



International
Standard

ISO/IEEE
11073-10425

**Health informatics — Device
interoperability —**

**Part 10425:
Personal health device
communication — Device
specialization — Continuous
Glucose Monitor (CGM)**

*Informatique de santé — Interopérabilité des dispositifs —
Partie 10425: Communication entre dispositifs de santé personnels
— Spécialisation des dispositifs — Glucomètre continu (CGM)*

**Third edition
2024-09**

STANDARDSISO.COM : Click to view the full PDF of ISO/IEEE 11073-10425:2024



COPYRIGHT PROTECTED DOCUMENT

© IEEE 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from IEEE at the address below.

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Foreword

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

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted (see www.iso.org/directives).

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.

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

ISO/IEEE 11073-10425 was prepared by the IEEE 11073 Standards Committee of the IEEE Engineering in Medicine and Biology Society (as IEEE Std 11073-10425) and drafted in accordance with its editorial rules. It was adopted, under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Technical Committee ISO/TC 215, *Health informatics*.

This third edition cancels and replaces the second edition (ISO/IEEE 11073-10425:2019), which has been technically revised.

The main changes are as follows:

- updated Normative Reference to refer to IEEE Std 11073-20601-2019;
- updated version of this device specialization;
- updated the association details based on new version;
- updated the wording in 6.3 regarding the Observational;
- added some text to 6.12 to further elaborate the DIM extensibility rule;
- corrected the use condition of GET MDS at E.4.1;

- updated the text in 8.5.2 regarding attribute-id-list, in order to be compliant with 20601-V4;
- added 4.3, Compliance with other standards;
- removed the year in bibliography to represent the latest version;
- updated the bit example in E.4.3 by inserting the Mds-Time-Info into MDS;
- made the ISO/IEEE 11073-10101 as normative reference;
- updated the wording at 1.3 and 4.1 regarding the precedence of nomenclature between 10101, 20601, 104xx and this standard;
- updated the usage of nomenclature-version. Tied it with the corresponding protocol-version;
- updated the examples in Annex E using protocol-version4.

A list of all parts in the ISO 11073 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEEE 11073-10425:2024

Abstract: Within the context of the ISO/IEEE 11073 family of standards for device communication, a normative definition of the communication between continuous glucose monitor (CGM) devices and managers (e.g., cell phones, personal computers, personal health appliances, set top boxes), in a manner that enables plug-and-play interoperability, is established in this standard. It leverages appropriate portions of existing standards including ISO/IEEE 11073 terminology and information models. 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 of CGM devices. In this context, CGM refers to the measurement of the level of glucose in the body on a regular (typically 5 minute) basis through a sensor continuously attached to the person.

Keywords: continuous glucose monitor, IEEE 11073-10425™, medical device communication, personal health devices

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

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 28 July 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9836-4 STD26255
Print: ISBN 978-1-5044-9837-1 STDPD26255

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, 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 makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. 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: THE NEED TO PROCURE 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 is 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, nor 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 the presenter’s views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, 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 evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#). An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.

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 constitute 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.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US 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 licensing fees, 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; <https://www.copyright.com/>. 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 10 years. When a document is more than 10 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 [IEEE Xplore](#) or [contact IEEE](#). For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#). Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).

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 <https://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.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users 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.

Participants

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

Daidi Zhong, *Chair*
Malcolm Clarke, *Vice-Chair*
Raymond Krasinski, *Secretary*

Karsten Aalders	Chia-Chin Chong	Raul Gonzalez Gomez
Charles R. Abbruscato	Jinhan Chung	Chris Gough
Nabil Abujbara	John A. Cogan	Channa Gowda
Maher Abuzaïd	John T. Collins	Charles M. Gropper
James Agnew	Cory Condek	Amit Gupta
Manfred Aigner	Todd H. Cooper	Jeff Guttmacher
Jorge Alberola	Sandra Costanzo	Rasmus Haahr
David Aparisi	Douglas Coup	Christian Habermann
Lawrence Arne	Nigel Cox	Michael Hagerty
Diego B. Arquillo	Hans Crommenacker	Jerry Hahn
Serafin Arroyo	Tomio Crosley	Robert Hall
Muhammad Asim	Allen Curtis	Shu Han
Kit August	Jesús Daniel Trigo	Nathaniel Hamming
Doug Baird	David Davenport	Rickey L. Hampton
David Baker	Russell Davis	Sten Hanke
Anindya Bakshi	Sushil K. Deka	Aki Harma
Ananth Balasubramanian	Ciro de la Vega	Jordan Hartmann
Sunlee Bang	Jim Dello Stritto	Kai Hassing
M. Jonathan Barkley	Kent Dicks	Avi Hauser
Gilberto Barrón	Hyoungdo Do	Nathaniel Heintzman
David Bean	Fangjie Dong	Charles Henderson
John Bell	Jonathan Dougherty	Jun-Ho Her
Olivia Bellamou-Huet	Xiaolian Duan	Timothy L. Hirou
Rudy Belliardi	Sourav Dutta	Allen Hobbs
Daniel Bernstein	Jakob Ehrensvarð	Alex Holland
George A. Bertos	Fredrik Einberg	Arto Holopainen
Chris Biernacki	Javier Escayola Calvo	Kris Holtzclaw
Ola Björnsne	Mark Estes	Robert Hoy
Thomas Blackadar	Leonardo Estevez	Anne Huang
Thomas Bluethner	Michael Faughn	Guiling Huang
Douglas P. Bogia	Bosco T. Fernandes	Haofei Huang
Xavier Boniface	Christoph Fischer	Zhiyong Huang
Shannon Boucousis	Morten Flintrup	David Hughes
Lyle G. Bullock, Jr.	Russell Foster	Robert D. Hughes
Bernard Burg	Eric Freudenthal	Jiyoung Huh
Chris Burns	Matthias Frohner	Hugh Hunter
Jeremy Byford-Rew	Kenneth Fuchs	Philip O. Isaacson
Satya Calloji	Jing Gao	Atsushi Ito
Carole C. Carey	Marcus Garbe	Michael Jaffe
Craig Carlson	John Garguilo	Praduman Jain
Santiago Carot-Nemesio	Liang Ge	Zongbo Jiang
Seungchul Chae	Rick Geimer	Hu Jin
Yao Chen	Igor Gejdos	Danny Jochelson
Jing Cheng	Ferenc Gerbovics	Akiyoshi Kabe
Peggy Chien	Alan Godfrey	Steve Kahle
David Chiu	Nicolae Goga	Tomio Kamioka
Jinyong Choi	Julian Goldman	James J. Kang

Kei Kariya
 Andy Kaschl
 Junzo Kashihara
 Ralph Kent
 Laurie M. Kermes
 Sanjay R. Kharche
 Ahmad Kheirandish
 Junhyung Kim
 Minh Kim
 Min-Joon Kim
 Taekon Kim
 Tetsuya Kimura
 Michael J. Kirwan
 Alfred Kloos
 Edward Koch
 Jeongmee Koh
 Jean-Marc Koller
 John Koon
 Patty Krantz
 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
 Jing Li
 Qiong Li
 Xiangchen Li
 Patrick Lichter
 Jisoon Lim
 Wei-Jung Lo
 Charles Lowe
 Don Ludolph
 Ling Luo
 Christian Lusztick
 Bob MacWilliams
 Srikanth Madhurbootheswaran
 Miriam L. Makhlouf
 M. Sabarimalai Manikandan
 Romain Marmot
 Sandra Martinez
 Miguel Martínez de EsproncedaCámara
 Peter Mayhew
 Jim McCain
 LászlóMeleg
 Alexander Mense
 Behnaz Minaei
 Jinsei Miyazaki
 Madhu Mohan
 Erik Moll
 Darr Moore
 Chris Morel

Carsten Mueglitz
 Soundharya Nagasubramanian
 Alex Neefus
 Trong-Nghia Nguyen-Dobinsky
 Michael E. Nidd
 Jim Niswander
 Hiroaki Niwamoto
 Thomas Norgall
 Yoshiteru Nozoe
 Abraham Ofek
 Brett Olive
 BegonyaOtal
 Marco Paleari
 Bud Panjwani
 Carl Pantiskas
 Harry P. Pappas
 Hanna Park
 Jong-Tae Park
 Myungeun Park
 Phillip E. Pash
 TongBi Pei
 Soren Petersen
 James Petisce
 Peter Piction
 Michael Pliskin
 Varshney Prabodh
 Jeff Price
 Harald Prinzhorn
 Lifei Qian
 Harry Qiu
 Tanzilur Rahman
 Lin Ran
 Phillip Raymond
 Terrie Reed
 Barry Reinhold
 Brian Reinhold
 Melvin I. Reynolds
 John G. Rhoads
 Jeffrey S. Robbins
 Chris Roberts
 Moskowitz Robert
 Stefan Robert
 Scott M. Robertson
 Timothy Robertson
 Sean Rocke
 David Rosales
 Bill Saltzstein
 Giovanna Sannino
 Jose A. Santos-Cadenas
 Stefan Sauermann
 John Sawyer
 Alois Schloegl
 Paul S. Schluter
 Mark G. Schnell
 Richard A. Schrenker
 Antonio Scorpiniti
 KwangSeok Seo
 Riccardo Serafin

Sid Shaw
 Frank Shen
 Min Shih
 Mazen Shihabi
 Redmond Shouldice
 Sternly K. Simon
 Marjorie Skubic
 Robert Smith
 Ivan Soh
 Motoki Sone
 Emily Sopensky
 Rajagopalan Srinivasan
 Nicholas Steblay
 Lars Steubesand
 John (Ivo) Stivorio
 Hermanni Suominen
 Lee Surprenant
 Ravi Swami
 Ray Sweidan
 Na Tang
 Haruyuyki Tatsumi
 Isabel Tejero
 Tom Thompson
 Jonas Tirén
 Janet Traub
 Gary Tschautscher
 Masato Tsuchid
 Ken Tubman
 Akib Uddin
 Sunil Unadkat
 Fabio Urbani
 Philipp Urbauer
 Laura Vanzago
 Alpo Värri
 Andrei Vasilateanu
 Dalimar Velez
 Martha Velez
 Rudi Voon
 Isobel Walker
 David Wang
 Jerry P. Wang
 Shiwei Wang
 Yao Wang
 Yi Wang
 Steve Warren
 Fujio Watanabe
 Toru Watsuji
 Kathleen Wible
 Paul Williamson
 Jia-Rong Wu
 Will Wykeham
 Ariton Xhafa
 Ricky Yang
 Melanie S. Yeung
 Qiang Yin
 Done-Sik Yoo
 Zhi Yu

Jianchao Zeng
Jason Zhang
Zerui Zhang
Shiwei Zhao

Liang Zheng
Daidi Zhong
Yuanhong Zhong
Qing Zhou

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.

Robert Aiello
Cheryl Alexander Wang
Bjoern Andersen
Pradeep Balachandran
Malcolm Clarke
Javier Espina
Michael Faughn
Kenneth Fuchs

Charles M. Gropper
Werner Hoelzl
Piotr Karocki
Stuart Kerry
Raymond Krasinski
H. Moll
Rajesh Murthy

Bansi Patel
Scott M. Robertson
Stefan Schlichting
Walter Struppler
John Vergis
Yu Yuan
Oren Yuen
Daidi Zhong

When the IEEE SA Standards Board approved this standard on 30 March 2023, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani
Doug Edwards
Ramy Ahmed Fathy
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick

Joseph S. Levy
Howard Li
Johnny Daozhuang Lin
Gui Lin
Xiaohui Liu
Kevin W. Lu
Daleep C. Mohla
Andrew Myles

Paul Nikolich
Annette D. Reilly
Robby Robson
Lei Wang
F. Keith Waters
Karl Weber
Philip B. Winston
Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 11073-10425-2023, Health Informatics—Device Interoperability—Part 10425: Personal Health Device Communication—Device Specialization—Continuous Glucose Monitor (CGM).

ISO/IEEE 11073 standards enable communication between medical devices and external computer systems. This document uses the optimized framework created in ISO/IEEE 11073-20601 and describes a specific, interoperable communication approach for continuous glucose monitors (CGMs).¹ 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 (PHDs).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEEE 11073-10425:2024

¹ Information on references can be found in Clause 2.

Contents

1. Overview	1
1.1 Scope	1
1.2 Purpose	1
1.3 Word usage	1
1.4 Context	1
2. Normative references	2
3. Definitions, acronyms, and abbreviations	2
3.1 Definitions	2
3.2 Acronyms and abbreviations	3
4. Introduction to ISO/IEEE 11073 personal health devices (PHDs)	4
4.1 General	4
4.2 Introduction to ISO/IEEE 11073-20601 modeling constructs	4
4.3 Compliance with other standards	5
5. Glucose monitoring concepts and modalities	5
5.1 General	5
5.2 Device types	6
5.3 CGM agent-to-manager communication	7
5.4 Collected data	7
5.5 Stored data	9
6. Continuous glucose monitor (CGM) domain information model (DIM)	9
6.1 Overview	9
6.2 Class extensions	9
6.3 Object instance diagram	9
6.4 Types of configuration	11
6.5 Profiles	11
6.6 MDS object	11
6.7 Numeric objects	14
6.8 Real-time sample array objects	24
6.9 Enumeration objects	24
6.10 PM-store objects	27
6.11 Scanner objects	31
6.12 Class extension objects	31
6.13 CGM information model extensibility rules	31
7. CGM service model	32
7.1 General	32
7.2 Object access services	32
7.3 Object access event report services	34
8. CGM communication model	34
8.1 Overview	34
8.2 Communication characteristics	34
8.3 Association procedure	35
8.4 Configuring procedure	36
8.5 Operating procedure	38
8.6 Time synchronization	39

9. Test associations	39
9.1 Behavior with standard configuration	39
9.2 Behavior with extended configurations	39
10. Conformance	39
10.1 Applicability	39
10.2 Conformance specification	39
10.3 Levels of conformance.....	40
Annex A (informative) Bibliography	46
Annex B (normative) Any additional ASN.1 definitions	47
Annex C (normative) Allocation of identifiers	51
Annex D (informative) Message sequence examples	56
Annex E (informative) Protocol data unit examples.....	58
Annex F (informative) Revision history	67

STANDARDSISO.COM : Click to view the full PDF of ISO/IEEE 11073-10425:2024

Health Informatics—Device Interoperability

Part 10425:

Personal Health Device Communication—Device Specialization—Continuous Glucose Monitor (CGM)

1. Overview

1.1 Scope

This standard establishes a normative definition of communication between personal health continuous glucose monitor (CGM) devices (agents) and managers (e.g., cell phones, personal computers, personal health appliances, set top boxes) in a manner that enables plug-and-play interoperability. It leverages work done in other ISO/IEEE 11073 standards including existing terminology, information profiles, 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 of CGM devices. In this context, CGM refers to the measurement of the level of glucose in the body on a regular (typically 5 minute) basis through a sensor continuously attached to the person.

1.2 Purpose

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

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall equals is required to*).^{2,3}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should equals is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may equals is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can equals is able to*).

1.4 Context

See ISO/IEEE 11073-20601 for an overview of the environment within which this standard is written.

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

² The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

³ The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

This standard is based on ISO/IEEE 11073-20601, which in turn draws information from both ISO/IEEE 11073-10201 [B6] and ISO/IEEE 11073-20101 [B7].⁴ The medical device encoding rules (MDER) used within this standard are fully described in ISO/IEEE 11073-20601.

The object classes and attributes in this standard are identified by nomenclature codes. Each code consists of a reference identifier (RefID) string and an integer code value. By using a consistent nomenclature, interoperability is enhanced as all implementations maintain the same semantic meaning for the numeric codes. This standard leverages the existing nomenclature codes in ISO/IEEE 11073-10101. Between this standard, ISO/IEEE 11073-10101, ISO/IEEE 11073-20601, and other IEEE Std 11073-104zz, all required nomenclature codes for implementation are documented. New codes may be defined in newer versions/ revisions of each of these documents. In the case of a conflict, when one term code has been assigned to two separate semantic concepts with different RefIDs, in general, the oldest definition in actual use should take precedence. The same policy applies when one RefID has two different code values assigned in different specifications. The resolution of such conflicts will be determined through joint action by the responsible working groups and other stakeholders, and any corrective action will be published as corrigenda.

NOTE—In this standard, IEEE Std 11073-104zz is used to refer to the collection of device specialization standards that utilize ISO/IEEE 11073-20601, where zz can be any number from 01 to 99, inclusive.⁵

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used; therefore, 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.

ISO/IEEE 11073-10101, Health informatics—Point-of-care medical device communication—Part 10101: Nomenclature.⁶

ISO/IEEE 11073-20601, Health informatics—Personal health device communication—Part 20601: Application Profile—Optimized Exchange Protocol.

See Annex A for all informative material referenced by this standard.

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.

blood glucose: Glucose concentration in the blood.

class: In object-oriented modeling, a term collectively describing the attributes, methods, and events utilized by objects instantiated from the class.

compute engine: See: **manager**.

continuous glucose monitor (CGM): A medical device to provide a series of estimates of blood glucose concentration; typically from body fluid.

⁴ The numbers in brackets correspond to those of the bibliography in Annex A.

⁵ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

⁶ ISO/IEEE publications are available from the ISO Central Secretariat (<http://www.iso.ch/>). ISO/IEEE publications are also available in the United States from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁷ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

device: A physical apparatus implementing either an agent or a manager role.

glucose: The major source of energy used by the body cells. Glucose is commonly referred to as *sugar*.

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

interstitial fluid (ISF): The thin layer of fluid that surrounds the body's cells.

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 a computer system.

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

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

personal health device (PHD): A device used in personal health applications.

personal telehealth device: *See:* **personal health device (PHD)**.

3.2 Acronyms and abbreviations

APDU	application protocol data unit
ASN.1	Abstract Syntax Notation One
AST	alternative site testing
BGM	blood glucose meter
CGM	continuous glucose monitor
DIM	domain information model
DM	disease management
EUI-64	extended unique identifier (64 bits)
HCP	health care professional
ICS	implementation conformance statement
ID	identifier
ISF	interstitial fluid
MDC	medical device communication
MDER	medical device encoding rules
MDS	medical device system
MOC	managed object class
OID	object identifier
PHD	personal health device
VMO	virtual medical object
VMS	virtual medical system

4. Introduction to ISO/IEEE 11073 personal health devices (PHDs)

4.1 General

This standard and the remainder of the series of ISO/IEEE 11073 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 ISO/IEEE 11073-20601 for a description of the guiding principles for this series of ISO/IEEE 11073 PHD standards.

ISO/IEEE 11073-20601 supports the modeling and implementation of an extensive set of PHDs. This standard defines aspects of the CGM device. It describes all aspects necessary to implement the application layer services and data exchange protocol between an ISO/IEEE 11073 PHD CGM agent and a manager. This standard defines a subset of the objects and functionality contained in ISO/IEEE 11073-20601, and it extends and adds definitions where appropriate. All new definitions are given in Annex B in Abstract Syntax Notation One (ASN.1) (ITU-T X.680 **Error! Reference source not found.**). Nomenclature codes referenced in this standard that are not defined in ISO/IEEE 11073-20601 are normatively defined in Annex C.

ISO/IEEE 11073-20601 supports the modeling and implementation of an extensive set of PHDs. This standard defines aspects of the weighing scale device. It describes all aspects necessary to implement the application layer services and data exchange protocol between an ISO/IEEE 11073 PHD weighing scale agent and a manager. This standard defines a subset of the objects and functionality contained in ISO/IEEE 11073-20601 and extends and adds definitions where appropriate. All new definitions are given in Annex B in ASN.1. All nomenclature codes referenced in this standard are collected in Annex C.

4.2 Introduction to ISO/IEEE 11073-20601 modeling constructs

4.2.1 General

The ISO/IEEE 11073 series of standards, and in particular ISO/IEEE 11073-20601, 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 ISO/IEEE 11073-20601 for a detailed description of the modeling constructs.

4.2.2 Domain information model (DIM)

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 the data that an agent can communicate to a manager. Communication between the agent and the manager is defined by the application protocol in ISO/IEEE 11073-20601.

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 ISO/IEEE 11073-20601 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 ISO/IEEE 11073-20601.

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.

4.3 Compliance with other standards

Devices that comply with this standard may also be required to comply with other domain- and device-specific standards that supersede the requirements of this standard with respect to issues including safety, reliability, and risk management. A user of this standard is expected to be familiar with all other such standards that apply and to comply with any higher specifications thus imposed.

Typically, medical devices should comply with the IEC 60601-1 [B1] base standards with respect to electrical and mechanical safety and any device-specific standard as might be defined in the IEC 60601-2 [B2] series of standards. Software aspects may apply through standards such as IEC 62304 [B3]. Devices that comply with this standard implement higher layers of network software and utilize lower layers as appropriate to the application. The requirements on performance of such applications and conformance are defined elsewhere and are outside the scope of this standard. Moreover, the use of any medical equipment is subject to risk assessment and risk management appropriate to the application. Relevant examples include ISO 14971 [B5] and IEC 80001-1 [B4]. The requirements of such risk assessment, risk management, and conformance are outside the scope of this standard. The applicable versions of the referenced safety-related standards may differ by country.

5. Glucose monitoring concepts and modalities

5.1 General

This clause presents the general concepts of CGMs. In the context of PHDs in this family of standards, a CGM is a device that estimates the concentration of glucose in the blood typically measured from interstitial fluid (ISF). The glucose concentration is available on a continual basis at a periodic interval from a sensor. A CGM improves therapy control as opposed to the single, episodic measurements of a blood glucose meter (BGM). Frequent measurements provided by a CGM give a patient greater insight as to the fluctuations in blood glucose levels throughout the day, and in turn, can help to reduce the risk of developing diabetic complications.

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). 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 ISF). Glucose levels rise after meals and are usually lowest in the morning, before the first meal of the day.

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. If left untreated or improperly managed, diabetes can lead to complications including cardiovascular disease, kidney failure, and eye disease.

Concentration of blood glucose uses either mmol/L or mg/dL as units. Countries that use the metric system generally use mmol/L. However, the United States as well as other countries use mg/dL. To convert blood glucose measurements between the two units, 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 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
ISF	Subcutaneous tissue	N/A
Control solution	N/A	N/A

NOTE—The blood glucose concentration may be indirectly derived from an ISF sample, which is a common technique used in continuous glucose monitoring. A control solution is normally used for glucose meter quality control.

ISF is the common source of the measurement made by a CGM, though new technologies on the horizon may employ other sources. BGMs may utilize other sample sources for their measurements, with the common source being capillary whole blood.

5.2 Device types

Continuous glucose monitor (CGM) devices are generally designed to be portable and permanently connected to the body.

The structural shape of CGMs may vary, but a CGM device typically includes the following components: the glucose sensor, a transmitter, and a receiver. These components may be enclosed in physical different devices.

With current technology at the time of writing, the sensor consists of a small metallic filament that is inserted into subcutaneous layer of fat tissue under the skin, where it measures an approximation of the blood glucose level from the ISF. Preferred sites for sensor insertion are the abdomen, lumbar region, and the upper arms. Typically, there is a mechanical means (e.g., an adhesive patch) used to keep the sensor in place. The sensor needs to be replaced periodically.

A transmitter connected to the sensor is used to wirelessly transmit the measurements to the receiver. This receiver is often a physically separate device that can display trend graphs and other statistics or notifications along with the current glucose measurement, as depicted in Figure 1(a). Insulin pumps, and other personal electronic devices, can also serve as the receiver of the CGM measurements, as depicted in Figure 1(b).

CGMs provide blood glucose approximation typically from ISF. To help ensure an accurate approximation, a CGM is periodically calibrated against a blood-based glucose measurement. While manual entry of the blood glucose measurement into the CGM receiver is possible, more sophisticated CGMs provide wireless communication between either the transmitter or receiver and a BGM.

For clarity, the terms *transmitter* and *receiver* were used; however, note that both these devices may actually be transceivers.

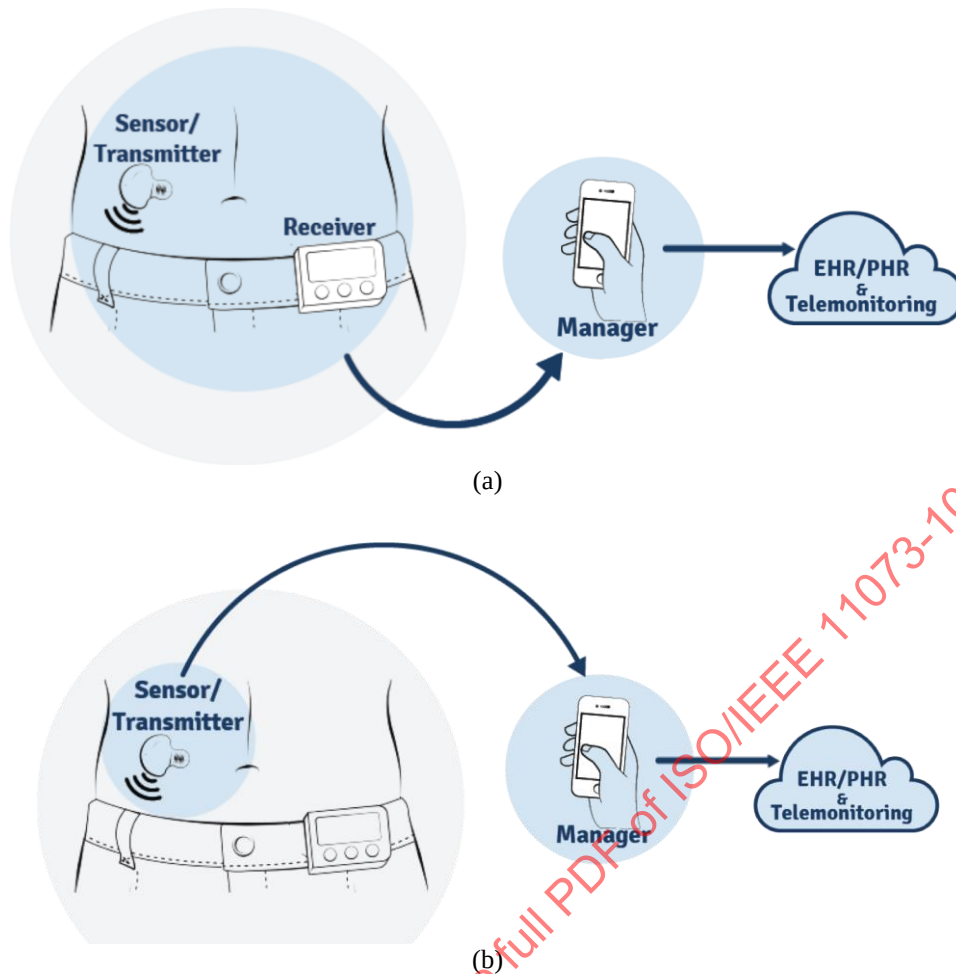


Figure 1—Agent-to-manager communication scenarios:
(a) CGM receiver to manager and (b) CGM sensor/transmitter to manager
 (there may be other scenarios not depicted in Figure 1)

5.3 CGM agent-to-manager communication

As described in 5.2, a CGM may consist of two physical parts, e.g., the sensor/transmitter and the receiver. This device specialization provides a standard of interoperability between only one of those physical parts and the computing device acting as the CGM manager. For example, this standard could be between the CGM receiver and manager, or it could be between the CGM sensor/transmitter and manager when the system does not include a specific CGM receiver. These scenarios are depicted in Figure 1. Other scenarios not discussed may exist.

The CGM agent may periodically send the measurement results to the manager upon availability or the exchange may take place after a CGM session (hours or days). Furthermore, the manager may request stored results of a dedicated time period. This functionality, in addition to the store and forward scenario, requires time stamps for each measurement result. It is the responsibility of the agent to resolve the time stamp of any measurements reported from CGM components. The manager as referenced here could be a personal computer, mobile phone, or other computing device.

5.4 Collected data

5.4.1 General

The CGM is a portable device and therefore may not be connected to a manager while collecting data. The two main use cases for a CGM agent to connect to a manager and send its data are the following:

- The CGM user visits a health care professional (HCP) to examine the adequacy of the insulin therapy. The HCP normally compares the historic data from the CGM with the corresponding data from an insulin pump device to derive

necessary adjustments to the insulin therapy. As the interval between such visits may constitute several months, CGMs are typically capable of storing data for such time periods.

- The CGM user connects the CGM agent to a manager as needed to examine the adequacy of the insulin therapy and to apply adjustments. This tends to happen on a more frequent basis (e.g., once per week).

In addition to the preceding two use cases, a CGM agent may also be continuously connected to a manager to report collected data (e.g., artificial pancreas).

5.4.2 Glucose

Glucose is a measurement of glucose concentration in the blood. Typically for CGM, this measurement is made from other body fluids than blood, and thus calibration is required to calculate the blood glucose levels.

5.4.3 Sensor calibration

A glucose measurement is typically needed to calibrate the CGM. This measurement could originate from a BGM, but would need to be stored in the memory of the continuous glucose meter, so as to have a log of the calibrations performed. Traditionally, the calibration glucose measurement is entered manually by the user, but may also be collected directly from a BGM.

5.4.4 Sensor run-time

CGM sensors deteriorate over time due to their method of collecting measurements, e.g., sensor embedded in the subcutaneous tissue receives build-up. Thus, CGM sensors need to be replaced periodically and each manufacturer specifies the life of the sensor. The sensor run-time indicates the suggested period of time the sensor should be used.

5.4.5 Glucose sampling interval

The glucose sampling interval indicates the frequency of glucose measurements.

5.4.6 Glucose trend

Glucose trend is the rate of change in glucose measurements at a time instant.

5.4.7 Patient low/high thresholds

The patient low/high thresholds are settings used to indicate a range of patient acceptable glucose concentrations. If glucose concentrations fall outside this range, a typical reaction is to notify the patient (e.g., as a CGM status message or other indicator) and record the event.

5.4.8 Device hypo/hyper thresholds

The device hypo/hyper thresholds are settings to indicate the critical glucose concentration range. If a glucose concentration crosses either of these thresholds, the CGM typically notifies the patient (e.g., as a CGM status message or other indicator) and records the event.

5.4.9 Glucose rate-of-change thresholds

The glucose rate-of-change thresholds are settings to indicate the maximum increase and decrease rate of glucose variation. If a glucose rate of change crosses either of these thresholds, the CGM typically notifies the patient (e.g., as a CGM status message or other indicator) and records the event.

5.4.10 PHD DM status

The “PHD DM Status” allows generic notification handling for PHDs within the Disease Management domain. The status of the device is represented in a number of bit flags covering warning, error, service, and undetermined messages.

5.4.11 CGM status

CGM status object represents the specific notifications given by the CGM device including, but not limited to, warnings, errors, and handling events.

5.5 Stored data

As stated in 5.4.1, a CGM may be used over several months of operation without being connected to a manager to send its data. Once a CGM is connected to a manager, the manager is able to select which of the agent's stored measurements or observations to retrieve. Depending on the agent's capabilities to organize its data into clusters of chronologically contiguous data, the manager may also select the time ranges of the stored data to retrieve. The agent then transmits the manager's selection in one or several blocks of messages for processing by a manager or other processing apparatus. The manager may also be able to choose a set of data clusters for deletion.

6. Continuous glucose monitor (CGM) domain information model (DIM)

6.1 Overview

This clause describes the DIM of the CGM.

6.2 Class extensions

In this standard, extensions are defined with respect to ISO/IEEE 11073-20601.

The following two attributes are an extension to the enumeration class:

- The Capability-Mask-Simple attribute indicates whether the corresponding bit in the Enum-Observed-Value-Simple-Bit-Str attribute is supported by the Agent.
- The State-Flag-Simple attribute indicates whether the corresponding bit in the Enum-Observed-Value-Simple-Bit-Str attribute is a state or event.

If the Enum-Observed-Value-Simple-Bit-Str attribute is used, the Capability-Mask-Simple and State-Flag-Simple attributes are mandatory. The values of the Capability-Mask-Simple and State-Flag-Simple attributes are implementation specific.

See Annex B for their ASN.1 structures.

The Capability-Mask-Simple attribute is dynamic, i.e., the attribute may change at some point after configuration. The State-Flag-Simple attribute is static, i.e., it shall remain unchanged after the configuration is agreed upon.

6.3 Object instance diagram

The object instance diagram of the CGM DIM, which is defined for the purposes of this standard, is shown in Figure 2. See 6.6 through 6.12 for descriptions of the different CGM objects [e.g., the CGM medical device system (MDS) object, the glucose numeric object, and the CGM status enumeration object]. See 6.13 for rules for extending the CGM DIM beyond elements as described in this standard. Each subclause that describes an object of the CGM 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 ISO/IEEE 11073-20601.

- 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 are defined as follows:

- M—attribute is mandatory.
- C—attribute is conditional and depends on the condition stated in the Remark or Value column (if ISO/IEEE 11073-20601 is referenced, then it contains the conditions).
- R—attribute is recommended.
- NR—attribute is not recommended.
- 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. For attributes with qualifiers set to R or NR, underlying requirements stated in the Remark and Value column in ISO/IEEE 11073-20601 shall be followed. If any attribute (from the DIM of ISO/IEEE 11073-20601) is not included in the definition of that object in this standard, it shall not be included in that object by an implementation unless it is a vendor-specific attribute extended according to 6.12.

An attribute is further qualified as static, dynamic, or observational. Static attributes shall not change value during the life of an association. Dynamic attributes have a value that may change during the life of an association. The dynamic attribute value should be sent at configuration time and shall be sent at or before the time when the value would be needed for interpreting a reported observation. Observational attributes have a value that may change during the life of an association. When a set of observational attribute values is received, these values are combined with the available context information (i.e., all related dynamic and static attribute values) to represent the observation at the observation time.

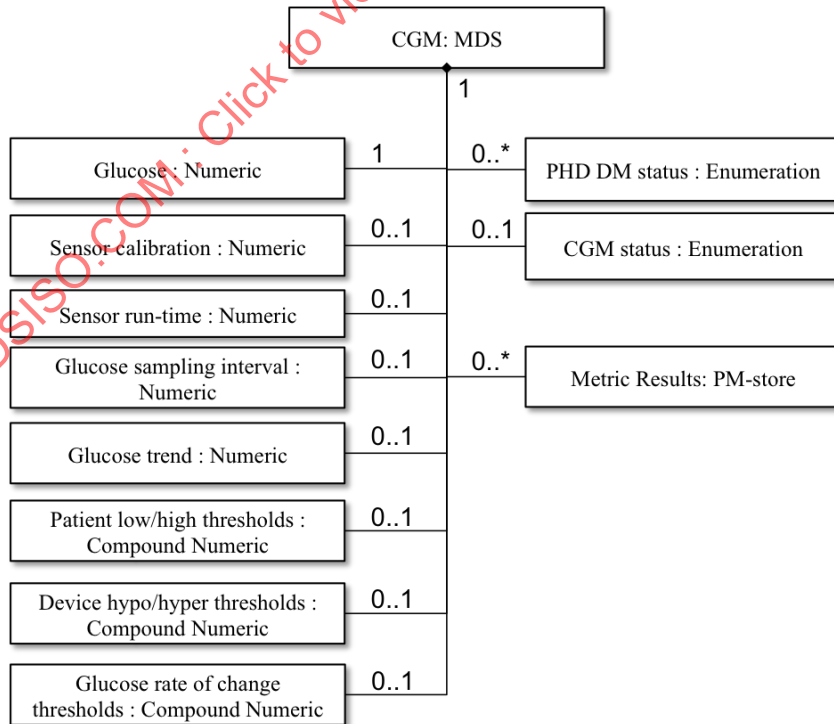


Figure 2—CGM DIM

6.4 Types of configuration

6.4.1 General

As specified in ISO/IEEE 11073-20601, 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.

One standard configuration is defined in this standard. The standard configuration 2500 (0x09C4) contains one glucose object (see 6.7.2).

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 Profiles

6.5.1 General

A profile further constrains the objects, services, and communication model of a specialization. By profiling the device specialization, the standard provides more guidance on the specific mandatory objects that shall be implemented, the objects that are optional, and the objects that are not required. This standard does not define profiles for the CGM device.

6.6 MDS object

6.6.1 MDS object attributes

Table 2 summarizes the attributes of the CGM 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	Qual.
Handle	0	M
System-Type	Attribute not present. See ISO/IEEE 11073-20601.	NR
System-Type-Spec-List	{MDC_DEV_SPEC_PROFILE_CGM, 3}	M
System-Model	{“Manufacturer”, “Model”}	M
System-Id	Extended unique identifier (64 bits) (EUI-64)	M
Dev-Configuration-Id	Standard config: 0x09C4 Extended configs: 0x4000–0x7FFF	M
Attribute-Value-Map	See ISO/IEEE 11073-20601.	C
Production-Specification	See ISO/IEEE 11073-20601.	C

Mds-Time-Info	See ISO/IEEE 11073-20601.	C
Date-and-Time	See ISO/IEEE 11073-20601.	C
Base-Offset-Time	See ISO/IEEE 11073-20601.	R
Relative-Time	See ISO/IEEE 11073-20601.	C
HiRes-Relative-Time	See ISO/IEEE 11073-20601.	C
Date-and-Time-Adjustment	See ISO/IEEE 11073-20601.	R
Power-Status	See ISO/IEEE 11073-20601.	R
Battery-Level	See ISO/IEEE 11073-20601.	R
Remaining-Battery-Time	See ISO/IEEE 11073-20601.	R
Reg-Cert-Data-List	See ISO/IEEE 11073-20601.	O
Confirm-Timeout	See ISO/IEEE 11073-20601.	O
Transport-Timeout	See ISO/IEEE 11073-20601.	O

NOTE—See ISO/IEEE 11073-20601 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 ISO/IEEE 11073-20601 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. For a CGM agent with extended configuration, this identifier is chosen in the range of extended-config-start to extended-config-end (see ISO/IEEE 11073-20601) 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.

As defined in ISO/IEEE 11073-20601, the production-specification attribute includes component serial numbers, revisions, and so on in manufacture specific format. For CGM MDS object, the production-specification attribute shall include the required information for all physical components, e.g., sensor, transmitter, receiver, etc., as applicable. When any one of these components are changed or replaced, the MDS production-specification attribute shall be updated accordingly.

6.6.2 MDS object methods

Table 3 defines the methods (actions) of the CGM 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 ISO/IEEE 11073-20601) 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 ISO/IEEE 11073-20601); the Parameters (action-info-args) column defines the associated ASN.1 data structure (see ISO/IEEE 11073-20601 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	—
	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 ISO/IEEE

11073-20601). Agents with an internal real-time clock shall indicate this capability by also setting the mds-time-capab-real-time-clock bit in the Mds-Time-Info attribute.

The Set-Time method can be supported only if the Absolute-Time-Stamp attribute is supported.

— **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 ISO/IEEE 11073-20601).

The Set-Base-Offset-Time method can be supported only if the Base-Offset-Time-Stamp attribute is supported.

6.6.3 MDS object events

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

Table 4—CGM 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	ScanReportInfo Fixed	—
	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 agent during the configuring procedure if the manager does not already know the agent's configuration from past associations or because the manager has not been implemented to recognize the configuration according to the CGM device specialization. The event provides static information about the supported measurement capabilities of the agent.

— **MDS-Dynamic-Data-Update-Var:**

This event provides dynamic measurement data from the agent for the numeric and enumeration objects. 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 agent for the numeric and enumeration objects. 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—ISO/IEEE 11073-20601 requires that managers support all of the MDS object events previously listed.

6.6.4 Other MDS services

6.6.4.1 GET service

A CGM 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 only after the manager has confirmed selection of the agent's configuration.

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

Table 5—CGM MDS object GET service

Service	Subservice type name	Mode	Subservice type	Parameters	Results
GET	<na>	<implied confirmed>	<na>	GetArgumentSimple = (obj-handle = 0), attribute-id-list <optional>	GetResultSimple = (obj-handle = 0), attribute-list

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

6.6.4.2 SET service

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

6.7 Numeric objects

6.7.1 General

The CGM DIM (see Figure 2) contains numeric objects that represent aspects of glucose concentration, sensor calibration, sensor run-time, measurement interval, trending, patient thresholds, hypo/hyper thresholds, and glucose rate-of-change thresholds. These are described in 6.7.2 through 6.7.9. Table 6 shows attributes that are common to all the numeric objects.

Table 6—Common numeric object attributes

Attribute name	Value	Qual.
Handle	See ISO/IEEE 11073-20601.	M
Type	Defined in the following subclauses.	M
Supplemental-Types	See ISO/IEEE 11073-20601.	O
Metric-Spec-Small	Defined in the following subclauses.	M
Metric-Structure-Small	See ISO/IEEE 11073-20601.	O
Measurement-Status	See ISO/IEEE 11073-20601.	C
Metric-Id	See ISO/IEEE 11073-20601.	O
Metric-Id-List	See ISO/IEEE 11073-20601.	C
Metric-Id-Partition	See ISO/IEEE 11073-20601.	O
Unit-Code	Defined in the following subclauses.	M
Attribute-Value-Map	See ISO/IEEE 11073-20601.	C
Source-Handle-Reference	See ISO/IEEE 11073-20601.	O
Label-String	See ISO/IEEE 11073-20601.	O
Unit-LabelString	See ISO/IEEE 11073-20601.	O
Absolute-Time-Stamp	See ISO/IEEE 11073-20601.	C
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Relative-Time-Stamp	See ISO/IEEE 11073-20601.	C
HiRes-Time-Stamp	See ISO/IEEE 11073-20601.	C
Measure-Active-Period	See ISO/IEEE 11073-20601.	O
Simple-Nu-Observed-Value	See ISO/IEEE 11073-20601.	C
Compound-Simple-Nu-Observed-Value	See ISO/IEEE 11073-20601.	C
Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	C
Compound-Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	C
Compound-Nu-Observed-Value	See ISO/IEEE 11073-20601.	C
Accuracy	See ISO/IEEE 11073-20601.	O

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

Each object represents a specific aspect of glucose measurement, patient settings, or sensor operations. The object 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 6.6.3) prior to reporting any of the dependent values.

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

The CGM specialization recommends the Base-Time-Offset for all numeric objects. Base-Time-Offset attribute allows for convenient time adjustments based on changing time zones.

6.7.2 Glucose

Glucose is a measurement of glucose concentration in the blood. Typically for CGM, this measurement is made from other body fluids than blood, and thus calibration is required to calculate the blood glucose levels. Table 7 summarizes the attributes of the glucose numeric object. The glucose numeric object shall be supported by a CGM agent.

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

The observed value reported in this object is a glucose measurement. Only non-negative numbers shall be used.

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

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.

The glucose numeric type attribute defines the type of fluid the CGM will sample. If the fluid type is unknown, then undetermined whole blood, MDC_CONC_GLU_UNDETERMINED_WHOLEBLOOD, or undetermined plasma, MDC_CONC_GLU_UNDETERMINED_PLASMA, should be chosen, as appropriate. The glucose numeric is further defined by the supplemental-type attribute, which indicates from which body site the CGM will be sampling. If the sample location is unknown, MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED shall be chosen; and if the sample location is not available in the codes provided, MDC_CTXT_GLU_SAMPLELOCATION_OTHER shall be chosen.

The measurement-status attribute is used to qualify the measurement or provide additional operational conditions and is recommended. A measurement-status of *calibration-ongoing* shall indicate that the CGM is in the process of calibration when the measurement was taken. A measurement-status of *invalid* shall indicate that the CGM is uncalibrated when the measurement was taken. A measurement-status of *questionable* shall indicate that the measurement is not reliable. A measurement-status of *validated-data* shall indicate that the CGM was calibrated when the measurement was taken and the measurement is reliable.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEEE 11073-10425:2024

Table 7—Glucose numeric object attributes

Attribute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x09C4)	
	Value	Qual.	Value	Qual.
Handle	See ISO/IEEE 11073-20601.	M	1	M
Type	{MDC_PART_SCADA, MDC_CONC_GLU_ISF or 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_CONTROL or MDC_CONC_GLU_UNDETERMINED_WHOLEBLOOD or MDC_CONC_GLU_UNDETERMINED_PLASMA}	M	{MDC_PART_SCADA, MDC_CONC_GLU_ISF}	M
Supplemental-Types	{MDC_PART_PHD_DM, MDC_CTXT_GLU_SAMPLELOCATION_FINGER or MDC_CTXT_GLU_SAMPLELOCATION_AST or MDC_CTXT_GLU_SAMPLELOCATION_EARLOBE or MDC_CTXT_GLU_SAMPLELOCATION_CTRLsolution or MDC_CTXT_GLU_SAMPLELOCATION_SUBCUTANEOUS or MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED or MDC_CTXT_GLU_SAMPLELOCATION_OTHER} See ISO/IEEE 11073-20601 and following text.	O	{MDC_PART_PHD_DM, MDC_CTXT_GLU_SAMPLELOCATION_ SUBCUTANEOUS}	M
Metric-Spec-Small	mss-avail-intermittent mss-avail-stored-data mss-acc-agent-initiated mss-cat-calculation	M	mss-avail-intermittent mss-avail- stored-data mss-acc-agent-initiated mss-cat-calculation	M
Measurement-Status	See ISO/IEEE 11073-20601 and the following text.	R	See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PER_L	M	MDC_DIM_MILLI_G_PER_DL	M
Attribute-Value-Map	See ISO/IEEE 11073-20601.	C	MDC_ATTR_NU_VAL_OBS_BASIC, then MDC_ATTR_TIME_STAMP_BO.	M
Base-Offset-Time	See ISO/IEEE 11073-20601.	R	If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from ISO/IEEE 11073-20601 apply.	M

(Table continues on the next page.)

Table 7—Glucose numeric object attributes (*continued*)

Attribute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x09C4)	
	Value	Qual.		Value
Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R	If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from ISO/IEEE 11073-20601 apply.	M
Measurement-Confidence-95	See following text.	O	See following text.	NR
Threshold-Notification-Text-String	See following text.	O	See following text.	NR

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

6.7.2.1 Measurement-confidence-95 attribute

The measurement-confidence-95 attribute specifies the upper and lower bounds for a range within which the manufacture is 95% confident that the actual measurement value resides. The lower and upper bounds have the same units as the measurement. The lower bound shall be less than or equal to the upper bound.

The measurement-confidence-95 attribute is not to be included in the standard configuration and is optional for extended configurations. Table 8 defines the measurement-confidence-95 attribute.

Table 8—Glucose measurement-confidence-95 attribute

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Measurement-Confidence-95	MDC_ATTR_MSMT_CONFIDENCE_95	Measurement Confidence95	This attribute defines the lower and upper bounds for a range within which the manufacture is 95% confident that the actual value resides. The unit for the lower bound and upper bound is the same as the measurement.	Optional Observational

NOTE—See Annex B for ASN.1 structure definition.

6.7.2.2 Glucose threshold and status attributes

One attribute extending the glucose numeric object is provided to report the agent's glucose threshold details, and a second reports whether the measurement has reached or crossed beyond the threshold boundaries. The Measurement-Status attribute has been extended (compatible with ISO/IEEE 11073-10201:2004 [B6]) from the definition in ISO/IEEE 11073-20601 in order to report the threshold status. Note that the patient low/high thresholds and device hypo/hyper thresholds objects, 6.7.7 and 6.7.8, respectively, store the glucose numeric threshold values. See Table 9 for addition details.

Table 9—Glucose threshold and status attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Threshold-Notification-Text-String	MDC_ATTR_THRES_NOTIF_TEXT_STRING	OCTET STRING	Text related to the current threshold notification.	Optional Observational
Measurement-Status	MDC_ATTR_MSMT_STAT	Measurement Status	Reports whether an observed value is at or outside threshold boundaries. If thresholding is to be used, this attribute is mandatory. Use bit msmt-value-exceed-boundaries(14), to indicate that the measurement is outside threshold boundaries. Use msmt-state-ann-inhibited(15) to indicate that the threshold indication is disabled and should not cause a displayed annunciation. If this bit is set, then bit 14 shall not be set. These are bits extended from the ISO/IEEE 11073-20601: 2016 definitions of MeasurementStatus. The definition of all other bits of Measurement-Status remains unchanged from their definition in ISO/IEEE 11073-20601:2010.	Conditional Observational

NOTE—See Annex B for ASN.1 bit mapping definition.

6.7.3 Sensor calibration

As previously described, a glucose measurement is typically needed to calibrate the CGM. This measurement could originate from a BGM, but would need to be stored in the memory of the continuous glucose meter, so as to have a log of the calibrations performed. Table 10 summarizes the attributes of the sensor calibration numeric object.

Table 10—Sensor calibration numeric object attributes

Attribute name	Extended configuration	
	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CGM_SENSOR_CALIBRATION}	M
Supplemental-Types	{MDC_PART_PHD_DM, MDC_CTXT_GLU_SAMPLELOCATION_FINGER or MDC_CTXT_GLU_SAMPLELOCATION_AST or MDC_CTXT_GLU_SAMPLELOCATION_EARLOBE or MDC_CTXT_GLU_SAMPLELOCATION_SUBCUTANEOUS or MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED or MDC_CTXT_GLU_SAMPLELOCATION_OTHER } See ISO/IEEE 11073-20601.	O
Metric-Spec-Small	mss-avail-stored-data mss-upd-aperiodic mss-acc-agent-initiated mss-cat-manual mss-cat-setting The mss-cat-manual shall only be set if, and only if, the reading is manually entered.	M
Measurement-Status	See ISO/IEEE 11073-20601 and following text.	R
Unit-Code	MDC_DIM_MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PER_L.	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

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

The measurement-status attribute is recommended. This attribute is used to qualify the calibration or provide additional calibration conditions. A measurement-status of *invalid* indicates that the CGM is uncalibrated. A measurement-status of *validated-data* indicates that the CGM was calibrated.

The sensor calibration numeric is further defined by the supplemental-type attribute, which indicates the body site used for the glucose calibration measurement. If the sample location is unknown, MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED shall be chosen; and if the sample location is not available in the codes provided, MDC_CTXT_GLU_SAMPLELOCATION_OTHER shall be chosen.

6.7.4 Sensor run-time

CGM sensors deteriorate over time due to their method of collecting measurements, e.g., sensor embedded in the subcutaneous tissue receives build-up. Thus, CGM sensors need to be replaced periodically and each manufacture specifies the life of their sensor. The sensor run-time numeric object indicates the suggested period of time CGM sensor should be used. Table 11 summarizes the attributes of the sensor run-time numeric object.

Table 11—Sensor run-time numeric object attributes

Attribute name	Extended configuration	
	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CGM_SENSOR_RUN_TIME}	M
Metric-Spec-Small	mss-upd-aperiodic mss-msmt-aperiodic mss-acc-agent-initiated mss-cat-calculation mss-avail-stored-data mss-cat-setting See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_HR	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The sensor run-time numeric object does not support any methods, events, or other services.

Using the time stamp attribute as the start time and the observed value attribute as the duration with units of hour, one can calculate date and time when the CGM sensor should be replaced. Typically, this object is only created during sensor insertion; however, if the CGM is able to determine the quality of the sensor, this object may be used to reflect a dynamic sensor run-time.

6.7.5 Glucose sampling interval

The glucose sampling interval numeric indicates the frequency of CGM glucose measurements. Table 12 summarizes the attributes of the glucose sampling interval numeric object.

Table 12—Glucose sampling interval numeric object attributes

Attribute name	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CGM_SENSOR_SAMPLE_INTERVAL}	M
Metric-Spec-Small	mss-upd-aperiodic mss-acc-agent-initiated mss-avail-stored-data mss-cat-manual mss-cat-setting See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_MIN	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

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

The glucose sampling interval numeric type is MDC_CGM_SENSOR_SAMPLE_INTERVAL and the unit-code attribute for the glucose sampling interval numeric is minutes.

6.7.6 Glucose trend

Therapy used to provide glycemic control may take into consideration the change in blood glucose over time, or its slope. The glucose trend numeric provides this metric, and its attributes are summarized in Table 13.

Table 13—Glucose trend numeric object attributes

Attribute name	Value	Qual.
Type	{MDC_PART_PHD_DM MDC_CONC_GLU_TREND}	M
Metric-Spec-Small	See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_MILLI_G_PER_DL_PER_MIN or MDC_DIM_MILLI_MOLE_PER_L_PER_MIN	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R
Threshold-Notification-Text-String	See following text.	O

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

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

The glucose trend numeric type is MDC_CONC_GLU_TREND and the units-code attribute shall be MDC_DIM_MILLI_G_PER_DL_PER_MIN or MDC_DIM_MILLI_MOLE_PER_L_PER_MIN, as appropriate. The observed value shall be the change in glucose concentration measurements per minute.

6.7.6.1 Glucose trend threshold and status attributes

One attribute extending the glucose trend numeric object is provided to report the agent's glucose rate-of-change threshold details, and a second reports whether the measurement has reached or crossed beyond the threshold boundaries. The Measurement-Status attribute has been extended (compatible with ISO/IEEE 11073-10201:2004 [B6]) from the definition in ISO/IEEE 11073-20601 in order to report the threshold status. Note that the glucose rate-of-change thresholds (see 6.7.9) store the glucose trend numeric threshold values. See Table 14 for addition details.

Table 14—Glucose trend threshold and status attributes

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Threshold-Notification-Text-String	MDC_ATTR_THRES_NOTIF_TEXT_STRING	OCTET STRING	Text related to the current threshold notification.	Optional Observational
Measurement-Status	MDC_ATTR_MSMT_STAT	Measurement Status	Reports whether an observed value is at or outside threshold boundaries. If thresholding is to be used, this attribute is mandatory. Use bit msmt-value-exceed-boundaries(14), to indicate that the measurement is outside threshold boundaries. Use msmt-state-ann-inhibited(15) to indicate that the threshold indication is disabled and should not cause a displayed annunciation. If this bit is set, bit 14 shall not be set. These are bits extended from the ISO/IEEE 11073-20601 definitions of MeasurementStatus. The definition of all other bits of MeasurementStatus remains unchanged from their definition in ISO/IEEE 11073-20601:2010.	Conditional Observational

NOTE—See Annex B for ASN.1 bit mapping definition.

6.7.7 Patient low/high threshold

The patient low/high threshold numeric is a setting used to indicate a range of patient acceptable glucose concentrations. If glucose concentrations fall outside this range, a typical reaction is to notify the patient and log the event. Table 15 summarizes the attributes of the patient low/high threshold numeric object.

Table 15—Patient low/high threshold numeric object attributes

Attribute name	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CONC_GLU_PATIENT_THRESHOLDS_LOW_HIGH}	M
Metric-Spec-Small	See ISO/IEEE 11073-20601.	M
Metric-Structure-Small	See ISO/IEEE 11073-20601.	M
Metric-Id-List	See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PER_L	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Compound-Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The patient low/high thresholds numeric object does not support any methods, events, or other services.

The patient low/high thresholds numeric type is MDC_CONC_GLU_PATIENT_THRESHOLDS_LOW_HIGH and the units-code attribute shall be MDC_DIM_MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PER_L, as appropriate. The patient low/high thresholds compound observed value attribute shall include first the patient low threshold, MDC_CONC_GLU_PATIENT_THRESHOLD_LOW, followed by the patient high threshold, MDC_CONC_GLU_PATIENT_THRESHOLD_HIGH.

6.7.8 Device hypo/hyper thresholds

The device hypo/hyper thresholds numeric is a setting to indicate the critical glucose concentration range. Table 16 summarizes the attributes of the device hypo/hyper thresholds numeric object.

Table 16—Device hypo/hyper thresholds numeric object attributes

Attribute name	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CONC_GLU_THRESHOLDS_HYPO_HYPER}	M
Metric-Spec-Small	See ISO/IEEE 11073-20601.	M
Metric-Structure-Small	See ISO/IEEE 11073-20601.	M
Metric-Id-List	See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PER_L	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Compound-Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The device hypo/hyper thresholds numeric object does not support any methods, events, or other services.

The device hypo/hyper threshold numeric type is MDC_CONC_GLU_PATIENT_THRESHOLDS_HYPO_HYPER and the units-code attribute shall be MDC_DIM_MILLI_G_PER_DL or MDC_DIM_MILLI_MOLE_PER_L, as appropriate. The device hypo/hyper thresholds compound observed value attribute shall include first the device hypo threshold,

MDC_CONC_GLU_PATIENT_THRESHOLD_HYPO, followed by the device hyper threshold, MDC_CONC_GLU_PATIENT_THRESHOLD_HYPER.

6.7.9 Glucose rate-of-change thresholds

The glucose rate-of-change thresholds numeric is a setting to indicate the maximum rate of glucose variation. Table 17 summarizes the attributes of the glucose rate-of-change thresholds numeric object.

Table 17 —Glucose rate-of-change thresholds numeric object attributes

Attribute name	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CONC_GLU_RATE_THRESHOLDS }	M
Metric-Spec-Small	See ISO/IEEE 11073-20601.	M
Metric-Structure-Small	See ISO/IEEE 11073-20601.	M
Metric-Id-List	See ISO/IEEE 11073-20601.	M
Unit-Code	MDC_DIM_MILLI_G_PER_DL_PER_MIN or MDC_DIM_MILLI_MOLE_PER_L_PER_MIN	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Compound-Basic-Nu-Observed-Value	See ISO/IEEE 11073-20601.	R

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The glucose rate-of-change thresholds numeric object does not support any methods, events, or other services.

The glucose rate-of-change thresholds numeric type is MDC_CONC_GLU_RATE_THRESHOLDS and the units-code attribute shall be MDC_DIM_MILLI_G_PER_DL_PER_MIN or MDC_DIM_MILLI_MOLE_PER_L_PER_MIN, as appropriate. The glucose rate-of-change thresholds compound observed value attribute shall include first the glucose rate increase threshold, MDC_CONC_GLU_RATE_THRESHOLD_INCREASE, followed by the glucose rate decrease threshold, MDC_CONC_GLU_RATE_THRESHOLD_DECREASE.

6.8 Real-time sample array objects

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

6.9 Enumeration objects

6.9.1 General

The CGM DIM (see Figure 2) contains enumeration objects that represent the general device status and CGM specific status. The nomenclature code to identify the enumeration class is MDC_MOC_VMO_METRIC_ENUM. Subclauses 6.9.2 and 6.9.3 define the precise definitions for both general and specific CGM status enumeration objects. Table 18 shows the common attributes for all the enumeration objects.

Enumeration objects do not support any methods, events, or other services.

Table 18 —Common enumeration object attributes

Attribute name	Value	Qual.
Handle	See ISO/IEEE 11073-20601.	M
Type	Defined in the following subclauses.	M
Supplemental-Types	See ISO/IEEE 11073-20601.	O
Metric-Spec-Small	Defined in the following subclauses.	M
Metric-Structure-Small	See ISO/IEEE 11073-20601.	O
Measurement-Status	See ISO/IEEE 11073-20601.	C

Attribute name	Value	Qual.
Metric-Id	See ISO/IEEE 11073-20601.	O
Metric-Id-List	See ISO/IEEE 11073-20601.	C
Metric-Id-Partition	See ISO/IEEE 11073-20601.	O
Unit-Code	See ISO/IEEE 11073-20601.	O
Attribute-Value-Map	See ISO/IEEE 11073-20601.	C
Source-Handle-Reference	See ISO/IEEE 11073-20601.	O
Label-String	See ISO/IEEE 11073-20601.	O
Unit-LabelString	See ISO/IEEE 11073-20601.	O
Absolute-Time-Stamp	See ISO/IEEE 11073-20601.	C
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	C
Relative-Time-Stamp	See ISO/IEEE 11073-20601.	C
HiRes-Time-Stamp	See ISO/IEEE 11073-20601.	C
Measure-Active-Period	See ISO/IEEE 11073-20601.	O
Enum-Observed-Value-Simple-OID	See ISO/IEEE 11073-20601.	C
Enum-Observed-Value-Simple-Bit-Str	See ISO/IEEE 11073-20601.	C
Enum-Observed-Value-Basic-Bit-Str	See ISO/IEEE 11073-20601.	C
Enum-Observed-Value-Simple-Str	See ISO/IEEE 11073-20601.	C
Enum-Observed-Value	See ISO/IEEE 11073-20601.	C
Enum-Observed-Value-Partition	See ISO/IEEE 11073-20601.	O

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

6.9.2 PHD DM status

The PHD DM status object allows generic device events to be recorded in order to track important events for the user and troubleshooting information for manufacturers. In the case where the CGM is more than one physical device, e.g., sensor, transmitter, or receiver, and these PHD DM status events are recorded in the CGM agent for each physical device, then there shall be only one instance of the PHD DM status object for each physical device, and the *Supplemental-Types* attribute shall be used to clarify which physical device. There shall not be two PHD DM status objects with the same supplemental-type. Table 19 defines the attributes for the object that represents the PHD DM status. The PHD DM status enumeration object may be supported by a CGM agent.

Table 19—PHD DM status enumeration object attributes

Attribute name	Extended configuration	Qual.
Type	{ MDC_PART_PHD_DM, MDC_PHD_DM_DEV_STAT }	M
Supplemental-Types	{ MDC_PART_PHD_DM, MDC_CGM_DEV_TYPE_SENSOR or MDC_CGM_DEV_TYPE_TRANSMITTER or MDC_CGM_DEV_TYPE_RECEIVER or MDC_CGM_DEV_TYPE_OTHER } See ISO/IEEE 11073-20601.	R
Metric-Spec-Small	mss-avail-intermittent mss-avail-stored-data mss-upd-aperiodic mss-acc-agent-initiated mss-acc-manager-initiated	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Enum-Observed-Value-Simple-Bit-Str	See following text.	M
Capability-Mask-Simple	See 6.2.	M
State-Flag-Simple	See 6.2.	M

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The observed value reported in this object is the general device status.

Since these are essentially event flags, the Unit-Code attribute is not appropriate for this object. Similarly, the Source-Handle-Reference is inappropriate, as this object monitors the status of the equipment.

The explicit expression of the existence of annunciations is realized by the setting of the appropriate bit in the Enum-Observed-Value-Simple-Bit-Str attribute, as defined in Table 20. If a manager supports this object, it shall be able to interpret the entire set of presented conditions. Anytime the status changes for any monitored condition, the agent shall report the status of all the monitored conditions.

The detection of the condition change may take time. In case there is a delay in detecting the start or stop of a condition, then the event shall be reported with a time stamp that is the time of the occurrence of the respective event rather than the time that the event is reported.

If an acceptable, existing bit is not available, device-status-undetermined shall be. 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 20—Mapping of PHD DM status to object Bit-Str attribute

Bit	PHD DM status condition	PHDDMStat mnemonic
0	Agent reports that an undetermined or not supported condition occurred.	device-status-undetermined
1	Agent reports that a reset has occurred.	device-status-reset
5	Agent reports that a general fault occurred.	device-status-error
6	Agent reports that a mechanical fault occurred.	device-status-error-mechanical
7	Agent reports that an electronic fault occurred.	device-status-error-electronic
8	Agent reports that a software error occurred.	device-status-error-software
9	Agent reports that a battery fault occurred.	device-status-error-battery
15	Agent reports that a general service is required.	device-status-service
16	Agent reports that a time synchronization is required.	device-status-service-time-sync-required
17	Agent reports that a calibration is required.	device-status-service-calibration-required
18	Agent reports that a component replenishment is required.	device-status-service-replenishment-required
25	Agent reports that battery power is low.	device-status-battery-low
26	Agent reports that battery is depleted.	device-status-battery-depleted
27	Agent reports that battery has been replaced.	device-status-battery-replaced
28	Agent reports that battery is interrupted.	device-status-battery-interrupted

NOTE 1—The bits in Table 20 are defined as: 0 = False and 1 = True.

NOTE 2—The specific bit mappings of PHDDMStat are defined in Annex B.

NOTE 3—All bits not defined in Table 20 or Annex B are reserved for future use.

NOTE 4—An agent is not required to implement all the features specified in Table 20.

6.9.3 CGM status

The CGM status enumeration object allows specific running status, calibration states, notifications, errors, etc., for the CGM system. This enumeration object differs from the PHD DM status in 6.9.2 as it provides additional status codes specific to the CGM system. An enumeration object fulfills this need. If this object is to be implemented, then the object type and bit assignments shall be implemented as described. Table 21 summarizes the attributes of the CGM status enumeration object.

Table 21 —CGM status attributes

Attribute name	Extended configuration	
	Value	Qual.
Type	{MDC_PART_PHD_DM, MDC_CGM_DEV_STAT}	M
Metric-Spec-Small	mss-avail-intermittent mss-avail-stored-data mss-upd-aperiodic mss-msmt-aperiodic mss-acc-agent-initiated	M
Base-Offset-Time-Stamp	See ISO/IEEE 11073-20601.	R
Enum-Observed-Value-Simple-Bit-Str	See following text.	M
Capability-Mask-Simple	See 6.2.	M
State-Flag-Simple	See 6.2.	M

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The CGM status enumeration object does not support any methods, events, or other services.

An agent explicitly expresses the existence of the CGM status by setting the appropriate bits in the Enum-Observed-Value-Simple-Bit-Str attribute, as defined in Table 22. It is recommended to use the Enum-Observed-Value-Simple-Bit-Str attribute as the currently available status options are greater than what the Enum-Observed-Value-Basic-Bit-Str attribute allows. 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 22 —Mapping of device, sensor, and signal status to object Bit-Str attribute

Bit	Device or sensor condition	CGMStat mnemonic
0	Session stopped	sensor-session-stopped
2	Sensor type incorrect for device	sensor-type-incorrect
3	Sensor malfunction	sensor-malfunction
4	Device Specific Alert	device-specific-alert
7	Calibration not allowed	sensor-calibration-not-allowed
8	Calibration recommended	sensor-calibration-recommended
9	Calibration required	sensor-calibration-required
10	Sensor temperature too high for valid test/result at time of measurement	sensor-temp-too-high
11	Sensor temperature too low for valid test/result at time of measurement	sensor-temp-too-low
12	Sensor result lower than the Patient Low level	sensor-result-below-patient-low
13	Sensor result higher than the Patient High level	sensor-result-above-patient-high
14	Sensor result lower than the Hypo level	sensor-low-hypo
15	Sensor result higher than the Hyper level	sensor-high-hyper
16	Sensor Rate of Decrease exceeded	sensor-rate-decrease-exceeded
17	Sensor Rate of Increase exceeded	sensor-rate-increase-exceeded
18	Sensor result lower than the device can process	sensor-result-too-low
19	Sensor result higher than the device can process	sensor-result-too-high
20	Sensor communication is out of range	sensor-com-out-of-range

NOTE 1—The bits in Table 22 are defined as: 0 = False and 1 = True.

NOTE 2—The specific bit mappings of CGMStat are defined in Annex B.

NOTE 3—All bits not defined in Table 22 or Annex B are reserved for future use.

6.10 PM-store objects

6.10.1 General

In the context of PHDs, CGMs are portable or mobile devices and are typically physically attached to the user. Thus, CGM agents may be used to collect measurements or observations at a time when out of the network and agent/manager associations cannot be established. It is also common that a given set of measurements made by CGM agents 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 HCPs.

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 ISO/IEEE 11073-20601 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 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.

6.10.2 Persistent store model

The PM-store model defined by this standard utilizes one or more PM-segments for the data of each object to be persistently stored (see Figure 3, for example). A segment holding glucose measurements shall be present if a PM-store is implemented. The other segments are optional and hold observations from the supporting objects that are implemented.

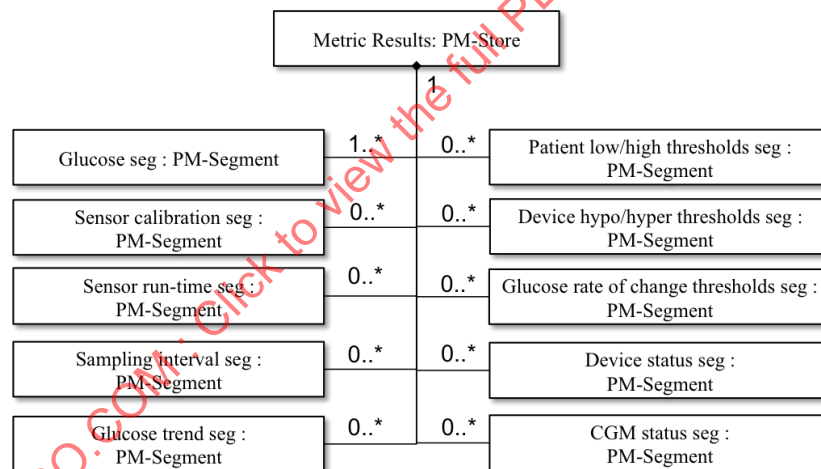


Figure 3—CGM persistent store model example

Each entry shall include one of the time formats in the segm-entry-header so a manager can correlate entries across the different segments. If a particular object is not supported, the corresponding segment is not required to exist. Each segment has a cardinality of zero-to-many or one-to-many, as PM-segments are required to contain data from a contiguous period of time (see ISO/IEEE 11073-20601). Therefore changing time and/or date on the agent typically results in the creation of new segment instances for the supported measurement or observation objects. Furthermore, a CGM agent may subdivide data from one contiguous period of time into several segments for further clustering of data (e.g., one segment per day or for an uninterrupted time span of the CGM being in operating mode). If a particular segment resulting from such time/date changes or clustering does not contain any entries, it is not required to exist.

Note that the PM-store object is not part of standard configurations defined in this standard.

Following the guides provided in this standard should enable an implementer to store and retrieve the data within this model, but the specifics for determining the specific nature of the data layout and the subsequent visualization, mining, or other managing of the retrieved data are outside the scope of this standard.

6.10.3 PM-store object attributes

Table 23 defines the attributes of the PM-store object that shall be implemented by the agent. The nomenclature code to identify the PM-store objects is MDC_MOC_VMO_PMSTORE.

Table 23 —PM-store object attributes

Attribute name	Extended configuration	
	Value	Qual.
Handle	See ISO/IEEE 11073-20601.	M
PM-Store-Capab	See ISO/IEEE 11073-20601.	M
Store-Sample-Algorithm	See ISO/IEEE 11073-20601.	M
Store-Capacity-Count	See ISO/IEEE 11073-20601.	M
Store-Usage-Count	See ISO/IEEE 11073-20601.	M
Operational-State	See ISO/IEEE 11073-20601.	M
PM-Store-Label	See ISO/IEEE 11073-20601.	O
Sample-Period	See ISO/IEEE 11073-20601.	NR
Number-Of-Segments	See ISO/IEEE 11073-20601.	M
Clear-Timeout	See ISO/IEEE 11073-20601.	M

The PM-Store-Capab attribute shall set the following bits as indicated:

— **pmssc-var-no-of-segm:**

If the agent creates new segments either due to storing data of multiple sessions or due to time changes as described in 8.12.2.2 of ISO/IEEE 11073-20601, then pmssc-var-no-of-segm shall be set.

— **pmssc-epi-seg-entries:**

The pmssc-epi-seg-entries bit shall be set.

— **pmssc-peri-seg-entries:**

The pmssc-peri-seg-entries bit shall not be set.

The remaining bits of the PM-Store-Capab attribute are agent specific and shall be set appropriately.

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

6.10.4 PM-store object methods

Table 24 defines the methods of the PM-store objects.

Table 24 —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	
	Get-Segment-Info	Confirmed	MDC_ACT_SEG_GET_INFO	SegmSelection	SegmentInfoList
	Trig-Segment-Data-Xfer	Confirmed	MDC_ACT_SEG_TRIG_XFER	TrigSegmDataXferReq	TrigSegmDataXferRsp

— **Clear-Segments:**

This method allows the manager to delete all data entries stored in a PM-segment object. The agent shall support the Clear-Segments method by setting the pmssc-clear-segm-by-all-sup bit for the PM-Store-Capab attribute. Deletion of PM-segments is not guaranteed by this method. See ISO/IEEE 11073-20601 for information on how the agent shall reply in case it decides to protect certain segments from deletion.

— **Get-Segment-Info:**

This method allows the manager to retrieve the PM-segment attributes.

— **Trig-Segment-Data-Xfer:**

This method allows the manager to initiate the transfer of the data entries stored in the PM-segment object.

Refer to ISO/IEEE 11073-20601 for details.

6.10.5 PM-store object events

Table 25 defines the events sent by the PM-store objects.

Table 25 —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. Refer to ISO/IEEE 11073-20601 for details.

6.10.6 PM-store object services

6.10.6.1 GET service

The GET service shall be provided by an agent implementing PM-store objects. This service shall be available only while the agent is in the Operating state. Refer to ISO/IEEE 11073-20601 for details.

6.10.6.2 SET service

There are currently no SET services defined for PM-store objects in this standard.

6.10.7 PM-segment objects

Table 26 defines the attributes of the periodic session PM-segment object contained in the periodic PM-store object managing the stored measurements or observations. The nomenclature code to identify the PM-segment class is MDC_MOC_PM_SEGMENT.

Table 26—Common PM-segment object attributes

Attribute name	Extended configuration	
	Value	Qual.
Instance-Number	See ISO/IEEE 11073-20601.	M
PM-Segment-Entry-Map	See ISO/IEEE 11073-20601.	M
PM-Seg-Person-Id	See ISO/IEEE 11073-20601.	C
Operational-State	See ISO/IEEE 11073-20601.	M
Sample-Period	See ISO/IEEE 11073-20601.	C
Segment-Label	See ISO/IEEE 11073-20601.	O
Segment-Start-Abs-Time	See ISO/IEEE 11073-20601.	C
Segment-End-Abs-Time	See ISO/IEEE 11073-20601.	C
Date-and-Time-Adjustment	See ISO/IEEE 11073-20601.	C
Segment-Start-BO-Time	See ISO/IEEE 11073-20601.	C
Segment-End-BO-Time	See ISO/IEEE 11073-20601.	C
Segment-Usage-Count	See ISO/IEEE 11073-20601.	M
Segment-Statistics	See ISO/IEEE 11073-20601.	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 ISO/IEEE 11073-20601.	O
Transfer-Timeout	See ISO/IEEE 11073-20601.	M

NOTE 1—See ISO/IEEE 11073-20601 for information on whether an attribute is static or dynamic.

NOTE 2—See 6.3 for a description of the qualifiers.

The Fixed-Segment-Data attribute serves as the container of the stored measurements or observations. 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 contains an optional header and one or more elements. Each element holds data from one or more metric measurements.

6.11 Scanner objects

Scanner objects are not required by this standard.

6.12 Class extension objects

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

6.13 CGM information model extensibility rules

The CGM DIM of this standard may be extended by including elements defined in ISO/IEEE 11073-20601 as well as vendor-specific metrics and attributes as required. Any object or attribute extensions implemented should follow the guidelines of this standard as closely as possible. Such vendor-specific objects and attributes shall be identified by assigning nomenclature codes from the private numbering space (0xF000 – 0xFFFF) within the appropriate partition as defined in ISO/IEEE 11073-20601.

A CGM 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 ISO/IEEE 11073-20601).

7. CGM 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 ISO/IEEE 11073-20601 for a detailed description of the PHD service model. Subclauses 7.2 and 7.3 define the specifics of object access and event reporting services for a CGM agent according to this standard.

7.2 Object access services

The object access services of ISO/IEEE 11073-20601 are used to access the objects defined in the DIM of the glucose device.

The following generic object access services are supported by a CGM 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 CGM MDS object attributes is given in 6.6.4.1, and the list of CGM PM Store attributes is given in 6.10.3.
- SET service: used by the manager to set the values of the agent object attributes. No settable attributes are defined for a CGM 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 CGM device specialization is given in 6.6.3.
- 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 27 summarizes the object access services described in this standard.

Table 27 —CGM object access services

Service	Subservice type name	Mode	Subservice type	Parameters	Result	Remarks
GET	<na>	<implied Confirmed>	<na>	GetArgumentSimple = (obj-handle = 0), attribute-id-list <optional>	GetResultSimple = (obj-handle = 0), attribute-list	Allows the manager to retrieve the value of attributes of the MDS object in the agent.
	<na>	<implied Confirmed>	<na>	GetArgumentSimple = (obj-handle = <i>handle of PM-store object</i>), attribute-id-list <optional>	GetResultSimple = (obj-handle = <i>handle of PM-store object</i>), attribute-list	Allows the manager to retrieve the values of attributes of a PM-store object in the agent.
EVENT REPORT	MDS-Configuration-Event	Confirmed	MDC_NOTI_CONFIG	ConfigReport	ConfigReportRsp	Configuration Report to inform manager of the configuration of the agent.
	MDS-Dynamic-Data-Update-Var	Confirmed	MDC_NOTI_SCAN_REPORT_VAR	ScanReportInfoVar	—	Data Report to provide dynamic data to manager for some or all of the agent's objects in variable format.
	MDS-Dynamic-Data-Update-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_FIXED	ScanReportInfoFixed	—	Data Report to provide dynamic data to manager for some or all of the agent's objects in fixed format.
	MDS-Dynamic-Data-Update-MP-Var	Confirmed	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	—	This is the same as MDS-Dynamic-Data-Update-Var but allows inclusion of data from multiple people.
	MDS-Dynamic-Data-Update-MP-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	—	This is the same as MDS-Dynamic-Data-Update-Fixed but allows inclusion of data from multiple people.
	Segment-Data-Event	Confirmed	MDC_NOTI_SEGMENT_DATA	SegmentDataEvent	SegmentDataResult	PM-store object event to provide data stored in the Fixed-Segment-Data of a PM-segment from the agent to the manager.
ACTION	Set-Time	Confirmed	MDC_ACT_SET_TIME	SetTimeInvoke	—	Manager method to invoke the agent to set time in absolute time format to requested value.
	Set-Base-Offset-Time	Confirmed	MDC_ACT_SET_BO_TIME	SetBOTimeInvoke	—	Manager method to invoke the agent to set time in base offset time format to requested value.
	Clear-Segments	Confirmed	MDC_ACT_SEG_CLR	SegmSelection	—	Allows the manager to delete data stored in selected PM-segments in the agent.
	Get-Segment-Info	Confirmed	MDC_ACT_SEG_GET_INFO	SegmSelection	SegmentInfoList	Allows the manager to retrieve the value of PM-segment attributes of one or more PM-segments in the agent.
	Trig-Segment-Data-Xfer	Confirmed	MDC_ACT_SEG_TRIG_XFER	TrigSegmDataXferReq	TrigSegmDataXferRsp	Allows the manager to start the transfer of the Fixed-Segment-Data attribute of a PM-segment in the agent.

7.3 Object access event report services

The event report service (see Table 27) is used by the agent to report its information (e.g., measurements). Event reports in this standard are a property of the MDS (see Table 4) and the PM-store object (see Table 25). The event reports used in this standard are defined in ISO/IEEE 11073-20601.

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

- MDS event reports shall be used in confirmed mode.
- Agent initiated mode shall be supported for measurement data transmission.
- Persistently stored metric mode may be supported for measurement data transmission.
- Manager initiated mode may be support for measurement data transmission.

A CGM agent, which is 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 helped reducing overhead.

A manager shall support both single-person and multiple-person event reports. A CGM 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 ISO/IEEE 11073-20601.

8. CGM communication model

8.1 Overview

This clause describes the general communication model and procedures of the CGM agent as defined in ISO/IEEE 11073-20601. Therefore, the respective parts of ISO/IEEE 11073-20601 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.

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 CGM agent are defined. Small limits allow for simple implementations in terms of low cost and complexity.

A CGM 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 896 octets. For this standard, N_{rx} shall be 224 octets.

For a CGM 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 ISO/IEEE 11073-20601, 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 CGM agent and manager according to this standard shall be pursued as specified in ISO/IEEE 11073-20601.

Table 28 lists the valid combinations of protocol version and nomenclature version. In the association procedure, an agent indicating support to a specific protocol version shall indicate support to the corresponding nomenclature version as well. In the association procedure, a manager selecting a specific protocol version shall select the corresponding nomenclature version.

To indicate support for multiple protocol versions, the bit values are combined. For example, if the agent supports protocol-version2, protocol-version3, and protocol-version4, it shall use protocol version bits 0x70000000 and nomenclature-version bits 0xE0000000.

Future versions of this specification may include further valid combinations that can be used by implementations that comply with that future version.

Table 28—Valid combinations of protocol and nomenclature version

Protocol version	Bit value	Corresponding nomenclature version	Bit value
1	0x80000000	1	0x80000000
2	0x40000000	1	0x80000000
3	0x20000000	2	0x40000000
4	0x10000000	3	0x20000000

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:
 - The version of the data exchange protocol shall be set to *protocol-version4* (i.e., *protocol-version* = 0x10000000). Support for any other version may be indicated by setting additional bits. When protocols lower than *protocol-version4* are used, the agent shall use only features in that protocol.
 - At least the *MDER* shall be supported (i.e., *encoding-rules* = 0x8000).
 - The protocol version bits and nomenclature version bits shall consist of valid combinations of bits as defined in Table 28.
 - The field *functional-units* may have the test association bits set but shall not have any other bits set.
 - The field *system-type* shall be set to *sys-type-agent* (i.e., *system-type* = 0x00800000).
 - The *system-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 CGM with which it is associating and, optionally, to implement a simple access restriction policy.
 - The *dev-config-id* field shall be set to the value of the *Dev-Configuration-Id* attribute of the *MDS* object of the agent.
 - If the agent supports only the CGM specialization, then the field indicating the data request modes (*data-req-mode-capab*) supported by the CGM agent shall be set to *data-req-supp-init-agent*.
 - If the agent supports only the CGM 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 ISO/IEEE 11073-20601. 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:
 - The manager following this specialization shall support *protocol-version4*. The manager may support additional protocol versions and select them if the agent offers them. When protocols lower than *protocol-version4* are used, the manager shall use only features in that protocol.
 - 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 MDER.
 - The manager shall select a valid combination of protocol version and nomenclature version as defined in Table 28.
 - The field *functional-units* shall have all bits reset except for those relating to a test association.
 - The field *system-type* shall be set to *sys-type-manager* (i.e., *system-type* = 0x80000000).
 - The *system-id* field shall contain the unique system ID of the manager, which shall be a valid EUI-64 type identifier.
 - The field *dev-config-id* shall be *manager-config-response* (0).
 - The field *data-req-mode-capab* shall be 0.
 - If the agent supports only the CGM specialization, *data-req-initagent-count* shall be 1 and *data-req-init-manager-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 ISO/IEEE 11073-20601 shall be followed. Subclause 8.4.2 specifies the configuration notification and response messages for a CGM agent with standard configuration ID 2500 (0x09C4). Normally, a manager would already know the standard configuration. However, for the purposes of this example, it does not.

8.4.2 CGM—standard configuration (0x09C4)

8.4.2.1 Agent procedure

The agent performs the configuration procedure using a “Remote Operation Invoke | Confirmed Event Report” message with an *MDC_NOTL_CONFIG* event to send its configuration to the manager (see ISO/IEEE 11073-20601). The *ConfigReport* structure is used for the *event-info* field (see Table 4). For a CGM agent with standard configuration ID 2500 (0x09C4), the format and contents of the configuration notification message are as follows:

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x50	CHOICE.length = 80
0x00 0x4E	OCTET STRING.length = 78
0x00 0x02	invoke-id = 2 (start of DataApdu. MDER encoded.)
0x01 0x01	CHOICE(Remote Operation Invoke Confirmed Event Report)
0x00 0x48	CHOICE.length = 72
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 0x3E	event-info.length = 62 (start of ConfigReport)
0x09 0xC4	config-report-id (Dev-Configuration-Id value)
0x00 0x01	config-obj-list.count = 1 Measurement object will be “announced”
0x00 0x38	config-obj-list.length = 56
0x00 0x06	obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01	obj-handle = 1 (→ 1 st Measurement is glucose)
0x00 0x05	attributes.count = 5
0x00 0x30	attributes.length = 48
0x09 0x2F	attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04	attribute-value.length = 4
0x00 0x02 0x71 0xD4	MDC_PART_SCADA MDC_CONC_GLU_ISF
0x0A 0x61	attribute-id = MDC_ATTR_SUPPLEMENTAL_TYPES
0x00 0x08	attribute-value.length = 8
0x00 0x01	SupplementalTypeList.count = 1
0x00 0x04	SupplementalTypeList.length = 4
0x00 0x80 0x72 0x39	MDC_PART_PHD_DM
	MDC_CTXT_GLU_SAMPLELOCATION_SUBCUTANEOUS
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0xC0 0x42	mss-avail-intermittent, mss-avail-stored-data, mss-acc-agent-initiated,
	mss-cat-calculation
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_VALUE_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

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 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)
0x00 0x00 0x00 0x00	currentTime = 0
0x0D 0x1C	event-type = MDC_NOTI_CONFIG
0x00 0x04	event-reply-info.length = 4
0x09 0xC4	ConfigReportRsp.config-report-id = 2500
0x00 0x00	ConfigReportRsp.config-result = accepted-config

8.5 Operating procedure

8.5.1 General

Measurement data and status information are communicated from the CGM 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 ISO/IEEE 11073-20601.

8.5.2 GET CGM MDS attributes

See Table 5 for a summary of the GET service.

Refer to the respective standard for the details of getting CGM MDS attributes for the specific version of the protocol.

8.5.3 Measurement data transmission

See Table 4 and Table 25 for a summary of the event report services available for measurement data transfer.

To limit the amount of data being transported within an APDU, the CGM 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.

8.6 Time synchronization

Time synchronization may be employed between a CGM 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

The test association provides a manufacturer with the mechanism to test or demonstrate features of a product in a comprehensive manner. This clause defines the behavior of the standard CGM agent during a test association. Support for test association is optional.

9.1 Behavior with standard configuration

An agent or manager entering a test association using the configuration ID for the standard CGM device of this standard shall enter the Operating state in test mode. When in test mode, where possible, this should be indicated visually to any user. Normal functionality shall be suspended, and any test data generated shall not be processed by the device as physiological data.

After the agent enters the operating state, it shall send a single simulated glucose measurement of 999 mg/dL (a value never seen in normal usage and outside normal range) within 30 s of entering the Operating state. If the measurement-status attribute of the numeric object is implemented, then the test-data bit shall be set.

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 ISO/IEEE 11073-20601.

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

- DIM 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 the following:

- 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 (ICSs) as detailed in 10.4.

This standard is used in conjunction with ISO/IEEE 11073-20601. It is recommended that the ICS for this standard be created first so that the ICS created for ISO/IEEE 11073-20601 may 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 ISO/IEEE 11073-20601 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 ISO/IEEE 11073-20601 and ISO/IEEE 11073-104zz documents.

10.3.3 Conformance level 2: Extended nomenclature (ASN.1 and/or ISO/IEEE 11073-10101)

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 structures defined using ASN.1 and/or nomenclature codes within the ISO/IEEE 11073-10101 framework (0xF000 through 0xFFFF). These extensions should be defined in ICS tables pointing toward their reference.

10.4 Implementation conformance statements (ICSs)

10.4.1 General format

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

Each ICS table has the following columns:

Index	Feature	Reference	Req/Status	Support	Comment
-------	---------	-----------	------------	---------	---------

The table column headings have the following meaning:

- **Index**, is an identifier (e.g., a tag) of a specific feature.
- **Feature**, briefly describes the characteristic for which a conformance statement is being made.
- **Reference**, is a reference to the clause or subclause within this standard or to an external source for the definition of the feature (may be empty).
- **Req/Status**, specifies the conformance requirement (e.g., mandatory, 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 to 10.4.6 specify the format of the specific ICS tables.

10.4.2 General ICS

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

Table 29 shows the general ICSs.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEEE 11073-10425:2024

Table 29 —IEEE 11073-10425 general ICS table

Index ^a	Feature	Reference	Req/Status	Support	Comment
GEN11073-10425-1	Implementation Description	—	Identification of the device/ application. Description of functionality.		
GEN11073-10425-2	Standards followed and their revisions	(Standard documents)	(Set of existing revisions)	(Set of supported revision)	
GEN11073-10425-3	Nomenclature document used and revision	(Standard documents)	(Set of existing revisions)	(Set of supported revisions)	
GEN11073-10425-4	Conformance Adherence— Level 1	See 10.3.3	Base conformance declaration that device meets the following IEEE 11073-10425 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)	
GEN11073-10425-5	Conformance Adherence— Level 2	See 6.3	In addition to GEN11073-10425-4, if the device implements extensions and/or additions, they shall conform to nomenclature codes from ASN.1 and/or ISO/IEEE 11073-10101 framework. These extensions should also be defined in ICS tables pointing toward their reference.	Yes/No	
GEN11073-10425-6	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.		
GEN11073-10425-7	Nomenclature document used and revision	(Standard documents)	(Set of existing revisions)	(Set of supported revision)	
GEN11073-10425-8	Data Structure Encoding	—	—	Description of encoding method(s) for ASN.1 data structures	

(Table continues on the next page.)

Table 28—IEEE 11073-10425 general ICS table (*continued*)

Index ^a	Feature	Reference	Req/Status	Support	Comment
GEN11073-10425-9	Use of Private Objects	—	Does the implementation use objects that are not defined in the DIM?	Yes/No (If yes, explain in Table 30)	
GEN11073-10425-10	Use of Private Nomenclature Extensions	—	Does the implementation use private extensions to the nomenclature (i.e., 0xF000-0xFFFF codes from ISO/IEEE 11073-10101)? Private Nomenclature extensions are allowed <i>only</i> if the standard nomenclature does not include the specific terms required by the application.	Yes/No (If yes: explain in Table 33)	
GEN11073-10425-11	11073-20601 Conformance		Provide the conformance report required by ISO/IEEE 11073-20601		

^a The prefix GEN11073-10425- is used for the index in the general ICS table.

10.4.3 DIM MOC ICS

The DIM MOC ICS defines which objects are implemented. Information on each object shall be provided as a separate row in the template of Table 30.

Table 30—Template for DIM MOC ICS table

Index	Feature	Reference	Req/Status	Support	Comment
MOC- <i>n</i>	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

The MOC attribute ICS defines which attributes, including any inherited attributes, are used/supported in each object of an implementation. Information on each attribute of an object shall be provided as a separate row in the template of Table 31. A separate MOC attribute ICS shall be provided for each object.

Table 31—Template for MOC attribute ICS table

Index	Feature	Reference	Req/Status	Support	Comment
ATTR- <i>n-x</i>	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 the following:

- 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
- Any other applicable 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*).

10.4.5 MOC notification ICS

The MOC notification ICS specifies all implemented notifications (typically in form of the event report service) that are emitted by the agent. Table 32 provides a template for use. One table has to be provided for each object that supports special object notifications.

Table 32—Template for MOC notification ICS table

Index	Feature	Reference	Req/Status	Support	Comment
NOTI- <i>n-x</i>	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 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 ICS

The MOC nomenclature ICS specifies all nonstandard nomenclature codes that are utilized by the agent. Table 33 provides a template for use. One row of the table is to be used for each nomenclature element.

Table 33—Template for MOC nomenclature ICS table

Index	Feature	Reference	Req/Status	Support	Comment
NOME- <i>n</i>	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. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

[B1] IEC 60601-1, Medical electrical equipment—Part 1: General requirements for basic safety and essential performance.⁸

[B2] IEC 60601-2, Medical electrical equipment—Part 2: Particular requirements for the basic safety and essential performance for specific device. (See the entire series of standards, Part 2-1 through Part 2-51.)

[B3] IEC 62304:2006, Medical device software—Software life-cycle processes.

[B4] IEC 80001-1, Application of risk management for IT-networks incorporating medical devices—Part 1: Roles, responsibilities, and activities.

[B5] ISO 14971:2007, Medical devices—Application of risk management to medical devices.⁹

[B6] ISO/IEEE 11073-10201, Health informatics—Point-of-care medical device communication—Part 10201: Domain information model.¹⁰

[B7] ISO/IEEE 11073-20101, Health informatics—Point-of-care medical device communication—Part 20101: Application profile—Base standard.

[B8] ITU-T Rec. X.680, Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation.¹¹

⁸ IEC publications are available from the International Electrotechnical Commission (<https://www.iec.ch>) and the American National Standards Institute (<https://www.ansi.org/>).

⁹ ISO publications are available from the International Organization for Standardization (<https://www.iso.org/>) and the American National Standards Institute (<https://www.ansi.org/>).

¹⁰ ISO/IEEE publications are available from the ISO Central Secretariat (<http://www.iso.ch/>). ISO/IEEE publications are also available in the United States from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

¹¹ ITU-T publications are available from the International Telecommunications Union (<https://www.itu.int/>).

Annex B

(normative)

Any additional ASN.1 definitions

B.1 PHD DM status, CGM status, and measurement status bit mappings

The extension to the enumeration class for PHD DM status requires the following ASN.1 structure definition:

```
PHDDMStat ::= BITS-32 {
    device-status-undetermined (0),
    device-status-reset (1),
    -- reserved for future extension (2),
    -- reserved for future extension (3),
    -- reserved for future extension (4),
    device-status-error (5),
    device-status-error-mechanical (6),
    device-status-error-electronic (7),
    device-status-error-software (8),
    device-status-error-battery (9),
    -- reserved for future extension (10),
    -- reserved for future extension (11),
    -- reserved for future extension (12),
    -- reserved for future extension (13),
    -- reserved for future extension (14),
    device-status-service (15),
    device-status-service-time-sync-required (16),
    device-status-service-calibration-required (17),
    device-status-service-replenishment-required (18),
    -- reserved for future extension (19),
    -- reserved for future extension (20),
    -- reserved for future extension (21),
    -- reserved for future extension (22),
    -- reserved for future extension (23),
    -- reserved for future extension (24),
    device-status-battery-low (25),
    device-status-battery-depleted (26),
    device-status-battery-replaced (27),
    device-status-battery-interrupted (28),
    -- reserved for future extension (29),
    -- reserved for future extension (30),
    -- reserved for future extension (31),
}
```

The CGM status enumeration object requires the following ASN.1 structure definition:

```
CGMStat ::= BITS-32 {
    sensor-session-stopped(0),
    sensor-type-incorrect(2),
    sensor-malfunction(3),
    device-specific-alert(4),
    sensor-calibration-not-allowed(7),
    sensor-calibration-recommended(8),
    sensor-calibration-required(9),
    sensor-temp-too-high(10),
    sensor-temp-too-low(11),
}
```

```

    sensor-result-below-patient-low(12),
    sensor-result-above-patient-high(13),
    sensor-low-hypo(14),
    sensor-high-hyper(15),
    sensor-rate-decrease-exceeded(16),
    sensor-rate-increase-exceeded(17),
    sensor-result-too-low(18),
    sensor-result-too-high(19),
    sensor-com-out-of-range(20)
}

```

The extension to the Metric Measurement-Status attribute requires the following ASN.1 structure definition:

```

MeasurementStatus ::= BITS-16 {
    invalid(0),
    questionable(1),
    not-available(2),
    calibration-ongoing(3),
    test-data(4),
    demo-data(5),
    validated-data(8),
    early-indication(9),
    msmt-ongoing(10),
    msmt-value-exceed-boundaries (14),      -- indicate that the measurement is outside threshold
                                            -- boundaries.
    msmt-state-ann-inhibited (15)          -- indicate that the threshold indication is disabled
                                            -- and should not cause a displayed annunciation. If
                                            -- this bit is set, the bit 14 shall not be set.
}

```

B.2 Numeric extension for measurement confidence

The measurement confidence extensions to the glucose object require the following ASN.1 structure definition:

```

--
-- Measurement-Confidence-95 attribute defines the lower and upper bounds for a range within which the
-- manufacture is 95% confident the actual measurement value resides
--
-- NOTE: The unit for the lower and upper bounds is the same as the measurement
--
MeasurementConfidence95 ::= SEQUENCE {
    lower-bound SFLOAT-type
    upper-bound SFLOAT-type
}

```

B.3 Capability-mask

-- The capability mask defines if the corresponding bit in the enumeration bit string is supported.

```
CapabMaskSimp ::= BITS-32 {
    bit_0_supported(0),      -- bit is supported: 1 | bit is not supported: 0
    bit_1_supported(1),      -- bit is supported: 1 | bit is not supported: 0
    bit_2_supported(2),      -- bit is supported: 1 | bit is not supported: 0
    bit_3_supported(3),      -- bit is supported: 1 | bit is not supported: 0
    bit_4_supported(4),      -- bit is supported: 1 | bit is not supported: 0
    bit_5_supported(5),      -- bit is supported: 1 | bit is not supported: 0
    bit_6_supported(6),      -- bit is supported: 1 | bit is not supported: 0
    bit_7_supported(7),      -- bit is supported: 1 | bit is not supported: 0
    bit_8_supported(8),      -- bit is supported: 1 | bit is not supported: 0
    bit_9_supported(9),      -- bit is supported: 1 | bit is not supported: 0
    bit_10_supported(10),    -- bit is supported: 1 | bit is not supported: 0
    bit_11_supported(11),    -- bit is supported: 1 | bit is not supported: 0
    bit_12_supported(12),    -- bit is supported: 1 | bit is not supported: 0
    bit_13_supported(13),    -- bit is supported: 1 | bit is not supported: 0
    bit_14_supported(14),    -- bit is supported: 1 | bit is not supported: 0
    bit_15_supported(15),    -- bit is supported: 1 | bit is not supported: 0
    bit_16_supported(16),    -- bit is supported: 1 | bit is not supported: 0
    bit_17_supported(17),    -- bit is supported: 1 | bit is not supported: 0
    bit_18_supported(18),    -- bit is supported: 1 | bit is not supported: 0
    bit_19_supported(19),    -- bit is supported: 1 | bit is not supported: 0
    bit_20_supported(20),    -- bit is supported: 1 | bit is not supported: 0
    bit_21_supported(21),    -- bit is supported: 1 | bit is not supported: 0
    bit_22_supported(22),    -- bit is supported: 1 | bit is not supported: 0
    bit_23_supported(23),    -- bit is supported: 1 | bit is not supported: 0
    bit_24_supported(24),    -- bit is supported: 1 | bit is not supported: 0
    bit_25_supported(25),    -- bit is supported: 1 | bit is not supported: 0
    bit_26_supported(26),    -- bit is supported: 1 | bit is not supported: 0
    bit_27_supported(27),    -- bit is supported: 1 | bit is not supported: 0
    bit_28_supported(28),    -- bit is supported: 1 | bit is not supported: 0
    bit_29_supported(29),    -- bit is supported: 1 | bit is not supported: 0
    bit_30_supported(30),    -- bit is supported: 1 | bit is not supported: 0
    bit_31_supported(31),    -- bit is supported: 1 | bit is not supported: 0
}
```

B.4 State-flag

-- The state flag defines if the corresponding bit in the enumeration bit string is a state or an event. For a bit -- that is a state, then the value is set to 1.

```
StateFlagSimp ::= BITS-32 {
    bit_0_state(0),  -- bit is state: 1 | bit is event: 0
    bit_1_state(1),  -- bit is state: 1 | bit is event: 0
    bit_2_state(2),  -- bit is state: 1 | bit is event: 0
    bit_3_state(3),  -- bit is state: 1 | bit is event: 0
    bit_4_state(4),  -- bit is state: 1 | bit is event: 0
    bit_5_state(5),  -- bit is state: 1 | bit is event: 0
    bit_6_state(6),  -- bit is state: 1 | bit is event: 0
}
```

```

bit_7_state(7), -- bit is state: 1 | bit is event: 0
bit_8_state(8), -- bit is state: 1 | bit is event: 0
bit_9_state(9), -- bit is state: 1 | bit is event: 0
bit_10_state(10), -- bit is state: 1 | bit is event: 0
bit_11_state(11), -- bit is state: 1 | bit is event: 0
bit_12_state(12), -- bit is state: 1 | bit is event: 0
bit_13_state(13), -- bit is state: 1 | bit is event: 0
bit_14_state(14), -- bit is state: 1 | bit is event: 0
bit_15_state(15), -- bit is state: 1 | bit is event: 0
bit_16_state(16), -- bit is state: 1 | bit is event: 0
bit_17_state(17), -- bit is state: 1 | bit is event: 0
bit_18_state(18), -- bit is state: 1 | bit is event: 0
bit_19_state(19), -- bit is state: 1 | bit is event: 0
bit_20_state(20), -- bit is state: 1 | bit is event: 0
bit_21_state(21), -- bit is state: 1 | bit is event: 0
bit_22_state(22), -- bit is state: 1 | bit is event: 0
bit_23_state(23), -- bit is state: 1 | bit is event: 0
bit_24_state(24), -- bit is state: 1 | bit is event: 0
bit_25_state(25), -- bit is state: 1 | bit is event: 0
bit_26_state(26), -- bit is state: 1 | bit is event: 0
bit_27_state(27), -- bit is state: 1 | bit is event: 0
bit_28_state(28), -- bit is state: 1 | bit is event: 0
bit_29_state(29), -- bit is state: 1 | bit is event: 0
bit_30_state(30), -- bit is state: 1 | bit is event: 0
bit_31_state(31) -- bit is state: 1 | bit is event: 0

```

```

}

```

Annex C

(normative)

Allocation of identifiers

C.1 General

This annex contains the nomenclature codes used in this document. For those not contained in this annex, the normative definition is found in ISO/IEEE 11073-20601 or ISO/IEEE 11073-10101.

C.2 Definitions of terms and codes

The format used here follows that of ISO/IEEE 11073-10101.

```

/*****
* From Communication Infrastructure (MDC_PART_INFRA)
*****/
#define MDC_DEV_SPEC_PROFILE_CGM          4121    /* Device specialization profile for continuous
glucose monitor */

/*****
* From Medical supervisory control and data acquisition (MDC_PART_SCADA)
*****/
#define MDC_CONC_GLU_CAPILLARY_WHOLEBLOOD 29112    /* Glucose concentration from capillary whole
blood */
#define MDC_CONC_GLU_CAPILLARY_PLASMA     29116    /* Glucose concentration from capillary plasma */
#define MDC_CONC_GLU_VENOUS_WHOLEBLOOD    29120    /* Glucose concentration from venous whole blood
*/
#define MDC_CONC_GLU_VENOUS_PLASMA        29124    /* Glucose concentration from venous plasma */
#define MDC_CONC_GLU_ARTERIAL_WHOLEBLOOD  29128    /* Glucose concentration from arterial whole blood
*/
#define MDC_CONC_GLU_ARTERIAL_PLASMA      29132    /* Glucose concentration from arterial plasma */
#define MDC_CONC_GLU_CONTROL              29136    /* Glucose concentration from control solution */
#define MDC_CONC_GLU_ISF                  29140    /* Glucose concentration from interstitial fluid */
#define MDC_CONC_GLU_UNDETERMINED_WHOLEBLOOD 29292 /* Glucose concentration from undetermined whole
blood */
#define MDC_CONC_GLU_UNDETERMINED_PLASMA  29296    /* Glucose concentration from undetermined plasma
*/

```

/******

* From Personal Health Device Disease Management (MDC_PART_PHD_DM)

*****/

#define MDC_PHD_DM_DEV_STAT	20000	/* General PHD Disease Mgmt. Device Status */
#define MDC_CTXT_GLU_SAMPLELOCATION_UNDETERMINED location is undetermined */	29237	/* Glucose measurement context indicating sample location is undetermined */
#define MDC_CTXT_GLU_SAMPLELOCATION_OTHER location is other (does not match an available option) */	29238	/* Glucose measurement context indicating sample location is other (does not match an available option) */
#define MDC_CTXT_GLU_SAMPLELOCATION_FINGER location is finger*/	29240	/* Glucose measurement context indicating sample location is finger*/
#define MDC_CTXT_GLU_SAMPLELOCATION_SUBCUTANEOUS location is subcutaneous*/	29241	/* Glucose measurement context indicating sample location is subcutaneous*/
#define MDC_CTXT_GLU_SAMPLELOCATION_AST location is an alternative site */	29244	/* Glucose measurement context indicating sample location is an alternative site */
#define MDC_CTXT_GLU_SAMPLELOCATION_EARLOBE location is earlobe*/	29248	/* Glucose measurement context indicating sample location is earlobe*/
#define MDC_CTXT_GLU_SAMPLELOCATION_CTRLsolution location is from control solution */	29252	/* Glucose measurement context indicating sample location is from control solution */
#define MDC_CONC_GLU_TREND	29400	/* Trending glucose concentration */
#define MDC_CONC_GLU_PATIENT_THRESHOLDS_LOW_HIGH concentration */	29404	/* Patient low and high thresholds for glucose concentration */
#define MDC_CONC_GLU_PATIENT_THRESHOLD_LOW concentration */	29405	/* Patient low threshold value for glucose concentration */
#define MDC_CONC_GLU_PATIENT_THRESHOLD_HIGH concentration */	29406	/* Patient high threshold value for glucose concentration */
#define MDC_CONC_GLU_THRESHOLDS_HYPO_HYPER concentration */	29408	/* Hypo and hyper thresholds for glucose concentration */
#define MDC_CONC_GLU_THRESHOLD_HYPO	29409	/* Hypo threshold value for glucose concentration */
#define MDC_CONC_GLU_THRESHOLD_HYPER */	29410	/* Hyper threshold value for glucose concentration */
#define MDC_CONC_GLU_RATE_THRESHOLDS concentration */	29412	/* Rate of change thresholds for glucose concentration */
#define MDC_CONC_GLU_RATE_THRESHOLD_INCREASE glucose concentration */	29413	/* Increase threshold value for rate of change of glucose concentration */
#define MDC_CONC_GLU_RATE_THRESHOLD_DECREASE glucose concentration */	29414	/* Decrease threshold value for rate of change of glucose concentration */
#define MDC_CGM_SENSOR_CALIBRATION	29428	/* Continuous glucose monitor sensor calibration */
#define MDC_CGM_SENSOR_RUN_TIME	29432	/* Continuous glucose monitor sensor run time */
#define MDC_CGM_SENSOR_SAMPLE_INTERVAL interval */	29436	/* Continuous glucose monitor sensor sample interval */
#define MDC_CGM_DEV_STAT	29452	/* Continuous glucose monitor device status */
#define MDC_CGM_DEV_TYPE_SENSOR	29460	/* Continuous glucose monitor device type sensor */
#define MDC_CGM_DEV_TYPE_TRANSMITTER type_transmitter */	29461	/* Continuous glucose monitor device type_transmitter */
#define MDC_CGM_DEV_TYPE_RECEIVER */	29462	/* Continuous glucose monitor device type_receiver */
#define MDC_CGM_DEV_TYPE_OTHER (does not match an available option) */	29463	/* Continuous glucose monitor device type_other (does not match an available option) */

/* **** */

* From Object Infrastructure (MDC_PART_OBJ)

/* **** */

#define MDC_ATTR_THRES_NOTIF_TEXT_STRING 2696 /* Numeric object attribute threshold notification text string */

#define MDC_ATTR_MSMT_CONFIDENCE_95 2700 /* Numeric object attribute measurement confidence */

#define MDC_ATTR_ENUM_CAPABILITY_MASK_SIMPLE 2704 /* */

#define MDC_ATTR_ENUM_CAPABILITY_MASK_BASIC 2705 /* */

#define MDC_ATTR_ENUM_STATE_FLAG_SIMPLE 2706 /* */

#define MDC_ATTR_ENUM_STATE_FLAG_BASIC 2707 /* */

/* **** */

* From Dimensions (MDC_PART_DIM)

/* **** */

#define MDC_DIM_MILLI_G_PER_DL 2130 /* General dimension mg/dL */

#define MDC_DIM_MILLI_MOLE_PER_L 4722 /* General dimension mmol/L */

#define MDC_DIM_HR 2240 /* General dimension hour */

#define MDC_DIM_MIN 2208 /* General dimension minute */

#define MDC_DIM_MILLI_G_PER_DL_PER_MIN 11698 /* General dimension mg/dL per minute */

#define MDC_DIM_MILLI_MOLE_PER_L_PER_MIN 11730 /* General dimension mmol/L per minute */

C.3 Systematic derivations of terms and codes

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