS object is not considered part of the configuration. A manager reassociating with an agent offering the same Dev-Configuration-Id cannot expect the MDS attribute values to be the same; for example, an agent may clear the manager-set-time bit as its clock has already been set.

7.4.3.3 Configuration event report

Change as shown and insert the NOTE after the third paragraph:

The configuration that the agent wants to use for the duration of the association to a manager is indicated by using the Dev-Configuration-Id value for the dev-config-id field in Association Request message. If the manager does not already know the agent's device configuration (e.g., based on a previous association phase), the manager asks for the agent's device configuration. Even if the manager knows the agent´s device configuration, the manager may ask to enter the configuring state in order to check attributes from the MDS object before deciding to accept the association. The agent transfers its configuration to the manager using a configuration event report. The report describes all the objects of the agent's device configuration along with the associated Dev-Configuration-Id value. For the duration of the association, the agent's configuration is fixed with respect to the number of objects. In case the agent intends to use a different configuration or wants to change the existing configuration by adding or removing objects, the agent shall release the association and re-associate with a new configuration.

For each object except the MDS, the configuration event report shall ~~also~~ contain~~s~~ the static attributes and should contain the dynamic attributes used by the object. These attributes are reported in a list of ConfigObject structures (A.11.5). The handle attribute value is reported in the obj-handle field of the ConfigObject and is not included in the attribute-list of the ConfigObject.~~Typically, the infrequently changing attributes are included within the configuration event report and dynamic values, such as measurements,~~ The observational attributes of the objects shall not be included in the ConfigObject unless the mss-cat-setting and/or mss-cat-manual bit of the Metric-Spec-Small attribute is set, in which case it may be included. An observational attribute may also be included if the mss-calculated bit of the Metric-Spec-Small attribute is set and all dependent attributes for the calculation are also allowed to be sent in the Config Report. Otherwise, the observational attribute values are sent in later ~~measurement~~ scan event reports (see 7.4.5 and 7.4.6). The agent may add new attributes to an object or change nonstatic attribute values while in the operating state~~during the association~~ without sending a new configuration.

Changes to any attribute values of metric and scanner objects shall be reported to the manager 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). Changes to any nonstatic attributes values on PM-stores or the MDS may be reported to the manager in event reports at the discretion of the agent. Adding new attributes can only be achieved using a variable format event report (see 7.4.5 for details on event report formats). Changing attribute values may use variable, fixed, or grouped event reports depending on the configuration.

NOTE—A manager can always poll PM-stores and the MDS object with the GET service; thus, it is not necessary for the agent to send updates or additions in scan event reports. However, if the agent application deems it is important for the manager to have this information at the time of the change, the agent is free to notify the manager with a scan event report.

Insert the following new subclause as shown:

7.4.3.4a Profiles

7.4.3.4a.1 General

A profile further constrains the objects, services, and communication model of a specialization. Some specializations are written to cover a broad category of device types. They define common objects that are useful to the set of devices and other objects that are specific to a limited number of devices. By further profiling the device specialization, the standard provides more guidance on the specific mandatory objects that shall be implemented, the objects that are optional, and the objects that are not needed. The profile can also define smaller application protocol data unit (APDU) sizes, specific standard configurations, and help the manager know exactly which device type has been implemented (e.g., step counter versus exercise cycle).

A profile is expected to be identified by a name and a nomenclature value; for example, within the Activity hub (ISO/IEEE Std 11073-10471) standard, an implementer could declare adherence to a smoke sensor profile or a carbon monoxide sensor profile.

Implementing a smoke sensor or carbon monoxide profile would choose from those objects, services, and communication facilities defined within the specialization. A device that claims conformance to these profiles adheres to the requirements stated in the clauses of the profiles for the domain information, service, and communication models.

7.4.3.4a.2 Domain information model constraints

Objects that are mandatory or conditional in the specialization remain mandatory or conditional in a profile. Objects that are optional in the specialization may either remain optional or become mandatory within a profile. A profile is not intended to do the following:

— Define additional objects

— Extend the objects with additional attributes and attribute value ranges

— Extend the conditions in addition to those that are already defined in the specialization

7.4.3.4a.3 Service model constraints

A profile is not expected to modify or extend the Association service or the object access services. In particular, a profile is not intended to do the following:

— Extend the set of events described by the specialization.

— Extend the set of methods described by the specialization.

A configuration event report pertaining to a particular profile may be distinct from other configuration event reports defined in the device specialization.

7.4.3.4a.4 Communication model constraints

A profile may reduce the maximum APDU size compared to the maximum APDU defined in the device specialization.

7.4.3.5 Types of configuration

7.4.3.5.1 Standard configuration

Change as shown:

A standard configuration is one that is specified in one of the ISO/IEEE 11073-104zz specializations and that has a Dev-Configuration-Id value assigned from the range between standard-config-start and standard-config-end, inclusive. That range is further subdivided by reserving 100 IDs for each ISO/IEEE 11073-104zz specialization in the range from zz × 100 to zz × 100 + 99, inclusive. For example, the range 1500–1599 is reserved for IEEE Std 11073-10415 [B4]. All unused values in the standard range are reserved for future use. A manager encountering such a reserved value shall assume the value to be an ~~extended~~unsupported~~unrecognized~~ standard configuration and ~~use it in the normal way~~process it as described in 8.7.3.3 and 8.8.3.

A manager that supports one (or more) of the ISO/IEEE 11073-104zz device specializations shall be able to accept ~~knows~~ all the standard device configurations specified for the profiles listed in conformance Table 22 under Gen-4. If there are standard configurations that are generally applicable for the supported specializations, then the manager shall be able to accept all of those as well~~in that particular specialization~~. Every time an agent requests to associate with that manager using a Dev-Configuration-Id value of a standard configuration, the manager may ~~can~~ accept the association without asking for the agent's configuration since it is already known. After successful association, both manager and agent enter the Operating mode. Alternatively, the manager may request the agent to send the standard configuration in order to enter the configuring state and check attributes from the MDS object prior to final acceptance (or rejection) of the agent.

7.4.3.5.2 Extended configuration

Change as shown:

In extended configurations, the agent's configuration is not standard; it might have a different set of objects, different attributes present, and/or different attribute values. An agent implementing extended configuration(s) shall select a unique Dev-Configuration-Id value from the range between extended-config-start and extended-config-end, inclusive for each extended configuration. At association time, the agent sends the Dev-Configuration-Id in the dev-config-id field to identify the agent's selected configuration for the duration of the association. If the manager already understands that configuration either because it was preloaded via an installation program or the agent previously associated with the manager, then the manager shall respond with the configuration accepted response, and no further configuration information needs to be sent. However, if the manager does not know the agent's configuration, the manager shall respond with an accepted-unknown-config response, and the agent shall transmit its configuration information by sending a configuration event report. See 8.7 and 8.8 for full details on associating and configuring procedures. Once the manager has the configuration, the agent may transmit measurement data. To save association time, the ~~same~~ Dev-Configuration-Id should be used by an agent for subsequent associations consistently. This has two implications, as follows:

a) The same Dev-Configuration-Id shall not be used by an agent for subsequent associations to identify a different device configuration.

b) An agent should use the same value for Dev-Configuration-Id in future Association Requests with the manager to denote the same configuration of the device~~ provided the device configuration remains the same~~.

7.4.4 Agent- and manager-initiated measurement data transmission

Change the final paragraph as shown and insert the two NOTEs:

In both agent- and manager-initiated measurement data transmission, as well as PM-segment transfers, event reports are used to carry ~~the measurement data~~attribute change sets and/or attribute additions. Attribute change sets and/or additions in these event reports are organized into one or more ObservationScans. A manager applies the changes in the ObservationScan as a unit to the respective objects attaching no semantic behavior to the order in which the attributes appear in these ObservationScans for that object.

NOTE 1—Example 1: If the ObservationScan for a temperature object contains the set of attribute values representing a temperature and a Metric-Id containing a body location, the correct sematic interpretation is that of a single measurement with those respective values. If the ObservationScan is from an RT-SA containing a stream of temperature values and a body location the correct interpretation is that the body location applies to the entire stream of temperature values. If the ObservationScan contains the set of attribute values representing a unit code and body location (both dyanamic attributes), the correct interpretation is that the new values will apply to the next observation received (assuming these two dynamic values are not further updated).

NOTE 2—Example 2: If an ObservationScan is received containing just attribute A followed by an ObservationScan for a temperature measurement, if attribute A is an observational attribute (for example measurement status), that value could NOT be applied to the temperature measurement. If attribute A is dynamic (for example a unit code), that value would be applied to the temperature measurement. If the attribute A and the temperature measurement were in one ObservationScan, the value of attribute A would apply to the temperature regardless of attribute A's qualifier.

7.4.5 Variable, fixed, and grouped format event reports

Change the last three paragraphs as shown:

The fixed format event report ~~is optimized~~ optimizes data transmission by defining a specific list of attributes to communicate and the order in which they appear in the message. ~~format in t~~ The Attribute-Value-Map of the object holds this definition, including the attribute identifiers and the length of the attribute's value. The specific choice of which attributes to include depends on which attribute values are expected to change. For instance, one weighing scale implementation may communicate the weight observation and time stamp, but another may send weight observation, measurement status, and timestamp, In the first case, the attribute-value-map would be set to include the weight observational attribute identifier (MDC_ATTR_NU_VAL_OBS_SIMP) and its length (4 bytes long) followed by the timestamp attribute identifier (MDC_ATTR_TIME_STAMP_ABS) and its length (8 bytes long). The second case would be similar but also would include the attribute identifier for measurement-status (MDC_ATTR_MSMT_STAT) and its length (2 bytes long). Th~~is~~e Attribute-Value-Map attribute shall be defined and transmitted to the manager ~~in a previous configuration message~~ before fixed format event report transfer commences. When an agent transmits data in a fixed format event report, it shall report the object handle and the attribute values in the same order and size as specified in the Attribute-Value-Map. In this way, the overhead of sending attribute identification and length in each event report is avoided. On receipt of a fixed format event report, the manager uses the object handle to retrieve the previously given Attribute-Value-Map ~~previously given at configuration time~~ to know how to extract the data. For example, in the first case described earlier, the manager knows that the weight observation is the first element in the fixed event report and that it is 4 bytes long so it can extract that into the Simple-Nu-Observed-Value attribute and then it extracts the final 8 bytes into the Absolute-Time-Stamp. The order of these elements is defined by the order in which the attribute identifiers are listed in the Attribute-Value-Map. The agent controls the order and communicates it to the manager via the Attribute-Value-Map attribute.

The grouped format event report is further optimized by defining an event report's message format, containing one or more objects, in the scanner object's Scan-Handle-Attr-Val-Map. The notion is similar to the Attribute-Value-Map, but the Scan-Handle-Attr-Val-Map allows the agent to report on multiple objects at once by referencing other object handles and the attributes within that object. This attribute shall be defined ~~in a separate configuration message~~ before grouped event report transfer commences. When an

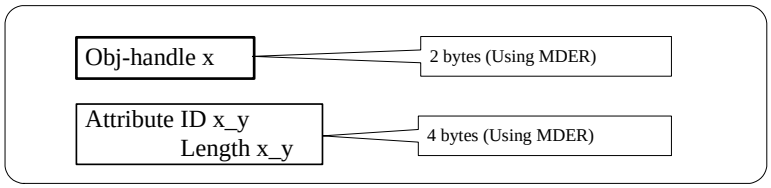
agent transmits data in a grouped format event report, it shall report the scanner object's handle along with the scanned objects' attribute values in the same order and size as specified in the Scan-Handle-Attr-Val-Map. In this way, the overhead of sending the scanned object's handles, their attribute identification, and data lengths in each event report is avoided. On receipt of a grouped format event report, the manager uses the scanner object's handle to retrieve the previously given Scan-Handle-Attr-Val-Map ~~previously given at configuration time~~ to know how to extract the data.

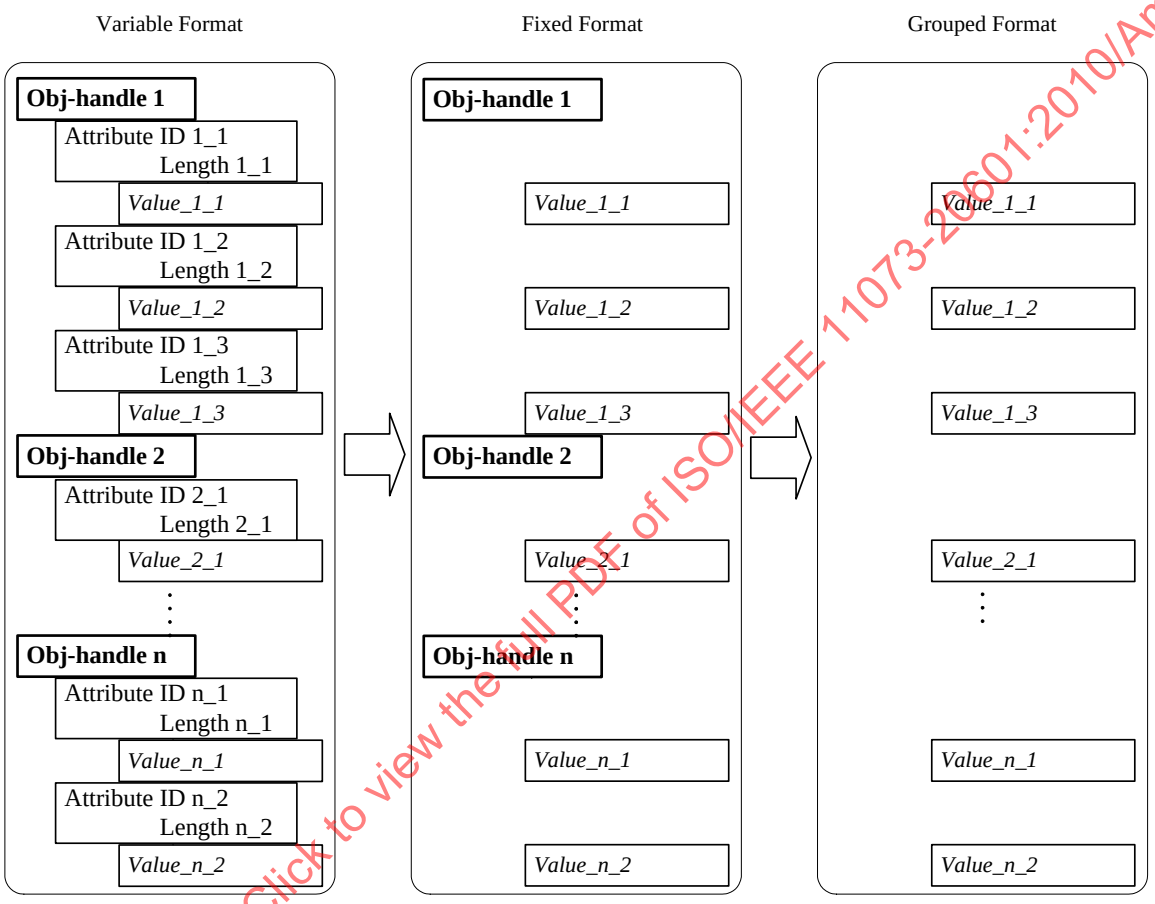
A manager shall support variable format~~,~~ and fixed format event reports, and if the manager supports scanners, it shall support~~, and~~ grouped format event reports. An agent may support any or all of variable format, fixed format, and grouped format event reports. The manager learns which format(s) the agent might use by inspecting the Attribute-Value-Map of objects ~~defined in the MDS Configuration Event from the agent~~ or by inspecting the Scan-Handle-Attr-Val-Map attribute for scanner objects ~~defined in the MDS Configuration Event~~.

7.4.6 Single-person and multiple-person event reports

Replace Figure 7 with the following figure:

Comparision between different reporting formats





7.4.7 Temporarily stored measurements

Change the fourth item as shown:

— Temporarily stored measurements are included in any of the ~~defined~~ following event report mechanisms: ~~(~~manager- or agent-initiated; ~~grouped,~~ fixed, or variable format; and single- or multiple-person~~)~~.

8. Communication model

8.3 Communications characteristics

8.3.2 Common communications characteristics

Change the following items in the lettered list as shown:

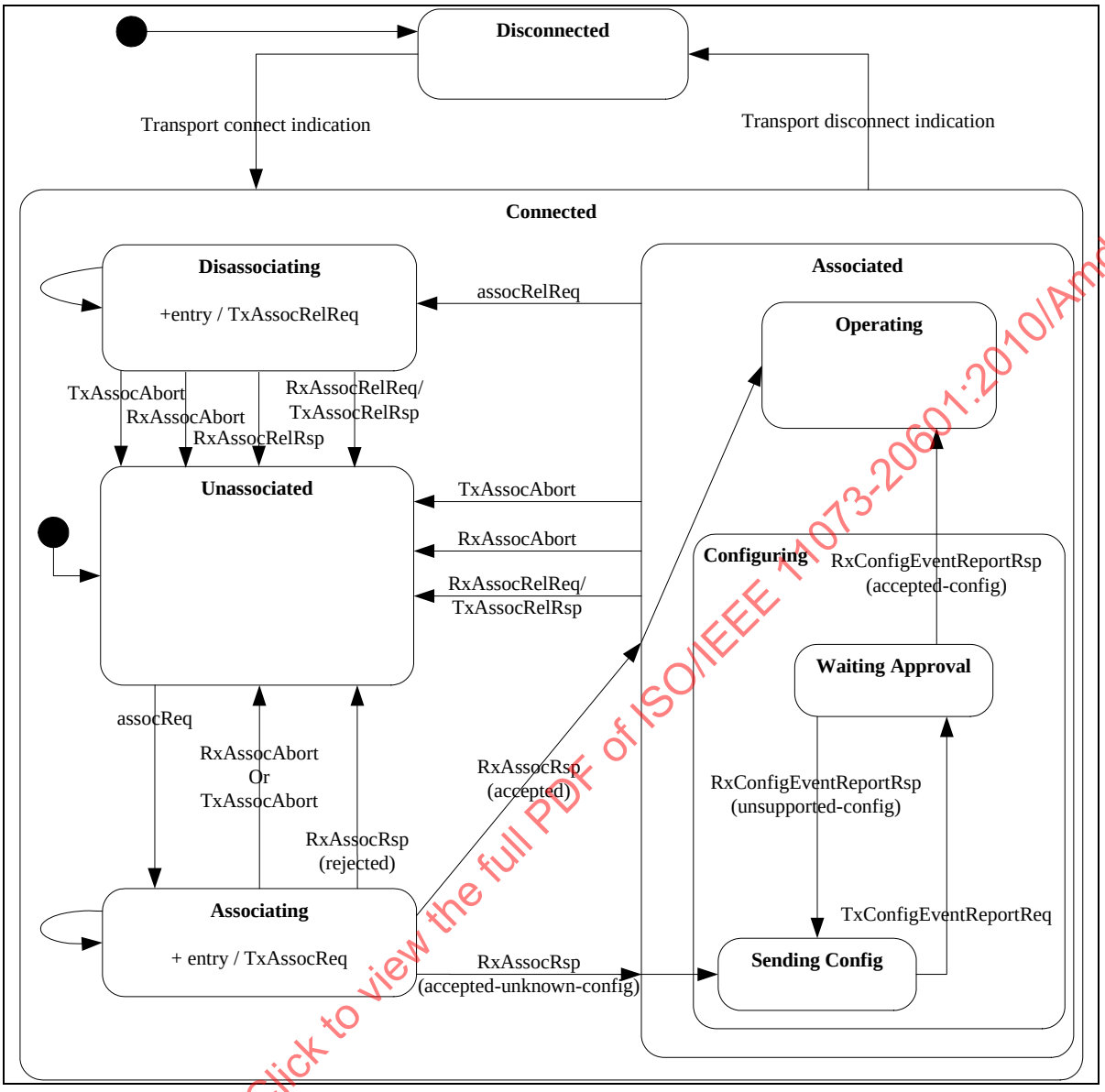
c) APDUs, in the agent-to-manager direction, shall be no larger than 63K (64 512) bytes in size. Specific device specializations, profiles, or implementations may evaluate the messages exchanged to determine a specific implementation size for a manager receive buffer that is smaller than the maximum agent-to-manager APDU size. If a manager receives an ~~larger~~ APDU that is larger than the manager's receive buffer, it shall reply with an error (roer) code of protocol-violation. The manager's receive buffer shall be at least as large as the largest buffer specified in the specializations the manager supports. The buffer size limitations in this item and the next one apply to all APDUs regardless of whether a standard or extended configuration is being used.

d) APDUs, in the manager-to-agent direction, shall be no larger than 8K (8192) bytes in size. Specific device specializations, profiles, or implementations may evaluate the messages exchanged to determine a specific implementation size for an agent receive buffer that is smaller than the maximum manager-to-agent APDU size. If an agent receives a larger APDU, it shall reply with an error (roer) code of protocol-violation.

8.4 State machines

8.4.1 Agent state machine

Replace Figure 10 with the following figure:



8.4.2 Manager state machine

Change the first item as shown:

— The manager shall wait in the Waiting for Config state for at least TO_{config} seconds before sending an ~~Association Release Request or~~ Association Abort message.

8.4.3 Timeout variables

Change Table 21 as shown:

Table 21—Timeout variables

	Communications service	Timeout		Subclause
		Variable	Value	
Associating procedure				
	Association	TO_{assoc}	10 s (and RC_{assoc} = 3)	8.7.5
	Configuration	TO_{config}	10 s	8.8.5
	Association Release	$TO_{release}$	3 s	8.10.5
Operating procedure				
MDS object	**Confirm action**	TO_{ca}	3 s	8.9.5.2
	Confirm event report	$TO_{cer-mds}$	MDS.Confirm-Timeout. If the attribute is not present, the agent and manager shall use the value 3 s.	8.9.5.3
	Get	TO_{get}	3 s	8.9.5.4
	Confirm set	TO_{cs}	3 s	8.9.5.5
	<inter-service timeout>	TO_{sp-mds}	3 s	8.9.5.6
PM-store object	**Confirm action**	TO_{ca}	3 s	8.9.5.2
	Confirm event report	$TO_{cer-pms}$	Segm.Confirm-Timeout. If the attribute is not present, the agent and manager shall use the value 3 s.	8.9.5.3
	Get	TO_{get}	3 s	8.9.5.4
	Confirm set	TO_{cs}	3 s	8.9.5.5
	<end of Segm timeout>	TO_{sp-pms}	Segm.Transfer-Timeout	8.9.5.6
	Confirm action – SegmClear	$TO_{clr-pms}$	PMS.Clear-Timeout	8.9.5.6
Scanner object	**Confirm set**	TO_{cs}	3 s	8.9.5.5
	Confirm event report	$TO_{cer-scan}$	Scan.Confirm-Timeout. If the attribute is not present, the agent and manager shall use the value 3 s.	8.9.5.3

8.7 Associating procedure

8.7.3 Normal procedures

Change the paragraph as shown:

Figure 12 and Figure 13 show sequence diagrams of the associating procedure between an agent and manager. Figure 12 shows the situation where the manager already knows about the agent's configuration due either to a prior connection with the agent or to the fact that the agent has a standard configuration (i.e., a predefined configuration that is specified in a specialization standard). Figure 13 shows the case where the manager does not know the agent's configuration and informs it that the association request is accepted, but that the configuration is unknown, or the case where the manager knows the agent´s configuration, but manager wants to enter configuring state in order to check attributes from the MDS object.

8.7.3.1 Agent procedure

8.7.3.1.1 General

Insert the following sentence at the end of the last paragraph:

The agent 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.

8.7.3.1.2 Data exchange protocol –defined by this standard

Change as shown:

If an agent sets the data-proto-id in A.8 to data-proto-id-20601, then it shall adhere to the abstract syntax definitions found in this standard for data types and message exchange. Further, the data-proto-info field shall be filled in with a PhdAssociationInformation structure, which defines the following information:

— The protocol-version field contains the version of the data exchange protocol.

— The encoding-rules field contains the specific DataApdu encoding rule(s) supported by the agent. The agent shall set one or more of the encoding-rules bits.

  — The agent shall always support MDER, i.e., the mder bit of the encoding-rules field shall be set by the agent.

  — The agent may offer other encoding rules, besides MDER, to the manager by setting other bits in the encoding-rules field.

— The nomenclature-version field contains the version of the nomenclature used.

— The functional-units~~A~~ field indicates~~ing~~ all functional units and optional features supported by the agent.

— The system-type field indicates the system type (agent in this case).

— The system-id field reports ~~A~~the unique System-Id attribute value (see Table 2) of the agent. The EUI-64 format is used to identify the agent. A manager may use this field to determine the identity of the agent with which it is communicating and optionally to implement a simple access restriction policy.

— The~~A~~ dev-config-id field~~, which~~ identifies the ~~current~~ configuration being offered for initial consideration during this association~~of the agent~~ as described in 7.4.3. For standard configurations, the value placed in the dev-config-id field ~~value~~ shall lie between standard-config-start and standard-config-end, inclusive. For extended configurations, the value placed in the dev-config-id field ~~value~~ shall lie between extended-config-start and extended-config-end, inclusive.

— The~~A~~ data-req-mode-capab~~, which~~ field defines the data request modes supported by the agent (see 8.9.3.3.3).

— ~~An~~ The option-list field ~~that~~ contains a list of additional attributes the agent wishes to communicate. The use of this field is not defined by this standard. This field should be an empty list unless otherwise defined in the ISO/IEEE 11073-104zz specializations.

8.7.3.2 Association response

Change the dashed list as shown:

— ~~A~~ The result field representing the outcome of the association procedure.

— The protocol-version field contains the version of the common data protocol chosen by the manager if the result field is equal to accepted or accepted-unknown-config.

— The encoding-rules field contains t~~T~~he one, and only one, DataApdu encoding rule chosen by the manager if the result field is equal to accepted or accepted-unknown-config.

  — The manager shall always support MDER to enable interoperability.

  — Alternatively, the manager may select one of the other encoding rules, besides MDER, that are offered by the agent.

NOTE—MDER is always supported by both the agent and the manager. However, if the agent offers additional encoding rules to the manager, it can be concluded that the agent had a valid reason to do so (i.e., the development of additional encoding rule support is not done without a compelling product reason). Thus, if an agent offers additional encoding rules beyond MDER, it is suggested that the manager honor one of the additional encoding rules offered if possible. For example, if an agent offers MDER and packed encoding rules (PER), it is suggested that the manager honor the PER encoding, if possible. If an agent offers MDER and XML encoding rules (XER), it is suggested that the manager honor the XER encoding rules, if possible. If an agent offers MDER, PER, and XER, this standard offers no suggestion as to the preferred encoding rule selection.

— The nomenclature-version field contains the version of the nomenclature chosen by the manager if the result field is equal to accepted or accepted-unknown-config.

— The functional-units ~~A~~ field indicat~~inges~~ the common functional units and optional features chosen by the manager if the result field is equal to accepted or accepted-unknown-config.

— The system-type field contains t~~T~~he system type (manager in this case since the message originated from the manager).

— The system-id field has t~~T~~he unique system ID of the manager. EUI-64 is used to uniquely identify the manager. An agent may use this field to determine whether it is communicating with the intended manager.

— The dev-config-id field shall be manager-config-response in the response.

— The data-req-mode-capab field shall be zero in the response.

— The option-list field is not currently used by this standard. This field should be an empty list.

8.7.3.3 Manager procedure

Change the second through the fourth paragraphs as shown:

A manager may reject the association for any of t~~T~~he possible rejection reasons ~~are~~ enumerated in 8.7.3.2. If the manager rejects the association, it shall transition to the Unassociated state.

If the request is not rejected by the manager, the result field in the Association Response message from the manager indicates whether the manager understands the configuration. If the manager recognizes the value in the dev-config-id field as representing a known standard device specialization or as a configuration from a previous association, the manager shall send an Association Response message with a result field of accepted and transition to the Operating state or may send an Association Response message with a result field of accepted-unknown-config to force the agent to enter configuring state in order to check attributes from the MDS object prior to final acceptance of the association.

If the manager does not recognize the value in the dev-config-id field, the manager shall send an Association Response message with the result field set to accepted-unknown-config and transition to the Configuring state.

8.7.5 Error conditions

Change the first paragraph as shown:

The agent shall wait for an Association Response message for a TO_{assoc} (timeout: association procedure) period. If the TO_{assoc} period expires, the agent shall retransmit the Association Request message ~~up to RCassoc (retry count: association procedure) times after the first timeout,~~ with a new TO_{assoc} period ~~in between each successive message~~. This process shall be repeated until an Association Response is received or RC_{assoc} (retry count: association procedure) attempts have been made after the first timeout, whichever comes first. This results in a maximum of RC_{assoc} + 1 Association Requests. If, after this retry sequence, the

agent does not successfully receive any Association Response messages, then the agent shall send an Association Abort message to the manager and transition back to the Unassociated state.

8.7.6 Test association

Change as shown:

In the first step, the agent passes the manager two bits of information in the functional-units field of the PhdAssociationInformation structure.

8.8 Configuring procedure

8.8.3 Normal procedures

Insert the following item as shown:

— The Dev-Configuration-Id attribute of the MDS object that corresponds to the configuration being described,

— All the objects supported by the agent except the MDS object, and

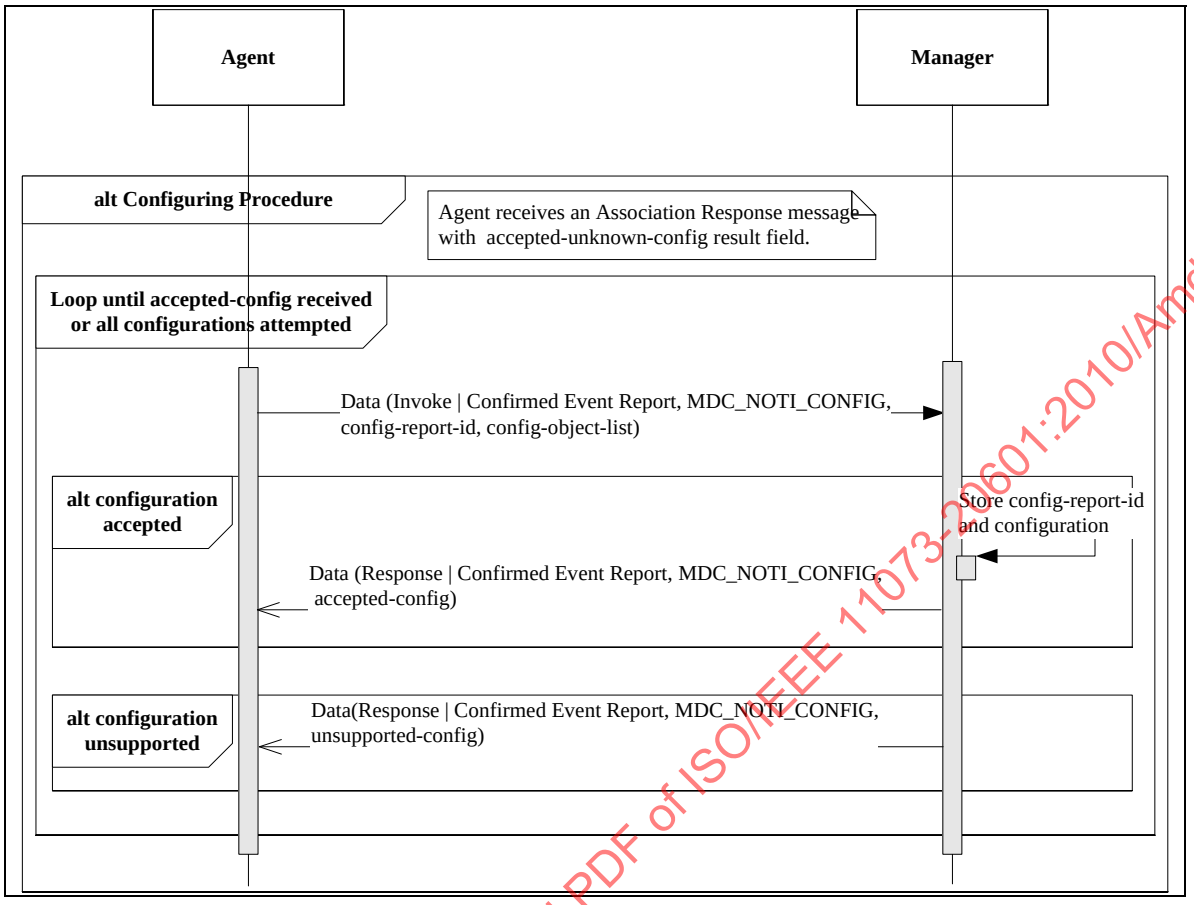
Change the second paragraph after the dashed list as shown:

For objects that report on the same attributes each time, the fixed format event report (see 7.4.5) is recommended, and the agent shall send an Attribute-Value-Map describing the message layout prior to sending a fixed format event report for the object. In the case of scanner objects that use the grouped format event reports, the agent shall send the Scan-Handle-Attr-Val-Map describing the layout prior to sending grouped format event reports from that scanner. Typically, the value maps are reported in the configuration event report.

Insert the following paragraph immediately before Figure 14:

The agent's configuration is identified by the Dev-Configuration-Id attribute of its MDS object, and the agent passes this value to the manager in the dev-config-id field in the Association Request message or the config-report-id field of the configuration event report message.

Replace Figure 14 with the following figure:



Change the paragraphs after Figure 14 as shown and insert the new NOTE and final paragraph:

An agent that conforms to one or more device specializations and/or profiles that define standard configurations (i.e., ISO/IEEE 11073-104zz specializations) shall support one or more of the standard configurations and may support one or more extended configurations. For interoperability, this agent shall send the supported standard configurations as a fall back if the extended configurations are unsupported. If an agent implements multiple ISO/IEEE 11073-104zz specializations, its System-Type-Spec-List attribute contains a list of type/version pairs, each referencing the respective device specialization and version of that specialization.

If the agent conforms to a standard configuration, it shall ~~use~~ set the value of a ~~d~~Dev-~~c~~Configuration-~~i~~Id as defined in the specific ISO/IEEE 11073-104zz device specialization. These standard configuration ~~d~~Dev-~~c~~Configuration-~~i~~Id values are assigned in the range between standard-config-start and standard-config-end, inclusive.

When an agent submits a ~~dev-config-id~~configuration event report corresponding to a standard configuration, the configuration message need not contain the configuration information and may send an event-type of MDC_NOTI_CONFIG with a standard configuration ID in the config-report-id field and an empty c~~C~~onfig-o~~O~~bject-l~~L~~ist. If the manager does not recognize the standard configuration (e.g., the manager was released prior to the device specialization being released), it shall send a response of standard-config-unknown. The agent may retry the configuration for the standard ~~device~~ configuration, but on the retry, it shall ~~by~~ send~~ing~~ the full configuration information instead of an empty config-obj-list.

NOTE—If the manager is able to interoperate with the provided standard configuration, it may accept that configuration. If the manager stores configurations, it may store this configuration for future reference when *any* agent uses this standard configuration identification and henceforth may treat the configuration as recognized.

An agent having a nonstandard configuration shall assign a unique ID to its configuration by generating a value for ~~d~~Dev-~~c~~Configuration-~~i~~Id in the range between extended-config-start and extended-config-end, inclusive.

An agent ~~may~~ should use the same value for D~~d~~ev-C~~c~~onfiguration-I~~i~~d in future Association Requests consistently. This has two implications. The same Dev-Configuration-Id shall not be used by an agent for subsequent associations to identify a different device configuration. An agent should use the same value for Dev-Configuration-Id in future Association Requests with the manager to denote the same configuration of the device. The selected value of dev-config-id shall be reported in the Dev-Configuration-Id attribute of the MDS object.

If the agent changes its configuration so that it can no longer support the old configuration or determines that a new configuration should be used in preference, it shall close any existing association by sending an Association Release message with a reason of configuration-changed. If the new configuration is a new extended configuration, the agent shall assign a new configuration ID. The next time the agent associates, it negotiates with the manager by stepping through each configuration in order of priority as described previously.

An agent may send an extended configuration with an empty config-object-list. This can happen, for instance, when an agent accepts plug-in components but presently does not have any inserted. The manager responds with either accepted-config or unsupported-config.

8.8.4 Exit conditions

Change the second paragraph as shown:

When the agent receives the accepted-config response from the manager, it shall transition to the Operating state. If the agent receives the unsupported-config response from the manager, it shall send the next configuration to the manager until no further configurations are available. Then it shall transition to the Disassociating state and send an Association Release Request message with a reason of no-more-configurations ~~and enter the Unassociated state~~.

8.9 Operating procedure

8.9.3 Normal procedures

8.9.3.2 MDS object attributes

Change the first paragraph as shown:

At any time in the Operating or Associated state, the manager may request the MDS object attributes of an agent by sending a data message with the “Remote Operation Invoke | Get” command and a reserved handle value of 0. The agent shall report its implemented MDS object attributes to the manager using a data message with the “Remote Operation Response | Get” response. See H.2.3 for example usages of this set of messages. Agents shall support a Get command that requests all attributes (i.e., the attribute-id-list is empty). Agents may support retrieval of a specific list of attributes ~~IDs~~. The handle is entered in the obj-handle field (A.10.4), and it is not included in the attribute ID list of the request or in the attribute list of the

response. If the manager requests specific MDS object attributes, indicated by the elements in attribute-id-list, and if the agent supports this capability, then the agent 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 agent to support this capability. If this capability is not implemented, then the agent shall respond with an error (roer) message with an error-value of not-allowed-by-object.

8.9.3.3 Measurement data transfer

8.9.3.3.3 Manager-initiated measurement data transmission overview

Change the third sentence in the third paragraph as shown:

When the manager starts a data mode, it provides a data-req-id that shall be used by the agent in all event reports related to that start request.

8.9.3.3.5 Manager-initiated time period mode

Change the first paragraph as shown:

The time period mode is used by the manager to enable an agent to send any data it collects for the duration of the requested time period (refer to Figure 18). When an agent receives the start DataRequest message from the manager, the agent shall send a DataResponse message acknowledging the result status of the request (DataReqResult) without transferring any measurement data in the response message. If DataReqResult is data-req-result-no-error, ~~anytime~~ then at any time that data becomes available~~,~~ the agent shall use the EVENT REPORT service to send event report(s) containing the measurement data to the manager until the time period as specified in the data request has expired, it receives a stop request from the manager, or the association between the agent and manager is terminated. The agent determines whether to use a Confirmed or Unconfirmed Event Report message to transfer the data.

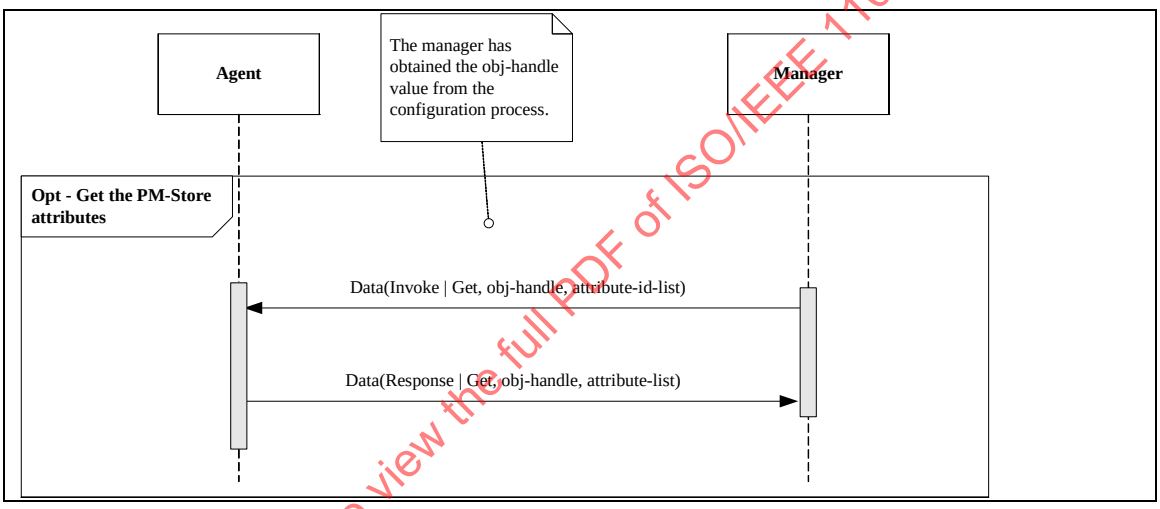
8.9.3.4 Persistently stored metric data transfer

8.9.3.4.2 Persistently stored metric data transmission

Change item a) as shown:

a) **Retrieving the PM-store attributes.** When the agent and manager are in the Operating state, the manager can inspect the configuration negotiated with the agent to determine the number of PM-store objects in the agent. The manager may query each PM-store to determine the number of PM-segments that exist within the PM-store. Figure 20 shows the sequence diagram of this operation. The manager sends a Get command to the agent requesting attribute information from a particular PM-store. The manager uses the handle number to reference the desired PM-store. The handle value is placed in the obj-handle field of the message (A.10.4) and is not present in the attribute-id list of the request or the attribute-list of the response. 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 an agent to support this capability. If this capability is not implemented, then the agent shall respond with an error (roer) message with an error-value of not-allowed-by-object. ~~The manager shall leave the attribute-id-list empty to request all attributes be returned. The agent responds with the values of the requested attributes.~~The manager can inspect the attributes to learn about how the store is configured. For instance, the PM-Store-Capab describes the capabilities of the store, and Number-Of-Segments defines how many segments are present in the store. See Table 9 for the full list of attributes and their definitions.

Replace Figure 20 with the following figure:



Change the first three paragraphs of item c) as shown:

c) **Transfer PM-segment content.** The manager retrieves specific PM-segments by using the ~~Trig-Segm-Data-Xfer~~ Trig-Segment-Data-Xfer ACTION method to initiate the data transfer (see Figure 22). In the first step, ~~T~~the manager sends the ACTION method to the agent with the handle of the PM-store object to access. The argument to this ACTION method is the instance number of the segment to transfer.~~shall pass on information about the PM-store handle to access and the segment instance number to transmit.~~

The agent shall decide whether the request can be honored. It checks for a valid segment number, available segment data (~~i.~~e.g., ~~they~~ the segment could be in the update process), or any other error conditions. If there is an error, the agent shall return an appropriate error code in the response and ignore the transmit request. Otherwise, the agent shall send a tsxr-successful response code to indicate that it has received the request and it can be honored.

The manager may send the Trig-Segment-Data-Xfer~~Trig-Segm-Data-Xfer~~ ACTION invoke message at any time. However, if the manager does send a Trig-Segment-Data-Xfer~~Trig-Segm-Data-Xfer~~ ACTION invoke message while a Clear-Segments ACTION invoke message is outstanding, the agent may generate a Trig-Segment-Data-Xfer~~Trig-Segm-Data-Xfer~~ ACTION

response message with a return code of trig-segm-xfer-rsp = tsxr-fail-clear-in-process. An example of when this return code might be sent is if the storage medium for the PM-store is a single Flash device. When a Flash device is being erased, it might cause the entire Flash device to be inaccessible.

Change item d) and insert new text as shown:

d) **Clear a PM-segment.** The agent may support PM-segment clearing. The manager determines whether the agent supports any of the clearing functions by inspecting 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.

The manager may clear a PM-segment at any time and uses the sequence shown in Figure 23. A typical time for clearing a segment is directly after the entire segment was transferred to the manager. The manager recognizes this condition when it receives a SegmEvtStatus with the sevtsta-last-entry bit set.

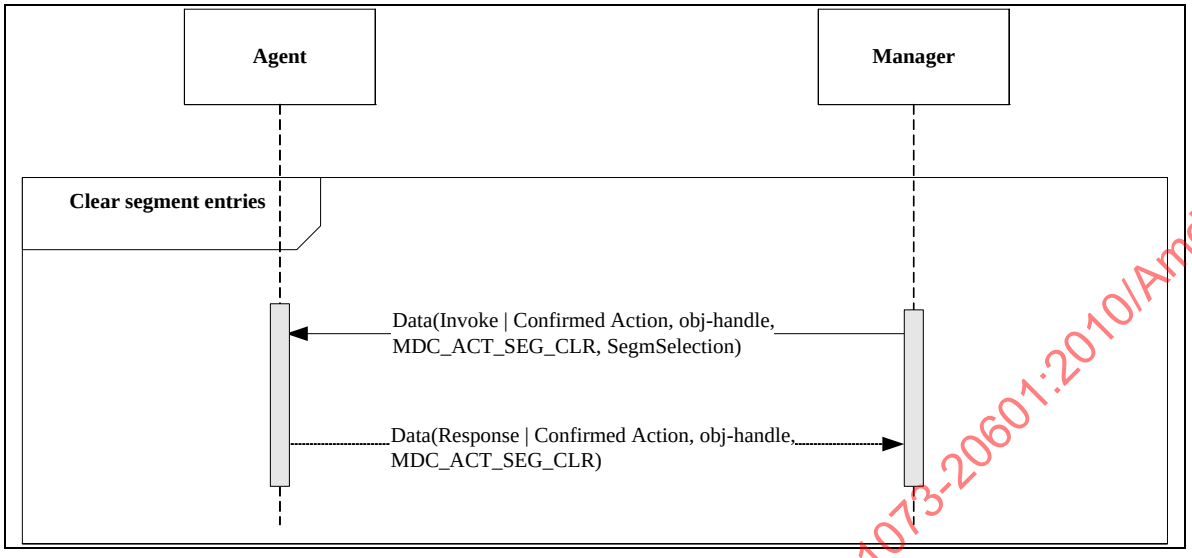
Whenever the manager decides to clear segment(s), it sends an ACTION command to the agent with the Clear-Segments method and segment selection criteria of all segments, a particular list of segments, or any segments within a given time range. ~~T~~If the agent supports this function, then it shall support clearing all segments (pmsc-clear-segm-all-sup), may~~,~~ ~~should~~ support clearing a particular list of segments (pmsc-clear-segm-by-list-sup), and may support the time range selection criteria (pmsc-clear-segm-by-time-sup). The manager determines which capabilities are supported by inspecting the individual PM-Store-Capab attribute bits.

If the manager invokes the Clear-Segments method but the agent does not support this function at all (no pmsc-clear-segm-* flags in the PM-Store-Capab attribute being set), then the agent shall respond with an roer DataApdu with an RoerErrorValue of “no-such-action”. If the manager invokes the Clear-Segments method but the agent does not support the particular action (list of segments or range of segments), then the agent shall respond with a roer DataApdu with an RoerErrorValue of “not-allowed-by-object”.

When the agent receives a Clear-Segment command, it may delete all present entries and leave the segment, or it can remove the segment. The manager determines which capabilities are supported by inspecting the pmsc-clear-segm-remove bit of the PM-Store-Capab attribute.

As described in 6.3.7.4, the clear method may not clear all of the selected PM-segments. For verification purposes, a manager may issue Get-Segment-Info actions and GET requests to monitor the actual clearing and/or removal of any segments.

Replace Figure 23 with the following figure:



8.9.5 Error conditions

8.9.5.2 Confirmed Action

Change as shown:

After sending a Confirmed Action invoke message, the manager shall wait for a Confirmed Action response message for a TO_{ca} (timeout: confirmed action service) period by default unless another timeout applies (e.g., $TO_{clr\text{-}pms}$ overrides TO_{ca} as described in 8.9.5.6). If the TO_{ca} expires, the manager shall send an Association Abort message to the agent and transition back to the Unassociated state.

8.9.5.3 Confirmed Event Report

Change as shown:

After sending a Confirmed Event Report invoke message, the agent shall wait for a Confirmed Event Report response message for a $TO_{cer\text{-}*}$ (timeout: confirmed event report service) period. If the $TO_{cer\text{-}*}$ expires, the agent shall send an Association Abort message to the manager and transition back to the Unassociated state.

The $TO_{cer\text{-}*}$ is defined on a per-object basis. Each of the objects in this standard that generate event reports has a separate timeout value that is reported by an appropriate attribute in each object:

— $TO_{cer\text{-}mds}$ (~~TOcer-~~timeout for the MDS object) MDS.Confirm~~ed-~~Timeout

— $TO_{cer\text{-}pms}$ (timeout ~~TOcer-~~for the PM-store object) Segm.Confirm~~ed-~~Timeout

— $TO_{cer\text{-}scan}$ (timeout ~~TOcer-~~for the scanner object) Scan.Confirm~~ed-~~Timeout

8.9.5.6 Special timeouts

Change the last paragraph as follows:

~~For TOsp-pms, a~~After sending a Confirmed Action (MDC_ACT_SEG_TRIG_XFER) invoke message and receiving the response, the manager shall wait up to a $TO_{sp\text{-}pms}$ (timeout: special segment transfer timeout of the PM-store object) period for a Confirmed Event Report (segm-evt-status=sevtsta-last-entry, semg-data-event-entries) invoke message~~ for a TOsp-pms (timeout: special segment transfer timeout of the PM-store object) period~~. If the $TO_{sp\text{-}pms}$ expires, the manager shall send an Association Abort message to the agent and transition back to the Unassociated state. The manager shall handle the Confirmed Action (MDC_ACT_SEG_TRIG_XFER) invoke message as it would any other action; it shall follow the timeout procedures described in 8.9.5.2 for Confirmed Actions.

8.10 Disassociating procedure

8.10.3 Normal procedures

Change the second item in the dashed list as shown:

— The configuration-changed reason is used by the agent during the Operating state to indicate that the agent's configuration changed and it is not possible to continue sending data with the previously agreed-upon configuration. Typically, the agent follows this message by sending a new Association Request with a ~~new~~ different Dev-Configuration-Id value in the dev-config-id field; however, this step is not required.

8.11 Message encoding

Change the last paragraph as shown:

All of the nomenclature codes used in this standard are defined using the Reference ID ~~MDC_...~~representation (i.e., MDC_...), but the nomenclature numeric codes shall be used during transmission regardless of which encoding rule was negotiated. Annex I contains a listing of the defined values for all the codes utilized in this standard.

8.12 Time coordination

8.12.1 General

Change as shown:

There are ~~three~~four types of clocks that an agent may implement: absolute time, base time with offset to local time, relative time, and high-resolution relative time. In all cases, information about the clock capabilities of the agent and whether one or more of the clocks are synchronized with an external time source can be found via the Mds-Time-Info attribute in Table 2. All bit references in the subclauses are part of this attribute. ~~All agents with any type of clock shall support this attribute.~~If an agent has ~~any~~a ~~type of~~ clock of any type and this attribute is not present, the manager shall ~~understand~~ assume ~~that~~ the default resolution ~~is~~ of 0 (i.e., unknown).

8.12.2 Absolute time

8.12.2.1 General

Change the second, third, and fourth paragraphs as shown:

Agents may support an independent method to synchronize the internal RTC to external clock sources. The synchronization method used is not in the scope of this standard. However, the agent shall indicate whether it synchronizes absolute time using the mds-time-capab-sync-abs-time bit. If synchronization is supported, the protocol used to synchronize the internal RTC (e.g., NTP and SNTP) is reported in the time-sync-protocol field using IDs such as ~~MDC_EXT_PROTO_TIME_NTP~~MDC_TIME_SYNC_NTPV3. The mds-time-state-abs-time-synced bit shall be set only when the agent believes its Date-and-Time attribute is synchronized with the external clock source.

Agents may wish to indicate to the manager whether ~~it should~~or not to set the time with the Set-Time action. If the agent is aware that its notion of the time is inaccurate, it shall set the mds-time-mgr-set-time bit, in which case ~~is set,~~ the manager shall ~~use~~ invoke the Set-Time action command to set the absolute time on the agent. The Set-Time shall be sent within a TO_{config} time period after ~~responding to the Association Request~~receiving the attribute from an MDS Get message. ~~If the bit is not set~~Upon receiving the Set-Time action, the agent shall clear the mds-time-mgr-set-time bit before sending the Set-Time confirmation. If on further occasions the agent needs to have the time set, the agent shall set the mds-time-mgr-set-time bit and either wait for a GET request to be issued by the manager or send the updated attribute in a scan event report. In some cases, the agent does not want the manager to set the clock. This situation can occur when the agent is synchronizing the clock via an external clock source or when the user has set the clock locally. In this case, neither the mds-time-mgr-set-time nor the mds-time-capab-set-clock bits shall be set, and the manager shall not attempt to set the clock.

The Date-and-Time and Absolute-Time-Stamp attributes report the agent's date and time. For some usages, it is important that the agent report the date and time as it was displayed to the person using the device (e.g., a glucose meter). For other usages, reporting a time that is coordinated to a universal time system such as coordinated universal time ~~coordinated~~(UTC) is important. For example, this situation can occur when the date and time are set in the factory to UTC and the device does not expose a method for altering the date and time. In some usages, it is advantageous if there is a continuous time basis, and with an offset to the local time that can be changed easily. For example, this situation can occur with devices that are used by people when travelling and must change the time on their device when in a different time zone. On the manager side, the ability to associate the measurement times with the manager's notion of time is critical for some usages.

8.12.2.2 Comparable time

Change the first paragraph as shown:

This standard utilizes a concept of "comparable time" to support all ~~three~~ four usages of managing the time stamp of measurements. The key concepts of comparable time are as follows:

Change the second item in the first dashed list as shown:

— If a set of measurements was collected when the current real-time clock was set differently, then the agent shall either discard the data or communicate the data along with the number of 1/100 of a second to add to each of the measurement times to place them on the same timeline as the agent's current Date-and-Time attribute.

Insert a new subclause as shown:

8.12.2a Base time with offset

This standard includes base time with an offset in minutes to local time as a format for timestamps. The time displayed to the user, or local time, of the agent is represented as two components: one that is a base time and a second that provides the offset in minutes from the base time to local time. This format provides the advantage over absolute time that it can maintain a contiguous time line and accommodate time changes, such as for daylight savings or time zones. This is more efficient than using the mechanisms of comparable time and would be recommended for use in devices that store observations taken over extended periods.

Note, however, that if the base time is changed, then the time adjustment shall be indicated using the same mechanisms as for absolute time.

In this standard, base time representation follows the definition of NTP. Base time is represented as the number of seconds since 00:00 on 1st January 1900 as a 32-bit unsigned integer (without accounting for leap seconds in a similar fashion to NTP), and a 16-bit unsigned fraction of seconds field expressed as x/65 536 s. A 16-bit signed integer field gives the offset from base time to local time in minutes.

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. If base time is aligned to 00:00 on 1st January 1900 in the UTC time zone (with an accuracy appropriate to the application), then the bit mds-time-bo-time-utc-aligned(14) in MdsTimeCapState shall be set.

8.12.3 Relative time

Change the first paragraph as shown:

Agents may implement a relative timer with time resolution down to 125 µs [least significant bit (LSB)]. This resolution is sufficient for sampling rates up to 8 kHz, permits high-resolution relative time periods to be measured, and spans time periods up to 6.2 days. If relative time is used with either temporarily stored measurements and/or a PM-store, agents shall ensure that the length of storage time never exceeds the resolution of the timer (i.e., 6.2 days). This assurance from an agent allows the manager to query the agent's current relative time and compute how long ago the measurement was taken. If longer storage times are required, either absolute time or high-resolution relative time attributes are used. Agents shall indicate support for relative time by setting the mds-time-capab-relative-time bit in the Mds-Time-Info attribute. This timer shall be initialized prior to association. With the exception of counter rollover, it shall monotonically increase its count and shall not have its value changed once initialized. The actual time resolution (i.e., internal update period) is defined by the agent, but should be appropriate for the purpose of that device.

9. Conformance model

9.2 Conformance specification

Change the third item in the dashed list as shown:

— Use of extended communication services (~~i.~~e.g., scanners~~)~~, scan periods, and scanner configuration~~bility~~)

9.4 General conformance

9.4.1 General ICS

Change the Support column text of the GEN-4 row of Table 22, change as shown:

(list the set of IEEE 11073-20601 device specializations and profiles that were followed and prepare the information specified in 9.5)

9.5 Device additions/extensions ICS

9.5.2 Personal health device DIM object and class (POC) ICS

Change Table 26 as shown:

Table 26—Template for POC ICS

Index	Feature	Reference	Status	Support	Comment
POC-*n*	Object Description	The class of the object (~~i.~~e.g., numeric, etc.)	Implemented	Specify restrictions (e.g., maximum number of supported instances)	

Annex A

(normative)

ASN.1 definitions

A.2 Common data types

A.2.1 Integer and bit string data types

Insert the following text after the line that says "INT-I32 ::= INTEGER (-2147483648..2147483647)":

```
--
-- Unless otherwise stated, all unused (reserved) bits in any of the BITS-* constructs shall be set to ~~zero~~0
by
-- the sender and unless otherwise stated, shall be ignored by the receiver if they are set to 1.
--
-- 8-bit bit string
--
BITS-8 ::= BIT STRING (SIZE(8))
```

A.2.5 Type ID data type

Change the NomPartition definition as shown:

```
NomPartition ::= INT-U16 {
                    nom-part-unspec(0),          -- unspecified
                    nom-part-obj(1),             -- object-oriented partition
                    nom-part-metric(2),          -- metric [supervisory control and data acquisition
                                                 -- (SCADA)] partition
                    nom-part-alert(3),           -- alerts/events partition
                    nom-part-dim(4),             -- dimensions partition
                    nom-part-vattr(5),           -- virtual attribute partition for operation objects
                    nom-part-pgrp(6),            -- parameter group ID partition
                    nom-part-sites(7),           -- measurement and body site locations
                    nom-part-infrastruct(8),     -- infrastructure elements partition
                    nom-part-fef(9),             -- file exchange format partition
                    nom-part-ecg-extn(10),       -- electrocardiogram extensions partition
                    nom-part-idco-extn(11),      -- implantable cardiac device observation extensions
                    nom-part-phd-dm(128),        -- disease management
                    nom-part-phd-hf(129),        -- health and fitness
                    nom-part-phd-ai(130),        -- aging independently
                    nom-part-ret-code(255),      -- return codes partition
                    nom-part-ext-nom(256),       -- IDs of other nomenclatures and dictionaries
                    nom-part-priv(1024)          -- private partition
}
```

A.2.13 Absolute time data type

Change the last paragraph as shown:

Note that the agent defines the actual time resolution used (~~i.~~e.g., if the clock resolution is 1 s, then sec-fractions is always zero). Agents should have a resolution of 1 s or better.

Insert a new annex subclause as shown:

A.2.13a Base time with offset data type

The base time with offset data type specifies the time of day and includes a time offset field to specify the difference in minutes between the base time and the local time. Base time is encoded as the number seconds since midnight of 1st January 1900 specified as INT-U32 and the fraction of x/65 536 s as INT-U16. The time offset field is specified as INT-I16. The base offset time data type is defined as follows:

```
--
BaseOffsetTime ::= SEQUENCE {
                    bo-seconds          INT-U32,
                    bo-fraction         INT-U16,
                    bo-time-offset      INT-I16
}
```

A.2.14 Operational state data type

Change OperationalState as shown:

```
--
OperationalState ::= INT-U16 {
                    disabled(0),
                    enabled(1),
                    notAvailable(2)  --the value notAvailable is not used in this standard
}
```

A.4 ACTION-method-related data types

Change the clause and insert new text as shown:

```
--
-- SetTimeInvoke selects the date and time to be set.
--
SetTimeInvoke ::= SEQUENCE {
                    date-time           AbsoluteTime,
                    accuracy            FLOAT-Type      -- accounts for set time (e.g., 2 min error);
                                                        -- value is defined in seconds. This parameter is
                                                        -- inherited from ISO/IEEE 11073-10201:2004
                                                        -- [B13], but not used. Thus, it shall be zero (0).
}

--
-- SetBOTimeInvoke selects the date and time to be set in base offset time format. If both seconds and
fraction fields are set to 0x0, they are ignored in the action and only the offset is changed.
```

```
--
SetBOTimeInvoke ::= SEQUENCE {
        date-time       BaseOffsetTime
}

--
-- SegmSelection selects the PM-segments that are subject to the method.
--
SegmSelection ::= CHOICE {
        all-segments    [1] INT-U16,    -- if this type is chosen to select all segments
                                        -- the actual contents of the field is “do not care”
                                        -- and shall be zero
        segm-id-list    [2] SegmIdList, -- using this list requires that the manager already
                                        -- knows the Instance-Number attributes of the
                                        -- PM-segments, e.g., from a previous
                                        -- Get-Segment-Info method call.
        abs-time-range  [3] AbsTimeRange
                                        -- support of abs-time-range is optional, indicated in
                                        -- the PM-Store-Capab attribute
        bo-time-range   [4] BOTimeRange
                                        -- support of bo-time-range is optional, indicated in
                                        -- the PM-Store-Capab attribute
}

--
-- SegmIdList selects PM-segments by ID.
--
SegmIdList ::= SEQUENCE OF InstNumber

--
-- AbsTimeRange allows selection of PM-segments by time period.
--
AbsTimeRange ::= SEQUENCE {
        from-time       AbsoluteTime,
        to-time         AbsoluteTime
}

--
-- BOTimeRange allows selection of PM-segments by time period specified as base offset time.
--
BOTimeRange ::= SEQUENCE {
        from-time       BaseOffsetTime,
        to-time         BaseOffsetTime
}
--
-- SegmentInfoList returns the object attributes (except the Fixed-Segment-Data) of all
-- selected PM-segment object instances in response to the Get-Segment-Info PM-store method.
-- This is required by the manager to retrieve the dynamic information about the segments.
--
SegmentInfoList ::= SEQUENCE OF SegmentInfo

SegmentInfo ::= SEQUENCE {
        seg-inst-no     InstNumber,
        seg-info        AttributeList
}
```

A.8 Association protocol definitions

Change ProtocolVersion as shown:

```
-- All unassigned " ProtocolVersion " bit values are reserved for future expansion and shall be set to zero.
ProtocolVersion ::= BITS-32 {
                    protocol-version1(0),           -- This bit shall be set if IEEE Std 11073-20601-2008
version 1 of the data
                                                    -- exchange protocol is supported
                    protocol-version2(1),           -- This bit shall be set if IEEE Std 11073-20601a
                                                    -- is supported
}
```

Change FunctionalUnits as shown:

```
-- All unassigned " FunctionalUnits " bit values are reserved for future expansion and shall be set to zero.
FunctionalUnits ::= BITS-32 {
                    fun-units-unidirectional(0),    -- Reserved for future use. This bit shall be set if
                                                    -- the agent is unidirectional
                    fun-units-havetestcap(1),       -- This bit indicates if the device can enter a
                                                    -- test association
                    fun-units-createtestassoc(2)    -- This bit is used to indicate an intention to
                                                    -- form a test association
}
```

A.10 Data protocol definitions

A.10.2 Data protocol frame

Change the comment for RejectResult as shown:

```
-- At any point, if a DataApdu invoked action (roiv-*) requires the receiver to reject an
-- operation prior to attempting to execute it, the receiver shall send
-- back a RejectResult. The invokeID is used to determine which invocation resulted in a
-- rejection condition. The problem field shall be filled in with a value from the RorjProblem list
-- below.
-- The RejectResult shall be used when a message is rejected.
RejectResult ::= SEQUENCE {
                    problem             RorjProblem
}
```

A.10.6 ACTION service

Change the first lines of comments as shown:

```
-- For action requests defined in this standard, obj-handle shall either be 0 to represent the MDS object or
-- a handle representing a PM-store object.
-- For the action-types defined in Table 3 and Table 10, the corresponding action-info-args
-- structures shall be used. Accordingly, action-info-args will be one of DataRequest,
-- SetTimeInvoke, SetBOTimeInvoke, SegmSelection, or TrigSegmDataXferReq.
```

A.11 Data types for new object attributes and object services

A.11.2 MDS-related data types

Change the MdsTimeInfo structure and insert new text as shown:

```
-- time-sync-accuracy allows an agent to report how closely synchronized its clock is with
-- respect to the clock sync master when time synchronization is used.
MdsTimeInfo ::= SEQUENCE {
        mds-time-cap-state              MdsTimeCapState,
        time-sync-protocol              TimeProtocolId,   -- this is a nomenclature code from
                                                          -- nom-part-infrastruct partition
        time-sync-accuracy              RelativeTime,     -- 0xFFFFFFFF if unknown
                                                          -- 0 if better than 1/8 ms
        time-resolution-abs-time        INT-U16,          -- if
                                                          -- mds-time-capab-real-time-clock
                                                          -- is set then this indicates the
                                                          -- rResolution of the agent's
                                                          -- absolute time clock.
                                                          -- 0 if unknown; otherwise,
                                                          -- the number of 1/100 s
                                                          -- that elapse with each clock
                                                          -- increment. For example, if an
                                                          -- agent has a clock that clicks at
                                                          -- 1 s intervals, this value
                                                          -- would be 100.
                                                          -- if mds-time-capab-bo-time
                                                          -- is set then this indicates the
                                                          -- resolution of the agent's
                                                          -- base time clock.
                                                          -- 0 if unknown; otherwise,
                                                          -- the number of 1/65 536 s
                                                          -- that elapse with each clock
                                                          -- increment. The value of 0xFFFF
                                                          -- is reserved to indicate an interval
                                                          -- 1 s.
        time-resolution-rel-time        INT-U16,          -- Resolution of the agent's
                                                          -- relative time clock. 0 if
                                                          -- unknown; otherwise, the number
                                                          -- of 125 µs that elapse
                                                          -- with each clock increment. For
                                                          -- example, if an agent has a clock
                                                          -- that clicks at 1 s intervals,
                                                          -- this value would be 8000.
        time-resolution-high-res-time   INT-U32           -- Resolution of the agent's
                                                          -- high-resolution time clock.
                                                          -- 0 if unknown; otherwise, the
                                                          -- the number of microseconds
                                                          -- that elapse with each clock
                                                          -- increment. For example, if an
                                                          -- agent has a clock that clicks
                                                          -- at 1 s intervals, this value
                                                          -- would be 1 000 000.
}
```

Change the MdsTimeCapState comments and insert new text as shown. Note that the last line of changes are strikeouts of the carriage return and tabs so "manager shall set the time" is all on one line after edits are completed.

-- Only one of mds-time-capab-real-time-clock and mds-time-capab-bo-time shall be specified.
-- Only one of mds-time-capab-sync-abs-time and mds-time-capab-sync-bo-time shall be specified.
-- Only one of mds-time-state-abs-time-synced and mds-time-state-bo-time-synced shall be specified.
-- All unassigned " MdsTimeCapState " bit values are reserved for future expansion and shall be set to zero.
MdsTimeCapState ::= BITS-16 {

mds-time-capab-real-time-clock(0),	-- device supports an internal RTC -- following absolute time
mds-time-capab-set-clock(1),	-- device supports Set-Time Action -- or Set-Base-Offset-Time Action
mds-time-capab-relative-time(2),	-- device supports RelativeTime
mds-time-capab-high-res-relative-time(3),	-- device supports -- HighResRelativeTime
mds-time-capab-sync-abs-time(4),	-- device syncs AbsoluteTime
mds-time-capab-sync-rel-time(5),	-- device syncs RelativeTime
mds-time-capab-sync-hi-res-relative-time(6),	-- device syncs HiResRelativeTime
mds-time-capab-bo-time(7),	-- device supports BaseOffsetTime
mds-time-state-abs-time-synced(8),	-- AbsoluteTime is synced
mds-time-state-rel-time-synced(9),	-- RelativeTime is synced
mds-time-state-hi-res-relative-time-synced(10),	-- HiResRelativeTime is synced
mds-time-mgr-set-time(11)	-- manager ~~is encouraged to~~ shall set the time
mds-time-capab-sync-bo-time(12),	-- device syncs BaseOffsetTime
mds-time-state- bo-time-synced(13),	-- Base time is synced
mds-time-state-bo-time-UTC-aligned(14)	-- Base time is aligned to UTC

}

A.11.3 Metric-related data types

Change the lead comments for MetricSpecSmall as shown:

--
-- The Metric Spec Small attribute is an abbreviated MetricSpec attribute as defined in ISO/IEEE
-- 11073-10201:2004 [B13]. It defines availability, periodicity, and category of the measurement.
-- The setting of bits 0 to 5 is primarily informational and shall be set if the condition is true but
-- a manager cannot assume that if they are set, the behavior will be observed.
~~-- If the metric corresponds to default values (no bits set), the attribute is not required.~~
-- All unassigned "MetricSpecSmall" bit values are reserved for future expansion and shall be set to zero.
--

Change the lead comments for MetricStructureSmall as shown:

-- This attribute is partly inherited from ISO/IEEE 11073-10201:2004 [B13], but enhanced by
-- value ms-struct::ms-struct-compound-fix~~fix-comp-no~~. For this release of the standard, ms-struct-compound
-- and ms-struct-compound-fix shall only be used for Numeric objects. Additional structures would
-- need to be introduced in RT-SA and Enumeration objects to allow the use of compound structures.
-- If compound is used, the agent shall send no more than ms-comp-no observed values.
--

A.11.5 MDS services

Change the comments about the ScanReport* definitions as shown:

```
-- The ScanReport* definitions are used when reporting information about measurements that wereobject attribute value
-- changes (Attribute Change Sets)sampled. There are two vectors: A) single person or multiple person and B) variable format,
-- fixed format, or grouped format. Combinations of these vectors lead to the six top-level definitions:
-- ScanReportInfoVar, ScanReportInfoFixed, ScanReportInfoGrouped,
-- ScanReportInfoMPVar, ScanReportInfoMPFixed, and ScanReportInfoMPGrouped.
-- The SEQUENCE OF ObservationScan or ObservationScanFixed may contain multiple instances
-- of the same handle as long as there is a time stamp to distinguish between the instances.
-- In all cases, scan-report-no shall be initialized to zero at association time and monotonically
-- increasing by one until roll-over occurs.
```

Change the ObservationScanFixed comment as shown:

```
ObservationScanFixed ::= SEQUENCE {
        obj-handle      HANDLE,         -- unique identification of metricthe object
        obs-val-data    OCTET STRING    -- observed value data defined by obj-handle
}
```

Change the ScanReportInfoGrouped definition as shown:

```
ScanReportInfoGrouped ::= SEQUENCE {
        data-req-id     DataReqIdINT-U16,
        scan-report-no  INT-U16,        -- counter for detection of missing scan reports
        obs-scan-grouped SEQUENCE OF ObservationScanGrouped
}
```

Change the comment for ScanReportPerVar as shown:

```
--
-- The value used for person-id is vendor determined (i.e.g., if an agent has two buttons
-- to distinguish between two people, the agent may use ID 1 and 2 or ID 35 and 97).
-- The process of mapping this ID to a specific person is outside the scope of this
-- standard.
--
```

Change the ScanReportInfoMPGrouped definition as shown:

```
ScanReportInfoMPGrouped ::= SEQUENCE {
        data-req-id     DataReqIdINT-U16,
        scan-report-no  INT-U16,        -- counter for detection of missing scan reports
        scan-per-grouped SEQUENCE OF ScanReportPerGrouped

}
```

A.11.8 PM-store and PM-segment related data types

Change the PmStoreCapab definition and insert new text as shown:

```
PmStoreCapab ::=BITS-16 {
        pmsc-var-no-of-segm(0),         -- indicates that the number of PM-segments
```

```
                                        -- contained in this PM-store is dynamic and may
                                        -- change
                pmsc-segm-id-list-select(3),    -- PM-segments in the SegmSelection data type can
                                        -- be selected by defining a rangelist of segment
                                        -- identifiers
                pmsc-epi-seg-entries(4),        -- Some/ all PM-segments contain
                                        -- episodic/aperiodic entries and therefore have
                                        -- to contain explicit time stamp information
                pmsc-peri-seg-entries(5),       -- Some/all PM-segments contain periodically
                                        -- sampled entries and therefore the PM-segment
                                        -- or PM-store shall support the
                                        -- Sample-Period attribute
                pmsc-abs-time-select(6),        -- PM-segments in the SegmSelection data type can
                                        -- be selected by defining an abs-time-range range or
                                        -- bo-time-range depending upon which time
                                        -- mode the device supports
                pmsc-clear-segm-by-list-sup(7), -- clearing a list of segments is supported
                pmsc-clear-segm-by-time-sup(8), -- clearing segments by abs time range or
                                        -- bo time range is supported depending
                                        -- upon which time mode the device supports
                pmsc-clear-segm-remove(9),      -- if this bit is set, the agent will completely remove
                                        -- the specified PM-segment instance as part of the
                                        -- Clear-Segment method. If this bit is not set, it will
                                        -- just remove all entries from the specified
                                        -- PM-segment.
                pmsc-clear-segm-all-sup(10) ,   -- clearing all segments is supported
                pmsc-multi-person(12)           -- The PM-store supports PM-segment for more
                                        -- than one person
}
```

Change the SegmEntryHeader definition and insert new text as shown:

```
--
-- The following bit string defines optional data items that are in front of each segment entry.
-- Multiple data items are definable. In this case, the data item with the lower bit number shall come
-- in front of items with higher bit numbers. The header allows definition of data items that are common
-- to all elements in the entry. If all bits are zero, the segment entry event report shall begin with data
-- from the first element.
-- All unassigned " SegmEntryHeader " bit values are reserved for future expansion and shall be set to zero.
-- If any bits are set to one beyond the expected bits (e.g., a new bit was added in a later version),
-- the data shall not be retrieved since the offset to the first data element cannot be calculated.
--
SegmEntryHeader ::= BITS-16 {
                seg-elem-hdr-absolute-time(0),  -- entry preceded by absolute time
                                        -- (data type AbsoluteTime)
                seg-elem-hdr-relative-time(1),  -- entry preceded by relative time
                                        -- (data type RelativeTime)
                seg-elem-hdr-hires-relative-time(2), -- entry preceded by high resolution relative time
                                        -- (data type HighResRelativeTime)
                seg-elem-hdr-bo-time(3)         -- entry preceded by base offset time
                                        -- (data type BaseOffsetTime)
                                         -- option (0) and option (3) are mutually exclusive
}
```

Change the SegmEvtStatus definition and insert new text as shown:

```
SegmEvtStatus ::= BITS-16 {
                    sevtsta-first-entry(0),          -- this event contains the first segment entry
                    sevtsta-last-entry(1),           -- this event contains the last segment entry (both first
                                                     -- and last bits can be set if all entries fit in one event)
                    sevtsta-agent-abort(4),          -- transfer aborted by agent (manager shall reply
                                                     -- with the same status)
                    sevtsta-manager-confirm(8),      -- set in reply if segment was received correctly (if
                                                     -- not set in reply, agent shall ~~repeat the last
~~event)~~stop the segment
                                                     -- transfer and respond with an error (roer) code
                                                     -- of protocol-violation).
                    sevtsta-manager-abort(12)        -- sent in reply by manager (agent shall stop sending
                                                     -- messages)
}
```

Annex E

(normative)

State tables

E.1 General

Insert the following paragraph before the current first paragraph:

The intended audience for Table E.1 is primarily users who maintain the standards and ensure consistent use of state numbers in Table E.2 and Table E.3.

Change Table E.1 as shown:

Table E.1—States

State number	State	Used by agent	Used by manager
1	Disconnected	Y	Y
2	Connected Unassociated	Y	Y
3	Connected Associating	Y	Y
4	Connected Associated Configuring Sending Config	Y	
5	Connected Associated Configuring Waiting Approval	Y	
6	Connected Associated Configuring Waiting		Y
7	Connected Associated Configuring Checking Config		Y
8	Connected Associated Operating	Y	Y
9	Connected Disassociating	Y	Y

Insert a new annex subclause as shown:

E.1a Events

The intended audience for Table E.1a is primarily users who maintain the standards and ensure consistent use of state numbers in Table E.2 and Table E.3.

All the events of the agent and manager are defined in Table E.1a.

Event Table Notations:

REQ – A request from the application software interfacing with the state machine

IND – A condition asserted by a lower layer of software through a well defined API

Rx – PDU (protocol data unit) that has arrived on the input data stream

Table E.1a—Events

Event number	Event
1	IND Transport connection
2	IND Transport disconnect
3	IND Timeout and maximum retry not reached
4	IND Timeout and maximum retry reached
5	REQ Assoc
6	REQ Assoc Rel
7	REQ Assoc Abort
8	Rx aarq (*)
9	~~Rx~~REQ ~~aarq (~~acceptable and known configuration~~)~~
10	~~Rx aarq (~~REQ acceptable and unknown configuration~~)~~
11	~~Rx aarq (~~REQ unacceptable configuration~~)~~
12	Rx aare (*)
13	Rx aare(accepted)
14	Rx aare(accepted-unknown-config)
15	Rx aare(rejected-*)
16	Rx rlrq
17	Rx rlre
18	Rx abrt
19	Rx apdu(*) Any APDU received not explicitly covered for this state (corrupt, unknown, unexpected, etc.)
20	REQ ConfigEventReport available
21	Rx roiv-*
22	Rx roiv-cmip-get, handle = 0
23	Rx roiv-* but not (roiv-cmip-get, handle = 0)
24	Rx roiv-confirmed-event-report
25	Rx roiv-* but not (roiv-confirmed-event-report)
26	Rx (rors-*, roer-*, or rorj-*)
27	Rx rors-cmip-confirmed-event-report (unsupported-config) and additional configurations available
28	Rx rors-cmip-confirmed-event-report (unsupported-config) and no additional configurations available
29	Rx rors-cmip-confirmed-event-report (accepted-config)
30	Rx (rors-*, roer-*, or rorj-*), but not Rx: rors-cmip-confirmed-event-report
31	REQ Unsupported-configuration asserted
32	REQ Supported-configuration asserted
33	IND App: ConfigEventReport available
34	REQ roiv-cmip-confirmed-*

E.2 Agent state table

Change the text as shown:

Tx – APDU that is sent~~e~~d on the output data stream
Signal ID – x.y = state.event where x is given in Table E.1 and in Table E.1a.

All timeout values indicate the length of time to wait prior to asserting an "IND Timeout" condition.

Change Table E.2 as shown:

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
1.1	Disconnected	IND Transport connection	Connected Unassociated	"Shall" indicate to application layer.	None
2.2	Connected Unassociated	IND Transport disconnect	Disconnected	"Should" indicate to application layer.	None
2.5	Connected Unassociated	REQ Assoc	Connected Associating	Timeout=TO$_{assoc}$~~reset~~, retry=reset.	Tx aarq
2.6	Connected Unassociated	REQ AssocRel	Connected Unassociated <no state transition>	None.	None
2.7	Connected Unassociated	REQ AssocAbort	Connected Unassociated <no state transition>	~~Should not happen~~Can be used to synchronize state on both sides.	Tx abrt(reason undefined)
2.8	Connected Unassociated	Rx aarq(*)	Connected Unassociated	Agent-agent association.	Tx aare(rejected-permanent)
2.12	Connected Unassociated	Rx aare(*)	Connected Unassociated <no state transition>	Should not happen.	Tx abrt(reason undefined)
2.16	Connected Unassociated	Rx rlrq	Connected Unassociated <no state transition>	Should not happen.	Tx abrt(reason undefined)
2.17	Connected Unassociated	Rx rlre	Connected Unassociated <no state transition>	Should not happen. Ignore.	None
2.18	Connected Unassociated	Rx abrt	Connected Unassociated <no state transition>	None.	None
2.19	Connected Unassociated	Rx ~~prst~~apdu(*) Any APDU not covered in 2.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated <no state transition>	Should not happen.	Tx abrt(reason undefined)
3.2	Connected Associating	IND Transport disconnect	Disconnected	None.	None
3.3	Connected Associating	IND Timeout and maximum retry not reached	Connected Associating <no state transition>	Time~~r~~out=TO$_{assoc}$~~reset~~, increment retry count~~++~~.	Tx aarq
3.4	Connected Associating	IND Timeout and maximum retry reached	Connected Unassociated	None.	Tx abrt(reason response-timeout)
3.6	Connected Associating	REQ AssocRel	Connected Unassociated	None.	Tx abrt(reason undefined)
3.7	Connected Associating	REQ AssocAbort	Connected Unassociated	None.	Tx abrt reason defined by application

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
3.8	Connected Associating	Rx aarq(*)	Connected Unassociated	Agent-agent association.	Tx aare(rejected-permanent)
3.13	Connected Associating	Rx aare(accepted)	Connected Associated Operating	This causes a direct transition to operating state.	None
3.14	Connected Associating	Rx aare(accepted-unknown-config)	Connected Associated Configuring Sending Config	The manager has accepted the association but does not have a configuration.	None
3.15	Connected Associating	Rx aare(rejected-*)	Connected Unassociated	No further attempts to connect.	None
3.16	Connected Associating	Rx rlrq	Connected Unassociated	Should not happen. The agent has received a request to release the association, but it has not yet established an association.	Tx abrt(reason undefined)
3.17	Connected Associating	Rx rlre	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
3.18	Connected Associating	Rx abrt	Connected Unassociated	None.	None
3.19	Connected Associating	Rx ~~prst~~apdu(*) Any APDU not covered in 3.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
4.2	Connected Associated Configuring Sending Config	IND Transport Disconnect	Disconnected	None.	None
4.4	Connected Associated Configuring Sending Config	IND Timeout	Connected Unassociated	No reply.	Tx abrt(reason response-timeout)
4.6	Connected Associated Configuring Sending Config	REQ AssocRel (*)	Connected Disassociating	Software requests association release. Timeout=$TO_{release}$ ~~reset~~.	Tx rlrq(*)
4.7	Connected Associated Configuring Sending Config	REQ AssocAbort	Connected Unassociated	Software abort.	Tx abrt reason defined by application
4.8	Connected Associated Configuring Sending Config	Rx aarq(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
4.12	Connected Associated Configuring Sending Config	Rx aare	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
4.16	Connected Associated Configuring Sending Config	Rx rlrq	Connected Unassociated	None.	Tx rlre(normal)
4.17	Connected Associated Configuring Sending Config	Rx rlre	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
4.18	Connected Associated Configuring Sending Config	Rx abrt	Connected Unassociated	None.	None
4.19	Connected Associated Configuring Sending Config	Rx ~~prst~~apdu(*) Any APDU not covered in 4.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	TX abrt(reason undefined)
4.22	Connected Associated Configuring Sending Config	Rx roiv-cmip-get, handle=0	Connected Associated Configuring Sending Config <no state transition>	Manager allowed to probe MDS. See 6.3.2.6.1.	rors-cmip-get.(MDS Attributes) or roer not-allowed-by-object~~unsupported-action~~ if request is not for all attributes and agent does not support the request
4.23	Connected Associated Configuring Sending Config	Rx roiv-* but not (roiv-cmip-get, handle=0)	Connected Associated Configuring Sending Config <no state transition>	Not allowed until operating state is reached.	Tx roer (no-such-object-instance)
4.26	Connected Associated Configuring Sending Config	Rx (rors-*, roer, or rorj)	Connected unassociated	Should not happen.	Tx abrt(reason undefined)
4.32	Connected Associated Configuring Sending Config	REQ Send(ConfigReport)	Connected Associated Configuring Waiting Approval	The agent has a configuration that it has not yet tried with the manager. Timeout=TO_{config}	Tx EventReport(ConfigReport)

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
5.2	Connected Associated Configuring Waiting Approval	IND Transport Disconnect	Disconnected	None.	None
5.4	Connected Associated Configuring Waiting Approval	IND Timeout	Connected Unassociated	No reply.	Tx abrt(reason configuration-timeout)
5.6	Connected Associated Configuring Waiting Approval	REQ AssocRel(*)	Connected Disassociating	Software request association release. Timeout=$TO_{release}$ ~~reset~~.	Tx rlrq(*)
5.7	Connected Associated Configuring Waiting Approval	REQ AssocAbort	Connected Unassociated	Software abort.	Tx abrt reason defined by application
5.8	Connected Associated Configuring Waiting Approval	Rx aarq(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
5.12	Connected Associated Configuring Waiting Approval	Rx aare(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
5.16	Connected Associated Configuring Waiting Approval	Rx rlrq	Connected Unassociated	None.	Tx rlre(normal)
5.17	Connected Associated Configuring Waiting Approval	Rx rlre	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
5.18	Connected Associated Configuring Waiting Approval	Rx abrt	Connected Unassociated	None.	None
5.19	Connected Associated Configuring Waiting Approval	Rx apdu(*) Any APDU not covered in 5.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	TX abrt(reason undefined)

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
5.22	Connected Associated Configuring Waiting Approval	Rx roiv-cmip-get, handle=0	Connected Associated Configuring Waiting Approval~~Sending Config~~ <no state transition>	Manager allowed to probe MDS. See 6.3.2.6.1.	rors-cmip-get (MDS Attributes) or roer not-allowed-by-object if request is not for all attributes and agent does not support the request
5.23	Connected Associated Configuring Waiting Approval	Rx roiv-* but not (roiv-cmip-get, handle=0)	Connected Associated Configuring Waiting Approval <no state transition>~~Sending Config~~	Not allowed until operating state is reached.	Tx roer (no-such-object-instance)
5.27	Connected Associated Configuring Waiting Approval	Rx rors-cmip-confirmed-event-report (unsupported-config) and additional configurations available	Connected Associated Configuring Sending Config	Manager has rejected the configuration and additional configurations are available.	None
5.28	Connected Associated Configuring Waiting Approval	RX rors-cmip-confirmed-event-report (unsupported-config) and no additional configurations available	Connected Disassociating	Manager has rejected the configuration and additional configurations are not available.	TX rlrq(reason no more-configs)
5.29	Connected Associated Configuring Waiting Approval	Rx rors-cmip-confirmed-event-report (accepted-config)	Connected Associated Operating	Manager has accepted configuration.	None
5.30	Connected Associated Configuring Waiting Approval	Rx (rors-*, roer, or rorj), but not Rx: rors-cmip-confirmed-event-report.	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
8.2	Connected Associated Operating	IND Transport Disconnect	Disconnected	None.	None

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
8.4	Connected Associated Operating	IND Timeout	Connected Unassociated	No reply.	Tx abrt(reason response-timeout)
8.6	Connected Associated Operating	REQ AssocRel	Connected Disassociating	None. Timeout=$TO_{release}$ ~~reset~~.	Tx rlrq(normal)[4]
8.7	Connected Associated Operating	REQ AssocAbort	Connected Unassociated	None.	Tx abrt reason defined by application
8.8	Connected Associated Operating	Rx aarq(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
8.12	Connected Associated Operating	Rx aare(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
8.16	Connected Associated Operating	Rx rlrq	Connected Unassociated	If the agent has any outstanding invoke-ids, it shall assume that it shall receive no response to its request.	Tx rlre(normal)
8.17	Connected Associated Operating	Rx rlre	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
8.18	Connected Associated Operating	Rx abrt	Connected Unassociated	None.	None
8.19	Connected Associated Operating	Rx apdu(*) Any APDU not covered in 8.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
8.21	Connected Associated Operating	Rx roiv-*	Connected Associated Operating <no state transition>	Normal processing of messages. This is the normal operating state.	Tx (rors-*, or roer, or rorj)
8.26	Connected Associated Operating	Rx (rors-*, roer, or rorj)	Connected Associated Operating <no state transition>	Normal processing of messages. This is the normal operating state.	None[5]
9.2	Connected Disassociating	IND Transport Disconnect	Disconnected	None.	None
9.4	Connected Disassociating	IND Timeout	Connected Unassociated	No reply to rlrq.	Tx abrt(reason response-timeout)
9.6	Connected Disassociating	REQ AssocRel	Connected Disassociating	Already disassociating. Ignore.	None
9.7	Connected Disassociating	REQ AssocAbort	Connected Unassociated	Abort the graceful disassociation process.	Tx abrt reason defined by application

[4] An AssocRel should not be sent until all outstanding invoke-ids are retired.
[5] If an rors-* is received with an unknown invoke-id, then the application layer shall cause an Abort message to be sent to the manager by sending an "REQ abrt" to the state machine.

Table E.2—Agent state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event output
9.8	Connected Disassociating	Rx aarq	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
9.12	Connected Disassociating	Rx aare(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
9.16	Connected Disassociating	Rx rlrq	Connected Disassociating <no state transition>	Both sides releasing connection. Respond and wait for own rlre.	Tx rlre(normal)
9.17	Connected Disassociating	Rx rlre	Connected Unassociated	Release process completed, exit to unassociated.	None
9.18	Connected Disassociating	Rx abrt	Connected Unassociated	None.	None
9.19	Connected Disassociating	Rx ~~prst~~apdu(*) Any APDU not covered in 9.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
9.21	Connected Disassociating	Rx roiv-*	Connected Disassociating <no state transition>	The manager sent an invoke message as the agent sent an rlrq. The agent has transitioned out of the Operating state and therefore will not provide a response.	None
9.26	Connected Disassociating	Rx (rors-*, roer, or rorj)	Connected Unassociated	Example 1: Application layer has outstanding invoke-ids but has previously issued a ReleaseRequest anyway. Example 2: Unsolicited rors-*.	Tx abrt(reason undefined)

E.3 Manager state table

Change Table E.3 as shown:

Table E.3—Manager state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event generated
1.1	Disconnected	IND Transport connection	Connected Unassociated	”Shall” indicate to application layer.	None
2.2	Connected Unassociated	IND Transport disconnect	Disconnected	”Should” indicate to application layer.	None
2.6	Connected Unassociated	REQ AssocRel	Connected Unassociated <no state transition>	Should not happen. Ignore.	None
2.7	Connected Unassociated	REQ AssocAbort	Connected Unassociated <no state transition>	Can be used to synchronize state on both sides~~Should not happen. Ignore~~.	~~None~~Tx abrt
2.8	Connected Unassociated	Rx aarq (*)	Connected Associating	Associate request	None
~~2.9~~	~~Connected Unassociated~~	~~Rx aarq (acceptable and known configuration)~~	~~Connected associated operating~~	~~The device and configuration are known to the manager.~~	~~Tx aare(accepted)~~
~~2.10~~	~~Connected Unassociated~~	~~Rx aarq (acceptable with unknown configuration)~~	~~Connected associated configuring waiting~~	~~The manager determines that the connection is acceptable, but does not have valid configuration information for the agent. Timeout=TOconfig~~	~~Tx aare(accepted-unknown-config)~~
~~2.11~~	~~Connected Unassociated~~	~~Rx aarq (unacceptable configuration)~~	~~Connected unassociated~~	~~The manager determines that the connection is unacceptable.~~	~~Tx aare(reject-*)~~
2.12	Connected Unassociated	Rx aare(*)	Connected Unassociated <no state transition>	Should not happen.	Tx abrt(reason undefined)
2.16	Connected Unassociated	Rx rlrq	Connected Unassociated <no state transition>	Should not happen.	Tx abrt(reason undefined)
2.17	Connected Unassociated	Rx rlre	Connected Unassociated <no state transition>	Should not happen. Ignore.	None
2.18	Connected Unassociated	Rx abrt	Connected Unassociated <no state transition>	None~~Should not happen. Ignore~~.	None
2.19	Connected Unassociated	Rx apdu(*) Any APDU not covered in 2.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated <no state transition>	Should not happen.	Tx abrt(reason undefined)

Table E.3—Manager state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event generated
3.2	Connected Associating	IND Transport disconnect	Disconnected	None.	None
~~3.4~~	~~Connected Associating~~	~~IND Timeout~~	~~Connected Unassociated~~	~~Application did not respond before the timeout expired~~	~~Tx abrt(reason response-timeout)~~
3.6	Connected Associating	REQ AssocRel	Connected Unassociated	None.	Tx abrt(reason undefined)
3.7	Connected Associating	REQ AssocAbort	Connected Unassociated	None.	Tx abrt reason defined by application
3.8	Connected Associating	Rx aarq(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
3.9	Connected Associating	REQ acceptable and known configuration	Connected Associated Operating	The agent and configuration are known to the manager.	Tx aare(accepted)
3.10	Connected Associating	REQ acceptable with unknown configuration	Connected Associated Configuring Waiting	The manager the association request, but does not have valid configuration information for the agent. Timeout=TO_{config}	Tx aare(accepted-unknown-config)
3.11	Connected Associating~~Connected Unassociated~~	REQ unacceptable configuration	Connected Unassociated	The manager determines that the association is unacceptable.	Tx aare(reject-*)
3.12	Connected Associating	Rx aare(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
3.16	Connected Associating	Rx rlrq	Connected Unassociated	Should not happen. The manager has received a request to release the association, but it has not yet established an association.	Tx abrt(reason undefined)
3.17	Connected Associating	Rx rlre	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
3.18	Connected Associating	Rx abrt	Connected Unassociated	None.	None
3.19	Connected Associating	Rx apdu(*) Any APDU not covered in 3.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
6.2	Connected Associated Configuring Waiting	IND Transport disconnect	Disconnected	None.	None
6.4	Connected Associated Configuring Waiting	IND Timeout	Connected Unassociated	No reply. Config Report or rors-cmp-get for MDS not received	Tx abrt(reason configuration-timeou~~not~~) if no Config Report; otherwise, (reason response-timeout)

Table E.3—Manager state table

Signal ID	Initial state	Event/input stream	Next state	Semantic behaviors/notes	Tx stream (output)/event generated
6.6	Connected Associated Configuring Waiting	REQ AssocRel	Connected Disassociating	None. Timeout=$TO_{release}$ ~~reset~~.	Tx rlrq(normal)
6.7	Connected Associated Configuring Waiting	REQ AssocAbort	Connected Unassociated	None.	Tx abrt reason defined by application
6.8	Connected Associated Configuring Waiting	Rx aarq(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
6.12	Connected Associated Configuring Waiting	Rx aare(*)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
6.16	Connected Associated Configuring Waiting	Rx rlrq	Connected Unassociated	The manager has received a request to release the association.	Tx rlre(normal)
6.17	Connected Associated Configuring Waiting	Rx rlre	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
6.18	Connected Associated Configuring Waiting	Rx abrt	Connected Unassociated	None.	None
6.19	Connected Associated Configuring Waiting	Rx apdu(*) Any APDU not covered in 6.* (corrupt, unknown, unexpected, etc.)	Connected Unassociated	Should not happen.	Tx abrt(reason undefined)
~~6.23~~	~~Connected Associated Configuring Waiting~~	~~Rx roiv-confirmed-event-report containing MDS info~~	~~Connected Associated Configuring Waiting <no state transition>~~	~~The manager receives the MDS information and sends an confirmed event report response.~~	~~Tx (rors-*, or roer, or rorj)~~
6.24	Connected Associated Configuring Waiting	Rx roiv-confirmed-event-report containing a configuration	Connected Associated Configuring Checking Config	Event report containing configuration from agent. Timeout=$TO_{config.}$	None