INTERNATIONAL STANDARD

ISO/IEEE 11073-10417

Third edition
2017-04

Health informatics — Personal health device communication —

Part 10417: **Device specialization — Glucose meter**

Informatique de santé — Communication entre dispositifs médicaux sur le site des soins —

Partie 10417: Spécialisation des dispositifs — Glucomètre

Reference number
ISO/IEEE 11073-10417:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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

stds.ipr@ieee.org
www.ieee.org

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.

ISO/IEEE 11073-10417 was prepared by the 11073 Committee of the Engineering in Medicine and Biology Society of the IEEE (as IEEE Std 11073-10417-2015). 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.

IEEE Std 11073-10417™-2015

(Revision of
IEEE Std 11073-10417-2011)

Health informatics—Personal health device communication

Part 10417: Device Specialization—Glucose Meter

Sponsor

IEEE 11073™ Standards Committee
of the
IEEE Engineering in Medicine and Biology Society

Approved 11 June 2015

IEEE-SA Standards Board

Abstract: Within the context of the ISO/IEEE 11073 family of standards for device communication, a normative definition of communication between personal telehealth glucose meter devices and compute engines (e.g., cell phones, personal computers, personal health appliances, and set top boxes) is established by this standard in a manner that enables plug-and-play interoperability. Appropriate portions of existing standards are leveraged, including ISO/IEEE 11073 terminology, information models, application profile standards, and transport standards. The use of specific term codes, formats, and behaviors in telehealth environments restricting optionality in base frameworks in favor of interoperability are specified. A common core of communication functionality for personal telehealth glucose meters is defined in this standard.

Keywords: glucose meter, IEEE 11073-10417™, 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 9746-3 STD20244
Print: ISBN 978-0-7381-9747-0 STDPD20244

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, 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.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. 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.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own 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.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of 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 of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

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

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document 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

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted 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 these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents 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.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, 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 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-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE–SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. 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 IEEE standard was completed, the Personal Health Devices Working Group had the following membership:

Daidi Zhong, *Chair*
Michael J. Kirwan, *Chair*
Raymond A. Strickland, *Vice-Chair*
Craig Carlson, *Vice-Chair*

Karsten Aalders
Charles R. Abbruscato
Nabil Abujbara
Maher Abuzaid
James Agnew
Haidar Ahmad
Manfred Aigner
Jorge Alberola
Murtaza Ali
Rolf Ambuehl
David Aparisi
Lawrence Arne
Diego B. Arquillo
Serafin Arroyo
Muhammad Asim
Merat Bagha
Doug Baird
David Baker
Anindya Bakshi
Ananth Balasubramanian
Sunlee Bang
M. Jonathan Barkley
Gilberto Barrón
David Bean
John Bell
Rudy Belliardi
Daniel Bernstein
George A. Bertos
Chris Biernacki
Ola Björsne
Thomas Blackadar
Marc Blanchet
Thomas Bluethner
Douglas P. Bogia
Xavier Boniface
Shannon Boucousis
Julius Broma
Lyle G. Bullock, Jr.
Bernard Burg
Chris Burns
Anthony Butt
Jeremy Byford-Rew
Satya Calloji
Carole C. Carey
Santiago Carot-Nemesio
Randy W. Carroll
Simon Carter
Seungchul Chae
Rahul Chauhan
James Cheng
Peggy Chien
David Chiu
Chia-Chin Chong
Saeed A. Choudhary
Jinhan Chung
Malcolm Clarke
John A. Cogan
John T. Collins
Cory Condek
Todd H. Cooper
David Cornejo
Douglas Coup
Nigel Cox
Hans Crommenacker
Tomio Crosley
David Culp
Allen Curtis
Eyal Dassau
David Davenport
Russell Davis
Sushil K. Deka
Ciro de la Vega
Pedro de-las-Heras-Quiros
Jim DelloStritto
Matthew d'Entremont
Lane Desborough
Kent Dicks
Hyoungho Do
Xiaolian Duan
Brian Dubreuil
Sourav Dutta
Jakob Ehrensvard
Fredrik Einberg
Roger M. Ellingson
Michihiro Enokida
Javier Escayola Calvo
Mark Estes
Leonardo Estevez
Roger Feeley
Bosco T. Fernandes
Christoph Fischer
Morten Flintrup
Joseph W. Forler
Russell Foster
Eric Freudenthal
Matthias Frohner
Ndifor Cyril Fru
Ken Fuchs
Jing Gao
Xuemei Gao
Marcus Garbe
John Garguilo
Rick Geimer
Igor Gejdos
Ferenc Gerbovics
Nicolae Goga
Julian Goldman
Raul Gonzalez Gomez
Chris Gough
Channa Gowda
Charles M. Gropper
Amit Gupta
Jeff Guttmacher
Rasmus Haahr
Christian Habermann
Michael Hagerty
Jerry Hahn
Robert Hall
Nathaniel Hamming
Rickey L. Hampton
Sten Hanke
Jordan Hartmann
Kai Hassing
Marc Daniel Haunschild
Wolfgang Heck
Nathaniel Heintzman
Charles Henderson
Jun-Ho Her
Takashi Hibino
Timothy L. Hirou
Allen Hobbs
Alex Holland
Arto Holopainen
Kris Holtzclaw
Robert Hoy
Frank Hsu
Anne Huang
Sen-Der Huang
Zhiqiang Huang
Ron Huby
David Hughes

Robert D. Hughes
Jiyoung Huh
Hugh Hunter
Hitoshi Ikeda
Yutaka Ikeda
Philip O. Isaacson
Atsushi Ito
Michael Jaffe
Praduman Jain
Wei Jin
Danny Jochelson
Chris Johnson
Phaneeth Junga
Akiyoshi Kabe
Steve Kahle
Tomio Kamioka
Kei Kariya
Andy Kaschl
Junzo Kashihara
Kohichi Kashiwagi
Ralph Kent
Laurie M. Kermes
Ikuo Keshi
Junhyung Kim
Minho Kim
Min-Joon Kim
Taekon Kim
Tetsuya Kimura
Alfred Kloos
Jeongmee Koh
Jean-Marc Koller
John Koon
Patty Krantz
Raymond Krasinski
Alexander Kraus
Ramesh Krishna
Geoffrey Kruse
Falko Kuester
Rafael Lajara
Pierre Landau
Jaechul Lee
JongMuk Lee
Kyong Ho Lee
Rami Lee
Sungkee Lee
Woojae Lee
Yonghee Lee
Joe Lenart
Kathryn A. Lesh
Qiong Li
Ying Li
Patrick Lichter
Jisoon Lim
Joon-Ho Lim
John Lin
Wei-Jung Lo
Charles Lowe
Don Ludolph
Christian Luszick
Bob MacWilliams
Srikkanth Madhurbootheswaran
Miriam L. Makhlouf

Romain Marmot
Sandra Martinez
Miguel Martínez de Espronceda Cámara
Peter Mayhew
Jim McCain
László Meleg
Alexander Mense
Ethan Metsger
Jinsei Miyazaki
Erik Moll
Darr Moore
Carsten Mueglitz
Piotr Murawski
Soundharya Nagasubramanian
Jae-Wook Nah
Alex Neefus
Trong-Nghia Nguyen-Dobinsky
Michael E. Nidd
Tetsu Nishimura
Jim Niswander
Hiroaki Niwamoto
Thomas Norgall
Anand Noubade
Yoshiteru Nozoe
Abraham Ofek
Brett Olive
Begonya Otal
Charles Palmer
Bud Panjwani
Carl Pantiskas
Harry P. Pappas
Mikey Paradis
Hanna Park
Jong-Tae Park
Myungeun Park
Soojun Park
Phillip E. Pash
TongBi Pei
Lucian Pestritu
Soren Petersen
James Petisce
Peter Piction
Michael Pliskin
Jeff Price
Harald Prinzhorn
John Quinlan
Arif Rahman
Tanzilur Rahman
Steve Ray
Phillip Raymond
Tim Reilly
Barry Reinhold
Brian Reinhold
Melvin I. Reynolds
John G. Rhoads
Jeffrey S. Robbins
Moskowitz Robert
Timothy Robertson
David Rosales
Bill Saltzstein
Benedikt Salzbrunn

Giovanna Sannino
Jose A. Santos-Cadenas
Stefan Sauermann
John Sawyer
Guillaume Schatz
Alois Schloegl
Paul S. Schluter
Lars Schmitt
Mark G. Schnell
Richard A. Schrenker
Antonio Scorpiniti
Kwang Seok Seo
Riccardo Serafin
Sid Shaw
Frank Shen
Liqun Shen
Bozhi Shi
Min Shih
Mazen Shihabi
Redmond Shouldice
Sternly K. Simon
Marjorie Skubic
Robert Smith
Ivan Soh
Motoki Sone
Emily Sopensky
Rajagopalan Srinivasan
Andreas Staubert
Nicholas Steblay
Beth Stephen
Lars Steubesand
John (Ivo) Stivoric
Raymond A. Strickland
Chandrasekaran Subramaniam
Hermanni Suominen
Lee Surprenant
Ravi Swami
Ray Sweidan
Jin Tan
Haruyuyki Tatsumi
John W. Thomas
Jonas Tirén
Alexandra Todiruta
James Tomcik
Janet Traub
Jesús Daniel Trigo
Gary Tschautscher
Masato Tsuchid
Ken Tubman
Yoshihiro Uchida
Sunil Unadkat
Fabio Urbani
Philipp Urbauer
Laura Vanzago
Alpo Värri
Dalimar Velez
Naveen Verma
Rudi Voon
Isobel Walker
David Wang
Jerry P. Wang
Yao Wang

Yi Wang
Steve Warren
Fujio Watanabe
Toru Watsuji
Mike Weng
Kathleen Wible
Paul Williamson
Jan Wittenber
Jia-Rong Wu
Will Wykeham
Ariton Xhafa
Yaxi Yan
Ricky Yang
Melanie S. Yeung
Done-Sik Yoo
Jianchao Zeng
Jason Zhang
Zhiqiang Zhang
Thomas Zhao
Miha Zoubek
Szymon Zyskoter

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

John Ballingall
Giberto Barrón
Lyle G. Bullock, Jr.
Keith Chow
Joseph El Youssef
Randall Groves
Kai Hassing
Werner Hoelzl
Noriyuki Ikeuchi
Atsushi Ito
Piotr Karocki
Patrick Keith-Hynes
Patrick Kinney
Robert Kircher
Michael J. Kirwan
Nick S. A. Nikjoo
Henry Pinto
Melvin I. Reynolds
Bartien Sayogo
Lars Schmitt
Raymond A. Strickland
Walter Struppler
Jan Wittenber
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 11 June 2015, it had the following membership:

John Kulick, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Jean-Philippe Faure
J. Travis Griffith
Gary Hoffman
Michael Janezic
Joseph L. Koepfinger*
David J. Law
Hung Ling
Andrew Myles
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Annette D. Reilly
Stephen J. Shellhammer
Adrian P. Stephens
Yatin Trivedi
Phillip Winston
Don Wright
Yu Yuan
Daidi Zhong

*Member Emeritus

Julie Alessi, *IEEE-SA Content Production and Management*

Kathryn Bennett, *IEEE-SA Operational Program Management*

Introduction

This introduction is not part of IEEE Std 11073-10417™-2015, Health informatics—Personal health device communication—Part 10417: Device Specialization—Glucose Meter.

ISO/IEEE 11073 standards enable communication between medical devices and external computer systems. This document uses the optimized framework created in IEEE Std 11073-20601-2015[a] and describes a specific, interoperable communication approach for glucose meters. These standards align with and draw on the existing clinically focused standards to provide support for communication of data from clinical or personal health devices.

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

Contents

Health informatics—Personal health device communication

Part 10417: Device Specialization—Glucose Meter

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 Scope

Within the context of the ISO/IEEE 11073 family of standards for device communication, this standard establishes a normative definition of communication between personal telehealth glucose meter devices and compute engines (e.g., cell phones, personal computers, personal health appliances, and set top boxes) in a manner that enables plug-and-play interoperability. It leverages appropriate portions of existing standards, including ISO/IEEE 11073 terminology, information models, application profile standards, and transport standards. It specifies the use of specific term codes, formats, and behaviors in telehealth environments restricting optionality in base frameworks in favor of interoperability. This standard defines a common core of communication functionality for personal telehealth glucose meters.

1.2 Purpose

This standard addresses the need for an openly defined, independent standard that support information exchange to and from personal health devices and compute engines (e.g., cell phones, personal computers, personal health appliances, and set top boxes). Interoperability is key to growing the potential market for these devices and enabling people to be better informed participants in the management of their heath.

1.3 Context

See IEEE Std 11073-20601™-2014[1] for an overview of the environment within which this standard is written.

This standard defines the device specialization for the insulin pump, being a specific agent type, and provides a description of the device concepts, its capabilities, and its implementation according to this standard.

This standard is based on IEEE Std 11073-20601™-2014[2] which in turn draws information from both ISO/IEEE 11073-10201:2004 [B3][3] and ISO/IEEE 11073-20101:2004 [B4]. The medical device encoding rules (MDERs) used within this standard are fully described in Annex F of IEEE Std 11073-20601-2014.

This standard reproduces relevant portions of the nomenclature found in ISO/IEEE 11073-10101:2004 [B2] and adds new nomenclature codes for the purposes of this standard. Among this standard and IEEE Std 11073-20601-2014, all required nomenclature codes for implementation are documented.

NOTE—In this standard, ISO/IEEE 11073-104zz is used to refer to the collection of device specialization standards that utilize IEEE Std 11073-20601-2014, where zz can be any number from 01 to 99, inclusive.[4]

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so that each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 11073-20601-2014, Health informatics—Personal health device communication—Application Profile—Optimized Exchange Protocol.[5, 6]

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[7]

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

class: In object-oriented modeling, it describes the attributes, methods, and events that objects instantiated from the class utilize.

[1] Information on references can be found in Clause 2.
[2] ISO/IEEE publications are available from the ISO Central Secretariat, 1, ch. de la Voie-Creuse, Case postale 56, CH-1211, Geneva 20, Switzerland (http://www.iso.ch/). ISO/IEEE publications are also available in the United States from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854-4141, USA (http://standards.ieee.org/).
[3] The numbers in brackets correspond to those of the bibliography in Annex A.
[4] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.
[5] The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.
[6] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854-4141, USA (http://standards.ieee.org/).
[7] *IEEE Standards Dictionary Online subscription* is available at: http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

compute engine: *See:* **manager.**

device: A term used to refer to a physical apparatus implementing either an agent or a manager role.

glucagon: Naturally occurring hormone released by the pancreas when glucose levels are low.

glucose: Commonly referred to as "sugar," it is the major source of energy used by the body cells.

glucose meter: A medical device for determining the approximate concentration of glucose in the blood.

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

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

obj-handle: *See:* **handle**.

object: In object-oriented modeling, a particular instantiation of a class. The instantiation realizes attributes, methods, and events from the class.

occlusion: Total obstruction of the infusion set that prevents administering insulin.

personal health device: A device used in personal health applications.

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

3.2 Acronyms and abbreviations

APDU	application protocol data unit
ASN.1	Abstract Syntax Notation One
AST	alternative site testing
DIM	domain information model
EUI-64	extended unique identifier (64 bits)
HbA1c	hemoglobin bound to glucose (the A1c form)
HCP	health care provider
ICS	implementation conformance statements
ISF	insulin sensitivity factor
MDC	medical device communication
MDER	medical device encoding rules
MDS	medical device system
MOC	managed object class
OID	object identifier
PDU	protocol data unit
PHD	personal health device
VMO	virtual medical object
VMS	virtual medical system

4. Introduction to ISO/IEEE 11073 personal health devices

4.1 General

This standard and the remainder of the series of ISO/IEEE 11073 personal health device (PHD) standards fit in the larger context of the ISO/IEEE 11073 series of standards. The full suite of standards enables agents to interconnect and interoperate with managers and with computerized health-care information systems. See IEEE Std 11073-20601-2014 for a description of the guiding principles for this series of ISO/IEEE 11073 Personal Health Device standards.

IEEE Std 11073-20601-2014 supports the modeling and implementation of an extensive set of personal health devices. This standard defines aspects of the glucose meter device. It describes all aspects necessary to implement the application layer services and data exchange protocol between an ISO/IEEE 11073 PHD glucose meter agent and a manager. This standard defines a subset of the objects and functionality contained in IEEE Std 11073-20601-2014 and extends and adds definitions where appropriate. All new definitions are given in Annex B in Abstract Syntax Notation One (ASN.1). Nomenclature codes referenced in this standard, which are not defined in IEEE Std 11073-20601-2014, are normatively defined in Annex C.

4.2 Introduction to IEEE 11073-20601 modeling constructs

4.2.1 General

The ISO/IEEE 11073 series of standards, and in particular IEEE Std 11073-20601-2014, is based on an object-oriented systems management paradigm. The overall system model is divided into three principal components: the domain information model (DIM), the service model, and the communication model. See IEEE Std 11073-20601-2014 for a detailed description of the modeling constructs.

4.2.2 Domain information model

The DIM is a hierarchical model that describes an agent as a set of objects. These objects and their attributes represent the elements that control behavior and report on the status of the agent and data that an agent can communicate to a manager. Communication between the agent and the manager is defined by the application protocol in the IEEE Std 11073-20601-2014.

4.2.3 Service model

The service model defines the conceptual mechanisms for the data exchange services. Such services are mapped to messages that are exchanged between the agent and the manager. Protocol messages within the ISO/IEEE 11073 series of standards are defined in ASN.1. The messages defined in IEEE Std 11073-20601-2014 can coexist with messages defined in other standard application profiles defined in the ISO/IEEE 11073 series of standards.

4.2.4 Communication model

In general, the communication model supports the topology of one or more agents communicating over logical point-to-point connections to a single manager. For each logical point-to-point connection, the dynamic system behavior is defined by a connection state machine as specified in IEEE Std 11073-20601-2014.

4.2.5 Implementing the models

An agent implementing this standard shall implement all mandatory elements of the information, service, and communication models as well as all conditional elements where the condition is met. The agent should implement the recommended elements, and it may implement any combination of the optional elements. A manager implementing this standard shall utilize at least one of the mandatory, conditional, recommended, or optional elements. In this context, *utilize* means to use the element as part of the primary function of the manager device. For example, a manager whose primary function is to display data would need to display a piece of data in the element in order to utilize it.

5. Glucose meter device concepts and modalities

5.1 General

This clause presents the general concepts of glucose meters. In the context of personal health devices in this family of standards, a glucose meter is a device that measures the concentration of glucose in the blood. Glucose, or the concentration of blood sugar in the blood, is the primary source of energy for the body's cells. The glucose level is tightly regulated in the human body and is normally maintained between approximately 70 mg/dL and 150 mg/dL (4 mmol/L and 8 mmol/L). The total amount of glucose in the circulating blood is, therefore, approximately 3.5 g to 7.5 g (assuming an ordinary adult blood volume of 5 L). Glucose levels rise after meals and are usually lowest in the morning, before the first meal of the day.

In a healthy adult male of 75 kg with a blood volume of 5 L, a blood glucose level of 100 mg/dL (5.5 mmol/L) corresponds to a total of approximately 5 g (1/5 oz and equivalent to a commercial sugar packet) of glucose in the blood and approximately 45 g (1.5 oz) in the total body fluid (which includes blood and interstitial fluid) and on average will be approximately 60% of the total body weight in men.

The failure to maintain blood glucose in the normal range leads to conditions of persistently high (hyperglycemia) or low (hypoglycemia) blood sugar. Diabetes mellitus, which is characterized by persistent hyperglycemia from several causes, is the most prominent disease related to the failure to regulate blood sugar. Fructose and galactose are also sugars found in the blood; however, only glucose levels are regulated via insulin and glucagon.

Countries that use the metric system generally use mmol/L. The United States uses mg/dL. To convert blood glucose readings, utilize the following conversions:

Divide the mg/dL by 18.02 to get mmol/L (or multiply by 0.0555)

Multiply the mmol/L by 18.02 to get mg/dL (or divide by 0.0555)

The glucose meters considered in this specialization are typically handheld instruments that require a sample of blood or body fluid to perform the glucose measurement. The glucose concentration measured by various techniques can be classified into different types defined by three elements: sample type, sample source, and concentration reference method. Table 1 shows all the glucose concentration types defined in this standard.

Table 1—Glucose concentration types

Sample type	Sample source	Reference method
Blood	Capillary	Whole blood
		Plasma
	Venous	Whole blood
		Plasma
	Arterial	Whole blood
		Plasma
	Undetermined	Whole blood
		Plasma
Interstitial fluid	N/A	N/A
Control solution	N/A	N/A
NOTE—The blood glucose concentration may be indirectly derived from an interstitial fluid (ISF) sample, which is a common technique used in continuous glucose monitoring. A control solution is normally used for glucose meter quality control.		

In addition to glucose measurement, glucose meters generally provide a means for the user to associate information on meals, exercise, and medications with a glucose measurement. Advanced devices may also allow users to customize device settings and to provide additional information related to their diabetes treatment and disease management.

Glucose meters are usually small, portable devices that are carried around by the user so that blood glucose measurements may be taken as needed, whether or not the user is in a network environment where communications with a manager can be established. Hence, measurement data are typically logged within the meter and can be retrieved at a later time, either in whole or in part, when the meter is connected to a manager network. There are two typical use cases for the transfer of measurement data to a manager:

- By the user in a non-health-care environment. A user may connect the meter to a personal manager device such as a personal computer or a portable communications device such as a cell phone. In this case, the user would download the most recent logged measurement data from the meter for near-term trend analysis or for transfer of data to an external electronic health network service. The transfer of data would happen immediately on connection with the manager application and require no initiation by the user. This is a temporary storage model of the meter.
- By a health care professional in an office or clinic environment. The health care professional would download all logged measurement data from the meter for longer-term analysis, such as the data logged since the patient’s last appointment. The transfer of data would be controlled entirely by the manager application and would not necessarily happen automatically when a connection is established. This is a long-term storage model of the meter.

This standard supports configurations that allow for support of both temporary and long-term storage models.

6. Glucose meter domain information model

6.1 Overview

This clause describes the domain information model of the glucose meter.

6.2 Class extensions

In this standard, no class extensions are defined with respect to IEEE Std 11073-20601-2014.

6.3 Object instance diagram

The object instance diagram of the glucose meter domain information model, which is defined for the purposes of this standard, is shown in Figure 1.

The objects of the DIM, as shown in Figure 1, are described in 6.5 to 6.11. See 6.5 through 6.11 for descriptions of the different glucose meter objects (e.g., the glucose meter medical device system (MDS) object, the glucose numeric object, and the enumeration object). See 6.12 for rules for extending the glucose meter information model beyond elements as described in this standard. Each clause that describes an object of the glucose meter contains the following information:

- The nomenclature code used to identify the class of the object. One example where this code is used is the configuration event, where the object class is reported for each object. This allows the manager to determine whether the class of the object being specified is a numeric, real-time sample array, enumeration, scanner, or PM-store class.
- The attributes of the object. Each object has attributes that represent and convey information on the physical device and its data sources. Each object has a Handle attribute that identifies the object instance within an agent. Attribute values are accessed and modified using methods such as GET and SET. Attribute types are defined using an ASN.1. The ASN.1 definitions for new attribute types specific to this standard are in Annex B, and the ASN.1 definitions for existing attribute types referenced in this standard are in IEEE Std 11073-20601-2014.
- The methods available on the object.
- The potential events generated by the object. The data are sent to the manager using events.
- The available services such as getting or setting attributes.

The attributes for each class are defined in tables that specify the name of the attribute, its value, and its qualifier. The qualifiers mean:

- M: Attribute is mandatory,
- C: Attribute is conditional and depends on the condition stated in the Remark or Value column (if IEEE Std 11073-20601-2014 is referenced, then it contains the conditions),
- R: Attribute is recommended,
- NR: Attribute is not recommended, and
- O: Attribute is optional.

Mandatory attributes shall be implemented by an agent. Conditional attributes shall be implemented if the condition applies and may be implemented otherwise. Recommended attributes should be implemented by the agent. Not recommended attributes should not be implemented by the agent. Optional attributes may be implemented on an agent.

The attributes can be either static, dynamic, or observational as specified in IEEE Std 11073-20601-2014.

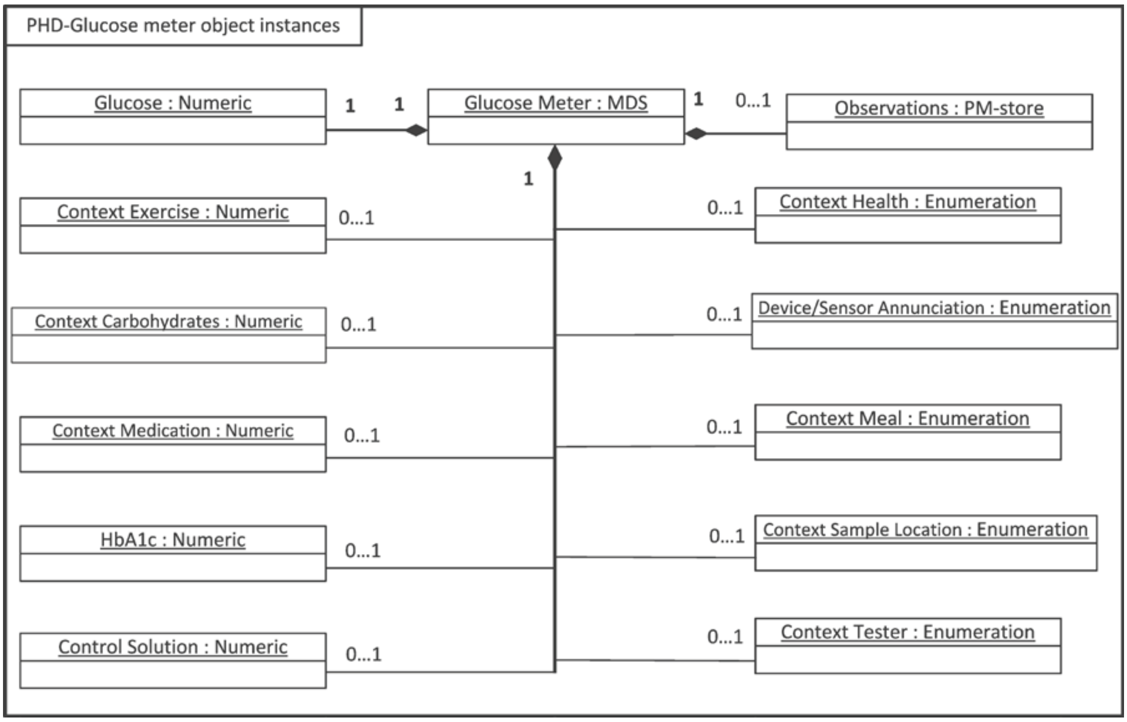
PHD-Glucose meter object instances
Glucose : Numeric
1
1
Glucose Meter : MDS
1
0...1
Observations : PM-store
1
Context Exercise : Numeric
0...1
0...1
Context Health : Enumeration
Context Carbohydrates : Numeric
0...1
0...1
Device/Sensor Annunciation : Enumeration
Context Medication : Numeric
0...1
0...1
Context Meal : Enumeration
HbA1c : Numeric
0...1
0...1
Context Sample Location : Enumeration
Control Solution : Numeric
0...1
0...1
Context Tester : Enumeration

Figure 1—Glucose meter—domain information model

6.4 Types of configuration

6.4.1 General

As specified in IEEE Std 11073-20601-2014, there are two styles of configuration available. Subclauses 6.4.2 and 6.4.3 briefly introduce standard and extended configurations.

6.4.2 Standard configuration

Standard configurations are defined in the ISO/IEEE 11073-104zz specializations (such as this standard) and are assigned a well-known identifier (Dev-Configuration-Id). The usage of a standard configuration is negotiated at association time between the agent and the manager. If the manager acknowledges that it understands and wants to operate using the configuration, then the agent can begin sending measurements immediately. If the manager does not understand the configuration, the agent provides the configuration prior to transmitting measurement information.

The standard configuration 1700 (0x06A4) is deprecated with the release of this version of the 10417 standard.

Standard configuration 1701 (0x06A5) contains a glucose numeric object and a control solution numeric object as defined in 6.6.7.

A new standard configuration 1702 (0x06A6) is introduced in this version of standard 10417. The new standard configuration contains a glucose numeric object, a control solution object, and a meal context object, and recommends the use of Base-Offset-Time attributes for time reporting.

As defined in 8.3.1 of IEEE Std 11073-20601-2014, if the extended configuration of the specialization implemented by an agent is one that makes use of PM stores and only reports measurements using segment data events and if the standard configuration does not support PM stores, an agent is not required to support the corresponding standard configuration. As defined in 7.4.3.6.1 of 20601, Managers released based on this specification shall support both standard configurations 1701 (0x06A5) and 1702 (0x06A6).

6.4.3 Extended configuration

In extended configurations, the agent's configuration is not predefined in a standard. The agent determines which objects, attributes, and values will be used in a configuration and assigns a configuration identifier. When the agent associates with a manager, it negotiates an acceptable configuration. Typically, the manager does not recognize the agent's configuration on the first connection, so the manager responds that the agent needs to send the configuration information as a configuration event report. If, however, the manager already understands the configuration, either because it was preloaded in some way or the agent had previously associated with the manager, then the manager responds that the configuration is known and no further configuration information needs to be sent.

6.5 Medical device system object

6.5.1 MDS object attributes

Table 2 summarizes the attributes of the glucose meter MDS object. The nomenclature code to identify the MDS object class is MDC_MOC_VMS_MDS_SIMP.

Table 2—MDS object attributes

Attribute name	Value	Qualifier
Handle	0	M
System-Type	Attribute not present. See IEEE Std 11073-20601-2014.	C
System-Type-Spec-List	{MDC_DEV_SPEC_PROFILE_GLUCOSE, 2}	M
System-Model	{"Manufacturer", "Model"}	M
System-Id	Extended unique identifier (64 bits) (EUI-64)	M
Dev-Configuration-Id	Extended configs: 0x4000–0x7FFF Standard config: 0x06A6 (1702) Standard config: 0x06A5 (1701)	M
Attribute-Value-Map	See IEEE Std 11073-20601-2014.	C
Production-Specification	See IEEE Std 11073-20601-2014.	O
Mds-Time-Info	See IEEE Std 11073-20601-2014.	C
Date-and-Time	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time	See IEEE Std 11073-20601-2014.	R
Relative-Time	See IEEE Std 11073-20601-2014.	C
HiRes-Relative-Time	See IEEE Std 11073-20601-2014.	C
Date-and-Time-Adjustment	See IEEE Std 11073-20601-2014.	C
Power-Status	*onBattery or onMains*	O
Battery-Level	See IEEE Std 11073-20601-2014.	O
Remaining-Battery-Time	See IEEE Std 11073-20601-2014.	O
Reg-Cert-Data-List	See IEEE Std 11073-20601-2014.	O
Confirm-Timeout	See IEEE Std 11073-20601-2014.	O
Transport-Timeout	See IEEE Std 11073-20601 2014.	O
NOTE—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.		

In the response to a Get MDS object command, only implemented attributes and their corresponding values are returned.

See IEEE Std 11073-20601-2014 for descriptive explanations of the individual attributes as well as for information on attribute ID and attribute type.

The Dev-Configuration-Id attribute holds a locally unique 16-bit identifier that identifies the device configuration instance. The Device Configuration Identifiers for standard configurations are listed in Table 2. For a glucose meter agent with extended configuration, this identifier is chosen in the range of extended-config-start to extended-config-end (see IEEE Std 11073-20601-2014) as shown in Table 2.

The agent sends the Dev-Configuration-Id during the Associating state (see 8.3) to identify its configuration for the duration of the association. If the manager already holds the configuration information relating to the Dev-Configuration-Id, it recognizes the Dev-Configuration-Id. Then the Configuring state (8.4) is skipped, and the agent and manager enter the Operating state. If the manager does not recognize the Dev-Configuration-Id, the agent and manager enter the Configuring state.

If an agent implements multiple IEEE 11073-104zz specializations, System-Type-Spec-List is a list of type/version pairs, each referencing the respective device specialization and version of that specialization.

6.5.2 MDS object methods

Table 3 defines the methods (actions) of the glucose meter agent's MDS object. These methods are invoked using the Action service. In Table 3, the Subservice type name column defines the name of the method; the Mode column defines whether the method is invoked as an unconfirmed action (i.e., roiv-cmip-action from IEEE Std 11073-20601-2014) or a confirmed action (i.e., roiv-cmip-confirmed-action); the Subservice type (action-type) column defines the nomenclature code to use in the action-type field of an action request and response (see IEEE Std 11073-20601-2014); the Parameters (action-info-args) column defines the associated ASN.1 data structure (see IEEE Std 11073-20601-2014 for ASN.1 definitions) to use in the action message for the action-info-args field of the request; and the Results (action-info-args) column defines the structure to use in the action-info-args of the response.

Table 3—MDS object methods

Service	Subservice type name	Mode	Subservice type (action-type)	Parameters (action-info-args)	Results (action-info-args)
ACTION	Set-Time	Confirmed	MDC_ACT_SET_TIME	SetTimeInvoke	—
ACTION	Set-Base-Offset-Time	Confirmed	MDC_ACT_SET_BO_TIME	SetBOTimeInvoke	—-

- **Set-Time**
  This method allows the manager to set a real-time clock in the agent with the absolute time. The agent indicates whether the Set-Time command is valid using the mds-time-capab-set-clock bit in the Mds-Time-Info attribute (see IEEE Std 11073-20601-2014).

- **Set-Base-Offset-Time**
  This method allows the manager to set a real-time clock in the agent with the base time and offset. The agent indicates whether the Set-Base-Offset-Time command is valid using the mds-time-capab-set-clock bit in the Mds-Time-Info attribute (see IEEE Std 11073-20601-2014).

  If the agent supports the Base-Offset-Time-Stamp attribute, this method shall be implemented.

Agents following only this device specialization and no others shall send event reports using agent-initiated measurement data transmission. Agents following this device specialization as well as others shall send event reports in the appropriate fashion. During the association procedure (see 8.3), data-req-mode-capab shall be set to the appropriate value for the event report style. As a result, the manager shall assume the glucose meter agent does not support any of the MDS-Data-Request features (see IEEE Std 11073-20601-2014 for additional information). Thus, implementation of the MDS-Data-Request method/action is not required in this standard and is not shown in Table 3.

6.5.3 MDS object events

Table 4 defines the events that can be sent by the glucose meter MDS object.

Table 4—Glucose meter MDS object events

Service	Subservice type name	Mode	Subservice type (event-type)	Parameters (event-info)	Results (event-reply-info)
EVENT REPORT	MDS-Configuration-Event	Confirmed	MDC_NOTI_CONFIG	ConfigReport	ConfigReport Rsp
	MDS-Dynamic-Data-Update-Fixed	Confirmed	MDC_NOTI_SCAN_ REPORT_FIXED	ScanReportInfoFixed	—
	MDS-Dynamic-Data-Update-Var	Confirmed	MDC_NOTI_SCAN_ REPORT_VAR	ScanReportInfoVar	—
	MDS-Dynamic-Data-Update-MP-Fixed	Confirmed	MDC_NOTI_SCAN_ REPORT_MP_FIXED	ScanReportInfoMP Fixed	—
	MDS-Dynamic-Data-Update-MP-Var	Confirmed	MDC_NOTI_SCAN_ REPORT_MP_VAR	ScanReportInfoMP Var	—

— **MDS-Configuration-Event**
This event is sent by the glucose meter agent during the configuring procedure if the manager does not already know the glucose meter agent's configuration from past associations or because the manager has not been implemented to recognize the configuration according to the glucose meter device specialization. The event provides static information about the supported measurement capabilities of the glucose meter agent.

— **MDS-Dynamic-Data-Update-Var**
This event provides dynamic measurement data from the glucose meter agent for the glucose numeric object. These data are reported using a generic attribute list variable format. The event is sent as an unsolicited message by the agent (i.e., an agent-initiated measurement data transmission). See 8.5.3 for more information on unsolicited event reporting.

— **MDS-Dynamic-Data-Update-Fixed**
This event provides dynamic measurement data from the glucose meter agent for the glucose numeric object. These data are reported in the fixed format defined by the Attribute-Value-Map attribute of the object(s). The event is sent as an unsolicited message by the agent (i.e., an agent-initiated measurement data transmission). See 8.5.3 for more information on unsolicited event reporting.

- **MDS-Dynamic-Data-Update-MP-Var**
  This is the same as MDS-Dynamic-Data-Update-Var but allows inclusion of data from multiple people.
- **MDS-Dynamic-Data-Update-MP-Fixed**
  This is the same as MDS-Dynamic-Data-Update-Fixed but allows inclusion of data from multiple people.

NOTE—IEEE Std 11073-20601-2014 requires that managers support all of the MDS object events listed Table 4.

6.5.4 Other MDS services

6.5.4.1 GET service

A glucose meter agent shall support the GET service, which is provided by the MDS object to retrieve the values of all implemented MDS object attributes. The GET service can be invoked as soon as the glucose meter agent receives the Association Response and moves to the Associated state, including the Operating and Configuring substates.

The manager may request the MDS object attributes of the glucose meter agent; in which case, the manager shall send the "Remote Operation Invoke | Get" message (see roiv-cmip-get in IEEE Std 11073-20601-2014) with the reserved MDS handle value of 0. The glucose meter agent shall report its MDS object attributes to the manager using the "Remote Operation Response | Get" message (see rors-cmip-get in IEEE Std 11073-20601-2014). See Table 5 for a summary of the GET service including some message fields.

Table 5—Glucose meter MDS object GET service

Service	Remark	Field	Value	Qualifier
roiv-cmip-get	Manager GET service request of agents MDS attributes.	obj-handle	0	M
		attribute-id-list	Dynamic	O
rors-cmip-get	Agent GET service response to GET service request by manager.	obj-handle	0	M
		attribute-list	Dynamic	M

See 8.5.2 for details on the procedure for getting the MDS object attributes.

6.5.4.2 SET service

The glucose meter specialization does not require an implementation to support the MDS object SET service.

6.6 Numeric objects

6.6.1 General

The glucose meter DIM (see Figure 1) contains numeric objects that represent aspects of blood glucose and associated medical context. The nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. Table 6 shows attributes that are common to all the numeric types of the glucose meter agent.

Table 6—Common glucose numeric object attributes

Attribute name	Extended configuration	
	Value	Qualifier
Handle	See IEEE Std 11073-20601-2014.	M
Type	Defined in each table below.	M
Supplemental-Types	See IEEE Std 11073-20601-2014.	NR
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual	M
Metric-Structure-Small	See IEEE Std 11073-20601-2014.	NR
Measurement-Status	See IEEE Std 11073-20601-2014.	NR
Metric-Id	See IEEE Std 11073-20601-2014.	C
Metric-Id-List	See IEEE Std 11073-20601-2014.	NR
Metric-Id-Partition	See IEEE Std 11073-20601-2014.	NR
Unit-Code	See IEEE Std 11073-20601-2014.	M
Attribute-Value-Map	See IEEE Std 11073-20601-2014.	C
Source-Handle-Reference	See IEEE Std 11073-20601-2014.	NR
Label-String	See IEEE Std 11073-20601-2014.	O
Unit-LabelString	See IEEE Std 11073-20601-2014.	O
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Relative-Time-Stamp	See IEEE Std 11073-20601-2014.	O
HiRes-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Measure-Active-Period	See IEEE Std 11073-20601-2014.	C
Simple-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	C
Compound-Simple-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	NR
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	C
Compound-Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	NR
Nu-Observed-Value	See IEEE Std 11073-20601-2014.	NR
Compound-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	NR
Accuracy	See IEEE Std 11073-20601-2014.	R

Subclauses 6.6.2 through 6.6.6 describe the numeric objects that are defined for use in the glucose meter. Each object represents a specific aspect of blood glucose measurement or an associated medical context. The class is denoted by the Type attribute. The description of each numeric object defines the data or events it produces, the possible states, and where appropriate, its behavior. The respective tables define the numeric values generated by the agent in response to a change in state.

Sometimes, the interpretation of one attribute value in an object depends on other attribute values in the same object. For example, Unit-Code and Unit-LabelString provide context for the observed values. Whenever a contextual attribute changes, the agent shall report these changes to the manager using an MDS object event (see 6.5.3) prior to reporting any of the dependent values.

The numeric object does not support any methods, events, or other services.

For standard configurations, the optional attributes are initially not present. See IEEE Std 11073-20601-2014 for descriptive explanations on the individual attributes as well as for information on attribute ID and attribute type.

6.6.2 Blood glucose

Table 7 summarizes the attributes of the blood glucose numeric object. The blood glucose numeric object shall be supported by a glucose meter agent.

Table 7—Blood glucose numeric object attributes

Attibute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x06A5)		Standard configuration (Dev-Configuration-Id = 0x06A6)	
	Value	Qualifier	Values	Qualifier	Values	Qualifier
Handle	See IEEE Std 11073-20601-2014	M	1	M	1	M
Type	{MDC_PART_SCADA, MDC_CONC_GLU_CAPILLARY_WHOLEBLOOD or MDC_CONC_GLU_CAPILLARY_PLASMA or MDC_CONC_GLU_VENOUS_WHOLEBLOOD or MDC_CONC_GLU_VENOUS_PLASMA or MDC_CONC_GLU_ARTERIAL_WHOLEBLOOD or MDC_CONC_GLU_ARTERIAL_PLASMA or MDC_CONC_GLU_UNDETERMINED_WHOLEBLOOD or MDC_CONC_GLU_UNDETERMINED_PLASMA or MDC_CONC_GLU_ISF}		{MDC_PART_SCADA, MDC_CONC_GLU_UNDETERMINED_PLASMA}	M	{MDC_PART_SCADA, MDC_CONC_GLU_UNDETERMINED_PLASMA}	M
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated. The mss-cat-manual shall only be set if the reading is manually entered.	M	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated	M	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated	M
Unit-Code	MDC_DIM_ MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PERL	M	MDC_DIM_MILLI_G_PER_DL	M	MDC_DIM_MILLI_G_PER_DL	M
Attribute-Value-Map	See IEEE Std 11073-20601-2014.	C	MDC_ATTR_NU_VAL_OBS_BASIC, then MDC_ATTR_TIME_STAMP_ABS	M	MDC_ATTR_NU_VAL_OBS_BASIC, then MDC_ATTR_TIME_STAMP_BO	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C	If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2014 apply.	C	If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2014 apply.	C

Table 7—Blood glucose numeric object attributes *(continued)*

Attibute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x06A5)		Standard configuration (Dev-Configuration-Id = 0x06A6)	
	Value	Qualifier	Values	Qualifier	Values	Qualifier
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R	See IEEE Std 11073-20601-2014.	C	See IEEE Std 11073-20601-2014.	C
Relative-Time-Stamp	See IEEE Std 11073-20601-2014.	NR	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	NR	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	NR
HiRes-Time-Stamp	See IEEE Std 11073-20601-2014.	C	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	C	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	C
Measure-Active-Period	See IEEE Std 11073-20601-2014.	NR	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	NR	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	NR
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	C	See IEEE Std 11073-20601-2014.	M	See IEEE Std 11073-20601-2014.	M
Simple-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	C	This type of observed value is not allowed because the Basic-Nu-Observed-Value is mandatory, and IEEE Std 11073-20601-2014 mandates that one and only one type of observed value can be used.	C	This type of observed value is not allowed because the Basic-Nu-Observed-Value is mandatory, and IEEE Std 11073-20601-2014 mandates that one and only one type of observed value can be used.	C
Compound-Simple-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	NR	This type of observed value is not allowed because the Basic-Nu-Observed-Value is mandatory, and IEEE Std 11073-20601-2014 mandates that one and only one type of observed value can be used.	C	This type of observed value is not allowed because the Basic-Nu-Observed-Value is mandatory, and IEEE Std 11073-20601-2014 mandates that one and only one type of observed value can be used.	C
Accuracy	See IEEE Std 11073-20601-2014.	R	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	NR	Attribute not initially present. If present, follow IEEE Std 11073-20601-2014.	NR

NOTE—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.

The blood glucose numeric object does not support any methods, events, or other services.

For a glucose agent with a standard configuration, the AttrValMap structure (see IEEE Std 11073-20601-2014) of the Attribute-Value-Map attribute shall contain the attribute ID and attribute length information of the Basic-Nu-Observed-Value and Absolute-Time-Stamp or Base-Offset-Time-Stamp attribute in the same order as indicated in Table 7.

In case a blood glucose measurement needs to be further associated with a meal, sample location, and tester information, the additional Enumeration objects (i.e., context carbohydrates, context sample location, and context tester) can be used. See 6.8 for detail.

In standard configurations 1701 (0x06A5) and 1702 (0x06A6), or in an extended configuration that does not include the Device and Sensor Status Annunciation object, a glucose measurement that is above the capabilities of the device sensor shall be indicated with an observed value of +INFINITY, and a glucose measurement that is below the capabilities of the device sensor shall be indicated with an observed value of –INFINITY.

For measurements that exceed the capabilities of the device sensor that are reported using the extended configuration, the glucose agent shall either set the appropriate flags in the device and sensor status (GlucoseDevStat) or report the measurement using the aforementioned +/− INFINITY values but the sensor shall not send over-range flags and an over-range measurement value.

6.6.3 Hemoglobin bound to glucose A1c form (HbA1c)

HbA1c, which is also known as A1c or glycated hemoglobin, is used as the long-term measure of blood sugar control. The A1c test measures how many A1c hemoglobin cells (a specific part of red blood cells) have sugar attached to them. Because these cells live for approximately 4 months, this test indicates how well blood sugar has been controlled over the previous few months. The American Diabetes Association [B1] recommends an A1c result of 7% or less to help reduce the risk of long-term complications of diabetes. Table 8 identifies the measured, calculated, or manually entered values for recording HbA1c data.

Table 8—HbA1c numeric object attributes

Attribute name	Extended configuration	
	Value	Qualifier
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_SCADA, MDC_CONC _HBA1C}.	M
Unit-Code	MDC_DIM_ PERCENT.	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014	R
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	M
NOTE—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.		

If the HbA1c value is entered manually, the mss-cat-manual bit shall be set in Metric-Spec-Small as well.

6.6.4 Context exercise

The level of exercise undertaken in a period can be important to record to balance food intake and insulin dose. Where there are issues over control of blood glucose, a review of exercise may be helpful in the care and management of the person. Table 9 identifies the values a person can enter on the glucose device to record his or her exercise regimen.

Table 9—Context exercise numeric object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_GLU_EXERCISE}	M
Unit-Code	MDC_DIM_ PERCENT	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Measure-Active-Period	See IEEE Std 11073-20601-2014—this indicates the duration of exercise.	M
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014—this indicates the intensity of exercise ranging from 0 to 100.	M
NOTE—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.		

The context exercise numeric object does not support any methods, events, or other services.

If the Exercise value is entered manually, the mss-cat-manual bit shall be set in Metric-Spec-Small as well.

6.6.5 Context medication

Treatment for diabetes is most effective when the effects of medication on blood glucose levels are monitored. The ability to track medication with a test result can inform a physician as to whether a particular medication, or a combination of medicines, is effective. Table 10 identifies the values a person can enter on the glucose device to record his or her medication regimen.

Table 10 —Context medication numeric object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_MEDICATION}	M
Metric-Id	MDC_CTXT_MEDICATION_RAPIDACTING or MDC_CTXT_MEDICATION_SHORTACTING or MDC_CTXT_MEDICATION_INTERMEDIATEACTING or MDC_CTXT_MEDICATION_LONGACTING or MDC_CTXT_MEDICATION_PREMIX	M
Unit-Code	MDC_DIM_MILLI_G or MDC_DIM_MILLI_L or MDC_DIM_INTL_UNIT	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	M
NOTE—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.		

If the Medication value is entered manually, the mss-cat-manual bit shall be set in Metric-Spec-Small as well.

6.6.6 Context carbohydrates

Recording carbohydrates intake can be an important aid in insulin dose management. Whereas there are issues over control of blood glucose, a review of carbohydrates may be helpful in the care and management of the person. Table 11 enables the person to record the amount of carbohydrates ingested as this can directly affect the level of glucose in the blood.

Table 11 —Context carbohydrates numeric object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_GLU_CARB}	M
Metric-Id	MDC_CTXT_GLU_CARB_BREAKFAST or MDC_CTXT_GLU_CARB_LUNCH or MDC_CTXT_GLU_CARB_DINNER or MDC_CTXT_GLU_CARB_SNACK or MDC_CTXT_GLU_CARB_DRINK or MDC_CTXT_GLU_CARB_SUPPER or MDC_CTXT_GLU_CARB_BRUNCH or MDC_CTXT_GLU_CARB_UNDETERMINED or MDC_CTXT_GLU_CARB_OTHER or MDC_CTXT_GLU_CARB_NO_ENTRY or MDC_CTXT_GLU_CARB_NO_INGESTION	M
Unit-Code	MDC_DIM_G	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	M
NOTE—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.		

If the Carbohydrates value is entered manually, the mss-cat-manual bit shall be set in Metric-Spec-Small as well.

6.6.7 Control solution

Table 12 summarizes the attributes of the control solution numeric object. In standard configurations 1701 (0x06A5) and 1702 (0x06A6), the control solution numeric object shall be supported by a glucose meter agent.

Table 12 —Control solution numeric object attributes

Attribute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x06A5)		Standard configuration (Dev-Configuration-Id = 0x06A6)	
	Value	**Qualifier**	**Value**	**Qualifier**	**Value**	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M	2	M	2	M
Type	{MDC_PART_SCADA, MDC_CONC_GLU_CONTROL}	M	{MDC_PART_SCADA, MDC_CONC_GLU_CONTROL}	M	{MDC_PART_SCADA, MDC_CONC_GLU_CONTROL}	M
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated	M	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated	M	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated	M
Metric-Id	MDC_CONC_GLU_CONTROL_ LEVEL_LOW or MDC_CONC_GLU_CONTROL_ LEVEL_MEDIUM or MDC_CONC_GLU_CONTROL_ LEVEL_HIGH or MDC_CONC_GLU_CONTROL_ LEVEL_UNDETERMINED	M	MDC_CONC_GLU_CONTROL_ LEVEL_UNDETERMINED	M	MDC_CONC_GLU_CONTROL_ LEVEL_UNDETERMINED	M
Unit-Code	MDC_DIM_ MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PERL	M	MDC_DIM_MILLI_G_PER_DL	M	MDC_DIM_MILLI_G_PER_DL	M
Attribute-Value-Map	See IEEE Std 11073-20601-2014.	C	MDC_ATTR_NU_VAL_OBS_BASIC, then MDC_ATTR_ID_PHYSIO, then MDC_ATTR_TIME_STAMP_ABS	M	MDC_ATTR_NU_VAL_OBS_ BASIC, then MDC_ATTR_ID_PHYSIO, then MDC_ATTR_TIME_STAMP_BO	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C	See IEEE Std 11073-20601-2014.	C	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R	See IEEE Std 11073-20601-2014.	R	See IEEE Std 11073-20601-2014.	R
Basic-Nu-Observed-Value	See IEEE Std 11073-20601-2014.	C	See IEEE Std 11073-20601-2014.	M	See IEEE Std 11073-20601-2014.	M

In standard configurations 1701 (0x06A5) and 1702 (0x06A6), or in an extended configuration that does not include the Device and Sensor Status Annunciation object, a control solution measurement that is above the capabilities of the device sensor shall be indicated with an observed value of +INFINITY, and a glucose measurement that is below the capabilities of the device sensor shall be indicated with an observed value of –INFINITY.

If a device supports the Context Sample Location enumeration object and reports an MDC_CONC_GLU_CONTROL measurement, then the device shall also report a corresponding MDC_CTXT_GLU_SAMPLELOCATION_CTRLSOLUTION measurement.

6.7 Real-time sample array objects

Real-time sample array objects are not required by this standard.

6.8 Enumeration objects

6.8.1 General

The blood glucose meter uses a number of optional enumeration objects to represent data that are related to blood glucose. The nomenclature code to identify the enumeration class is MDC_MOC_VMO_METRIC_ENUM. The attribute structure shown in Table 13 is common to all enumeration types. Subclauses 6.8.2 through 6.8.6 define the precise definitions for each enumeration type and take precedence.

Table 13 —Common glucose enumeration object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	Defined in each enumeration table below.	M
Supplemental-Types	See IEEE Std 11073-20601-2014.	NR
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual	M
Metric-Structure-Small	See IEEE Std 11073-20601-2014.	NR
Measurement-Status	See IEEE Std 11073-20601-2014.	NR
Metric-Id	See IEEE Std 11073-20601-2014.	NR
Metric-Id-List	See IEEE Std 11073-20601-2014.	NR
Metric-Id-Partition	See IEEE Std 11073-20601-2014.	NR
Unit-Code	See IEEE Std 11073-20601-2014.	NR
Attribute-Value-Map	See IEEE Std 11073-20601-2014.	C
Source-Handle-Reference	See IEEE Std 11073-20601-2014.	NR
Label-String	See IEEE Std 11073-20601-2014.	O
Unit-LabelString	See IEEE Std 11073-20601-2014.	O
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014	R
Relative-Time-Stamp	See IEEE Std 11073-20601-2014.	O
HiRes-Time-Stamp	See IEEE Std 11073-20601-2014.	O
Measure-Active-Period	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value-Simple-OID	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value-Simple-Bit-Str	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value-Basic-Bit-Str	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value-Simple-Str	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value-Partition	See IEEE Std 11073-20601-2014.	O

Subclauses 6.8.2 through 6.8.6 describe the possible uses of the glucose enumeration object. Each use is an instance of the enumeration class with a particular Type value. The interpretation of associated values is dependent on the Type value. The description of each enumeration object defines all the possible states and, where appropriate, its behavior. The respective tables define the glucose enumeration types generated by the agent in response to a change in state.

The enumeration object does not support any methods, events, or other services. Event reports for these objects are generated by the MDS object.

See IEEE Std 11073-20601-2014 for descriptive explanations on the individual attributes as well as for information on attribute id and attribute type.

6.8.2 Device and sensor status annunciation

The device and sensor status annunciation object allows glucose meter specific errors to be recorded to track important troubleshooting information for manufacturers. Faulty sensor detection and signal irregularities are closely associated with a glucose meter, applicable to telehealth use cases. An Enumeration object fulfills this need.

An additional Enumeration object containing the running status of the device and sensor assembly is also provided. If this object is to be implemented, then the object identifier (OID) Type and bit assignments shall be implemented as described in this clause. The nomenclature code to identify the enumeration object class is MDC_MOC_VMO_METRIC_ENUM. Refer to Table 14 for the set of attributes of this object.

This object is instantiated only in extended configurations. A manager should support the interpretation of this object to enable reporting of these occurrences. An agent should support this object to transmit these occurrences.

Table 14 —Device and sensor annunciation status attributes

Attribute name	Extended configuration	
	Value	Qualifier
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_GLU_METER_DEV_STATUS}	M
Supplemental-Types	See IEEE Std 11073-20601-2014.	NR
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated	M
Metric-Structure-Small	See IEEE Std 11073-20601-2014.	NR
Measurement-Status	See IEEE Std 11073-20601-2014.	O
Metric-Id	See IEEE Std 11073-20601-2014.	NR
Metric-Id-List	See IEEE Std 11073-20601-2014.	NR
Metric-Id-Partition	See IEEE Std 11073-20601-2014.	NR
Unit-Code	See IEEE Std 11073-20601-2014.	NR
Attribute-Value-Map	See IEEE Std 11073-20601-2014.	C
Source-Handle-Reference	See IEEE Std 11073-20601-2014.	NR
Label-String	See IEEE Std 11073-20601-2014.	O
Unit-LabelString	See IEEE Std 11073-20601-2014.	O
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Ste 11073-20601-2014	R
Relative-Time-Stamp	See IEEE Std 11073-20601-2014.	C
HiRes-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Enum-Observed-Value-Simple-OID	See IEEE Std 11073-20601-2014.	O
Enum-Observed-Value-Simple-Bit-Str	See following text.	NR
Enum-Observed-Value-Basic-Bit-Str	See following text.	R
Enum-Observed-Value-Simple-Str	See IEEE Std 11073-20601-2014.	NR
Enum-Observed-Value	See following text.	NR
Enum-Observed-Value-Partition	See IEEE Std 11073-20601-2014.	O

NOTE 1—See IEEE Std 11073-20601-2014 for information on whether an attribute is static or dynamic.
NOTE 2—See 6.3 for a description of the qualifiers.

An agent explicitly expresses the existence of annunciations by setting the appropriate bits in the Enum-Observed-Value-Basic-Bit-Str attribute, as defined in Table 15. It is recommended to use the Enum-Observed-Value-Basic-Bit-Str attribute as it consumes fewer payload octets than the Enum-Observed-Value-Simple-Bit-Str attribute. The Enum-Observed-Value attribute should not be used, as it unnecessarily complicates the modeling of the object. If a manager supports the interpretation of this object, it shall be able to interpret the entire set of presented conditions. An agent may implement any subset of these same conditions. Note that a manager shall interpret these bits only within the context of this attribute and only within this device specialization as other specializations may use corresponding terms for different purposes.

Table 15 —Mapping of device, sensor, and signal status to object Bit-Str attribute

Device or sensor condition	GlucoseDevStat mnemonic
Agent reports that the battery is low and needs replacing.	device-battery-low
Agent reports that the sensor is malfunctioning or faulting.	sensor-malfunction
Agent reports that there is not enough blood/control solution on the strip.	sensor-sample-size-insufficient
Agent reports that the strip was inserted incorrectly.	sensor-strip-insertion
Agent reports that the strip is not the right type for the device.	sensor-strip-type-incorrect
Agent reports that the reading or value is higher than the sensor can process.	sensor-result-too-high
Agent reports that the reading or value is lower than the sensor can process.	sensor-result-too-low
Agent reports that the ambient temperature is too high for a valid test/result.	sensor-temp-too-high[a]
Agent reports that the ambient temperature is too low for a valid test/result.	sensor-temp-too-low[a]
Agent reports that the reading was interrupted or the strip was pulled too soon.	sensor-read-interrupt
A general device fault has occurred in the agent.	device-gen-fault
Agent reports that the ambient temperature is out of range for a valid test/result.	sensor-temp-out-of-range[a]

[a] The agent shall not indicate both sensor-temp-too-high and sensor-temp-too-low at the same time. If either the sensor-temp-too-high or sensor-temp-too-low bits are set, the agent shall set the sensor-temp-out-of-range bit as well. The agent shall only use sensor-temp-out-of-range by itself if sensor-temp-too-high or sensor-temp-too-low cannot be determined by the device.

The specific bit mappings of GlucoseDevStat are defined in Annex B.

6.8.3 Context meal

A blood sugar measurement (or referred to as a reading) may be further associated with information on meal relationship when this measurement is taken. The blood sugar level can be significantly affected by the testing time relative to the meal time. Table 16 defines the meal condition at the time when the blood sugar measurement is taken.

Table 16 —Context meal enumeration object attributes

Attribute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x06A6)	
	Value	Qualifier	Value	Qualifier
Handle	See IEEE Std 11073-20601 2014.	M	3	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_GLU_MEAL}.	M	{MDC_PART_PHD_DM, MDC_CTXT_GLU_MEAL}.	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R	See IEEE Std 11073-20601-2014.	R
Enum-Observed-Value-Simple-OID	One of the following nomenclature value shall be used: MDC_CTXT_GLU_MEAL_PREPRANDIAL MDC_CTXT_GLU_MEAL_POSTPRANDIAL MDC_CTXT_GLU_MEAL_FASTING MDC_CTXT_GLU_MEAL_BEDTIME MDC_CTXT_GLU_MEAL_CASUAL	M	One of the following nomenclature value shall be used: MDC_CTXT_GLU_MEAL_PREPRANDIAL MDC_CTXT_GLU_MEAL_POSTPRANDIAL MDC_CTXT_GLU_MEAL_FASTING MDC_CTXT_GLU_MEAL_BEDTIME MDC_CTXT_GLU_MEAL_CASUAL	M

6.8.4 Context sample location

A blood sugar measurement may be further characterized by the blood sample location. Table 17 defines the possible blood locations where the blood sample may be taken.

Table 17 —Context sample location enumeration object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_GLU_SAMPLELOCATION}	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Enum-Observed-Value-Simple-OID	One of the following nomenclature value shall be used: MDC_CTXT_GLU_SAMPLELOCATION_FINGER MDC_CTXT_GLU_SAMPLELOCATION_AST MDC_CTXT_GLU_SAMPLELOCATION_EARLOBE MDC_CTXT_GLU_SAMPLELOCATION_CTRLSOLUTION MDC_CTXT_GLU_SAMPLELOCATION_OTHER MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED	M

6.8.5 Context tester

The precision (or validity) of a blood sugar measurement can be impacted by whom and where the measurement is performed. Table 18 defines the possible cases of testers who may perform the blood sugar measurement.

Table 18 —Context tester enumeration object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_GLU_TESTER}	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Enum-Observed-Value-Simple-OID	One of the following nomenclature value shall be used: MDC_CTXT_GLU_TESTER _SELF MDC_CTXT_GLU_TESTER _HCP MDC_CTXT_GLU_TESTER _LAB	M

6.8.6 Context health

Stress is an important factor when monitoring blood sugar levels, as excessive stress can cause these levels to rise. Stress hormones such as epinephrine and cortisol are released and, as a result, can significantly raise the blood sugar levels. Table 19 defines the various levels of health a person feels when taking a glucose measurement. Although the state of health is a matter of perception, it enables a dialog between patient and physician to discuss matters that may affect therapy. *Minor* and *major* are related to the general health or the level of illness of the person. *Menses* refers to the female menstrual cycle, and *stress* refers to physiological or psychological stress. In the event of multiple readings, the most impactful on the result shall be noted.

Table 19 —Context health enumeration object attributes

Attribute name	Extended configuration	
	Value	**Qualifier**
Handle	See IEEE Std 11073-20601-2014.	M
Type	{MDC_PART_PHD_DM, MDC_CTXT_GLU_HEALTH}	M
Absolute-Time-Stamp	See IEEE Std 11073-20601-2014.	C
Base-Offset-Time-Stamp	See IEEE Std 11073-20601-2014.	R
Enum-Observed-Value-Simple-OID	One of the following nomenclature value shall be used: MDC_CTXT_GLU_HEALTH_MINOR MDC_CTXT_GLU_HEALTH_MAJOR MDC_CTXT_GLU_HEALTH_MENSES MDC_CTXT_GLU_HEALTH_STRESS MDC_CTXT_GLU_HEALTH_NONE	M

6.9 PM-store objects

6.9.1 General

In the context of personal health devices, glucose meters are highly portable due to their convenient sizes and are normally carried around with users so that blood glucose measurements may be taken as needed. More often than not, blood glucose measurements are taken at a time when out of the network and when manager/agent associations cannot be established. It is also common that a given set of measurements made by glucose meters may need to be uploaded to more than one manager, for example, in the home and at a medical facility.

To support dual usage, an agent may provide two or more configurations. One configuration may use a temporary measurement storage model that uploads the most recent data immediately on association (agent initiated) with little user intervention, such as might be used by a typical home user that uploads measurements frequently to a personal computer or a mobile device such as a cell phone. Another configuration may use a long-term measurement storage model that uploads data at the request of the manager, such as might be used by the patient's physician or other health care professionals.

The long-term storage model is realized using PM-stores. Any configuration that does not include a PM-store object utilizes agent-initiated event reports to transmit the observations. The use of temporarily stored data as defined in IEEE Std 11073-20601-2014 is most useful for small numbers of measurements and is subject to automatic deletion during upload.

Alternatively, in the case where a large number of measurements may be stored or if automatic deletion is to be avoided, a PM-store configuration should be used. Any configuration with a PM-store for persistent storage shall disable agent-initiated transmission and shall enable access to the PM-store transmissions. As a result, this standard describes a mechanism using PM-store to hold measurements for longer durations. The data held in PM-store objects are deleted by user actions via the manager or user interface on the device, and the capacity is limited only by the amount of memory.

6.9.2 Persistent store model

The PM-store model utilizes a PM-segment for each type of object to be persistently stored (see Figure 2). The segment holding glucose readings shall be present if the PM-store is implemented. The other segments are optional and hold observations from the supporting medical context objects that are implemented. Each entry shall include one of the time formats in the segm-entry-header so a manager can correlate entries across the different segments. Entries in the supported medical context PM-segments do not require a correlating result in the GlucoseSeg PM-segment, and can therefore be independent of a glucose measurement. This model was selected to reduce the transmission sizes as much as possible. If a particular medical context object is not supported, then the segment is not required to exist. If a person does not enter any data of that type when a reading is taken (e.g., the person does not respond to a prompt for what was eaten recently), then the entry in the PM-segment is not created. The PM-segments that are zero to many in Figure 2 behave this way because of the unknown number of timestamp changes, which are used to determine a common notion of time.

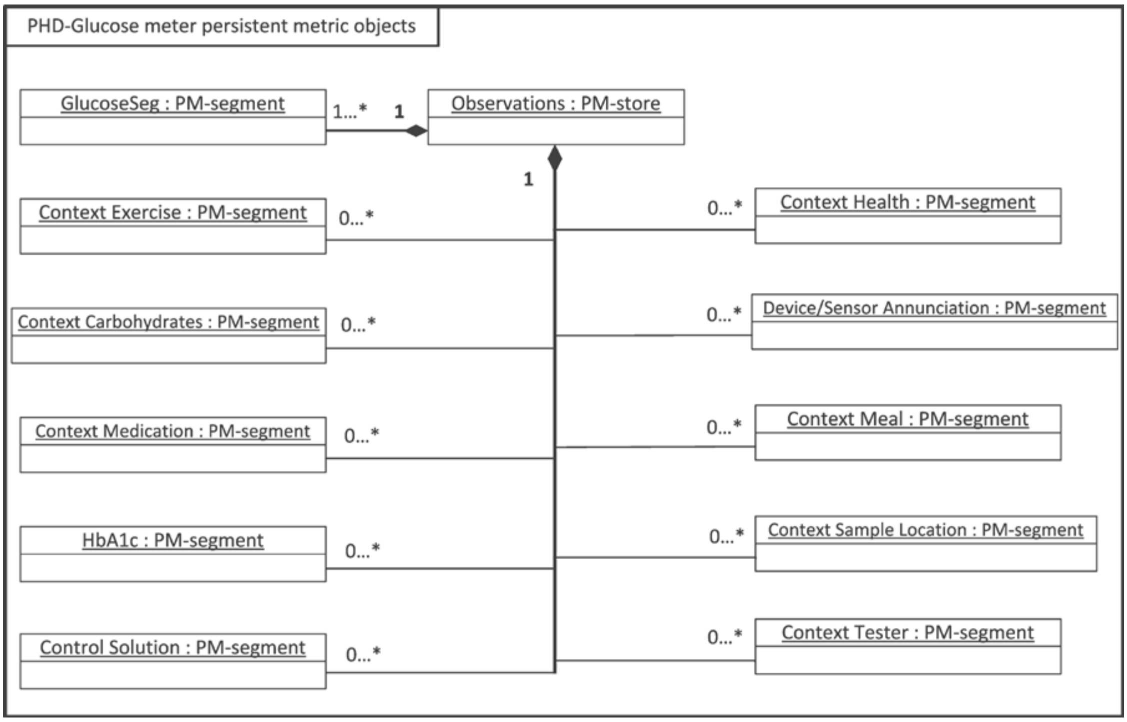
PHD-Glucose meter persistent metric objects
GlucoseSeg : PM-segment
1...*
1
Observations : PM-store
1
Context Exercise : PM-segment
0...*
Context Health : PM-segment
0...*
Context Carbohydrates : PM-segment
0...*
Device/Sensor Annunciation : PM-segment
0...*
Context Medication : PM-segment
0...*
Context Meal : PM-segment
0...*
HbA1c : PM-segment
0...*
Context Sample Location : PM-segment
0...*
Control Solution : PM-segment
0...*
Context Tester : PM-segment
0...*

Figure 2—Glucose meter—persistent store model

6.9.3 PM-store object attributes

Table 20 defines the attributes of the PM-store objects that shall be implemented by the agent. The nomenclature code to identify the PM-store objects is MDC_MOC_VMO_PMSTORE.

Table 20 —Glucose meter PM-store object attributes

Attribute name	Extended configuration	
	Value	Qualifier
Handle	See IEEE Std 11073-20601-2014.	M
PM-Store-Capab	See IEEE Std 11073-20601-2014.	M
Store-Sample-Algorithm	See IEEE Std 11073-20601-2014.	M
Store-Capacity-Count	See IEEE Std 11073-20601-2014.	M
Store-Usage-Count	See IEEE Std 11073-20601-2014.	M
Operational-State	See IEEE Std 11073-20601-2014.	M
PM-Store-Label	See IEEE Std 11073-20601-2014.	M
Sample-Period	See IEEE Std 11073-20601-2014.	NR
Number-Of-Segments	Number dependent on context support or time changes.	M
Clear-Timeout	See IEEE Std 11073-20601-2014.	C

Some considerations when using the PM-Store-Capab are as follows:

- If the agent creates new segments because of time changes as described in the “Comparable time” clause of IEEE Std 11073-20601-2014, then pmsc-var-no-of-segm shall be set.
- If the agent is recording episodic data in the PM-store, then the pmsc-epi-seg-entries shall be set.
- The remaining bits are agent specific.

6.9.4 PM-store object methods

Table 21 defines the methods of the PM-store objects.

Table 21 —Glucose meter PM-store object methods

Service	Subservice type name	Mode	Subservice type (action-type)	Parameters (action-info-args)	Results (action-info-args)
ACTION	Clear-Segments	Confirmed	MDC_ACT_SEG_ CLR	SegmSelection	(empty)
ACTION	Get-Segment-Info	Confirmed	MDC_ACT_SEG_ GET_INFO	SegmSelection	SegmentInfoList
ACTION	Get-Segment-Id-List	Confirmed	MDC_ACT_SEG_ GET_ID_LIST	(empty)	SegmIdList
ACTION	Trig-Segment-Data-Xfer	Confirmed	MDC_ACT_SEG_ TRIG_XFER	TrigSegmDataXferReq	TrigSegmDataXfer Rsp

- **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.
- **Get-Segment-Info**
  This method allows the manager to retrieve the PM-segment attributes.
- **Get-Segment-Id-List**
  This method allows the manager to retrieve a list of the instance numbers of all the PM-segments of a PM-store.
- **Trig-Segment-Data-Xfer**
  This method allows the manager to initiate the transfer of the data entries stored in the PM-segment object.

6.9.5 PM-store object events

Table 22 defines the events sent by the PM-store objects.

Table 22 —Glucose meter PM-store object events

Service	Subservice type name	Mode	Subservice type (event-type)	Parameters (event-info)	Results (event-reply-info)
EVENT REPORT	Segment-Data-Event	Confirmed	MDC_NOTI_SEGMENT _DATA	SegmentDataEvent	SegmentDataResult

— **Segment-Data-Event**
This event allows the agent to send the data entries stored in the PM-segment object. This event is triggered by the manager using the Trig-Segment-Data-Xfer action.

6.9.6 PM-store object services

6.9.6.1 GET service

The PM-store object supports the GET service to retrieve the values of all PM-store object attributes. The GET service may be invoked as soon as the agent is in the Operating state.

6.9.7 PM-segment objects

Each of the PM-store objects contains a corresponding PM-segment object.

Table 23 defines the attributes of the PM-segment object contained in the PM-store object that represents the stored glucose measurements. The nomenclature code to identify the PM-segment object class is MDC_MOC_PM_SEGMENT.

Table 23 —Common PM-segment object attributes

Attribute name	Extended configuration	
	Value	Qualifier
Instance Number	See IEEE Std 11073-20601-2014.	M
PM-Segment-Entry-Map	See IEEE Std 11073-20601-2014.	M
PM-Seg-Person-Id	See IEEE Std 11073-20601-2014.	C
Sample-Period	See IEEE Std 11073-20601-2014.	C
Operational-State	See IEEE Std 11073-20601-2014.	M
Segment-Label	See IEEE Std 11073-20601-2014.	M
Segment-Start-Abs-Time	See IEEE Std 11073-20601-2014.	C
Segment-End-Abs-Time	See IEEE Std 11073-20601-2014.	C
Segment-Start-BO-Time	See IEEE Std 11073-20601-2014.	R
Segment-End-BO-Time	See IEEE Std 11073-20601-2014.	R
Segment-Date-and-Time-Adjustment	See IEEE Std 11073-20601-2014.	C
Segment-Usage-Count	See IEEE Std 11073-20601-2014.	M
Segment-Statistics	See IEEE Std 11073-20601-2014.	O
Fixed-Segment-Data	Segment data transferred as an array of entries in a format as specified in the PM-Segment-Entry-Map attribute.	M
Confirm-Timeout	See IEEE Std 11073-20601-2014.	O
Transfer-Timeout	See IEEE Std 11073-20601-2014.	M

The Fixed-Segment-Data attribute stores the actual measurements or context logs. When the Fixed-Segment-Data attribute is transmitted, all entries in the event report are formatted according to the PM-Segment-Entry-Map. Each entry stores a single sample point, which may consist of a set of attributes.

6.10 Scanner objects

Scanner objects are not required by this standard.

6.11 Class extension objects

In this standard, no class extension objects are defined with respect to IEEE Std 11073-20601-2014.

6.12 Glucose meter information model extensibility rules

The glucose meter domain information model of this standard may be extended by including vendor-specific metrics and attributes as required. For example, a vendor might include a cholesterol measurement in addition to the glucose measurement. Any object or attribute extensions implemented should follow the guidelines of this standard as closely as possible.

A glucose meter agent having a configuration with extensions beyond the standard configuration, as specified in this standard, shall use a configuration ID in the range of IDs reserved for extended configurations (see IEEE Std 11073-20601-2014).

7. Glucose meter service model

7.1 General

The service model defines the conceptual mechanisms for data exchange services. These services are mapped to messages that are exchanged between the agent and the manager. Protocol messages within the ISO/IEEE 11073 series of standards are defined in ASN.1. See IEEE Std 11073-20601-2014 for a detailed description of the personal health device service model. Subclauses 7.2 and 7.3 define the specifics of object access and event reporting services for a glucose meter agent according to this standard.

7.2 Object access services

The object access services of IEEE Std 11073-20601-2014 are used to access the objects defined in the domain information model of the glucose device.

The following generic object access services are supported by a glucose meter agent according to this standard:

- **GET service**: used by the manager to retrieve the values of the agent MDS and PM-store object attributes. The list of glucose meter MDS object attributes is given in 6.5.4.1, and the list of glucose meter PM Store attributes is given in 6.9.3.

- **SET service**: used by the manager to set the values of the agent object attributes. No settable attributes are defined for a glucose meter agent according to this standard.
- **Event report service**: used by the agent to send configuration reports and measurement data to the manager. The list of event reports for the glucose meter device specialization is given in 6.5.3.
- **Action service**: used by the manager to invoke actions (or methods) supported by the agent. An example is Set-Time action, which is used to set a real-time clock with the absolute time at the agent.

Table 24 summarizes the object access services described in this standard.

7.3 Object access event report services

The event report service (see Table 24) is used by the agent to report its information (e.g., measurements). Event reports in this standard are a property of the MDS object only. The event reports used in this standard are defined in IEEE Std 11073-20601-2014.

The following conditions apply for a glucose meter agent according to this standard:

- Event reports shall be used in confirmed mode.
- Agent-initiated mode shall be supported for measurement data transmission.

A glucose meter agent designed to operate in an environment where data may be collected from multiple people may use one of the multiple-person event report styles to transmit all the data from each person in a single event. If this functionality is not required, the agent may use the single-person event report styles, which have reduced overhead.

A manager shall support both single-person and multiple-person event reports. A glucose meter agent may support either one or both single-person and multiple-person event reports. The formats for single- and multiple-person reports are described in IEEE Std 11073-20601-2014.

Table 24—Glucose meter object access services

Service	Subservice type name	Mode	Subservice type	Parameters	Result	Remarks
GET	<na>	<implied Confirmed>	<na>	GetArgument Simple = (obj-handle = 0), attribute-id-list *<optional>*	GetResultSimple = (obj-handle = 0), attribute-list	Allows the manager to retrieve the value of an attribute of an object in the agent.
	<na>	<implied Confirmed>	<na>	GetArgument Simple = (obj-handle = *handle of PM-Store object*), attribute-id-list *<optional>*	GetResultSimple = (obj-handle = *handle of PM-Store object*), attribute-list	Allows the manager to retrieve the values of all PM-store object attributes.
EVENT REPORT	MDS-Configuration-Event	Confirmed	MDC_NOTI_CONFIG	ConfigReport	ConfigReportRsp	Configuration Report to inform manager of the agent's configuration.
	MDS-Dynamic-Data-Update-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_FIXED	ScanReport InfoFixed	—	Data Report to provide dynamic data to manager for some or all of the agent's objects in fixed format.
	MDS-Dynamic-Data-Update-Var	Confirmed	MDC_NOTI_SCAN_REPORT_VAR	ScanReport InfoVar	—	Data Report to provide dynamic data to manager for some or all of the agent's objects in variable format.
	MDS-Dynamic-Data-Update-MP-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReport InfoMPFixed	—	This is the same as MDS-Dynamic-Data-Update-Fixed but allows inclusion of data from multiple people.
	MDS-Dynamic-Data-Update-MP-Var	Confirmed	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReport InfoMPVar	—	This is the same as MDS-Dynamic-Data-Update-Var but allows inclusion of data from multiple people.
ACTION	Set-Time	Confirmed	MDC_ACT_SET_TIME	SetTime Invoke	—	Manager method to invoke the agent to set time to requested value.
	Set-Base-Offset-Time	Confirmed	MDC_ACT_SET_BO_TIME	SetBOTime Invoke	—	Manager method to invoke the agent to set time to requested value.
	Clear-Segments	Confirmed	MDC_ACT_SEG_CLR	SegmSelection	—	Allows the manager to clear some or all of the entries contained in PM-segment objects.
	Get-Segment-Info	Confirmed	MDC_ACT_SEG_GET_INFO	SegmSelection	SegmentInfoList	Allows the manager to retrieve information about some or all PM-segment objects.
	Get-Segment-Id-List	Confirmed	MDC_ACT_SEG_GET_ID_LIST	(empty)	SegmIdList	Allows the manager to retrieve a list of the instance numbers of all the PM-segments of a PM-store.
	Trig-Segment-Data-Xfer	Confirmed	MDC_ACT_SEG_TRIG_XFER	TrigSegmData XferReq	TrigSegmDataXfer Rsp	Allows the manager to begin sending segment data.

8. Glucose meter communication model

8.1 Overview

This clause describes the general communication model and procedures of the glucose meter agent as defined in IEEE Std 11073-20601-2014. Therefore, the respective parts of IEEE Std 11073-20601-2014 are not reproduced; rather, the specific choices and restrictions with respect to optional elements (e.g., objects, attributes, and actions) and specific extensions (e.g., nomenclature terms) are specified.

For an illustrative overview of the various message transactions during a typical measurement session, see the sequence diagram for the example use case in Annex D and the corresponding protocol data unit (PDU) examples in Annex E.

8.2 Communication characteristics

In this subclause, limits on the size of an application protocol data unit (APDU) transmitted or to be received by a glucose agent are defined. Small limits allow for simple implementations in terms of low cost and complexity.

A glucose agent implementing only this device specialization shall not transmit any APDU larger than N_{tx} and shall be capable of receiving any APDU up to a size of N_{rx}. For this standard, N_{tx} shall be 64 512 octets for implementations supporting persistent metric storage. In the absence of the persistent metric storage capability, N_{tx} shall be 5120 octets. For this standard, N_{rx} shall be 224 octets.

For a glucose agent implementing functions from other device specializations, an upper bound estimation of the APDU sizes brings the following: An agent shall not transmit any APDU larger than the sum of N_{tx} of all the device specializations implemented and shall be capable of receiving any APDU up to the sum of N_{rx} of all the device specializations implemented. If these numbers are higher than the maximum size determined in IEEE Std 11073-20601-2014, the latter shall be applied.

In case the APDU size limit does not allow for the inclusion of a certain amount of multiple pending measurements at the agent, they shall be sent using multiple event reports. See 8.5.3 for the maximum number of measurements allowed for inclusion in a single event report.

8.3 Association procedure

8.3.1 General

Unless otherwise stated, the association procedure for a glucose meter agent and manager according to this standard shall be pursued as specified in IEEE Std 11073-20601-2014.

8.3.2 Agent procedure—association request

In the association request sent by the agent to the manager:

- The version of the association procedure used by the agent shall be set to assoc-version1 (i.e., *assoc-version* = 0x80000000).

- The DataProtoList structure element of the data protocol identifier shall be set to data-proto-id-20601 (i.e., *data-proto-id* = 0x5079).
- The *data-proto-info* field shall contain a PhdAssociationInformation structure that shall contain the following parameter values:
  1) If a glucose agent relies on a feature introduced in a specific version of the 20601 protocol, the agent shall only indicate support for that specific version of 20601 protocol or higher. The agent's supported protocol version(s) shall be indicated by setting the appropriate protocol-version bit(s).
     i) Major features introduced with version 2 of the protocol are:
        BOT to handle multiple time changes efficiently
     ii) Major features introduced in version 3 of the protocol are:
        Get Segment Id List to support reporting a large number of PM-Segments
        Reporting time faults using the Date-Time-Adjustment value 7FFFFFFF
  2) At least the MDERs shall be supported (i.e., *encoding-rules* = 0x8000).
  3) The version of the nomenclature used shall be set to nom-version1 (i.e., *nomenclature-version* = 0x80000000).
  4) The field *functional-units* may have the test association bits set but shall not have any other bits set.
  5) The field *system-type* shall be set to sys-type-agent (i.e., *system-type* = 0x00800000).
  6) The s*ystem-id* field shall be set to the value of the System-Id attribute of the MDS object of the agent. The manager may use this field to determine the identity of the glucose meter with which it is associating and, optionally, to implement a simple access restriction policy.
  7) The *dev-config-id* field shall be set to the value of the Dev-Configuration-Id attribute of the MDS object of the agent.
  8) If the agent supports only the glucose meter specialization, then the field indicating the data request modes (*data-req-mode-capab*) supported by the glucose meter agent shall be set to *data-req-supp-init-agent*.
  9) If the agent supports only the glucose meter specialization, then *data-req-init-manager-count* shall be set to zero, and *data-req-init-agent-count* shall be set to 1.

8.3.3 Manager procedure—association response

In the association response message sent by the manager:

- The *result* field shall be set to an appropriate response from those defined in IEEE Std 11073-20601-2014. For example, if all other conditions of the association protocol are satisfied, *accepted* is returned when the manager recognizes the *dev-config-id* of the agent and *accepted-unknown-config* otherwise.
- In the DataProtoList structure element, the data protocol identifier shall be set to data-proto-id-20601 (i.e., *data-proto-id* = 0x5079).
- The *data-proto-info* field shall be filled in with a PhdAssociationInformation structure that shall contain the following parameter values:
  1) The manager shall respond to protocol versions supported by setting the appropriate bit in the protocol-version.
  2) The manager shall respond with a single selected encoding rule that is supported by both agent and manager. The manager shall support at least the MDERs.

3) The version of the nomenclature used shall be set to nom-version1 (i.e., *nomenclature-version* = 0x80000000).

4) The field *functional-units* shall have all bits reset except for those relating to a test association.

5) The field *system-type* shall be set to sys-type-manager (i.e., *system-type* = 0x80000000)

6) The *system-id* field shall contain the unique system ID of the manager, which shall be a valid EUI-64 type identifier.

7) The field *dev-config-id* shall be manager-config-response (0).

8) The field *data-req-mode-capab* shall be 0.

9) The fields *data-req-init-*-count* shall be 0.

8.4 Configuring procedure

8.4.1 General

The agent enters the Configuring state if it receives an association response of accepted-unknown-config. In this case, the configuration procedure as specified in IEEE Std 11073-20601-2014 shall be followed. Subclause 8.4.2 specifies the configuration notification and response messages for a glucose meter agent with standard configuration ID 0x06A5 (1701), and 8.4.2 specifies the configuration notification and response messages for a glucose meter agent with a standard configuration ID 0x06A6 (1702) as introduced in this version of 10417. Normally, a manager would already know the standard configuration. However, standard configuration devices are required to send their configuration, if requested. This covers a case where an agent associates with a manager that does not have preconfigured knowledge of the standard configuration (e.g., due to a version mismatch between agent and manager).

The agent shall support the GET service for the MDS attributes during the Configuring state.

8.4.2 Glucose meter—standard configuration (0x06A5)

8.4.2.1 Agent procedure

The agent performs the configuration procedure using a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the manager (see IEEE Std 11073-20601-2014). The ConfigReport structure is used for the *event-info* field (see Table 4). For a glucose meter agent with standard configuration 1701 with ID 0x06A5, the format and contents of the configuration notification message are as follows:

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x70	CHOICE.length = 112
0x00 0x6E	OCTET STRING.length = 110
0x00 0x01	invoke-id (differentiates this message from any other outstanding)
0x01 0x01	CHOICE (Remote Operation Invoke | Confirmed Event Report)
0x00 0x68	CHOICE.length = 104
0x00 0x00	obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF	event-time (set to 0xFFFFFFFF if RelativeTime is not supported)
0x0D 0x1C	event-type = MDC_NOTI_CONFIG
0x00 0x5E	event-info.length = 94 (start of ConfigReport)

```
0x06 0xA5               config-report-id (Dev-Configuration-Id value)
0x00 0x02               config-obj-list.count = 2 Measurement objects will be “announced”
0x00 0x58               config-obj-list.length = 88
0x00 0x06               obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01               obj-handle = 1     (→ 1st Measurement is blood glucose)
0x00 0x04               attributes.count = 4
0x00 0x24               attributes.length = 36
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x02               MDC_PART_SCADA |
0x72 0x70               MDC_CONC_GLU_UNDETERMINED_PLASMA
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0xF0 0x40               intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x08 0x52               MDC_DIM_MILLI_G_PER_DL
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C               attribute-value.length = 12
0x00 0x02               AttrValMap.count = 2
0x00 0x08               AttrValMap.length = 8
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x09 0x90 0x00 0x08     MDC_ATTR_TIME_STAMP_ABS | value length = 8
0x00 0x06               obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x02               obj-handle = 2     (→ 2nd Measurement is control solution)
0x00 0x04               attributes.count = 4
0x00 0x24               attributes.length = 36
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x02               MDC_PART_SCADA |
0x71 0xD0               MDC_CONC_GLU_CONTROL
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0xF0 0x40                intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x08 0x52               MDC_DIM_MILLI_G_PER_DL
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C               attribute-value.length = 12
0x00 0x02               AttrValMap.count = 2
0x00 0x08               AttrValMap.length = 8
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x09 0x90 0x00 0x08     MDC_ATTR_TIME_STAMP_ABS | value length = 8
```

8.4.2.2 Manager procedure

The manager shall respond to a configuration notification message using a “Remote Operation Response | Confirmed Event Report” data message with an MDC_NOTI_CONFIG event using the ConfigReportRsp structure for the *event-info* field (see Table 4). As a response to the standard configuration notification message in 8.4.2.1, the format and contents of the manager's configuration notification response message are as follows:

```
0xE7 0x00               APDU CHOICE Type (PrstApdu)
0x00 0x16               CHOICE.length = 22
0x00 0x14               OCTET STRING.length = 20
0x00 0x01               invoke-id (differentiates this message from any other outstanding)
0x02 0x01               CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E               CHOICE.length = 14
0x00 0x00               obj-handle = 0 (MDS object)
0x05 0x14 0xDB 0x12     currentTime
0x0D 0x1C               event-type = MDC_NOTI_CONFIG
0x00 0x04               event-reply-info.length = 4
0x06 0xA5               ConfigReportRsp.config-report-id = 1701
0x00 0x00               ConfigReportRsp.config-result = accepted-config
```

8.4.3 Glucose meter—standard configuration (0x06A6)

8.4.3.1 Agent procedure

The agent performs the configuration procedure using a "Remote Operation Invoke | Confirmed Event Report" message with an MDC_NOTI_CONFIG event to send its configuration to the manager (see IEEE Std 11073-20601-2014). The ConfigReport structure is used for the *event-info* field (see Table 4). For a glucose meter agent with standard configuration ID 0x06A6, the format and contents of the configuration notification message are as follows:

```
0xE7 0x00               APDU CHOICE Type (PrstApdu)
0x00 0x96               CHOICE.length = 150
0x00 0x94               OCTET STRING.length = 148
0x00 0x02               invoke-id = 2 (differentiates this message from any other outstanding)
0x01 0x01               CHOICE (Remote Operation Invoke | Confirmed Event Report Simple)
0x00 0x8E               CHOICE.length = 142
0x00 0x00               obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF     event-time (set to 0xFFFFFFFF if RelativeTime is not supported)
0x0D 0x1C               event-type = MDC_NOTI_CONFIG
0x00 0x84               event-info.length = 132 (start of ConfigReport)
0x06 0xA6               config-report-id = 06A6 (Dev-Configuration-Id value=1702)
0x00 0x03               config-obj-list.count = 3 objects will be "announced"
0x00 0x7E               config-obj-list.length = 126
0x00 0x06               obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01               obj-handle = 1 (→ 1st Measurement is blood glucose)
0x00 0x04               attributes.count = 4
0x00 0x24               attributes.length = 36
0x0A0x46                attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0xF0 0x40               intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C               attribute-value.length = 12
0x00 0x02               AttrValMap.count = 2
0x00 0x08               AttrValMap.length = 8
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
```

```
0x0A 0x82 0x00 0x08     MDC_ATTR_TIME_STAMP_BO | value length = 8
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x08 0x52               MDC_DIM_MILLI_G_PER_DL
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x02               MDC_PART_SCADA |
0x72 0x70               MDC_CONC_GLU_UNDETERMINED_PLASMA
0x00 0x06               obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x02               obj-handle = 2 (→ 2nd Measurement is control solution)
0x00 0x04               attributes.count = 4
0x00 0x24               attribute-value.length = 36
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0xF0 0x40               intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C               attribute-value.length = 12
0x00 0x02               AttrValMap.count = 2
0x00 0x08               AttrValMap.length = 8
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x0A 0x82 0x00 0x08     MDC_ATTR_TIME_STAMP_BO | value length = 8
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x08 0x52               MDC_DIM_MILLI_G_PER_DL
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x02               MDC_PART_SCADA |
0x71 0xD0               MDC_CONC_GLU_CONTROL
0x00 0x05               obj-class = MDC_MOC_VMO_METRIC_ENUM
0x00 0x03               obj-handle = 3 (→ 3rd object is Context-Meal)
0x00 0x03               attributes.count = 3
0x00 0x1E               attributes.length = 30
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0xF0 0x40               intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C               attribute-value.length = 12
0x00 0x02               AttrValMap.count = 2
0x00 0x08               AttrValMap.length = 8
0x0A 0x49               attribute-id = MDC_ATTR_ENUM_OBS_VAL_SIMP_OID
0x00 0x04               attribute-value.length = 4
0x0A 0x82 0x00 0x08     MDC_ATTR_TIME_STAMP_BO | value length = 8
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x80 0x72 0x48     MDC_PART_PHD_DM | MDC_CTXT_GLU_MEAL
```

8.4.3.2 Manager procedure

The manager shall respond to a configuration notification message using a “Remote Operation Response | Confirmed Event Report” data message with an MDC_NOTI_CONFIG event using the ConfigReportRsp structure for the *event-info* field (see Table 4). As a response to the standard configuration notification message in 8.4.2.1, the format and contents of the manager's configuration notification response message are as follows:

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x16                CHOICE.length = 22
0x00 0x14                OCTET STRING.length = 20
0x00 0x02                invoke-id (differentiates this message from any other outstanding)
0x02 0x01                CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E                CHOICE.length = 14
0x00 0x00                obj-handle = 0 (MDS object)
0x05 0x14 0xDB 0x12      currentTime
0x0D 0x1C                event-type = MDC_NOTI_CONFIG
0x00 0x04                event-reply-info.length = 4
0x06 0xA6                ConfigReportRsp.config-report-id = 1702
0x00 0x00                ConfigReportRsp.config-result = accepted-config
```

8.5 Operating procedure

8.5.1 General

Measurement data and status information are communicated from the glucose meter agent during the Operating state. If not stated otherwise, the operating procedure for a glucose meter agent of this standard shall be as specified in IEEE Std 11073-20601-2014.

8.5.2 GET glucose meter MDS and PM-store attributes

The glucose meter shall support the GET service for MDS and PM-store objects (if it has PM-store objects) according to IEEE Std 11073-20601-2014.

See Table 5 for a summary of the GET service.

If the manager leaves the *attribute-id-list* field in the roiv-cmip-get service message empty, the glucose meter agent shall respond with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object. This allows the glucose meter agent to store a predefined transmission template for its response message and to modify just the varying parts at fixed locations before sending.

If the manager requests specific MDS object attributes, indicated by the elements in *attribute-id-list*, and the agent supports this capability, the glucose meter agent shall respond with a rors-cmip-get service 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 a glucose meter agent to support this capability. If this capability is not implemented, the glucose meter agent shall respond with a "Remote Operation Error Result" (roer) service message (see IEEE Std 11073-20601-2014) with the error-value field set to no-such-action (9).

8.5.3 Measurement data transmission

See Table 4 and Table 22 for a summary of the event report services available for measurement data transfer.

For temporary store measurement, data transfer for a glucose meter agent of this standard shall always be initiated by the glucose meter (see agent-initiated measurement data transmission in IEEE Std 11073-20601-2014). To limit the amount of data being transported within an APDU, the glucose meter agent shall not include more than 25 temporarily stored measurements in a single event report. If more than 25 pending measurements are available for transmission, they shall be sent using multiple event reports. If multiple glucose measurements are available, up to 25 measurements should be transmitted within a single event report. Alternatively, they may be transmitted using a single event report for each glucose measurement. However, the former strategy is recommended to reduce overall message size and power consumption.

For PM-store transfers, the manager inspects the available segments and retrieves all applicable data.

8.6 Time synchronization

Time synchronization may be employed between a glucose meter and a manager to coordinate the clocks used when reporting physiological events. Note that the mechanism for synchronizing an agent to a manager is outside the scope of this standard. If time synchronization is used, then this shall be reported in the Mds-Time-Info attribute of the MDS object.

9. Test associations

A glucose meter may implement a wide range of behaviors in a test association that enable a manufacturer to test features of a product in a comprehensive manner. It is also possible for a glucose meter not to support test associations at all. This clause defines a simple behavior that simulates the generation of a measurement in the context of a standard device configuration.

9.1 Behavior with standard configuration

To facilitate automated standardized test processes, a glucose meter that presents the standard configuration ID and enters into a test association should be able to simulate the arrival of measurement data from the device measurement engine. It should not be necessary for an operator to perform a test for the measurement data to be generated.

After the agent enters the Operating state, it simulates the reception of an event from the measurement engine representing a blood glucose measurement of 999 mg/dL. To the extent possible, this measurement is observed only by those components of the agent that understand the test association. When the event is propagated into a numeric object, the test-data bit of the measurement-status attribute shall be set if the measurement-status attribute is supported. An agent is not required to use the measurement-status attribute if it would not normally do so outside of a test association.

The agent should send the events reports for all simulated measures within 30 s of entering the Operating state. The test association is terminated in a manner consistent with the agent’s normal behavior for terminating an association.

9.2 Behavior with extended configurations

This specification does not define a test association that uses an extended configuration.

10. Conformance

10.1 Applicability

This standard shall be used in conjunction with IEEE Std 11073-20601-2014.

An implementation or a system can conform to the following elements of this standard:

- Domain information model class hierarchy and object definitions (object attributes, notifications, methods, and data type definitions)
- Nomenclature code values
- Protocol and service models
- Communication service model (association and configuration)

10.2 Conformance specification

This standard offers levels of conformance with respect to strict adherence to the standard device and the use of extensions for:

- Information model of a specific device
- Use of attributes, value ranges, and access methods

A vendor shall specify the level of conformance for an implementation based on this standard and provide details of the way in which the definitions of this standard and any extensions are applied.

Specifications shall be provided in the form of a set of implementation conformance statements (ICS) as detailed in 10.4.

Because this standard is used in conjunction with IEEE Std 11073-20601-2014, the ICS should be created for this standard first. The ICS created for IEEE Std 11073-20601-2014 may then refer to the ICS for this standard where applicable.

10.3 Levels of conformance

10.3.1 General

This standard defines the following levels of conformance.

10.3.2 Conformance level 1: base conformance

The application uses elements of the information, service, and communication models (object hierarchy, actions, event reports, and data type definitions) and the nomenclature scheme defined in IEEE Std 11073-20601-2014 and ISO/IEEE 11073-104zz standards. All mandatory features defined in the object definition tables and in the ICS tables are implemented. Furthermore, any conditional, recommended, or optional features that are implemented shall follow the requirements in IEEE Std 11073-20601-2014 and ISO/IEEE 11073-104zz documents.

10.3.3 Conformance level 2: extended nomenclature (ASN.1 and/or ISO/IEEE 11073-10101:2004 [B2])

Conformance level 2 meets conformance level 1 but also uses or adds extensions in at least one of the information, service, communication, or nomenclature models. These extensions shall conform to nomenclature codes from ASN.1 and/or within the ISO/IEEE 11073-10101:2004 [B2] framework (0xF000 through 0xFFFF). These extensions should be defined in ICS tables pointing toward their reference.

10.4 Implementation conformance statements

10.4.1 General format

The ICS are provided as an overall conformance statement document that comprises a set of tables in the form given by the templates in the following clauses.

Each ICS table has the following columns:

Index	Feature	Reference	Req./Status	Support	Comment

The table column headings have the following meaning:

- Index: an identifier (e.g., a tag) of a specific feature.
- Feature: briefly describes the characteristic for which a conformance statement is being made.
- Reference: to the clause/paragraph within this document or to an external source for the definition of the feature (may be empty).
- Req./Status: specifies the conformance requirement (e.g., mandatory or recommended)—in some cases, this standard does not specify conformance requirements but requests the status of a particular feature be provided.
- Support: specifies the presence or absence of a feature and any description of the characteristics of the feature in the implementation. This column is to be filled out by the implementer.
- Comment: contains any additional information on the feature. This column is to be filled out by the implementer.

Subclauses 10.4.2 through 10.4.6 specify the format of the specific ICS tables.

10.4.2 General implementation conformance statement

The general ICS specify the versions/revisions that are supported by the implementation and high-level system behavior.

Table 25 shows the general ICS.

Table 25 —11073-10417 general ICS table

Index	Feature	Reference	Req./Status	Support	Comment
EN 11073-10417-1	Implementation description	—	Identification of the device/ application. Description of functionality.		
GEN 11073-10417-2	Standards followed and their revisions	(Standard documents)	(Set of existing revisions)	(Set of supported revision)	
GEN 11073-10417-3	Nomenclature document used and revision	(Standard documents)	(Set of existing revisions)	(Set of supported revisions)	
GEN 11073-10417-4	Conformance adherence—level 1	See 10.3.3	Base conformance declaration that device meets the following IEEE Std 11073-10417 conformance requirements: a) All mandatory requirements shall be implemented. b) If implemented, conditional, recommended, and optional requirements shall conform to standard.	Yes/No (No is not expected as No implies that the implementation is non-conformant)	
GEN 11073-10417-5	Conformance adherence—level 2	See 6.3	In addition to GEN 11073-10417-4, if the device implements extensions and/or additions, they shall conform to nomenclature codes from ASN.1 and/or 10101 framework. These extensions should also be defined in ICS tables pointing toward their reference.	Yes/No	
GEN 11073-10417-6	Object containment tree	See 6.3	Provide object containment diagram showing relations between object instances used by the application. A conforming implementation uses only object relations as defined in the DIM.		
GEN 11073-10417-7	Nomenclature document used and revision	(Standard documents)	(Set of existing revisions)	(Set of supported revision)	
GEN 11073-10417-8	Data structure encoding	—	—	Description of encoding method(s) for ASN.1 data structures	
GEN 11073-10417-9	Use of private objects	—	Does the implementation use objects that are not defined in the DIM?	Yes/No (If yes: explain in Table 26)	
GEN 11073-10417-10	Use of private nomenclature extensions	—	Does the implementation use private extensions to the nomenclature (i.e., 0xF000 through 0xFFFF codes from ISO/IEEE 11073-10101:2004 [B2])? Private Nomenclature extensions are *only* allowed if the standard nomenclature does not include the specific terms required by the application.	Yes/No (If yes: explain in Table 29)	
GEN 11073-10417-11	11073-20601 conformance		Provide the conformance report required by IEEE Std 11073-20601-2014.		

[a] The prefix GEN11073-10417- is used for the index in the general ICS table.

10.4.3 DIM MOC implementation conformance statement

The DIM MOC ICS define which objects are implemented. Information on each object shall be provided as a separate row in the template of Table 26.

Table 26 —Template for DIM MOC ICS table

Index	Feature	Reference	Req./Status	Support	Comment
MOC-*n*	Object description	Reference to the clause in the standard or other location where the object is defined.	Implemented	Specify restrictions, e.g., max. number of supported instances	

The *n* in the Index column should be the object handle for implementations that have predefined objects. Otherwise the Index column shall simply be a unique number (1..*m*).

All private objects should be specified and include either a reference to the definition for the object, or where no publicly available reference is available, the definition of the object should be appended to the conformance statement.

The Support column should indicate any restrictions for the object implementation.

An object containment diagram (class instance diagram) should be provided as part of the DIM MOC ICS.

10.4.4 MOC attribute ICS

For each supported object as defined in the DIM MOC ICS, a MOC attribute ICS has to be provided that defines which attributes are used/supported by the implementation, including any inherited attributes. Table 27 is a template only.

Table 27 —Template for MOC attribute ICS table

Index	Feature	Reference	Req./Status	Support	Comment
ATTR-*n*-*x*	Attribute name. Extended attributes shall include the Attribute ID also.	Fill in the reference to the ASN.1 structure if the attribute is not defined in this standard.	M = Mandatory C = Conditional R = Recommended O = Optional (as per definition in Attribute Definition Tables)	Implemented? Yes/No Static/Dynamic Specify restrictions (e.g., value ranges). Describe how attribute is accessed (e.g., Get, Set, sent in config event report, sent in a data event report). Describe any specific restrictions.	

All private attributes should be specified and include reference to the definition for the attribute. Where no publicly available reference is available, the definition of the attribute should be appended to the conformance statement.

The Support column shall specify whether the attribute is implemented; for extension attributes, whether the attribute value is static or dynamic; any value ranges; restrictions on attribute access or availability; and any other information.

The *n* in the Index column refers to the ID of the managed object for which the table is supplied (i.e., the index of the managed object as specified in the MOC ICS). There is one separate table for each supported managed object.

The *x* in the Index column is a unique serial number (1..*m*).

NOTE—The attribute definition tables in the standard define a minimum mandatory set of attributes for each object.

10.4.5 MOC notification implementation conformance statement

The MOC notification ICS specify all implemented notifications (typically in form of the event report service) that are emitted by the agent. Table 28 provides a template for use. One table has to be provided for each object that supports special object notifications.

Table 28 —Template for MOC notification ICS table

Index	Feature	Reference	Req./Status	Support	Comment
NOTI-*n*-*x*	Notification name and Notification ID	Reference to the clause in the standard or other location where the event is defined.		The Support column shall specify how the notification is sent and any restrictions.	

The *n* in the Index column refers to the ID of the managed object for which the table is supplied (i.e., the index of the managed object as specified in the POC ICS). There is one separate table for each managed object that supports specific object notifications (i.e., events).

The *x* in the Index column of Table 28 is a unique serial number (1..*m*).

All private notifications should be specified and include reference to the definition for the notification. Where no publicly available reference is available, the definition of the notification should be appended to the conformance statement.

10.4.6 MOC nomenclature conformance statement

The MOC nomenclature ICS specify all nonstandard nomenclature codes that are utilized by the agent. Table 29 provides a template for use. One row of the table is to be used for each nomenclature element.

Table 29 —Template for MOC nomenclature ICS table

Index	Feature	Reference	Req./Status	Support	Comment
NOME-*n*	Nomenclature name and nomenclature value	Reference to the clause in the standard or other location where the nomenclature is defined or used.		Describe how the nomenclature is used. Describe any specific restrictions.	

The *n* in the Index column is a unique serial number (1..*m*).

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] American Diabetes Association, “Standards of medical care in diabetes—2006,” *Diabetes Care*, vol. 29, suppl. 1, Jan. 2006.

[B2] ISO/IEEE 11073-10101:2004, Health informatics—Point-of-care medical device communication—Part 10101: Nomenclature.[9]

[B3] ISO/IEEE 11073-10201:2004, Health informatics—Point-of-care medical device communication—Part 10201: Domain information model.

[B4] ISO/IEEE 11073-20101:2004, Health informatics—Point-of-care medical device communication—Part 20101: Application profile—Base standard.

[B5] ITU-T Rec. X.680-2002, Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation.[10]

[9] ISO/IEEE publications are available from the ISO Central Secretariat, 1, ch. de la Voie-Creuse, Case postale 56, CH-1211, Geneva 20, Switzerland (http://www.iso.ch/). ISO/IEEE publications are also available in the United States from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854-4141, USA (http://standards.ieee.org/).

[10] ITU publications are available from the International Telecommunications Union, Place des Nations, 1211 Geneva 20, Switzerland (http://www.itu.in/).

Annex B

(normative)

Any additional ASN.1 definitions

B.1 Device and sensor status bit mapping

The threshold extensions to the enumeration class require the following ASN.1 structure definitions:

```
GlucoseDevStat::= BITS-16 {
                    device-battery-low(0),
                    sensor-malfunction(1),
                    sensor-sample-size-insufficient(2),
                    sensor-strip-insertion(3),
                    sensor-strip-type-incorrect(4),
                    sensor-result-too-high(5),
                    sensor-result-too-low(6),
                    sensor-temp-too-high(7),
                    sensor-temp-too-low(8),
                    sensor-read-interrupt(9),
                    device-gen-fault(10),
                    sensor-temp-out-of-range(11)
                      }
```

Annex C

(normative)

Allocation of identifiers

C.1 General

This annex contains the nomenclature codes used in this document and not found in IEEE Std 11073-20601-2014. For those not contained in this annex, the normative definition is found in IEEE Std 11073-20601-2014.

C.2 Definitions of terms and codes

The format used here follows that of ISO/IEEE 11073-10101:2004 [B2].

```
/*******************************************************************************
* From Medical supervisory control and data acquisition (MDC_PART_SCADA)
*******************************************************************************/
#define MDC_CONC_GLU_CAPILLARY_WHOLEBLOOD    29112/*                            */
#define MDC_CONC_GLU_CAPILLARY_PLASMA        29116/*                            */
#define MDC_CONC_GLU_VENOUS_WHOLEBLOOD       29120/*                            */
#define MDC_CONC_GLU_VENOUS_PLASMA           29124/*                            */
#define MDC_CONC_GLU_ARTERIAL_WHOLEBLOOD     29128/*                            */
#define MDC_CONC_GLU_ARTERIAL_PLASMA         29132/*                            */
#define MDC_CONC_GLU_UNDETERMINED_WHOLEBLOOD  29292/*                           */
#define MDC_CONC_GLU_UNDETERMINED_PLASMA     29296/*                            */
#define MDC_CONC_GLU_CONTROL                 29136/*                            */
#define MDC_CONC_GLU_CONTROL_LEVEL_LOW       29300/*                            */
#define MDC_CONC_GLU_CONTROL_LEVEL_MEDIUM    29304/*                            */
#define MDC_CONC_GLU_CONTROL_LEVEL_HIGH      29308/*                            */
#define MDC_CONC_GLU_ISF                     29140/*                            */
#define MDC_CONC_GLU_CONTROL_LEVEL_UNDETERMINED  29312/*                        */
#define MDC_CONC_HBA1C                       29148/*                            */

/*******************************************************************************
* From Personal Health Device Disease Management (MDC_PART_PHD_DM)
*******************************************************************************/
#define MDC_GLU_METER_DEV_STATUS             29144/*                            */
#define MDC_CTXT_GLU_EXERCISE                29152/*                            */
#define MDC_CTXT_GLU_CARB                    29156/*                            */
#define MDC_CTXT_GLU_CARB_UNDETERMINED       29157/*                            */
#define MDC_CTXT_GLU_CARB_OTHER              29158/*                            */
#define MDC_CTXT_GLU_CARB_NO_ENTRY           29159/*                            */
#define MDC_CTXT_GLU_CARB_BREAKFAST          29160/*                            */
#define MDC_CTXT_GLU_CARB_NO_INGESTION       29161/*                            */
#define MDC_CTXT_GLU_CARB_LUNCH              29164/*                            */
#define MDC_CTXT_GLU_CARB_DINNER             29168/*                            */
#define MDC_CTXT_GLU_CARB_SNACK              29172/*                            */
#define MDC_CTXT_GLU_CARB_DRINK              29176/*                            */
#define MDC_CTXT_GLU_CARB_SUPPER             29180/*                            */
```

```
#define MDC_CTXT_GLU_CARB_BRUNCH                       29184/*                                  */
#define MDC_CTXT_MEDICATION                            29188/*                                  */
#define MDC_CTXT_MEDICATION_RAPIDACTING                29192/*                                  */
#define MDC_CTXT_MEDICATION_SHORTACTING                29196/*                                  */
#define MDC_CTXT_MEDICATION_INTERMEDIATEACTING   29200/*                                        */
#define MDC_CTXT_MEDICATION_LONGACTING                 29204/*                                  */
#define MDC_CTXT_MEDICATION_PREMIX                     29208/*                                  */
#define MDC_CTXT_GLU_HEALTH                            29212/*                                  */
#define MDC_CTXT_GLU_HEALTH_MINOR                      29216/*                                  */
#define MDC_CTXT_GLU_HEALTH_MAJOR                      29220/*                                  */
#define MDC_CTXT_GLU_HEALTH_MENSES                     29224/*                                  */
#define MDC_CTXT_GLU_HEALTH_STRESS                     29228/*                                  */
#define MDC_CTXT_GLU_HEALTH_NONE                       29232/*                                  */
#define MDC_CTXT_GLU_SAMPLELOCATION                    29236/*                                  */
#define MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED   29237/*                                      */
#define MDC_CTXT_GLU_SAMPLELOCATION_OTHER      29238/*                                          */
#define MDC_CTXT_GLU_SAMPLELOCATION_FINGER     29240/*                                          */
#define MDC_CTXT_GLU_SAMPLELOCATION_AST        29244/*                                          */
#define MDC_CTXT_GLU_SAMPLELOCATION_EARLOBE 29248/*                                             */
#define MDC_CTXT_GLU_SAMPLELOCATION_CTRLSOLUTION   29252/*                                      */
#define MDC_CTXT_GLU_MEAL                              29256/*                                  */
#define MDC_CTXT_GLU_MEAL_PREPRANDIAL                  29260/*                                  */
#define MDC_CTXT_GLU_MEAL_BEDTIME                      29261/*                                  */
#define MDC_CTXT_GLU_MEAL_POSTPRANDIAL                 29264/*                                  */
#define MDC_CTXT_GLU_MEAL_FASTING                      29268/*                                  */
#define MDC_CTXT_GLU_MEAL_CASUAL                       29272/*                                  */
#define MDC_CTXT_GLU_TESTER                            29276/*                                  */
#define MDC_CTXT_GLU_TESTER_SELF                       29280/*                                  */
#define MDC_CTXT_GLU_TESTER_HCP                        29284/*                                  */
#define MDC_CTXT_GLU_TESTER_LAB                        29288/*                                  */

/*****************************************************************************************
* From Dimensions (MDC_PART_DIM)
*****************************************************************************************/
#define MDC_DIM_MILLI_L                          1618  /* mL                                    */
#define MDC_DIM_MILLI_G                          1746  /* mg                                    */
#define MDC_DIM_MILLI_G_PER_DL                   2130  /* mg dL-1                               */
#define MDC_DIM_MILLI_MOLE_PER_L                 4722  /* mmol L-1                              */
#define MDC_DIM_G                                1728  /* g                                     */
#define MDC_DIM_INTL_UNIT                        5472  /* International unit                    */
```

C.3 Systematic derivations of terms and codes

Systematic derivations of terms and codes are outlined in Table C.1

Table C.1—Systematic derivations of terms and codes

Systematic name	Common term	Acronym	Description/definition	Reference ID	Code
Concentration | Total, Glucose | CapillaryWholeBlood | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from capillary whole blood	MDC_CONC_GLU_ CAPILLARY_WHOLEBLOOD	29112
Concentration | Total, Glucose | CapillaryPlasma | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from capillary blood plasma	MDC_CONC_GLU_ CAPILLARY_PLASMA	29116
Concentration | Total, Glucose | VenousWhole Blood | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from venous whole blood	MDC_CONC_GLU_VENOUS_ WHOLEBLOOD	29120
Concentration | Total, Glucose | VenousPlasma | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from venous blood plasma	MDC_CONC_GLU_VENOUS_ PLASMA	29124
Concentration | Total, Glucose | ArterialWhole Blood | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from arterial whole blood	MDC_CONC_GLU_ARTERIAL_ WHOLEBLOOD	29128
Concentration | Total, Glucose | ArterialPlasma | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from arterial blood plasma	MDC_CONC_GLU_ARTERIAL_ PLASMA	29132
Concentration | Total, Glucose | Undetermined WholeBlood | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from whole blood	MDC_CONC_GLU_ UNDETERMINED_ WHOLEBLOOD	29292
Concentration | Total, Glucose | Undetermined Plasma | FluidChemistry	Blood glucose result		Object containing glucose measurement produced from blood plasma	MDC_CONC_GLU_ UNDETERMINED_PLASMA	29296
Concentration | Glucose | ControlSolution | FluidChemistry	Control result		Object containing measurement produced from control solution	MDC_CONC_GLU_ CONTROL	29136
Concentration | Glucose | ControlSolution | FluidChemistry	Control level: low		Object containing control solution level (range)	MDC_CONC_GLU_CONTROL_ LEVEL_LOW	29300
Concentration | Glucose | ControlSolution | FluidChemistry	Control level: medium		Object containing control solution level (range)	MDC_CONC_GLU_CONTROL_ LEVEL_MEDIUM	29304
Concentration | Glucose | ControlSolution | FluidChemistry	Control level: high		Object containing control solution level (range)	MDC_CONC_GLU_CONTROL_ LEVEL_HIGH	29308
Concentration | Glucose | InterstitialFluid | FluidChemistry	Interstitial fluid glucose		Object containing glucose measurement obtained from interstitial fluid	MDC_CONC_GLU_ISF	29140
Concentration | Glucose | ControlSolution | FluidChemistry	Control level: undetermined		Object containing control solution level (range)	MDC_CONC_GLU_CONTROL_ LEVEL_UNDETERMINED	29312
Concentration | HbA1c | FluidChemistry	HbA1c		Object containing A1c or glycated hemoglobin	MDC_CONC_HBA1C	29148
Status | value | FunctionalStatus | Device	Device status		Object containing glucose device specific status flags	MDC_GLU_METER_DEV_ STATUS	29144
Glucose | Context | Exercise	Exercise		Object containing contextual information related to the effect of exercise on glucose level	MDC_CTXT_GLU_EXERCISE	29152

Table C.2—Systematic derivations of terms and codes *(continued)*

Systematic name	Common term	Acronym	Description/definition	Reference ID	Code
Glucose | Context | Carb	Carbohydrates		Object containing contextual information related to the effect of carbohydrates on glucose level	MDC_CTXT_GLU_CARB	29156
Medication | Context	Medication		Object containing contextual information related to the effect of medication on glucose level	MDC_CTXT_MEDICATION	29188
Glucose | Context | Health	Level of health		Object containing contextual information related to the effect of health on glucose level	MDC_CTXT_GLU_HEALTH	29212
Glucose | Context | SampleLocation	Sample location		Object specifying location of test performed	MDC_CTXT_GLU_ SAMPLELOCATION	29236
Glucose | Context |Samplelocation | Alternativesitetest	Alternative site test of sample location		Object specifying that the location of test performed was from an alternative site on the body	MDC_CTXT_GLU_ SAMPLELOCATION_AST	29244
Glucose | Context | Meal	Meal relationship		Object containing contextual information related to the effect of food intake on glucose level	MDC_CTXT_GLU_MEAL	29256
Glucose | Context | Meal | BeforeMeal	Pre-meal relationship		Object containing contextual information related to the effect of between-meal food intake on glucose level	MDC_CTXT_GLU_MEAL_ PREPRANDIAL	29260
Glucose | Context | Meal | AfterMeal	Post-meal relationship		Object containing contextual information related to the effect of diet on glucose level	MDC_CTXT_GLU_MEAL_ POSTPRANDIAL	29264
Glucose | Context | Meal | Fasting	Fasting relationship		Object containing contextual information related to the effect of long-term absence of food intake (overnight) on glucose level	MDC_CTXT_GLU_MEAL_ FASTING	29268
Glucose | Context | Meal | Bedtime	Bedtime relationship		Object containing contextual information related to the effect of evening food intake on glucose level	MDC_CTXT_GLU_MEAL_ BEDTIME	29261
Glucose | Context | Tester	Tester		Object containing contextual information related to a third party conducting the test	MDC_CTXT_GLU_TESTER	29276
Glucose | Context | Tester | HealthCare Professional	Health care professional tester		Object containing contextual information related to a health care professional conducting the test	MDC_CTXT_GLU_TESTER_ HCP	29284

Annex D

(informative)

Message sequence examples

Figure D.1 shows a sequence diagram of the messaging procedure corresponding to the following use case. The user of a glucose meter agent device intends to connect it to a manager for the first time. The glucose meter is capable of performing glucose measurements. Thus, it operates as an extended configuration.

a) When the user connects the glucose meter, the manager does not yet know the agent's configuration and sends a response to the agent's association request with the result *accepted-unknown-config*. See E.2.2.2 and E.2.2.3 for the corresponding PDU examples.

b) As a consequence of this, the agent negotiates its configuration information to the manager. After getting confirmation from the manager accepting the agent's configuration, the agent device is ready to send measurements. Both devices enter the Operating state. See E.3.2.2 and E.3.2.3 for the corresponding PDU examples.

c) Subsequently, the manager may request the MDS object attributes of the agent by sending a data message with the "Remote Operation Invoke | Get" command. As a response, the agent reports its MDS object attributes to the manager using a data message with the "Remote Operation Response | Get" command. See E.4.1.2 and E.4.1.3 for the corresponding PDU examples.

d) If the measurement was not taken prior to association, as a next step, the user of the agent device takes a single measurement. The measurement data are transmitted to the manager using a confirmed event report. After having successfully received the measurement data, the manager sends a confirmation to the agent. See E.5.1 and E.5.2 for the corresponding PDU examples.

e) The user ends the measurement session (e.g., by pushing a proper button on the device, or just by not using the device for a duration longer than a certain time period). As a consequence, the agent disassociates from the manager by sending an association release request. The manager responds with an association release response. See E.6.1 and E.6.2 for the corresponding PDU examples.

f) When the agent requests to associate to the manager for the next measurement session (e.g., the next day), the result in the manager's response is *accepted*, as it already knows the agent's configuration from the previous measurement session. Both devices transition directly to the Operating state.

g) Finally, the last two steps shown are similar as in item d) and item e). The user takes a single confirmed measurement followed by releasing the association.

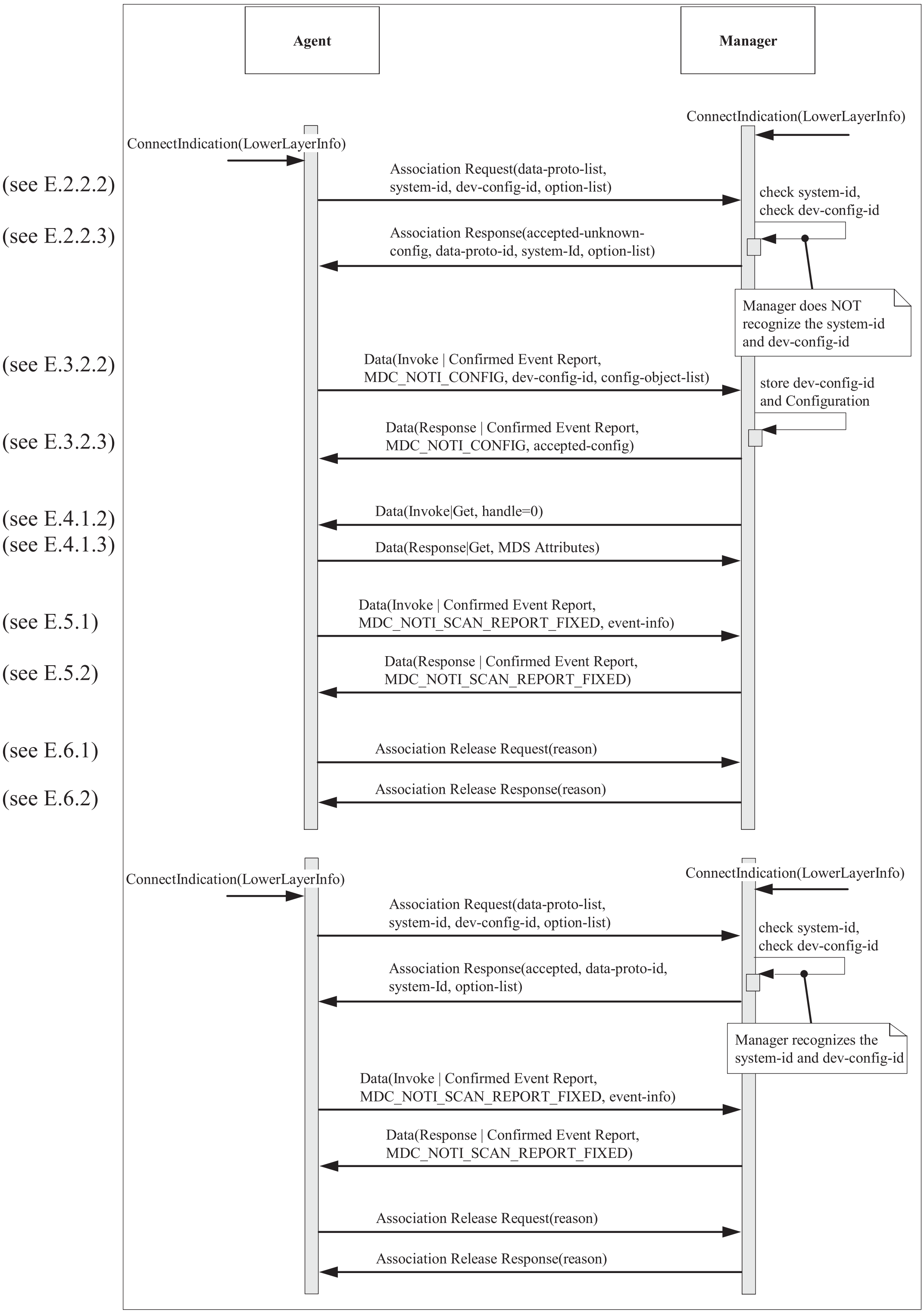
Agent
Manager
ConnectIndication(LowerLayerInfo)
ConnectIndication(LowerLayerInfo)
(see E.2.2.2)
Association Request(data-proto-list, system-id, dev-config-id, option-list)
check system-id, check dev-config-id
(see E.2.2.3)
Association Response(accepted-unknown-config, data-proto-id, system-Id, option-list)
Manager does NOT recognize the system-id and dev-config-id
(see E.3.2.2)
Data(Invoke | Confirmed Event Report, MDC_NOTI_CONFIG, dev-config-id, config-object-list)
store dev-config-id and Configuration
(see E.3.2.3)
Data(Response | Confirmed Event Report, MDC_NOTI_CONFIG, accepted-config)
(see E.4.1.2)
Data(Invoke|Get, handle=0)
(see E.4.1.3)
Data(Response|Get, MDS Attributes)
(see E.5.1)
Data(Invoke | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED, event-info)
(see E.5.2)
Data(Response | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED)
(see E.6.1)
Association Release Request(reason)
(see E.6.2)
Association Release Response(reason)
ConnectIndication(LowerLayerInfo)
ConnectIndication(LowerLayerInfo)
Association Request(data-proto-list, system-id, dev-config-id, option-list)
check system-id, check dev-config-id
Association Response(accepted, data-proto-id, system-Id, option-list)
Manager recognizes the system-id and dev-config-id
Data(Invoke | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED, event-info)
Data(Response | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED)
Association Release Request(reason)
Association Release Response(reason)

Figure D.1 —Sequence diagram for glucose meter example use case

Annex E

(informative)

Protocol data unit examples

E.1 General

This annex shows binary examples of messages exchanged between a glucose meter agent and a manager. Three different scenarios containing the association and configuration information exchanges are presented in E.2.2, E.2.3, and E.2.4. The first scenario illustrates the case when the agent intends to operate using an extended configuration. The manager does not have the configuration declared by the agent from a prior association. The second illustrates the agent presenting the same extended configuration to the manager, and the manager does have the configuration from the previously transferred configuration exchange. Finally, the agent presents a standard configuration to the manager, and the manager has the configuration because the manager has been preprogrammed with this configuration.

E.2 Association information exchange

E.2.1 General

When the transport connection is established between the manager and the agent, they both enter the Unassociated state. When the agent sends an association request, both manager and agent enter the Associating state.

E.2.2 Extended configuration

E.2.2.1 General

In this exchange, the agent sends an association request intending to use an extended configuration during measurement transfer. However, the manager does not have this configuration.

E.2.2.2 Association request

The glucose meter agent sends the following message to the manager. The agent intends to associate using an extended configuration.

```
0xE2 0x00                APDU CHOICE Type (AarqApdu)
0x00 0x32                CHOICE.length = 50
0x80 0x00 0x00 0x00      assoc-version
0x00 0x01 0x00 0x2A      data-proto-list.count = 1 | length = 42
0x50 0x79                data-proto-id = 20601
0x00 0x26                data-proto-info length = 38
0x80 0x00 0x00 0x02      protocolVersion
0xA0 0x00                encoding rules = MDER or PER
0x80 0x00 0x00 0x00      nomenclatureVersion
0x00 0x00 0x00 0x00      functionalUnits – no Test Association capabilities
```

```
0x00 0x80 0x00 0x00     systemType = sys-type-agent
0x00 0x08               system-id length = 8 and value (manufacturer- and device- specific)
0x11 0x22 0x33 0x44
0x55 0x66 0x77 0x88
0x40 0x00               dev-config-id – extended configuration
0x00 0x01               data-req-mode-flags
0x01 0x00               data-req-init-agent-count, data-req-init-manager-count
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

E.2.2.3 Association response

A manager responds to the agent that it can associate but does not have the glucose meter extended configuration (i.e., there is the need for the agent to send its configuration).

```
0xE3 0x00               APDU CHOICE Type (AareApdu)
0x00 0x2C               CHOICE.length = 44
0x00 0x03               result = accepted-unknown-config
0x50 0x79               data-proto-id = 20601
0x00 0x26               data-proto-info length = 38
0x80 0x00 0x00 0x02     protocolVersion
0x80 0x00               encoding rules = MDER
0x80 0x00 0x00 0x00     nomenclatureVersion
0x00 0x00 0x00 0x00     functionalUnits – normal Association
0x80 0x00 0x00 0x00     systemType = sys-type-manager
0x00 0x08               system-id length = 8 and value (manufacturer- and device- specific)
0x88 0x77 0x66 0x55
0x44 0x33 0x22 0x11
0x00 0x00               Manager’s response to config-id is always 0
0x00 0x00               Manager’s response to data-req-mode-flags is always 0
0x00 0x00               data-req-init-agent-count and data-req-init-manager-count are always 0
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

E.2.3 Previously known extended configuration

E.2.3.1 General

This exchange illustrates a transaction that takes place after a session beginning with an exchange like E.2.2 has occurred.

E.2.3.2 Association request

The glucose meter agent sends the following message to the manager. The agent intends to associate using an extended configuration.

```
0xE2 0x00               APDU CHOICE Type (AarqApdu)
0x00 0x32               CHOICE.length = 50
0x80 0x00 0x00 0x00     assoc-version
0x00 0x01 0x00 0x2A     data-proto-list.count = 1 | length = 42
0x50 0x79               data-proto-id = 20601
```

```
0x00 0x26                data-proto-info length = 38
0x80 0x00 0x00 0x01      protocolVersion
0xA0 0x00                encoding rules = MDER or PER
0x80 0x00 0x00 0x00      nomenclatureVersion
0x00 0x00 0x00 0x00      functionalUnits, no Test Association capabilities
0x00 0x80 0x00 0x00      systemType = sys-type-agent
0x00 0x08                system-id length = 8 and value (manufacturer- and device-specific)
0x11 0x22 0x33 0x44
0x55 0x66 0x77 0x88
0x40 0x00                dev-config-id – extended configuration
0x00 0x01                data-req-mode-flags
0x01 0x00                data-req-init-agent-count, data-req-init-manager-count
0x00 0x00 0x00 0x00      optionList.count = 0 | optionList.length = 0
```

E.2.3.3 Association response

A manager responds to the agent that it can associate with, recognizes, accepts, and has the glucose meter's extended configuration (i.e., there is no need for the agent to send its configuration).

```
0xE3 0x00                APDU CHOICE Type (AareApdu)
0x00 0x2C                CHOICE.length = 44
0x00 0x00                result = accepted
0x50 0x79                data-proto-id = 20601
0x00 0x26                data-proto-info length = 38
0x80 0x00 0x00 0x01      protocolVersion
0x80 0x00                encoding rules = MDER
0x80 0x00 0x00 0x00      nomenclatureVersion
0x00 0x00 0x00 0x00      functionalUnits – normal Association
0x80 0x00 0x00 0x00      systemType = sys-type-manager
0x00 0x08                system-id length = 8 and value (manufacturer- and device-specific)
0x88 0x77 0x66 0x55
0x44 0x33 0x22 0x11
0x00 0x00                Manager's response to config-id is always 0
0x00 0x00                Manager's response to data-req-mode-flags is always 0
0x00 0x00                data-req-init-agent-count and data-req-init-manager-count are always 0
0x00 0x00 0x00 0x00      optionList.count = 0 | optionList.length = 0
```

E.2.4 Standard configuration

E.2.4.1 General

This transaction would occur if an agent presents an association request incorporating the dev-config-id corresponding to a standard configuration. The manager has the configuration because it has been programmed with this configuration according to the information presented in this standard.

E.2.4.2 Association request

The glucose meter agent sends the following message to the manager. The agent intends to associate using a standard configuration. The agent is willing to enter into a test association as defined in Clause 10.

```
0xE2 0x00               APDU CHOICE Type (AarqApdu)
0x00 0x32               CHOICE.length = 50
0x80 0x00 0x00 0x00     assoc-version
0x00 0x01 0x00 0x2A     data-proto-list.count = 1 | length = 42
0x50 0x79               data-proto-id = 20601
0x00 0x26               data-proto-info length = 38
0x80 0x00 0x00 0x00     protocolVersion
0x80 0x00               encoding rules = MDER or PER
0x80 0x00 0x00 0x00     nomenclatureVersion
0x40 0x00 0x00 0x00     functionalUnits, has Test Association capabilities
0x00 0x80 0x00 0x00     systemType = sys-type-agent
0x00 0x08               system-id length = 8 and value (manufacturer- and device-specific)
0x11 0x22 0x33 0x44
0x55 0x66 0x77 0x88
0x06 0xA4               dev-config-id – standard configuration
0x00 0x01               data-req-mode-flags
0x01 0x00               data-req-init-agent-count, data-req-manager-count
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

E.2.4.3 Association response

A manager responds to the agent that it can associate with, recognizes, accepts, and has the glucose meter standard configuration (i.e., there is no need for the agent to send its configuration). The manager does not start a test association.

```
0xE3 0x00               APDU CHOICE Type (AareApdu)
0x00 0x2C               CHOICE.length = 44
0x00 0x00               result = accepted
0x50 0x79               data-proto-id = 20601
0x00 0x26               data-proto-info length = 38
0x80 0x00 0x00 0x00     protocolVersion
0x80 0x00               encoding rules = MDER
0x80 0x00 0x00 0x00     nomenclatureVersion
0x00 0x00 0x00 0x00     functionalUnits, normal Association
0x80 0x00 0x00 0x00     systemType = sys-type-manager
0x00 0x08               system-id length = 8 and value (manufacturer- and device-specific)
0x88 0x77 0x66 0x55
0x44 0x33 0x22 0x11
0x00 0x00               Manager’s response to config-id is always 0
0x00 0x00               Manager’s response to data-req-mode-flags is always 0
0x00 0x00               data-req-init-agent-count and data-req-init-manager-count are always 0
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

E.3 Configuration information exchange

E.3.1 General

If the association is not rejected or aborted, the agent and manager transition from the Associating state into one of two states. If the manager’s AssociateResult code is accepted, the agent and manager enter the Operating state. If the manager’s AssociateResult code is accepted-unknown-config, the agent and manager enter the Configuring state.

E.3.2 Extended configuration

E.3.2.1 General

This exchange takes place when the manager returns the AssociateResult code of accepted-unknown-config. The agent presents a description of its configuration corresponding to the dev-config-id it presented in the association request.

E.3.2.2 Remote operation invoke event report configuration

The glucose meter agent sends the description of its extended configuration. It does this by sending a confirmed event report of type MDC_NOTI_CONFIG.

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x9A                CHOICE.length = 154
0x00 0x98                OCTET STRING.length = 152
0x43 0x21                invoke-id = 0x4321 (start of DataApdu. MDER encoded.)
0x01 0x01                CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x92                CHOICE.length = 146
0x00 0x00                obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF      event-time = 0xFFFFFFFF
0x0D 0x1C                event-type = MDC_NOTI_CONFIG
0x00 0x88                event-info.length = 136 (start of ConfigReport)
0x40 0x00                config-report-id = Extended
0x00 0x03                config-obj-list.count = 3 Measurement objects will be “announced”
0x00 0x82                config-obj-list.length = 130
0x00 0x06                obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01                obj-handle = 1     (→ 1st Measurement is blood glucose)
0x00 0x04                attributes.count = 4
0x00 0x24                attributes.length = 36
0x09 0x2F                attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04                attribute-value.length = 4
0x00 0x02                MDC_PART_SCADA |
0x71 0xB8                MDC_CONC_GLU_CAPILLARY_WHOLEBLOOD
0x0A 0x46                attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02                attribute-value.length = 2
0xF0 0x40                 intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x09 0x96                attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02                attribute-value.length = 2
0x12 0x72                MDC_DIM_MILLI_MOLE_PER_L
0x0A 0x55                attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C                attribute-value.length = 12
0x00 0x02                AttrValMap.count = 2
0x00 0x08                AttrValMap.length = 8
0x0A 0x4C 0x00 0x02      MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x0A 0x82 0x00 0x08      MDC_ATTR_TIME_STAMP_BO | value length = 8
0x00 0x05                obj-class = MDC_MOC_VMO_METRIC_ENUM
0x00 0x02                obj-handle = 2     (→2nd Measurement is context meal)
0x00 0x03                attributes.count = 3
0x00 0x1E                attributes.length = 30
0x09 0x2F                attribute-id = MDC_ATTR_ID_TYPE
```

```
0x00 0x04                 attribute-value.length = 4
0x00 0x80 0x72 0x48       MDC_PART_PHD_DM | MDC_CTXT_GLU_MEAL
0x0A 0x46                 attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02                 attribute-value.length = 2
0xF0 0x48                 intermittent, stored data, upd & msmt aperiodic, agent init, manual
0x0A 0x55                 attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C                 attribute-value.length = 12
0x00 0x02                 AttrValMap.count = 2
0x00 0x08                 AttrValMap.length = 8
0x0A 0x82 0x00 0x08       MDC_ATTR_TIME_STAMP_BO, 8
0x0A 0x49 0x00 0x02       MDC_ATTR_ENUM_OBS_VAL_SIMP_OID, 2
0x00 0x06                 obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x03                 obj-handle = 3     (→3rd Measurement is context exercise)
0x00 0x04                 attributes.count = 4
0x00 0x28                 attributes.length = 40
0x09 0x2F                 attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04                 attribute-value.length = 4
0x00 0x80 0x71 0xE0       MDC_PART_PHD_DM | MDC_CTXT_GLU_EXERCISE
0x0A 0x46                 attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02                 attribute-value.length = 2
0xF0 0x48                 intermittent, stored data, upd & msmt aperiodic, agent init, manual
0x09 0x96                 attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02                 attribute-value.length = 2
0x02 0x20                 MDC_DIM_PERCENT
0x0A 0x55                 attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x10                 attribute-value.length = 16
0x00 0x03                 AttrValMap.count = 3
0x00 0x0C                 AttrValMap.length = 12
0x0A 0x82 0x00 0x08       MDC_ATTR_TIME_STAMP_BO, 8
0x0A 0x59 0x00 0x04       MDC_ATTR_TIME_PD_MSMT_ACTIVE, 4
0x0A 0x4C 0x00 0x02       MDC_ATTR_NU_VAL_OBS_SIMP, 2
```

E.3.2.3 Remote operation response event report configuration

The manager responds that it can utilize the agent’s configuration. The manager does by sending the confirmed event report response with a config-result of accepted-config.

```
0xE7 0x00                 APDU CHOICE Type (PrstApdu)
0x00 0x16                 CHOICE.length = 22
0x00 0x14                 OCTET STRING.length = 20
0x43 0x21                 invoke-id = 0x4321 (mirrored from invocation)
0x02 0x01                 CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E                 CHOICE.length = 14
0x00 0x00                 obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00       currentTime = 0
0x0D 0x1C                 event-type = MDC_NOTI_CONFIG
0x00 0x04                 event-reply-info.length = 4
0x40 0x00                 ConfigReportRsp.config-report-id = 0x4000
0x00 0x00                 ConfigReportRsp.config-result = accepted-config
```

E.3.3 Known configuration

E.3.3.1 General

This exchange takes place when the manager returns the AssociateResult code of accepted because the manager had previously received and processed the configuration corresponding to the dev-config-id sent by the agent. In this case, there is no exchange of configuration information, and the manager and agent have moved into the Operating state.

E.3.3.2 Remote operation invoke event report configuration

Since the manager was already aware of the agent's configuration, the Configuring state is skipped, and no event report invocation is generated by the agent.

E.3.3.3 Remote operation response event report configuration

The Configuring state has been skipped. No event report invocation is generated by the agent, so the manager does not generate any response.

E.3.4 Standard configuration

E.3.4.1 General

This exchange takes place when the manager returns the AssociateResult code of accepted because the manager had previously been programmed with the documented standard configuration corresponding to the dev-config-id sent by the agent. In this case, there is no exchange of configuration information, and the manager and agent have moved into the Operating state.

E.3.4.2 Remote operation invoke event report configuration

Since the manager had been programmed with the agent's configuration, the Configuring state is skipped, and no event report invocation is generated by the agent.

E.3.4.3 Remote operation response event report configuration

The Configuring state has been skipped. No event report invocation is generated by the agent, so the manager does not generate any response.

E.4 GET MDS attributes service

E.4.1.1 General

The GET MDS attributes is invoked at any time, when a device is in Associated state.

E.4.1.2 Get all medical device system attributes request

The manager queries the agent for its MDS object attributes.

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x0E                CHOICE.length = 14
0x00 0x0C                OCTET STRING.length = 12
0x00 0x03                invoke-id = 0x0003 (differentiates this message from any other
                         outstanding, choice is implementation specific)
0x01 0x03                CHOICE (Remote Operation Invoke | Get)
0x00 0x06                CHOICE.length = 6
0x00 0x00                handle = 0 (MDS object)
0x00 0x00                attribute-id-list.count = 0 (all attributes)
0x00 0x00                attribute-id-list.length = 0
```

E.4.1.3 Get response with all MDS attributes

The glucose meter agent responds to the manager with its attributes. Furthermore, some optional fields are communicated as well.

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x6E                CHOICE.length = 110
0x00 0x6C                OCTET STRING.length = 108
0x00 0x03                invoke-id = 0x0003 (mirrored from request)
0x02 0x03                CHOICE (Remote Operation Response | Get)
0x00 0x66                CHOICE.length = 102
0x00 0x00                handle = 0 (MDS object)
0x00 0x06                attribute-list.count = 6
0x00 0x60                attribute-list.length = 96
0x0A 0x5A                attribute id = MDC_ATTR_SYS_TYPE_SPEC_LIST
0x00 0x08                attribute-value.length = 8
0x00 0x01                TypeVerList count = 1
0x00 0x04                TypeVerList length = 4
0x10 0x11                type = MDC_DEV_SPEC_PROFILE_GLUCOSE
0x00 0x01                version = version 1 of the specialization
0x09 0x28                attribute-id = MDC_ATTR_ID_MODEL
0x00 0x1A                attribute-value.length = 26
0x00 0x0A 0x54 0x68      string length = 10 | “TheCompany”
0x65 0x43 0x6F 0x6D
0x70 0x61 0x6E 0x79
0x00 0x0C 0x47 0x6C      string length = 12 | “GlucoseMeter”
0x75 0x63 0x6F 0x73
0x65 0x4D 0x65 0x74 0x65 0x72
0x09 0x84                attribute-id = MDC_ATTR_SYS_ID
0x00 0x0A                attribute-value.length = 10
0x00 0x08 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 octet string length = 8 | EUI-64
0x0a 0x44                attribute-id = MDC_ATTR_DEV_CONFIG_ID
0x00 0x02                attribute-value.length = 2
0x40 0x00                dev-config-id = 16384 (extended-config-start)
0x09 0x2D                attribute-id = MDC_ATTR_ID_PROD_SPECN
0x00 0x12                attribute-value.length = 18
0x00 0x01                ProductionSpec.count = 1
0x00 0x0E                ProductionSpec.length = 14
```

```
0x00 0x01                 ProdSpecEntry.spec-type = 1 (serial-number)
0x00 0x00                 ProdSpecEntry.component-id = 0
0x00 0x08 0x44 0x45       string length = 8 | prodSpecEntry.prod-spec = “DE124567”
0x31 0x32 0x34 0x35
0x36 0x37
0x09 0x87                 attribute-id =MDC_ATTR_TIME_ABS
0x00 0x08                 attribute-value.length = 8
0x20 0x07 0x02 0x01       Absolute-Time-Stamp = 2007-02-01T12:05:0000
0x12 0x05 0x00 0x00
```

E.5 Data reporting

E.5.1 Confirmed measurement data transmission

The agent sends a spontaneous event report to the manager with measurement observations.

```
0xE7 0x00                 APDU CHOICE Type (PrstApdu)
0x00 0x64                 CHOICE.length = 100
0x00 0x62                 OCTET STRING.length = 98
0x12 0x36                 invoke-id = 0x1236
0x01 0x01                 CHOICE (Remote Operation Invoke | Confirmed Event Report)
0x00 0x5C                 CHOICE.length = 92
0x00 0x00                 obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF       event-time = 0xFFFFFFFF
0x0D 0x1D                 event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x52                 event-info.length = 82
0xF0 0x00                 ScanReportInfoFixed.data-req-id = 0xF000
0x00 0x00                 ScanReportInfoFixed.scan-report-no = 0
0x00 0x05                 ScanReportInfoFixed.obs-scan-fixed.count = 5
0x00 0x4A                 ScanReportInfoFixed.obs-scan-fixed.length = 74
0x00 0x01                 ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00 0x0A                 ScanReportInfoFixed.obs-scan-fixed.value[0]. obs-val-data.length = 10
0xF3 0x52                 Basic-Nu-Observed-Value = 85.0 (mmol/L)
0x20 0x08 0x09 0x19       Absolute-Time-Stamp = 2008-09-19T17:30:0000
0x17 0x30 0x00 0x00
0x00 0x01                 ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 1
0x00 0x0A                 ScanReportInfoFixed.obs-scan-fixed.value[1]. obs-val-data.length = 10
0xF4 0x4C                 Basic-Nu-Observed-Value = 110.0 (mmol/L)
0x20 0x08 0x09 0x19       Absolute-Time-Stamp = 2008-09-19T19:00:0000
0x19 0x00 0x00 0x00
0x00 0x02                 ScanReportInfoFixed.obs-scan-fixed.value[2].obj-handle = 2
0x00 0x0A                 ScanReportInfoFixed.obs-scan-fixed.value[2]. obs-val-data.length = 10
0x20 0x08 0x09 0x19       Absolute-Time-Stamp = 2008-09-19T17:30:0000
0x17 0x30 0x00 0x00
0x72 0x4C                 OID=MDC_CTXT_GLU_MEAL_PREPRANDIAL
0x00 0x02                 ScanReportInfoFixed.obs-scan-fixed.value[3].obj-handle = 2
0x00 0x0A                 ScanReportInfoFixed.obs-scan-fixed.value[3]. obs-val-data.length = 10
0x20 0x08 0x09 0x19       Absolute-Time-Stamp = 2008-09-19T19:00:0000
0x19 0x00 0x00 0x00
0x72 0x50                 OID= MDC_CTXT_GLU_MEAL_POSTPRANDIAL
0x00 0x03                 ScanReportInfoFixed.obs-scan-fixed.value[4].obj-handle = 3
```

0x00 0x0E	ScanReportInfoFixed.obs-scan-fixed.value[4]. obs-val-data.length = 14
0x20 0x08 0x09 0x19	Absolute-Time-Stamp = 2008-09-19T16:15:0000
0x16 0x15 0x00 0x00	
0x01 0x2B 0xF2 0x00	Measure-Active-Period=1 hour (28800000 counts of 1/8 milliseconds)
0xF3 0x84	Basic-Nu-Observed-Value = 90 (%)

E.5.2 Response to confirmed measurement data transmission

The manager confirms receipt of the agent's event report.

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x12	CHOICE.length = 18
0x00 0x10	OCTET STRING.length = 16
0x12 0x36	invoke-id = 0x1236 (mirrored from invocation)
0x02 0x01	CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0A	CHOICE.length = 10
0x00 0x00	obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00	currentTime = 0
0x0D 0x1D	event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x00	event-reply-info.length = 0

E.6 Disassociation

E.6.1 Association release request

When the agent is finished, it sends the following message to the manager to release the association.

0xE4 0x00	APDU CHOICE Type (RlrqApdu)
0x00 0x02	CHOICE.length = 2
0x00 0x00	reason = normal

E.6.2 Association release response

A manager responds to the agent that it can release association.

0xE5 0x00	APDU CHOICE Type (RlreApdu)
0x00 0x02	CHOICE.length = 2
0x00 0x00	reason = normal

ICS 35.240.80

Price based on 62 pages