# INTERNATIONAL STANDARD ISO/IEEE 11073-10404

First edition
2010-05-01

# Health informatics — Personal health device communication —

## Part 10404:
## Device specialization — Pulse oximeter

*Informatique de santé — Communication entre dispositifs de santé personnels —*

*Partie 10404: Spécialisation des dispositifs — Oxymètre de pouls*

Reference number
ISO/IEEE 11073-10404:2010(E)

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. Neither the ISO Central Secretariat nor IEEE accepts any liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies and IEEE members. In the unlikely event that a problem relating to it is found, please inform the ISO Central Secretariat or IEEE at the address given below.

**COPYRIGHT PROTECTED DOCUMENT**

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

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

Published in Switzerland

# Contents

Page

Foreword ..... v

Introduction ..... vii

1. Overview ..... 1

1.1 Scope ..... 1
1.2 Purpose ..... 1
1.3 Context ..... 2

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 ..... 3

4.1 General ..... 3
4.2 Introduction to IEEE 11073-20601 modeling constructs ..... 4

5. Pulse oximeter device concepts and modalities ..... 4

5.1 General ..... 4
5.2 Device types ..... 5
5.3 General concepts ..... 5
5.4 Collected data ..... 6
5.5 Derived data ..... 8
5.6 Stored data ..... 8
5.7 Device configurations ..... 8

6. Pulse oximeter DIM ..... 9

6.1 Overview ..... 9
6.2 Class extensions ..... 9
6.3 Object instance diagram ..... 9
6.4 Types of configuration ..... 10
6.5 MDS object ..... 11
6.6 Numeric objects ..... 14
6.7 Real-time sample array (RT-SA) objects ..... 24
6.8 Enumeration objects ..... 25
6.9 PM-store objects ..... 29
6.10 Scanner objects ..... 33
6.11 Class extension objects ..... 37
6.12 Pulse oximeter information model extensibility rules ..... 37

7. Pulse oximeter service model ..... 37

7.1 General ..... 37
7.2 Object access services ..... 37
7.3 Object access EVENT REPORT services ..... 40

8. Pulse oximeter communication model ..... 41

8.1 Overview ..... 41
8.2 Communications characteristics ..... 41
8.3 Association procedure ..... 42
8.4 Configuring procedure ..... 43
8.5 Operating procedure ..... 45
8.6 Time synchronization ..... 46

9. Test associations ..... 46

9.1 Behavior with standard configuration ..... 46
9.2 Behavior with extended configurations ..... 46

10. Conformance ..... 46

10.1 Applicability ..... 46
10.2 Conformance specification ..... 47
10.3 Levels of conformance ..... 47
10.4 Implementation conformance statements (ICSs) ..... 48

Annex A (informative) Bibliography ..... 52

Annex B (normative) Additional ASN.1 definitions ..... 53

Annex C (normative) Allocation of identifiers ..... 55

Annex D (informative) Message sequence examples ..... 57

Annex E (informative) PDU examples ..... 59

## Foreword

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

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

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

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

ISO/IEEE 11073-10404 was prepared by the 11073 Committee of the Engineering in Medicine and Biology Society of the IEEE (as IEEE Std 11073-10404-2008). It was adopted by Technical Committee ISO/TC 215, *Health informatics*, in parallel with its approval by the ISO member bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE. Both parties are responsible for the maintenance of this document.

ISO/IEEE 11073 consists of the following parts, under the general title *Health informatics — Personal health device communication (text in parentheses gives a variant of subtitle)*:

— *Part 10101: (Point-of-care medical device communication) Nomenclature*

— *Part 10201: Domain information model*

— *Part 10404: Device specialization — Pulse oximeter*

— *Part 10407: Device specialization — Blood pressure monitor*

- *Part 10408: (Point-of-care medical device communication) Device specialization — Thermometer*
- *Part 10415: (Point-of-care medical device communication) Device specialization — Weighing scale*
- *Part 10417: Device specialization — Glucose meter*
- *Part 10471: (Point-of-care medical device communication) Device specialization — Independant living activity hub*
- *Part 20101: (Point-of-care medical device communication) Application profiles — Base standard*
- *Part 20601: (Point-of-care medical device communication) Application profile — Optimized exchange protocol*
- *Part 30200: (Point-of-care medical device communication) Transport profile — Cable connected*
- *Part 30300: (Point-of-care medical device communication) Transport profile — Infrared wireless*

# Introduction

ISO/IEEE 11073 standards enable communication between medical devices and external computer systems. This standard uses the optimized framework created in IEEE Std 11073-20601™-2008[a] and describes a specific, interoperable communication approach for pulse oximeters. These standards align with, and draw upon, the existing clinically focused standards to provide support for communication of data from clinical or personal health devices.

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

# Health informatics—Personal health device communication—

# Part 10404: Device specialization—Pulse oximeter

***IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection in all circumstances. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.***

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

## 1. Overview

### 1.1 Scope

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

### 1.2 Purpose

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

### 1.3 Context

See IEEE Std 11073-20601-2008[1] for an overview of the environment within which this standard is written.

This standard, IEEE Std 11073-10404-2008, defines the device specialization for the pulse oximeter, being a specific agent type, and provides a description of the device concepts, its capabilities, and its implementation according to this standard.

This standard is based on IEEE Std 11073-20601-2008, which in turn draws information from both ISO/IEEE 11073-10201:2004 [B3][2] and ISO/IEEE 11073-20101:2004 [B4]. The medical device encoding rules (MDER) used within this standard are fully described in IEEE Std 11073-20601-2008.

This standard reproduces relevant portions of the nomenclature found in ISO/IEEE 11073-10101:2004 [B2] and adds new nomenclature codes for the purposes of this standard. Between this standard and IEEE Std 11073-20601-2008, all required nomenclature codes for implementation are documented.

NOTE—In this standard, ISO/IEEE P11073-104zz is used to refer to the collection of device specialization standards that utilize IEEE Std 11073-20601-2008, where zz can be any number from 01 to 99, inclusive.[3]

## 2. Normative references

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

IEEE Std 11073-20601-2008, Health informatics—Personal health device communication—Part 20601: Application profile—Optimized Exchange Profile.[4]

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

## 3. Definitions, acronyms, and abbreviations

### 3.1 Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards* [B1] should be referenced for terms not defined in this clause.

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

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

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

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

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

[1] Information on references can be found in Clause 2.

[2] The numbers in brackets correspond to the numbers in the bibliography in Annex A.

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

[4] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

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

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

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

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

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

**3.1.11 plethysmogram, plethysmographic,** or **photoplethysmographic waveform:** Sequence of samples related to the sequential time-varying light absorption due to effects of pulsatile blood flow.

**3.1.12 SpO$_2$:** Percentage oxygen saturation of haemoglobin as measured by a pulse oximeter, where this measurement is an estimate of the fraction of functional haemoglobin (or hemoglobin) in arterial blood that is saturated with oxygen.

NOTE—For more information about SpO$_2$, see ISO 9919 [B6].

## 3.2 Acronyms and abbreviations

| | |
|---|---|
| APDU | application protocol data unit |
| ASN.1 | Abstract Syntax Notation One |
| DIM | domain information model |
| ECG | electrocardiograph |
| EUI-64 | extended unique identifier (64 bits) |
| ICS | implementation conformance statement |
| ID | identifier |
| MDC | medical device communication |
| MDER | medical device encoding rules |
| MDS | medical device system |
| MOC | managed object class |
| OID | object identifier |
| PDU | protocol data unit |
| PHD | personal health device |
| PnP | plug-and-play |
| RT-SA | real-time sample array |
| SpO$_2$ | percentage oxygen saturation of haemoglobin |
| VMO | virtual medical object |
| VMS | virtual medical system |

# 4. Introduction to ISO/IEEE 11073 personal health devices

## 4.1 General

This standard and the remainder of the series of ISO/IEEE 11073 personal health device 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 healthcare information systems. See IEEE Std 11073-20601-2008 for a description of the guiding principles for this series of ISO/IEEE 11073 personal health device standards.

IEEE Std 11073-20601-2008 supports the modeling and implementation of an extensive set of personal health devices. IEEE Std 11073-10404-2008 (this standard) defines aspects of the pulse oximeter device. It describes all aspects necessary to implement the application layer services and data exchange protocol between an ISO/IEEE 11073 personal health device pulse oximetry agent and a manager. This standard defines a subset of the objects and functionality contained in IEEE Std 11073-20601-2008, extending and adding definitions where appropriate. All new definitions are given in Annex B in Abstract Syntax Notation One (ASN.1). Nomenclature codes referenced in this standard, which are not defined in IEEE Std 11073-20601-2008, are normatively defined in Annex C.

## 4.2 Introduction to IEEE 11073-20601 modeling constructs

### 4.2.1 General

The ISO/IEEE 11073 series of standards, and in particular IEEE Std 11073-20601-2008, is based on an object-oriented systems management paradigm. The overall system model is divided into three principal components: the domain information model (DIM), the service model, and the communication model. See IEEE Std 11073-20601-2008 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 data that an agent can communicate to a manager. Communication between the agent and manager is defined by the application protocol in IEEE Std 11073-20601-2008.

### 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 manager. Protocol messages within the ISO/IEEE 11073 series of standards are defined in ASN.1. The messages defined in IEEE Std 11073-20601-2008 can coexist with messages defined in other standard application profiles defined in the ISO/IEEE 11073 series of standards.

### 4.2.4 Communication model

In general, the communication model supports the topology of one or more agents communicating over logical point-to-point connections to a single manager. For each logical point-to-point connection, the dynamic system behavior is defined by a connection state machine as specified in IEEE Std 11073-20601-2008.

### 4.2.5 Implementing the models

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

# 5. Pulse oximeter device concepts and modalities

## 5.1 General

This clause presents the general concepts of pulse oximeter equipment. In the context of personal health devices in the ISO/IEEE 11073 family of standards, a pulse oximeter, also called an oximeter, provides a noninvasive estimate of functional oxygen of arterial haemoglobin ($SpO_2$) from a light signal interacting with tissue, by using the time-dependent changes in tissue optical properties that occur with pulsatile blood flow (see Draft Guidance for Industry and FDA Staff [B5]). Applying the Beer-Lambert law of light absorption through such an arterial network, the fraction of oxygenation of arterial haemoglobin can be estimated. This estimate, normally expressed as a percentage by multiplying that fraction by 100, is known as $SpO_2$. Occasionally, this estimate may be referenced as $\%SpO_2$. ISO 9919 [B6] contains additional information applicable to pulse oximetry.

## 5.2 Device types

Pulse oximeter systems with applicability in the personal health space may take on a variety of configurations and sensor compositions, and their configurations have suitability in different personal health application spaces. Pulse oximeter equipment comprises a pulse oximeter monitor, a pulse oximeter probe, and a probe cable extender, if provided. Some oximeters are all-in-one assemblies, where the optical probe, processing, and display components are in a single package. Other oximeters may consist of separate sensor and processing/display components. Still others may place the sensor and signal processing in one component, and send that information into an external component for display and storage. In addition, other configurations may add storage capability into the system. This implies that different information models may be best suited for each particular device configuration.

## 5.3 General concepts

### 5.3.1 Noninvasive measurement

The scope of this specialization covers the intended use of pulse oximeter equipment, which includes, but is not limited to, the estimation of arterial oxygen haemoglobin saturation and pulse rate. This standard is not applicable to pulse oximeter equipment intended for use in laboratory research applications or to oximeters that require a blood sample (see ISO 9919 [B6]). This standard does not cover measurement of oxygenation via blood extraction. This standard is not applicable to pulse oximeter equipment solely intended for foetal use.

The sensing mechanism may use either transmissive or reflective methods to measure blood oxygenation. In addition, blood oxygenation is usually determined as a ratio of the absorbance of two different wavelengths of light, although more wavelengths may be used.

### 5.3.2 Acquisition modes

#### 5.3.2.1 General

Pulse oximeters are used to measure $SpO_2$ within a variety of use scenarios.

#### 5.3.2.2 Spot-check

In a spot-check scenario, a user may simply want to take a single, fully processed reading for transmission to a manager. For example, the user would attach the oximeter, whereupon the agent would take an oximetry and pulse rate reading. The agent would then begin communication with a manager and send that single reading. The manager may acknowledge the transmission so the agent can subsequently disassociate and return to its prior state.

#### 5.3.2.3 Continuous monitoring

A continuous monitoring situation involves the pulse oximeter device measuring the user’s oxygenation for some period of time greater than that needed to acquire a single measurement. Multiple measurements may be taken to acquire trending information.

#### 5.3.2.4 Stored-and-forwarded measurements

Stored-and-forwarded measurements could be considered as a specialized, continuous monitoring application where the pulse oximetry device is not always in communication with a manager, and the oximeter records data over several minutes or hours. In this case, oximetry data are stored in the device for the duration of the study session and subsequently transferred to the manager at an appropriate time. This measurement communication style is distinct from the situation where temporarily stored measurements are transferred when the communication link is restored.

## 5.4 Collected data

### 5.4.1 General

This subclause describes the nature of the data that have been collected based on the acquisition modes described in 5.3.2.

### 5.4.2 Percentage of arterial haemoglobin oxygen saturation

#### 5.4.2.1 $SpO_2$

Every oximeter sends at least one expression of $SpO_2$. This is the primary measurement of a pulse oximeter. It is important to note that this measurement is determined through various signal processing techniques and can be expressed in different ways. Each method and expression has its applicability in particular application spaces (e.g., vital signs monitoring and diagnostic sleep studies). Often the reported $SpO_2$ has been processed with a variety of techniques in order to present the data for use in a number of ways.

In response to the various physiological phenomena and situations, $SpO_2$ measurements may be expressed in a variety of ways. Additional modalities for expressing $SpO_2$ are often used that are better suited to expose or suppress various physiological or environmental phenomena, as seen in 5.4.2.2. The following subclause outlines three expressions of $SpO_2$ that may be used by a device manufacturer to convey blood oxygenation level.

It is also conceivable that pulse oximeter equipment may deliver a single $SpO_2$ that is determined by one of these modalities. Furthermore, several of these distinct expressions may be transmitted concurrently during a measurement session. The manager, upon receiving this collection of information, may choose to display another subset of these expressions. It is required for a pulse oximeter agent to support at least one instance of this measurement.

#### 5.4.2.2 Alternative expressions of $SpO_2$

One case of $SpO_2$ measurement involves a user wearing a sensor during unintentional or moderate activity. The result of this activity may be intermittent loss of signal acquisition. The most common expression of $SpO_2$ may be too sensitive to these effects and could result in a fluctuating (and, therefore, misleading) reading. An $SpO_2$ measurement modality known as “slow-response” modality has a characteristic that “smoothes out” a series of measurements in some fashion, perhaps by changing an averaging parameter or by employing a different algorithm. This modality is defined in this standard.

During a sleep study, an apnea event results in a rapid desaturation of blood oxygenation. This $SpO_2$ measurement can be expressed by a “fast-response” modality that uses a technique that more effectively captures such events. The technique may vary among device manufacturers, but a distinct expression able to capture these rapid changes is defined in this standard.

The terms *slow-response* and *fast-response* are relative to a particular implementation and are not intended to show a comparison across devices or vendors. Note that these are descriptive terms intentionally left unspecific to allow more flexible interpretations within a particular implementation.

A pulse oximeter will often send $SpO_2$ measurements periodically; e.g., once every second. In addition, pulse oximeters may begin outputting measurements as soon as it has a reasonable estimate of functional haemoglobin oxygenation. Subsequent measurements may, in some fashion, converge on the oximeter's best estimate. An additional modality, the "spot-check" modality, fulfills the desire to be able to perform and display a single $SpO_2$ measurement that is also its best estimate of functional haemoglobin oxygenation. In other words, a spot-check is not simply the first measurement, but the first *best* measurement. The specific manner in which this measurement is produced is specific to the pulse oximeter implementation. Once that measurement is transmitted, the measurement session is complete.

### 5.4.3 Pulse rate

The heart rate measured by a pulse oximeter is produced by a heartbeat, but also requires ejection of blood by the heart and generation of an arterial and tissue pressure wave that is detectable by photoplethysmographic means. Therefore, the pulse rate may be a less reliable measure of heart rate than that of directly measuring by electrocardiograph (ECG). As described in 5.4.2.1 and 5.4.2.2, the reported value or values may be determined in a variety of ways, and corresponding modalities of "slow-response," "fast-response," and "spot-check" are defined for pulse rate measurements. It is required for a pulse oximeter agent to support at least one instance of this feature.

### 5.4.4 Pulsatile occurrence

If a precisely timestamped occurrence of a pulse is transmitted to a manager, that information can be used in conjunction with other reported physiological events to derive another physiological measurement. Other application spaces may wish to indicate pulsatile occurrence with less precision for purposes of displaying, for instance, a flashing heart icon. It is not required for a pulse oximeter agent to support this feature.

### 5.4.5 Plethysmogram

There are applications where it is desired to visualize the sequence of samples related to the time-varying light absorption due to the effects of pulsatile blood flow. Often these samples are taken from a single wavelength light source, usually the wavelength less affected by changes in oxygen saturation. It is not required for a pulse oximeter agent to support this feature.

### 5.4.6 Pulsatile quality and signal characterization

Pulse oximeter manufacturers have many ways to characterize the quality of the pulsatile wave. Unfortunately, no industry-wide standard currently exists to quantify the characteristics of the signal. However, signal amplitude metrics among the different vendors provide quantities that can be found to have a linear relationship. One notable characteristic is the amplitude of the signal modulation. Other methods to characterize the quality of the pulsatile wave may be employed. It is not required for a pulse oximeter to support this feature.

## 5.5 Derived data

### 5.5.1 Limit indications

Pulse oximeters may implement indicators based on monitoring physiological values as falling outside predefined limits. The commonly implemented indicators include reaching the thresholds of a high or low $SpO_2$, or reaching the thresholds of a high or low pulse rate.

### 5.5.2 Pulsatile status

Pulse oximeters may provide status indications of certain characteristics of a pulsatile wave or irregularities in the waveform.

### 5.5.3 Device and sensor status

Pulse oximeters may provide status indications pertaining to sensor malfunction or dislodgement as well as various signal anomalies.

## 5.6 Stored data

As stated in 5.3.2.4, a pulse oximeter may be used over one or more sessions of several hours without being in contact with a manager to send its data. After the session or sessions are completed, the pulse oximeter agent connects to a manager. The manager is able to select which of the agent's stored sessions 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 is also able to choose a set of sessions for deletion.

## 5.7 Device configurations

Although agents typically have a static configuration, it is permissible and desirable for an agent to support multiple configurations, one of which would be active at any given time. Pulse oximeters may have a rich set of features that can be combined into a collection of different configurations, one of which can be selected by the manager during configuration.

Two general categories of configurations exist. The first category is known as the set of standard configurations. These are intended to describe a relatively limited feature set of a single device specialization, which have predefined configuration ID codes. Managers may be pre-loaded with these configurations, in which case the configuration process is eliminated and immediate operation is allowed. The second category involves the set of extended configurations. These configurations are more flexible in that they may include concepts particular to one or more device specializations or include other features as defined in this standard.

# 6. Pulse oximeter DIM

## 6.1 Overview

This clause describes the DIM of the pulse oximeter.

## 6.2 Class extensions

In this standard, the $SpO_2$ and pulse rate numeric objects are extended with respect to IEEE Std 11073-20601-2008 to support threshold capabilities (see 6.6.2.1.1 and 6.6.3.1.1).

## 6.3 Object instance diagram

The object instance diagram of the pulse oximeter DIM's numeric objects, defined for the purposes of this standard, is shown in Figure 1.

The objects of the DIM, as shown in Figure 1, are described in the following subclauses: medical device system (MDS) object (see 6.5), the numeric objects (see 6.6), the real-time sample array (RT-SA) objects (see 6.7), and the enumeration objects (see 6.8). Figure 2 illustrates the PM-store objects (see 6.9), and the scanner objects (see 6.10) are shown in Figure 4. See 6.11 for rules for extending the pulse oximeter information model beyond elements as described in this standard. Each subclause that describes an object of the pulse oximeter 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, RT-SA, 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 ASN.1. The ASN.1 definitions for new attribute types specific to this standard are in Annex B, and the ASN.1 definitions for existing attribute types referenced in this standard are in IEEE Std 11073-20601-2008.
- The methods available on the object.
- The potential events generated by the object. Data are sent to the manager using events.
- The available services such as getting or setting attributes.

The attributes for each class are defined in tables that specify the name of the attribute, its value, and its qualifier. The qualifiers mean M – Attribute is Mandatory, C – Attribute is Conditional and depends on the condition stated in the Remark or Value column (if IEEE Std 11073-20601-2008 is referenced, then it contains the conditions), R – Attribute is Recommended, O – Attribute is Optional, NR – Attribute is Not Recommended. 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. Optional attributes may be implemented on an agent. Not recommended attributes should not be implemented by the agent.

The attributes can be either static, meaning that they shall remain unchanged after the configuration is agreed upon, or dynamic, meaning that the attribute may change at some point after configuration.

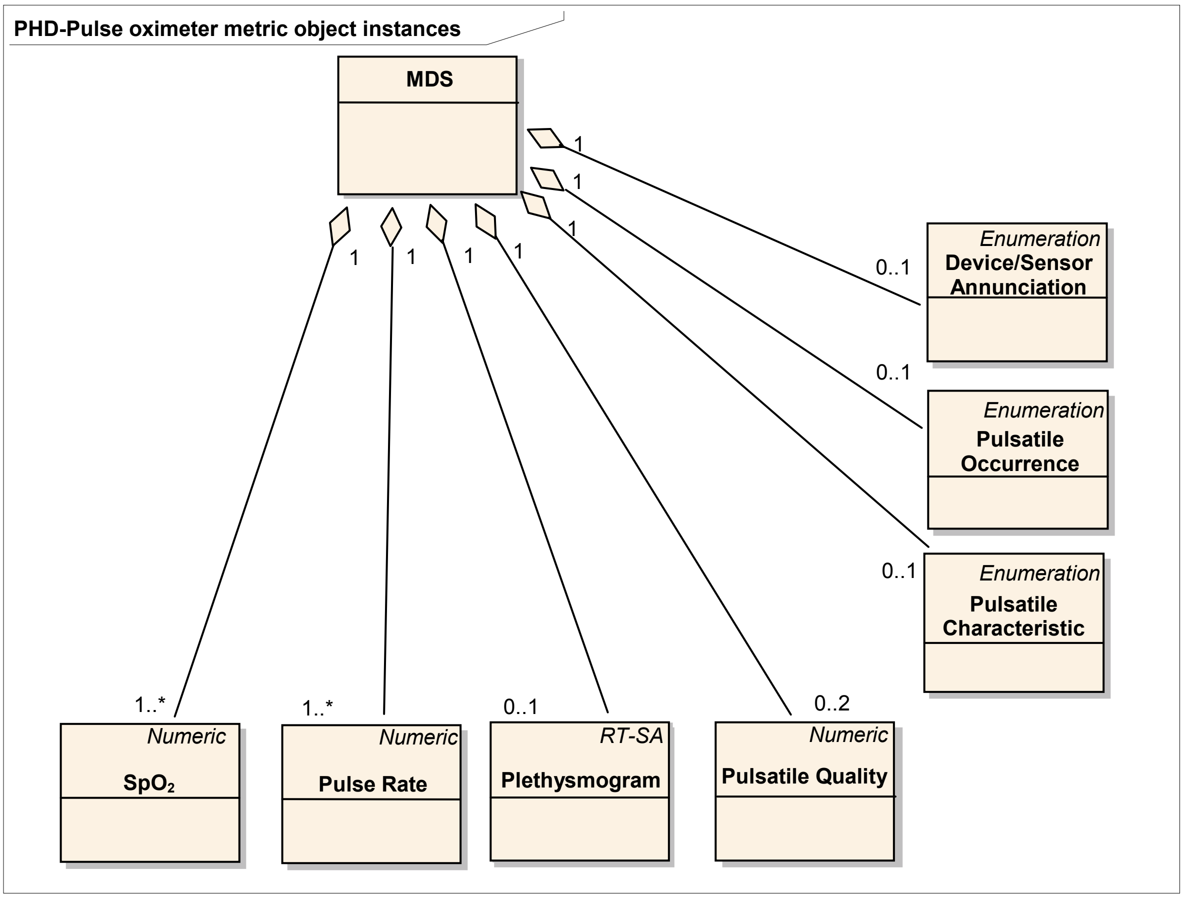

**Figure 1—Pulse oximeter DIM for metric objects**

## 6.4 Types of configuration

### 6.4.1 General

As specified in IEEE Std 11073-20601-2008, two styles of configuration are available. The following subclauses 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 configuration time between the agent and manager. If the manager acknowledges that it recognizes 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.

### 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 that it wants to use 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 know the agent's configuration on the first connection. Therefore, the manager responds that it needs the configuration and the agent transmits its configuration information by sending a configuration event report. If the manager already recognizes the configuration, either because it was preloaded via an installation program or the agent previously associated with the manager, then the manager responds that the configuration is known and no further configuration information needs to be sent.

If the device uses concepts and terms outside of this device specialization or terms defined in the private partition of ISO/IEEE 11073-10101:2004 [B2], it is considered a proprietary device.

## 6.5 MDS object

### 6.5.1 MDS object attributes

Table 1 summarizes the attributes of the pulse oximeter MDS object. The nomenclature code to identify the MDS class is MDC_MOC_VMS_MDS_SIMP.

**Table 1—MDS object attributes**

| Attribute name | Value | Qual. |
|---|---|---|
| Handle | 0 | M |
| System-Type | Attribute not present. See IEEE Std 11073-20601-2008 | C |
| System-Model | {“Manufacturer”,”Model”} | M |
| System-Id | EUI-64 | M |
| Dev-Configuration-Id | Standard config: 0x0190 (400) or 0x191 (401)<br>Extended configs: 0x4000–0x7FFF | M |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008 | C |
| Production-Specification | See IEEE Std 11073-20601-2008 | O |
| Mds-Time-Info | See IEEE Std 11073-20601-2008 | C |
| Date-and-Time | See IEEE Std 11073-20601-2008 | C |
| Relative-Time | See IEEE Std 11073-20601-2008 | C |
| HiRes-Relative-Time | See IEEE Std 11073-20601-2008 | C |
| Date-and-Time-Adjustment | See IEEE Std 11073-20601-2008 | C |
| Power-Status | onBattery or onMains | R |
| Battery-Level | See IEEE Std 11073-20601-2008 | R |
| Remaining-Battery-Time | See IEEE Std 11073-20601-2008 | R |
| Reg-Cert-Data-List | See IEEE Std 11073-20601-2008 | O |
| System-Type-Spec-List | { {MDC_DEV_SPEC_PROFILE_PULS_OXIM, 1} } | M |
| Confirm-Timeout | See IEEE Std 11073-20601-2008 | O |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

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

See IEEE Std 11073-20601-2008 for descriptive explanations of the individual attributes as well as information on attribute identity (ID) and attribute type.

The Dev-Configuration-Id attribute holds a locally unique 16-bit identifier that identifies the device configuration. For a pulse oximetry agent with extended configuration, this identifier is chosen in the range of extended-config-start to extended-config-end (see IEEE Std 11073-20601-2008) as shown in Table 1.

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 (see 8.4) is skipped, and the agent and manager enter the Operating state. If the manager does not recognize the Dev-Configuration-Id, the agent and manager enter the Configuring state.

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

### 6.5.2 MDS object methods

Table 2 defines the methods (actions) of the MDS object. These methods are invoked using the ACTION service. In Table 2, the Subservice type name column defines the name of the method. The Mode column defines whether the method is invoked as an unconfirmed action (i.e., roiv-cmip-action from IEEE Std 11073-20601-2008) or a confirmed action (i.e., roiv-cmip-confirmed-action). The Subservice type (action-type) column defines the nomenclature code to use in the action-type field of an action request and response (see IEEE Std 11073-20601-2008). The Parameters (action-info-args) column defines the associated ASN.1 data structure (see IEEE Std 11073-20601-2008 for ASN.1 definitions) to use in the action message for the action-info-args field of the request. The Results (action-info-args) column defines the structure to use in the action-info-args of the response.

**Table 2—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 | - |
| | MDS-Data-Request | Confirmed | MDC_ACT_DATA_REQUEST | DataRequest | DataResponse |

— **Set-Time:**
This method allows the manager to set a real-time clock in the agent with the absolute time. The agent indicates whether the Set-Time command is valid using the mds-time-capab-set-clock bit in the Mds-Time-Info attribute (see IEEE Std 11073-20601-2008).

— **MDS-Data-Request:**
This method allows the manager system to enable or disable measurement data transmission from the agent (see IEEE Std 11073-20601-2008 for a description).

Agents following only this device specialization and no others may send event reports either by using agent-initiated or manager-initiated measurement data transmission. During the association procedure (see 8.3), DataReqModeCapab shall be set to the appropriate value for the event report style. As a result, the manager shall assume that if the pulse oximetry agent supports any of the MDS-Data-Request features, then it may use them to access the object value only if the object's Metric-Spec-Small attribute has its acc-manager-initiated bit set (see IEEE Std 11073-20601-2008 for additional information).

### 6.5.3 MDS object events

Table 3 defines the events that can be sent by the pulse oximeter MDS object.

**Table 3—Pulse oximeter 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 or Unconfirmed | MDC_NOTI_CONFIG | ConfigReport | ConfigReportRsp |
| | MDS-Dynamic-Data-Update-Var | Confirmed or Unconfirmed | MDC_NOTI_SCAN_REPORT_VAR | ScanReportInfoVar | — |
| | MDS-Dynamic-Data-Update-Fixed | Confirmed or Unconfirmed | MDC_NOTI_SCAN_REPORT_FIXED | ScanReportInfoFixed | — |
| | MDS-Dynamic-Data-Update-MP-Var | Confirmed or Unconfirmed | MDC_NOTI_SCAN_REPORT_MP_VAR | ScanReportInfoMPVar | — |
| | MDS-Dynamic-Data-Update-MP-Fixed | Confirmed or Unconfirmed | MDC_NOTI_SCAN_REPORT_MP_FIXED | ScanReportInfoMPFixed | — |

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

- **MDS-Dynamic-Data-Update-Var:**
  This event provides dynamic measurement data from the pulse oximetry agent for the numeric and enumeration object(s). This is 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 pulse oximeter agent for the numeric and enumeration objects. This is reported in the fixed format defined by the Attribute-Value-Map attribute of the object(s). The event is sent as an unsolicited message by the agent (i.e., an agent-initiated measurement data transmission). See 8.5.3 for more information on unsolicited event reporting.

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

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

NOTE—IEEE Std 11073-20601-2008 requires that managers support all of the MDS object events listed above.

### 6.5.4 Other MDS services

#### 6.5.4.1 GET service

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

The manager may request the MDS object attributes of the pulse oximeter agent, in which case the manager shall send the "Remote Operation Invoke | Get" message (see roiv-cmip-get in IEEE Std 11073-20601-2008) with the reserved MDS handle value of 0. The pulse oximeter agent shall report either a list or its entire set of attributes of its MDS object attributes to the manager using the "Remote Operation Response | Get" message (see rors-cmip-get in IEEE Std 11073-20601-2008). See Table 4 for a summary of the GET service including some message fields.

**Table 4—Pulse oximeter MDS object GET service**

| Service | Subservice type name | Mode | Subservice type | Parameters | Result |
|---|---|---|---|---|---|
| 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.

#### 6.5.4.2 SET service

The pulse oximeter specialization does not require an implementation to support the MDS object SET service. However, a pulse oximetry agent vendor may implement a private SET service for the limited purposes of setting private attributes, which avoids any notion of remote control of the pulse oximetry agent. The implementation conformance statement (ICS) table should include information related to accessing private attributes in this manner.

## 6.6 Numeric objects

### 6.6.1 General

The pulse oximeter DIM for metric objects (see Figure 1) contains one mandatory numeric object for expressing $SpO_2$, one mandatory numeric object for pulse rate, and several optional numeric objects for additional $SpO_2$ and pulse rate modalities, pulse amplitude, and the reporting of current settings of physiological threshold limits. These are described in 6.6.2, 6.6.3, and 6.6.4.

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

### 6.6.2 $SpO_2$

Table 5 summarizes the attributes for reporting an $SpO_2$ measurement. The nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. At least one $SpO_2$ numeric object shall be supported by a pulse oximeter agent.

### Table 5—SpO$_2$ numeric object attributes

| Attribute name | Extended configuration | | Standard Configuration (Dev-Configuration-Id = 0x0190) | |
|---|---|---|---|---|
| | Value | Qual. | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M | 1 | M |
| Type | {MDC_PART_SCADA, MDC_PULS_OXIM_SAT_O2} | M | {MDC_PART_SCADA, MDC_PULS_OXIM_SAT_O2} | M |
| Supplemental-Types | See following text. | C | See IEEE Std 11073-20601-2008 and following text. | NR |
| Metric-Spec-Small | mss-avail-stored-data, mss-acc-manager-initiated, mss-acc-agent-initiated. See following text. | M | mss-avail-stored-data, mss-acc-agent-initiated. See following text. | M |
| Metric-Structure-Small | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Measurement-Status | See IEEE Std 11073-20601-2008 and following text. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present. follow IEEE Std 11073-20601-2008. | NR |
| Metric-Id-List | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Metric-Id-Partition | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Unit-Code | MDC_DIM_PERCENT | M | MDC_DIM_PERCENT | M |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008 and following text | C | MDC_ATTR_NU_VAL_OBS_BASIC | M |
| Source-Handle-Reference | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Label-String | See IEEE Std 11073-20601-2008. | O | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | O |
| Unit-LabelString | See IEEE Std 11073-20601-2008. | O | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | O |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2008 apply. | C |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Measure-Active-Period | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |

Table 5—$SpO_2$ numeric object attributes

| Attribute name | Extended configuration | | Standard Configuration (Dev-Configuration-Id = 0x0190) | |
|---|---|---|---|---|
| | Value | Qual. | Value | Qual. |
| Simple-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2008 apply. | C |
| Compound-Simple-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Basic-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | See IEEE Std 11073-20601-2008. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2008 apply. | C |
| Compound-Basic-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Compound-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Accuracy | See IEEE Std 11073-20601-2008. | R | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | R |
| Alert-Op-State | See following text. | O | See following text. | NR |
| Current-Limits | See following text. | O | See following text. | NR |
| Alert-Op-Text-String | See following text. | O | See following text. | NR |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

#### 6.6.2.1 $SpO_2$ — extended configuration

A pulse oximeter agent may instantiate more than one $SpO_2$ numeric object. Typically, a pulse oximeter contains a single $SpO_2$ numeric object. However, a pulse oximeter may contain multiple $SpO_2$ numeric objects if it is designed to transmit multiple modalities of $SpO_2$ either successively or concurrently.

For a pulse oximeter agent with extended configuration, the AttrValMap structure (see IEEE Std 11073-20601-2008) of the Attribute-Value-Map attribute may need to accommodate information pertaining to threshold status information in addition to other attributes such as the observed value and timestamp information.

The Supplemental-Types attribute is used to distinguish the modality of a particular $SpO_2$ measurement. In order to express the fast-response SpO2 measurement, MDC_MODALITY_FAST shall be used as the value for the Supplemental-Types attribute. In order to express the slow-response $SpO_2$ measurement, MDC_MODALITY_SLOW shall be used as the value for the Supplemental-Types attribute. In order to express the spot-check $SpO_2$ measurement, MDC_MODALITY_SPOT shall be used as the value for the Supplemental-Types attribute. If there is no desire to distinguish a modality, the Supplemental-Types attribute shall not be used.

Certain combinations of Supplemental-Types are allowed. If it is desired to express that a spot-check measurement also use a fast-response technique, the SupplementalTypeList structure of the Supplemental-Types attribute should contain the two values MDC_MODALITY_SPOT and MDC_MODALITY_FAST. Similarly, a spot-check using a slow-response measurement should contain the values MDC_MODALITY_SPOT and MDC_MODALITY_SLOW in its SupplementalTypeList. It is not recommended to combine the values MDC_MODALITY_SLOW and MDC_MODALITY_FAST.

The Metric-Spec-Small attribute may comprise several values, and one or more of these bits may be set:

- **mss-avail-stored-data:**
  If this bit is set, the pulse oximeter agent may send up to 25 temporarily stored measurements in an event report.
- **mss-msmt-aperiodic:**
  This bit is set if the measurement is not sent at a fixed interval. If a spot-check modality is used, this bit shall be set in addition to the use of the appropriate setting of the Supplemental-Types attribute. This bit may also be set when the spot-check modality is not used.
- **mss-acc-manager-initiated:**
  This bit is set if the object allows the use of a manager-initiated transfer.
- **mss-acc-agent-initiated:**
  This bit is set if the object’s reportable values are transmitted via event reports without the manager requesting the measurement data. This does not mean that an agent shall send data in this manner, only that it is capable of doing so. This bit shall be set if this object reports its measurement data with an agent-initiated event report issued by the MDS object at any time during the association. If this object is scanned only by a scanner object (see 6.10), this bit shall not be set as the manager controls the data flow through the use of the Operational-State attribute. An agent implementation should make careful use of agent-initiated transfers that are not scanner objects, as the manager has little, if any, control of the bandwidth utilization of the data link. If agent-initiated transfers are used, they should be for either intermittent or periodic event reports of a small number of object values.

The Absolute-Time-Stamp attribute in the $SpO_2$ numeric object shall be present when using the spot-check modality.

#### 6.6.2.1.1 Threshold settings and status attributes

Three attributes extending the $SpO_2$ numeric object are provided to report the agent’s threshold settings, and a fourth 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 [B3]) from the definition in IEEE Std 11073-20601-2008 in order to report the threshold status. Note that the Current-Limits attribute stores the threshold values as FLOAT-Type values. If a Basic-Nu-Observed-Value attribute is used to express the measurement value, the values in the Current-Limits attribute shall be expressed within the range and precision of an SFLOAT value. See Table 6.

### Table 6—SpO$_2$ threshold settings and status attributes

| Attribute name | Attribute ID | Attribute type | Remark | Qual. |
|---|---|---|---|---|
| Alert-Op-State | MDC_ATTR_AL_OP_STAT | CurLimAlStat | Reflects the current state of the threshold annunciation masking bits. If thresholding is to be used, this attribute is mandatory. | C |
| Current-Limits | MDC_ATTR_LIMIT_CURR | CurLimAlVal | Current threshold values. If thresholding is to be used, this attribute is mandatory. | C |
| Alert-Op-Text-String | MDC_ATTR_AL_OP_TEXT_STRING | AlertOpTextString | Individual text for upper and lower threshold. | O |
| Measurement-Status | MDC_ATTR_MSMT_STAT | MeasurementStatus | Dynamically reflects whether observed value is at or outside threshold boundaries. If thresholding is to be used, this attribute is mandatory. Use bit msmt-state-in-alarm(14) to indicate that the measurement is outside threshold boundaries. Use msmt-state-al-inhibited(15) to indicate that the threshold indication is disabled and should not cause a displayed annunciation. These are bits extended from the IEEE 11073-20601 definitions of MeasurementStatus. | C |

Further description of the attribute types may be found in Annex B.

#### 6.6.2.2 SpO$_2$ — standard configurations

The first standard configuration defined in this standard contains two numeric objects, one of which is the SpO$_2$ numeric object, described in the Standard Configuration column in Table 5. This standard configuration is provided to describe a most basic pulse oximeter implementation.

A second standard configuration is intended for spot-check use cases. In addition to assigning the Dev-Configuration-Id to 0x191, the SpO$_2$ object attributes are modified from Dev-Configuration-Id = 0x190 as follows:

- The Supplemental-Types attribute shall contain a single entry in its SupplementalTypeList, and its value shall be MDC_MODALITY_SPOT.
- The AttrValMap structure (see IEEE Std 11073-20601-2008) of the Attribute-Value-Map attribute differs from Table 5 in that it shall contain the attribute ID and attribute length information of the Basic-Nu-Observed-Value and Absolute-Time-Stamp attribute in this stated order.
- The Metric-Spec-Small attribute shall set the following two bits appropriately:
  - **mss-avail-stored-data:**
    If this bit is set, the pulse oximeter agent may send up to 25 temporarily stored measurements in an event report.
  - **mss-acc-agent-initiated:**
    Since the standard configuration contains two numeric objects, the bandwidth requirements are intended to be relatively light.

If an implementation is more capable than that provided by either of these standard configurations, it shall use an extended configuration.

#### 6.6.2.3 $SpO_2$ — methods, events, services

The $SpO_2$ numeric object does not support any methods, events, or other services.

See IEEE Std 11073-20601-2008 for descriptive explanations of the individual attributes as well as information on attribute ID and attribute type.

### 6.6.3 Pulse rate

Table 7 summarizes the attributes for reporting a pulse rate measurement. The nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. At least one pulse rate numeric object shall be supported by a pulse oximeter agent.

**Table 7—Pulse rate numeric object attributes**

| Attribute name | Extended configuration | | Standard configuration (Dev-Configuration-Id = 0x0190) | |
|---|---|---|---|---|
| | Value | Qual. | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M | 10 | M |
| Type | {MDC_PART_SCADA, MDC_PULS_OXIM_PULS_RATE} | M | {MDC_PART_SCADA, MDC_PULS_OXIM_PULS_RATE} | M |
| Supplemental -Types | See following text. | C | See IEEE Std 11073-20601-2008 and following text. | NR |
| Metric-Spec-Small | mss-avail-stored-data, mss-acc-manager-initiated, mss-acc-agent-initiated. See following text. | M | mss-avail-stored-data, mss-acc-agent-initiated. See following text. | M |
| Metric-Structure-Small | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Measurement -Status | See IEEE Std 11073-20601-2008 and Table 6. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Metric-Id-List | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Metric-Id-Partition | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Unit-Code | MDC_DIM_BEAT_PER_MIN | M | MDC_DIM_BEAT_PER_MIN | M |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008. | C | MDC_ATTR_NU_VAL_OBS_BASIC | M |
| Source-Handle-Reference | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Label-String | See IEEE Std 11073-20601-2008. | O | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | O |
| Unit-LabelString | See IEEE Std 11073-20601-2008. | O | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | O |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2008 apply. | C |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |

### Table 7—Pulse rate numeric object attributes

| Attribute name | Extended configuration | | Standard configuration (Dev-Configuration-Id = 0x0190) | |
|---|---|---|---|---|
| | **Value** | **Qual.** | **Value** | **Qual.** |
| Measure-Active-Period | See IEEE Std 11073-20601-2008. | NR | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | NR |
| Simple-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | See IEEE Std 11073-20601-2008. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2008 apply. | C |
| Compound-Simple-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Basic-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | See IEEE Std 11073-20601-2008. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601-2008 apply. | C |
| Compound-Basic-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Compound-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | C |
| Accuracy | See IEEE Std 11073-20601-2008. | R | Attribute not initially present. If present, follow IEEE Std 11073-20601-2008. | R |
| Alert-Op-State | See 6.6.2.1.1. | O | See 6.6.2.1.1. | NR |
| Current-Limits | See 6.6.2.1.1. | O | See 6.6.2.1.1. | NR |
| Alert-Op-Text-String | See 6.6.2.1.1. | O | See 6.6.2.1.1. | NR |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

### 6.6.3.1 Pulse rate—extended configuration

A pulse oximeter agent may instantiate more than one pulse rate numeric object. Typically, a pulse oximeter contains a single pulse rate numeric object. However, a pulse oximeter may contain multiple pulse rate numeric objects if it is designed to transmit multiple modalities of pulse rate either successively or concurrently.

For a pulse oximeter agent with extended configuration, the AttrValMap structure (see IEEE Std 11073-20601-2008) of the Attribute-Value-Map attribute may need to accommodate information pertaining to threshold status information in addition to other attributes such as the observed value and timestamp information.

The Supplemental-Types attribute is used to distinguish the modality of a particular pulse rate measurement. In order to express a fast-response pulse rate measurement, MDC_MODALITY_FAST shall be used in the Supplemental-Types attribute. In order to express a slow-response pulse rate measurement, MDC_MODALITY_SLOW shall be used in the Supplemental-Types attribute. In order to express the single spot-check pulse rate measurement, MDC_MODALITY_SPOT shall be used in the Supplemental-Types attribute. If there is no desire to distinguish a modality, the Supplemental-Types attribute shall not be used.

Certain combinations of Supplemental-Types are allowed. If it is desired to illustrate that a spot-check measurement also uses a fast-response technique, the SupplementalTypeList structure of the Supplemental-Types attribute should contain the two values MDC_MODALITY_SPOT and MDC_MODALITY_FAST. Similarly, a spot-check using a slow-response measurement should contain the values MDC_MODALITY_SPOT and MDC_MODALITY_SLOW in its SupplementalTypeList. It is not recommended to combine the values MDC_MODALITY_SLOW and MDC_MODALITY_FAST.

The Metric-Spec-Small attribute may comprise several values, and one or more of these bits may be set:

- **mss-avail-stored-data:**
  If this bit is set, the pulse oximeter agent may send up to 25 temporarily stored measurements in an event report.
- **mss-msmt-aperiodic:**
  This bit is set if the measurement is not sent at a fixed interval. If a spot-check modality is used, this bit shall be set in addition to the use of the appropriate setting of the Supplemental-Types attribute. This bit may also be set when the spot-check modality is not used.
- **mss-acc-manager-initiated:**
  This bit is set if the object allows the use of a manager-initiated transfer.
- **mss-acc-agent-initiated:**
  This bit is set if the object's reportable values are transmitted via event reports without the manager requesting the measurement data. This does not mean that an agent shall send data in this manner, only that it is capable of doing so. This bit shall be set if this object reports its measurement data with an agent-initiated event report issued by the MDS object at any time during the association. If this object is scanned only by a scanner object (see 6.10), this bit shall not be set as the manager controls the data flow through the use of the Operational-State attribute. An agent implementation should make careful use of agent-initiated transfers that are not scanner objects, as the manager has little, if any, control of the bandwidth utilization of the data link. If agent-initiated transfers are used, they should be for either intermittent or periodic event reports of a small number of object values.

The Absolute-Time-Stamp attribute in the pulse rate numeric object shall be present when using the spot-check modality.

#### 6.6.3.1.1 Threshold settings and status attributes

As described in 6.6.2.1.1, three attributes extending the pulse rate numeric object are provided to report the agent's threshold settings, and a fourth reports whether the measurement has reached or crossed beyond the threshold boundaries. See Table 6 for a detailed description of the additional attributes.

### 6.6.3.2 Pulse rate—standard configuration

The first standard configuration defined in this standard contains two numeric objects, one of which is the pulse rate numeric object, described in the Standard Configuration column in Table 7. The standard configuration is provided to describe a most basic pulse oximeter implementation.

A second standard configuration is intended for spot-check use cases. In addition to assigning the Dev-Configuration-Id to 0x191, the pulse rate object attributes are modified from Dev-Configuration-Id = 0x190 as follows:

- The Supplemental-Types attribute shall contain a single entry in its SupplementalTypeList, and its value shall be MDC_MODALITY_SPOT.
- The AttrValMap structure (see IEEE Std 11073-20601-2008) of the Attribute-Value-Map attribute differs from Table 7 in that it shall contain the attribute ID and attribute length information of the Basic-Nu-Observed-Value and Absolute-Time-Stamp attribute in this stated order.
- The Metric-Spec-Small attribute shall set the following two bits appropriately:
  - **mss-avail-stored-data:**
    If this bit is set, the pulse oximeter agent may send up to 25 temporarily stored measurements in an event report.
  - **mss-acc-agent-initiated:**
    Since the standard configuration contains two numeric objects, the bandwidth requirements are intended to be relatively light.

If an implementation is more capable than that provided by this standard configuration, it shall use an extended configuration.

### 6.6.3.3 Pulse rate—methods, events, services

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

See IEEE Std 11073-20601-2008 for descriptive explanations on the individual attributes as well as information on attribute ID and attribute type.

### 6.6.4 Pulsatile quality

Pulse oximeter vendors have various methods to express pulse amplitude, perfusion factor, or similar measure. These methods may comprise complex averaging formulas, scaling factors, and the like. In many cases, these various methods result in measurements that differ by a constant scaling value, but no standardized methodology currently exists to unify the notion of such a metric.

This numeric object serves as a container for expressing a quantifying element, which could be the modulation of the pulsatile signal or some other formulation, and agent implementers may choose the methods and attributes to express this numeric.

Since there are multiple ways to quantify the pulsatile waveform, more than one of these objects may be instantiated. For instance, one object can contain a pulse amplitude measurement, and a second object could contain some formulation of a defined pulsatile index.

Agent implementers should describe the implementation in the ICS tables described in Table 25 and Table 26. Information suitable for the DIM MOC ICS table should include the following:

- General implementation support of the measurement
- Commentary relating to guiding the use of the measurement value

Information suitable for the MOC attribute ICS table should include the following:

- Background on any unit code that may be used
- Description of any averaging method used
- Description of normalization values (e.g., the number reported is related to a root-mean-square value of peak)
- Information on which wavelength or wavelengths are used to calculate the measurement

Table 8 summarizes the attributes for building a framework to assist in expressing pulsatile quality.

**Table 8—Pulsatile quality numeric object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Type | {MDC_PART_SCADA,<br>MDC_PULS_OXIM_PERF_REL}<br>or {MDC_PART_SCADA,MDC_SAT_O2_QUAL} | M |
| Supplemental-Types | See IEEE Std 11073-20601-2008. | NR |
| Metric-Spec-Small | mss-acc-manager-initiated, mss-acc-agent-initiated | M |
| Metric-Structure-Small | See IEEE Std 11073-20601-2008. | NR |
| Measurement-Status | See IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | O |
| Metric-Id-List | See IEEE Std 11073-20601-2008. | NR |
| Metric-Id-Partition | See IEEE Std 11073-20601-2008. | NR |
| Unit-Code | See following text. | O |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008. | O |
| Source-Handle-Reference | See IEEE Std 11073-20601-2008. | O |
| Label-String | See following text. | R |
| Unit-LabelString | See following text. | R |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | O |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | O |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | O |
| Measure-Active-Period | See IEEE Std 11073-20601-2008. | O |
| Simple-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C |
| Compound-Simple-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C |
| Basic-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C |
| Compound-Basic-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C |
| Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C |
| Compound-Nu-Observed-Value | See IEEE Std 11073-20601-2008. | C |
| Accuracy | See IEEE Std 11073-20601-2008. | O |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

The unit code MDC_DIM_DIMLESS is the recommended dimension if this object is expressing a form of pulse amplitude, indicated by the type field set to MDC_PULS_OXIM_PERF_REL. Private unit codes may be used at the implementer's discretion, but could hinder the interoperability of the personal health infrastructure. If the type field contains MDC_SAT_O2_QUAL, the recommended dimension is MDC_DIM_PERCENT.

Similarly, the Label-String and Unit-LabelString attributes would be useful in conveying to the user more information regarding the characteristic being conveyed.

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

See IEEE Std 11073-20601-2008 for descriptive explanations on the individual attributes as well as information on attribute ID and attribute type.

## 6.7 Real-time sample array (RT-SA) objects

### 6.7.1 Plethysmographic waveform

A representation of the pulsatile wave may be transmitted as a series of samples.

Table 9 defines the attributes of the plesythmogram RT-SA object. This object is optional for this standard. The nomenclature code to identify the RT-SA object class is MDC_MOC_VMO_METRIC_SA_RT.

**Table 9—Plethysmogram object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Type | {MDC_PART_SCADA,<br>MDC_PULS_OXIM_PLETH } | M |
| Supplemental-Types | See IEEE Std 11073-20601-2008. | NR |
| Metric-Spec-Small | See following text. | M |
| Measurement-Status | See IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | NR |
| Unit-Code | See following text. | O |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008. | C |
| Source-Handle-Reference | See following text. | O |
| Label-String | See IEEE Std 11073-20601-2008. | O |
| Unit-LabelString | See IEEE Std 11073-20601-2008. | O |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Sample-Period | See IEEE Std 11073-20601-2008. | M |
| Simple-Sa-Observed-Value | See IEEE Std 11073-20601-2008. | M |
| Scale-and-Range-Specification | See IEEE Std 11073-20601-2008. | M |
| Sa-Specification | See IEEE Std 11073-20601-2008. | M |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

An agent should use the unit code of MDC_DIM_DIMLESS units as the basic value, as the sample points are often expressed as dimensionless units.

Several attributes in the plethysmogram are optional, and such flexibility allows implementations reasonable latitude in expressing details that the various vendors deem significant.

The plethysmogram data should be made available only through a scanner object. Consequently, the Metric-Spec-Small attribute should not have the mss-acc-manager-bit asserted. Asserting the mss-acc-agent-initiated bit is not recommended as this implies that this object's data are transmitted via MDS event reports. In other words, asserting neither the mss-acc-manager-initiated nor the mss-acc-acc-agent-initiated bits indicates that this object's data are transmitted only through a scanner object.

NOTE—The values in the RT-SA are intended to represent a plethysmogram, and these values may be related to the same values used to convey pulse amplitude information. The common practice of displaying a plethysmogram is such that the profile of the waveform will be an inversion of the absorbance waveform, in that a pulsatile inrush is seen as an upward deflection on a display. The manager is not required to perform any post-processing in order to invert the waveform, but the manager may need to be aware of this difference in order to present a more viewable waveform.

## 6.8 Enumeration objects

### 6.8.1 General

The pulse oximeter DIM (see Figure 1) contains three optional enumeration objects. The Pulsatile Occurrence enumeration object informs the user when a pulsatile wave is detected. The Pulsatile Characteristic object provides additional information about the pulsatile occurrence. Finally, an enumeration object is provided to enable an agent to report additional conditions concerning sensor status, general signal conditions, and device status in the device and sensor annunciation status object.

### 6.8.2 Pulsatile occurrence

Pulse oximeters often convey that they are detecting a pulsatile occurrence. One application of this object is to facilitate the flashing of a real-time display icon each time a pulsatile event is reported. Another application involves the ability to precisely timestamp the maximal inrush of a pulsatile wave. The nomenclature code to identify the enumeration object class is MDC_MOC_VMO_METRIC_ENUM. Refer to Table 10 for the set of attributes of this object.

This object is instantiated only in extended configurations, and it is optional to instantiate, but enables the ability to report the occurrence of a pulse. A manager should support the interpretation of this object in order to indicate the occurrence of a pulse.

**Table 10 —Pulsatile occurrence enumeration object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Type | {MDC_PART_SCADA, MDC_TRIG } | M |
| Supplemental-Types | See IEEE Std 11073-20601-2008. | O |
| Metric-Spec-Small | mss-acc-manager-initiated, mss-acc-agent-initiated | M |
| Metric-Structure-Small | See IEEE Std 11073-20601-2008. | NR |
| Measurement-Status | See IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | NR |
| Metric-Id-List | See IEEE Std 11073-20601-2008. | NR |
| Metric-Id-Partition | See IEEE Std 11073-20601-2008. | NR |
| Unit-Code | See following text. | NR |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008. | C |
| Source-Handle-Reference | See following text. | O |
| Label-String | See IEEE Std 11073-20601-2008. | O |
| Unit-LabelString | See IEEE Std 11073-20601-2008. | O |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | C |

**Table 10 —Pulsatile occurrence enumeration object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Enum-Observed-Value-Simple-OID | See IEEE Std 11073-20601-2008. | R |
| Enum-Observed-Value-Simple-Bit-Str | See following text. | NR |
| Enum-Observed-Value-Basic-Bit-Str | See following text. | NR |
| Enum-Observed-Value-Simple-Str | See IEEE Std 11073-20601-2008. | NR |
| Enum-Observed-Value | See following text. | NR |
| Enum-Observed-Value-Partition | See IEEE Std 11073-20601-2008. | O |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

Since this is essentially an event flag, the Unit-Code attribute is not appropriate for this object. If the Source-Handle-Reference is defined, it should point to either the Pulsatile Quality numeric object or the Plethysmogram RT-SA object.

Explicit expression of the existence of a pulsatile event is realized by sending the appropriate value in the Enum-Observed-Value-Simple-OID attribute. When reporting that a pulsatile occurrence has occurred, the value sent shall be MDC_TRIG_BEAT. If reporting that the maximal inrush of the pulsatile wave has occurred, the value sent shall be MDC_TRIG_BEAT_MAX_INRUSH.

The precise definition of “maximal inrush” in terms of slope or amplitude is vendor-dependent and should be defined in the appropriate ICS section.

If this object is sent within a periodic configurable scanner object and its object elements are reported using fixed format value messages (see 6.10.2), a placeholder value of MDC_METRIC_NOS may be sent to indicate that no pulsatile event occurred within the period that the Buf-Scan-Report-* event was emitted.

### 6.8.3 Pulsatile characteristic

Pulse oximeters may convey additional information about the pulsatile wave. The Pulsatile Characteristic object includes indication of adequate perfusion. The nomenclature code to identify the enumeration object class is MDC_MOC_VMO_METRIC_ENUM. Refer to Table 11 for the set of attributes of this object.

This object is instantiated only in extended configurations, and it is optional to instantiate, but enables the ability to report certain pulsatile conditions or artifacts. A manager should support the interpretation of this object in order to indicate characteristic information about a pulsatile wave.

**Table 11 —Pulse characteristic enumeration object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Type | {MDC_PART_SCADA,<br>MDC_PULS_OXIM_PULS_CHAR } | M |
| Supplemental-Types | See IEEE Std 11073-20601-2008. | O |
| Metric-Spec-Small | mss-acc-manager-initiated, mss-acc-agent-initiated | M |
| Metric-Structure-Small | See IEEE Std 11073-20601-2008. | NR |
| Measurement-Status | See IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | NR |

### Table 11 —Pulse characteristic enumeration object attributes

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Metric-Id-List | See IEEE Std 11073-20601-2008. | NR |
| Metric-Id-Partition | See IEEE Std 11073-20601-2008. | NR |
| Unit-Code | See following text. | NR |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008. | C |
| Source-Handle-Reference | See following text. | O |
| Label-String | See IEEE Std 11073-20601-2008. | O |
| Unit-LabelString | See IEEE Std 11073-20601-2008. | O |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Enum-Observed-Value-Simple-OID | See IEEE Std 11073-20601-2008. | O |
| Enum-Observed-Value-Simple-Bit-Str | See following text. | NR |
| Enum-Observed-Value-Basic-Bit-Str | See following text. | R |
| Enum-Observed-Value-Simple-Str | See IEEE Std 11073-20601-2008. | NR |
| Enum-Observed-Value | See following text. | NR |
| Enum-Observed-Value-Partition | See IEEE Std 11073-20601-2008. | O |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

Since these are essentially event flags, the Unit-Code attribute is not appropriate for this object. If the Source-Handle-Reference is defined, it should point to either the Pulse Amplitude numeric object or the Plethysmogram RT-SA object.

Explicit expression of the existence of characteristic events is realized by the setting of the appropriate bit in the Enum-Observed-Value-Basic-Bit-Str attribute, as defined in Table 12. It is recommended to use the Enum-Observed-Value-Basic-Bit-Str attribute as it consumes fewer payload octets than the Enum-Observed-Value-Simple-Bit-Str attribute. The Enum-Observed-Value attribute should not be used as it complicates the modeling of the object.

If a manager supports the interpretation of this object, it shall be able to interpret the entire set of presented characteristics, defined by the PulseOxPulsChar entity. An agent is not required to implement all the features specified in Table 12. An agent shall report each time there is a change in status in a condition. 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 12 —Mapping of pulse characteristics to object Bit-Str attribute

| Pulsatile characteristic | PulseOxPulsChar mnemonic |
|---|---|
| Quality of the detected pulse is nominal, in that there are no recognized abnormalities in the detected pulse. | pulse-qual-nominal |
| Perfusion or quality of the detected pulse is marginal. | pulse-qual-marginal |
| Perfusion or quality of the detected pulse is minimal. | pulse-qual-minimal |
| Perfusion or quality of the detected pulse is unacceptable. | pulse-qual-unacceptable |

The specific bit mappings of PulseOxPulseChar are defined in B.2.

### 6.8.4 Device and sensor annunciation conditions

The device and sensor annunciation status object reports on several aspects of the pulse oximeter equipment, including sensor displacement, faulty sensor detection, signal irregularities and loss-of-signal–tracking, and low-perfusion determination. The status of each aspect is reported by a specific status bit.

The device and sensor annunciation object containing the running status of the device, general signal condition, and sensor assembly is also provided. If this object is to be implemented, then the OID-Type and bit assignments shall be implemented as described in this subclause. Status bits in this object are distinguished from similar bits in the Pulsatile Characteristic object in that information in this object pertain to the general signal condition, whereas those in the Pulsatile Characteristic object are limited to a single pulsatile occurrence. The nomenclature code to identify the enumeration object class is MDC_MOC_ VMO_METRIC_ENUM. Refer to Table 13 for the set of attributes of this object.

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

**Table 13 —Device and sensor annunciation status object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Type | {MDC_PART_SCADA,<br>MDC_PULS_OXIM_DEV_STATUS }. | M |
| Supplemental-Types | See IEEE Std 11073-20601-2008. | O |
| Metric-Spec-Small | mss-acc-manager-initiated, mss-acc-agent-initiated. | M |
| Metric-Structure-Small | See IEEE Std 11073-20601-2008. | NR |
| Measurement-Status | See IEEE Std 11073-20601-2008. | O |
| Metric-Id | See IEEE Std 11073-20601-2008. | NR |
| Metric-Id-List | See IEEE Std 11073-20601-2008. | NR |
| Metric-Id-Partition | See IEEE Std 11073-20601-2008. | NR |
| Unit-Code | See following text. | NR |
| Attribute-Value-Map | See IEEE Std 11073-20601-2008. | C |
| Source-Handle-Reference | See following text. | NR |
| Label-String | See IEEE Std 11073-20601-2008. | O |
| Unit-LabelString | See IEEE Std 11073-20601-2008. | O |
| Absolute-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Relative-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| HiRes-Time-Stamp | See IEEE Std 11073-20601-2008. | C |
| Enum-Observed-Value-Simple-OID | See IEEE Std 11073-20601-2008. | O |
| Enum-Observed-Value-Simple-Bit-Str | See following text. | R |
| Enum-Observed-Value-Basic-Bit-Str | See following text. | NR |
| Enum-Observed-Value-Simple-Str | See IEEE Std 11073-20601-2008. | NR |
| Enum-Observed-Value | See following text. | NR |
| Enum-Observed-Value-Partition | See IEEE Std 11073-20601-2008. | O |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

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.

Explicit expression of the existence of annunciations is realized by the setting of the appropriate bit in the Enum-Observed-Value-Basic-Bit-Str attribute, as defined in Table 14. It is recommended to use the Enum-Observed-Value-Basic-Bit-Str attribute as it consumes fewer payload octets than the Enum-Observed-Value-Simple-Bit-Str attribute. The Enum-Observed-Value attribute should not be used as it unnecessarily complicates the modeling of the object. If a manager supports the interpretation of this object, it shall be able to interpret the entire set of presented conditions, defined by the PulseOxDevStat entity. An agent is not required to implement all the features specified in Table 14. An agent shall report each time there is a change in status in a condition. 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 14—Mapping of device, sensor and signal status to object Bit-Str attribute**

| Device or sensor condition | PulseOxDevStat mnemonic |
|---|---|
| Agent reports that the sensor is disconnected from the instrument. | sensor-disconnected |
| Agent reports that the sensor is malfunctioning or faulting. | sensor-malfunction |
| Agent reports that the sensor is not properly attached or has been dislodged, and accurate measurement is, therefore, prevented. | sensor-displaced |
| An unsupported sensor is connected to the agent. | sensor-unsupported |
| Agent reports that sensor is not connected to the user. | sensor-off |
| Agent reports that there is interference due to ambient light or electrical phenomena. | sensor-interference |
| Signal analysis is currently in progress prior to measurement availability. | signal-searching |
| Agent determines that a questionable pulse is detected. | signal-pulse-questionable |
| Agent detects a nonpulsatile signal. | signal-non-pulsatile |
| Agent reports that the signal is erratic or is not plausible. | signal-erratic |
| Agent reports a consistently low perfusion condition exists. | signal-low-perfusion |
| Agent reports a poor signal exists, possibly affecting accuracy. | signal-poor |
| Agent reports that the incoming signal cannot be analyzed or is inadequate for producing a meaningful result. | signal-inadequate |
| Agent has determined that some irregularity has been detected while processing the signal. | signal-processing-irregularity |
| A general device fault has occurred in the agent. | device-equipment-malfunction |
| An extended display update is currently active. | device-extended-update |

The specific bit mappings of PulseOxDevStat are defined in B.3.

## 6.9 PM-store objects

### 6.9.1 General

Several use cases illustrate that a pulse oximetry agent may store many minutes or hours worth of oximetry data while unable to remain in communication with a manager, or it is impractical to send multiple event reports of blocks of temporarily stored measurements. After the long-term acquisition is complete, the manager retrieves the data from the agent. This mechanism is allowed in extended configurations.

### 6.9.2 Persistent store model

The wide range of potential combinations of data layouts makes it impractical to provide a specification for a single normative persistent store data model. As such, a pulse oximetry agent has considerable latitude in selecting the format and set of data elements to transmit. If a pulse oximetry agent supports this function, the framework in the following subclause should be followed. The intent of this approach is to provide a "file system description" of the data layout, as opposed to a "file format specification." In other words, 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 is outside the scope of this standard.

The pulse oximeter in this use case stores oximetry data in a number of different ways, depending on the particular needs of the acquisition. The information model for the persistent store hierarchy is shown in Figure 2. As an example and pattern, Figure 3 illustrates the relationship between the various objects for the PM-store implementation. The PM-segments could group data in different ways. It could contain all varieties of data within one session, or multiple PM-segments could be created, with one containing all $SpO_2$ measurements for the session and a second containing all pulse rate measurements for that same session. However, the hierarchy of the PM-segment, entry, and elements should take the form seen in Figure 3.

This example illustrates a PM-store realization with two PM-segments. In this case, each PM-segment stores data from a distinct contiguous session. Figure 3 shows each PM-segment entry containing two data elements: the first representing an $SpO_2$ measurement and the second representing a pulse rate measurement. Since each entry contains the set of consistently ordered data sampled at a single point in time, one could place timestamp information in the SegEntryHdr, indicating the occurrence of each reading. If the samples are taken at fixed intervals, then the start time and sampling interval should be stored in the PM-segment attributes MDC_ATTR_TIME_START_SEG and MDC_ATTR_TIME_PD_ SAMP, and the SegEntryHdr may be left empty, whereas if the samples are not taken at fixed intervals, the timestamp of each sample must be stored in each SegEntryHdr.

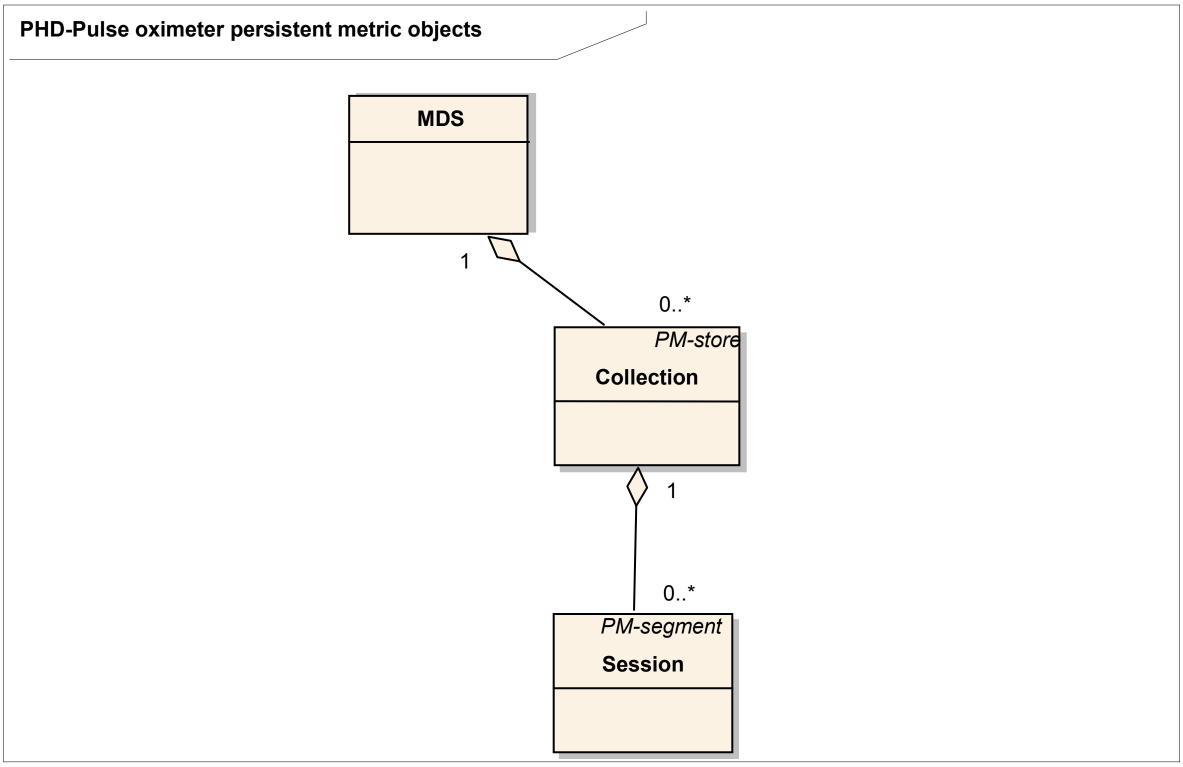

**Figure 2— Pulse oximeter DIM for the PM-store hierarchy**

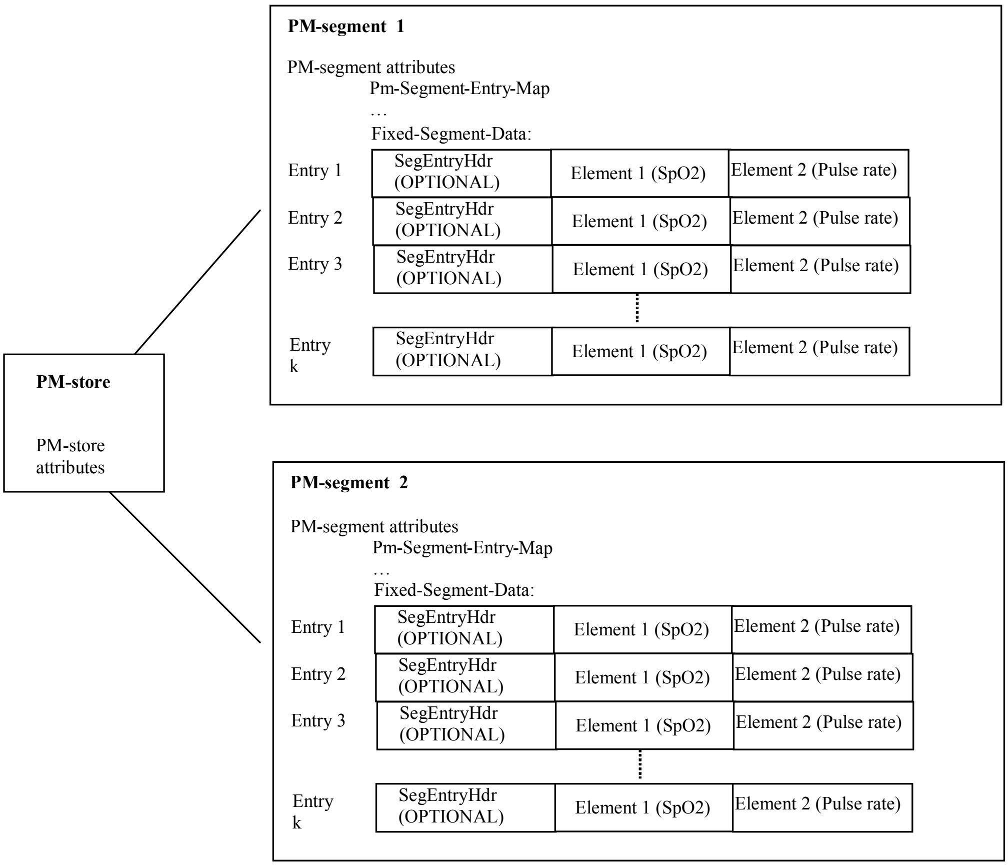

**Figure 3—PM-store usage for pulse oximeter**

### 6.9.3 PM-store object attributes

Table 15 lists the attributes for the PM-store object.

**Table 15—PM-store object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| PM-Store-Capab | See IEEE Std 11073-20601-2008. | M |
| Store-Sample-Algorithm | See IEEE Std 11073-20601-2008. | M |
| Store-Capacity-Count | See IEEE Std 11073-20601-2008. | O |
| Store-Usage-Count | See IEEE Std 11073-20601-2008. | O |
| Operational-State | See IEEE Std 11073-20601-2008. | M |
| PM-Store-Label | See IEEE Std 11073-20601-2008. | O |
| Sample-Period | See IEEE Std 11073-20601-2008. | C |
| Number-Of-Segments | See IEEE Std 11073-20601-2008. | M |
| Clear-Timeout | See IEEE Std 11073-20601-2008. | M |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

### 6.9.4 PM-store object methods

Table 16 defines the methods used by the PM-store object.

Table 16 —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 |

### 6.9.5 PM-store object events

Table 17 defines the events sent by the PM-store object.

Table 17 —PM-store object events

| Event | Mode | Event type | Event info parameter | Event-reply-info |
|---|---|---|---|---|
| Segment-Data-Event | Confirmed | MDC_NOTI_SEGMENT_DATA | SegmentDataEvent | SegmentDataResult |

To facilitate a practical level of device support, a Segment-Data-Event report size shall be no larger than 8192 octets. A PM-segment containing data in excess of this size shall transfer its data using multiple Segment-Data-Event reports as described in IEEE Std 11073-20601-2008.

### 6.9.6 PM-store object services

The GET service shall be provided by an agent implementing one or more PM-store objects. This service shall be available while the agent is in the Operating state. Refer to IEEE Std 11073-20601-2008 for further details.

### 6.9.7 PM-segment objects

Table 18 defines the attributes of the PM-segment object contained in the PM-store object managing the stored measurements. The nomenclature code to identify the PM-segment class is MDC_MOC_PM_ SEGMENT.

Table 18 —PM-segment object attributes

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Instance Number | See IEEE Std 11073-20601-2008. | M |
| PM-Segment-Entry-Map | See IEEE Std 11073-20601-2008. | M |
| PM-Seg-Person-Id | See IEEE Std 11073-20601-2008. | C |
| Sample-Period | See IEEE Std 11073-20601-2008. | C |
| Operational-State | See IEEE Std 11073-20601-2008. | M |
| Segment-Label | See IEEE Std 11073-20601-2008. | O |
| Segment-Start-Abs-Time | See IEEE Std 11073-20601-2008. | M |
| Segment-End-Abs-Time | See IEEE Std 11073-20601-2008. | M |
| Date-and-Time-Adjustment | See IEEE Std 11073-20601-2008. | C |
| Segment-Usage-Count | See IEEE Std 11073-20601-2008. | O |
| Segment-Statistics | See IEEE Std 11073-20601-2008. | O |
| Fixed-Segment-Data | Specified by vendor. | M |
| Confirm-Timeout | See IEEE Std 11073-20601-2008. | O |
| Transfer-Timeout | See IEEE Std 11073-20601-2008. | M |

The Fixed-Segment-Data attribute serves as the container of the stored measurements. The exact data format or data type of this attribute is vendor-specific.

## 6.10 Scanner objects

### 6.10.1 General

The scanner object class is a powerful construct that enables efficient grouping of several metrics into a single payload. It is also helpful in conveying the continuous nature of annunciations expressed within enumeration objects, as the scanner object can periodically query or observe the enumeration object dedicated to a particular part of status recording. The information model for the scanner hierarchy is shown in Figure 4. The periodic or episodic configurable scanner objects are instantiated only in extended configurations. The nomenclature code to identify the periodic configurable scanner object class is MDC_MOC_SCAN_CFG_PERI, and the code to identify the episodic configurable scanner is MDC_ MOC_SCAN_CFG_EPI.

Figure 5 illustrates an example collection of data that would be transmitted as an associated block of information from a periodic configurable scanner. This construct enables the packaging of data as an associated set of measurements. The periodicity of these measurements also allows the expression of the continuous nature of what might be considered to be indicator statuses, as the appropriate bits would continue to be asserted for the entire interval that the condition is valid. A simplified example of the scanner configuration and data reporting can be found in E.6.

The episodic configurable scanner can be used to send episodic observations, e.g., the pulse occurrence, only when these observations happen.

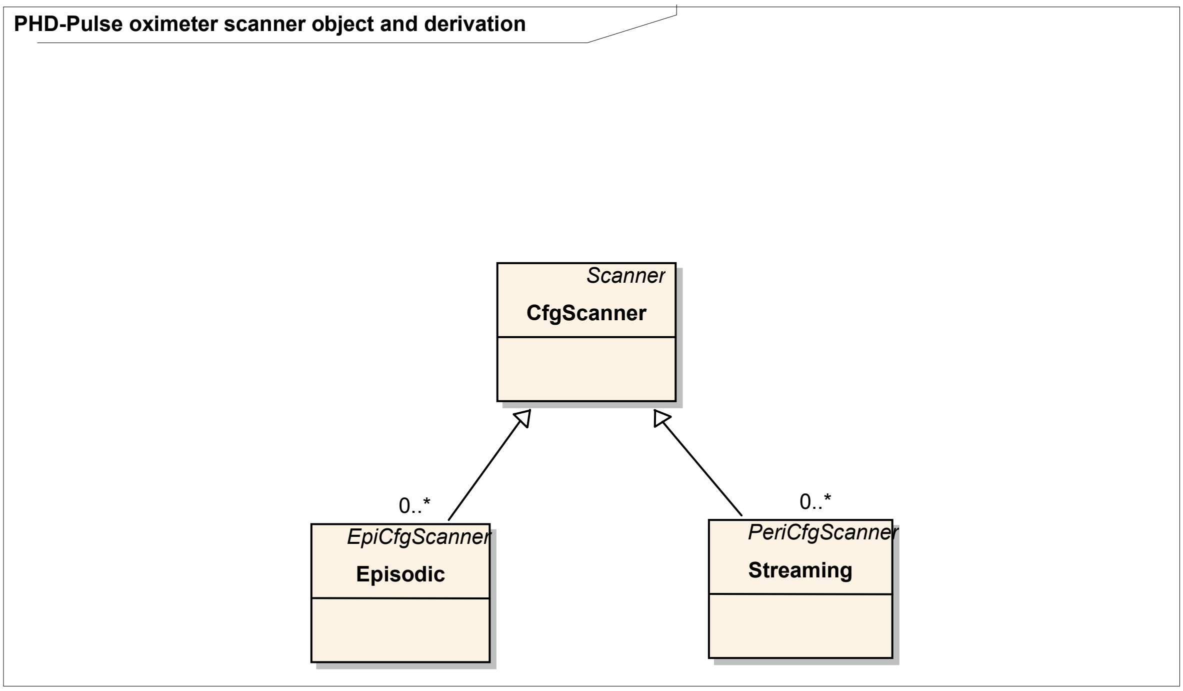

**Figure 4— Pulse oximeter DIM for the periodic configurable scanner object**

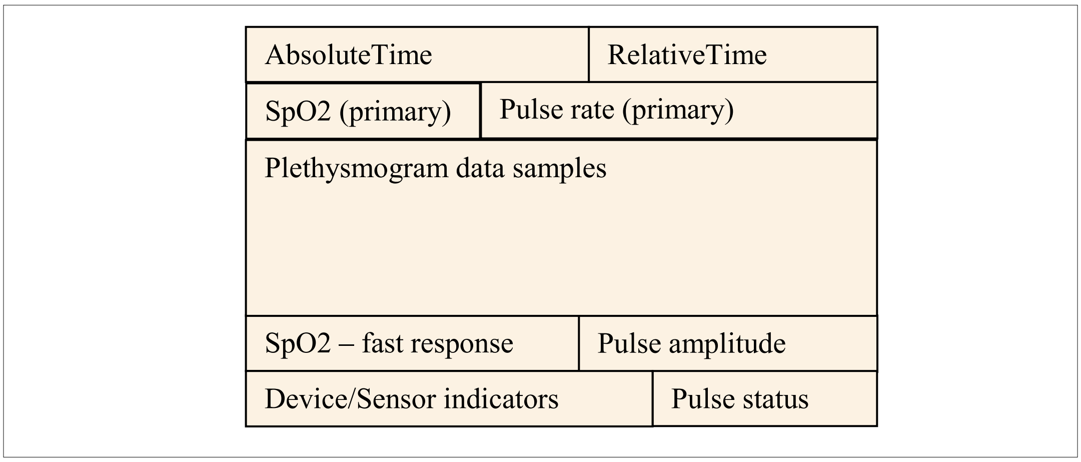

**Figure 5— Pulse oximeter scanner object usage example**

Since IEEE Std 11073-20601-2008 requires the manager to support grouped-format event reports, a manager must support the interpretation of this object class if the agent transmits data using periodic or episodic scanner objects. Otherwise, if the agent presents the bulk of its data with scanner objects, the manager cannot receive the data presented by such an agent.

### 6.10.2 Periodic configurable scanner attributes

Table 19 shows the attributes applicable to the periodic configurable scanner object.

**Table 19—Periodic configurable scanner object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Operational-State | See IEEE Std 11073-20601-2008. | M |
| Scan-Handle-List | See IEEE Std 11073-20601-2008. | C |
| Scan-Handle-Attr-Val-Map | See IEEE Std 11073-20601-2008. | C |
| Confirm-Mode | See IEEE Std 11073-20601-2008. | M |
| Confirm-Timeout | See IEEE Std 11073-20601-2008. | C |
| Transmit-Window | See IEEE Std 11073-20601-2008. | C |
| Reporting-Interval | See IEEE Std 11073-20601-2008. | M |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

With regard to the Confirm-Mode attribute, an agent may support either or both confirmed or unconfirmed scan reports; the manager shall support both confirmed and unconfirmed scan reports.

One or more periodic configurable scanner objects may be employed by a pulse oximeter in order to greatly increase the efficiency of delivering physiological and device information to the manager.

The events in Table 20 define the events sent by the pulse oximeter's periodic configurable scanner object.

**Table 20—Periodic configurable scanner object events**

| Event | Mode | Event type | Event info parameter | Event-reply-info |
|---|---|---|---|---|
| Buf-Scan-Report-Var | Confirmed or unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_VAR | ScanReportInfoVar | — |
| Buf-Scan-Report-Fixed | Confirmed or unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_FIXED | ScanReportInfoFixed | — |
| Buf-Scan-Report-Grouped | Confirmed or unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_GROUPED | ScanReportInfoGrouped | — |
| Buf-Scan-Report-MP-Var | Confirmed or unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_MP_VAR | ScanReportInfoMPVar | — |
| Buf-Scan-Report-MP-Fixed | Confirmed or unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_MP_FIXED | ScanReportInfoMPFixed | — |
| Buf-Scan-Report-MP-Grouped | Confirmed or unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_MP_GROUPED | ScanReportInfoMPGrouped | — |

### 6.10.3 Episodic configurable scanner attributes

Table 21 shows the attributes applicable to the episodic configurable scanner object.

**Table 21 —Episodic configurable scanner object attributes**

| Attribute name | Extended configuration | |
|---|---|---|
| | Value | Qual. |
| Handle | See IEEE Std 11073-20601-2008. | M |
| Operational-State | See IEEE Std 11073-20601-2008. | M |
| Scan-Handle_list | See IEEE Std 11073-20601-2008. | C |
| Scan-Handle-Attr-Val-Map | See IEEE Std 11073-20601-2008. | C |
| Confirm-Mode | See IEEE Std 11073-20601-2008. | M |
| Confirm-Timeout | See IEEE Std 11073-20601-2008. | C |
| Transmit-Window | See IEEE Std 11073-20601-2008. | C |
| Min-Reporting-Interval | See IEEE Std 11073-20601-2008. | M |

NOTE—See IEEE Std 11073-20601-2008 for information on whether an attribute is static or dynamic.

NOTE—See 6.3 for a description of the qualifiers.

With regard to the Confirm-Mode attribute, an agent may support either or both confirmed or unconfirmed scan reports; the manager shall support both confirmed and unconfirmed scan reports.

One or more episodic configurable scanner objects may be employed by a pulse oximeter in order to greatly increase the efficiency of delivering physiological and device information to the manager.

The events in Table 22 define the events sent by the pulse oximeter's episodic configurable scanner object.

**Table 22 —Episodic configurable scanner object events**

| Event | Mode | Event type | Event info parameter | Event-reply-info |
|---|---|---|---|---|
| Unbuf-Scan-Report-Var | Confirmed or unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_VAR | ScanReportInfoVar | — |
| Unbuf-Scan-Report-Fixed | Confirmed or unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_FIXED | ScanReportInfoFixed | — |
| Unbuf-Scan-Report-Grouped | Confirmed or unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED | ScanReportInfoGrouped | — |
| Unbuf-Scan-Report-MP-Var | Confirmed or unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_MP_VAR | ScanReportInfoMPVar | — |
| Unbuf-Scan-Report-MP-Fixed | Confirmed or unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_MP_FIXED | ScanReportInfoMPFixed | — |
| Unbuf-Scan-Report-MP-Grouped | Confirmed or unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_MP_GROUPED | ScanReportInfoMPGrouped | — |

### 6.11 Class extension objects

In this standard, no class extension objects are defined with respect to IEEE Std 11073-20601-2008.

### 6.12 Pulse oximeter information model extensibility rules

The pulse oximeter DIM of this standard may be extended by including vendor-specific metrics and attributes as required. For example, a vendor might include a blood pressure measurement in addition to $SpO_2$ and pulse rate measurements. Any object or attribute extensions implemented should follow the guidelines of this standard as closely as possible.

A pulse oximeter agent having a configuration with extensions beyond the standard configuration, as specified in this standard, shall use a configuration ID in the range of IDs reserved for extended configurations (see IEEE Std 11073-20601-2008).

## 7. Pulse oximeter 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 manager. Protocol messages within the ISO/IEEE 11073 series of standards are defined in ASN.1. See IEEE Std 11073-20601-2008 for a detailed description of the personal health device service model. The following subclauses define the specifics of object access and event reporting services for a pulse oximeter agent according to this standard.

### 7.2 Object access services

The object access services of IEEE Std 11073-20601-2008 are used to access the objects defined in the DIM of the pulse oximeter.

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

- GET service: used by the manager to retrieve the values of the agent MDS object and PM-store object attributes. The list of pulse oximeter MDS object attributes is given in 6.5.4.1.
- SET service: used by the manager to set the values of the agent object attributes. If the agent supports scanner objects, then the SET service shall be supported. If the agent does not support scanner objects, the SET service is optional. Note that although MDS attributes may be set, this standard does not allow the use of settable attributes as a mechanism for remote control.
- EVENT REPORT service: used by the agent to send configuration reports and measurement data to the manager. The list of event reports for the pulse oximeter device specialization is given in Table 3, Table 17, and Table 20.
- 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 23 summarizes the object access services described in this standard.

### Table 23 —Pulse oximeter 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 an attribute of an object in the agent. |
| | <na> | <implied Confirmed> | <na> | GetArgumentSimple = (obj-handle = handle of PM-store object), attribute-id-list <optional> | GetResultSimple = (obj-handle = handle of PM-store object), attribute-list | Allows the manager to retrieve the values of all PM-store object attributes. |
| EVENT REPORT | MDS-Configuration-Event | Confirmed | MDC_NOTI_ CONFIG | ConfigReport | ConfigReport Rsp | Configuration report to inform manager of the configuration of the agent. |
| | MDS-Scan-Report-Var | Confirmed or Unconfirmed | 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-Scan-Report-Fixed | Confirmed or Unconfirmed | 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-Scan-Report MP-Var | Confirmed or Unconfirmed | 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-Scan-Report-MP-Fixed | Confirmed or Unconfirmed | MDC_NOTI_ SCAN_ REPORT_ MP_FIXED | ScanReportInfoMPFixed | | |
| | Segment-Data-Event | Confirmed | MDC_NOTI_ SEGMENT_ DATA | SegmentDataEvent | SegmentData Resul | |
| | Buf-Scan-Report-Var | Confirmed or Unconfirmed | MDC_NOTI_ BUF_SCAN_ REPORT_ VAR | ScanReportInfoVar | | |

**Table 23 —Pulse oximeter object access services**

| Service | Subservice type name | Mode | Subservice type | Parameters | Result | Remarks |
|---|---|---|---|---|---|---|
| | Buf-Scan-Report-Fixed | Confirmed or Unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_FIXED | ScanReportInfoFixed | - | |
| | Buf-Scan-Report-Grouped | Confirmed or Unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_GROUPED | ScanReportInfoGrouped | - | |
| | Buf-Scan-Report-MP-Var | Confirmed or Unconfirmed | MDC_NOTI_BUF_SCAN_REPORT MP_ VAR | ScanReportInfoMPVar | | |
| | Buf-Scan-Report-MP-Fixed | Confirmed or Unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_MP_FIXED | ScanReportInfoMPFixed | - | |
| | Buf-Scan-Report-MP-Grouped | Confirmed or Unconfirmed | MDC_NOTI_BUF_SCAN_REPORT_MP_GROUPED | ScanReportInfoMPGrouped | - | |
| | Unbuf-Scan-Report-Var | Confirmed or Unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_VAR | ScanReportInfoVar | | |
| | Unbuf-Scan-Report-Fixed | Confirmed or Unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_FIXED | ScanReportInfoFixed | - | |
| | Unbuf-Scan-Report-Grouped | Confirmed or Unconfirmed | MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED | ScanReportInfoGrouped | - | |

Table 23 —Pulse oximeter object access services

| Service | Subservice type name | Mode | Subservice type | Parameters | Result | Remarks |
|---|---|---|---|---|---|---|
| | Unbuf-Scan-Report-MP-Var | Confirmed or Unconfirmed | MDC_NOTI_ UNBUF_ SCAN_ REPORT_ MP_VAR | ScanReportInfoMPVar | | |
| | Unbuf-Scan-Report-MP-Fixed | Confirmed or Unconfirmed | MDC_NOTI_ UNBUF_ SCAN_ REPORT_ MP_FIXED | ScanReportInfoMPFixed | - | |
| | Unbuf-Scan-Report-MP-Grouped | Confirmed or Unconfirmed | MDC_NOTI_ UNBUF_ SCAN_ REPORT_ MP_GROUPED | ScanReportInfoMPGrouped | - | |
| ACTION | Set-Time | Confirmed | MDC_ACT_ SET_TIME | SetTimeInvoke | | Manager method to invoke the agent to set time to requested value. |
| | MDS-Data-Request | Confirmed | MDC_ACT_ DATA_ REQUEST | DataRequest | DataResponse | Allows the manager to enable or disable measurement data transmission from the agent. |
| | 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 | Allows the manager to begin sending segment data. |
| SET | <na> | | <na> | SetArgumentSimple | | Allows the manager to control the operational state of a scanner object. |

## 7.3 Object access EVENT REPORT services

The EVENT REPORT service (see Table 23) is used by the agent to report its information (e.g., measurements). Event reports in this standard are a property of the MDS object only. The event reports used in this standard are defined in IEEE Std 11073-20601-2008.

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

- Event reports transmitting measurement data may be used either in confirmed or unconfirmed mode.
- Some agent implementations support agent-initiated mode only through MDS object emission. These implementations should take care to not send any relatively high-bandwidth measurements such as plethysmogram data in this manner.
- Some agent implementations support agent-initiated mode only through the use of periodic or episodic configurable scanner objects. The manager shall be aware of the behavior of scanner object data transfer by controlling the Operational-State attribute of the appropriate scanner object.
- Manager-initiated mode may be supported for measurement data transmission.

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

A manager shall support both single-person and multiple-person event reports. A pulse oximeter agent shall support single-person event reports and may support multiple-person event reports. The formats for single and multiple-person reports are described in IEEE Std 11073-20601-2008.

# 8. Pulse oximeter communication model

## 8.1 Overview

The following subclauses describe the general communication model and procedures of the pulse oximeter agent as defined in IEEE Std 11073-20601-2008. Therefore, the respective parts of IEEE Std 11073-20601-2008 are not reproduced, rather the specific choices and restrictions with respect to optional elements (e.g., attributes and procedures) and specific extensions (e.g., nomenclature terms) are specified.

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

## 8.2 Communications characteristics

For a pulse oximeter agent implementing no other device specialization except this standard, the maximum size of an application protocol data unit (APDU) sent using medical device encoding rules (MDER) shall be not larger than $N_{tx}$. For this standard, $N_{tx}$ = 9216 octets, based on the maximum APDU size that could be expected from a pulse oximetry agent implementing persistent metric capability (see 6.9.5). In the absence of the persistent metric capability, $N_{tx}$ = 5120 octets, based on the maximum APDU size that could be expected from a pulse oximetry agent allowing sufficient capacity for several OCTET STRING entities within the MDS Production Specification attribute. An agent according to this definition shall be capable of receiving an APDU using MDER up to the size of at least $N_{rx}$. For this standard, $N_{rx}$ = 256 octets, based on an Association Response with an allowance for a set of information in the option-list element of the PHDAssociationInformation structure of an Association Request.

If an agent implementing this device specialization implements functions from other device specializations, the following upper bound estimates for transmitted and received APDUs may serve as guidance:

An agent shall not transmit any APDU larger than the sum of $N_{tx}$ of all the device specializations implemented and shall be capable of receiving any APDU up to the sum of $N_{rx}$ of all the device specializations implemented. If these numbers are higher than the maximum size determined in IEEE Std 11073-20601-2008, the latter shall be applied.

## 8.3 Association procedure

### 8.3.1 General

Unless otherwise stated, the association procedure for a pulse oximetry agent and manager according to this standard shall be pursued as specified in IEEE Std 11073-20601-2008.

### 8.3.2 Agent procedure—association request

In the Association Request message sent by the agent to the manager, the following points apply:

- 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, which shall contain the following parameter values:
  - The version of the data exchange protocol shall be set to protocol-version1 (i.e., *protocol-version* = 0x80000000).
  - At least the MDER shall be supported (i.e., *encoding-rules* = 0x8000).
  - The version of the nomenclature used shall be set to nom-version1 (i.e., *nomenclature-version* = 0x80000000).
  - The *functional-units* field may have the test association bits set, but shall not have any other bits set.
  - The *system-type* field 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 pulse oximeter 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 pulse oximeter specialization, then the field indicating the data request modes (data-req-mode-capab) supported by the pulse oximeter agent shall be set based on the communication capabilities of the agent.

If the agent supports agent-initiated measurement transfer, then data-req-mode-capab shall have the data-req-supp-init-agent bit set. The data-req-init-manager-count shall be set to zero, and data-req-init-agent-count shall be set to 1.

If the agent supports manager-initiated measurement transfer, then data-req-mode-capab shall have the data-req-supp-stop, data-req-supp-scope-handle, data-req-supp-mode-single-rsp, and data-req-supp-mode-time-no-limit bits appropriately set. The data-req-init-manager-count shall be set to the maximum number of concurrent manager-initiated flows supported by the agent, and data-req-init-agent-count shall be set to 0 or 1.

### 8.3.3 Manager procedure—association response

In the Association Response message sent by the manager, the following points apply:

- The result field shall be set to an appropriate response from the responses defined in IEEE Std 11073-20601-2008. 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, which shall contain the following parameter values:
    - The version of the data exchange protocol shall be set to protocol-version1 (i.e., *protocol-version* = 0x80000000).
    - 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 version of the nomenclature used shall be set to nom-version1 (i.e., *nomenclature-version* = 0x80000000).
    - The *functional-units* field shall have all bits reset except for those relating to a test association.
    - The *system-type* field 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 device, which shall be a valid EUI-64 type identifier.
    - The *dev-config-id* field shall be manager-config-response (0).
    - The *data-req-mode-capab-flags* field shall be 0.
    - The *data-req-init-*-count* fields shall be 0.

## 8.4 Configuring procedure

### 8.4.1 General

The agent enters the Configuring state if it receives an Association Response of *accepted-unknown-config*. In this case, the configuration procedure as specified in IEEE Std 11073-20601-2008 shall be followed. The following subclauses specify the configuration notification and response messages for a pulse oximetry agent with standard configuration ID 0x0190. Normally, a manager would already know the standard configuration. However, for the purposes of this example, it does not.

### 8.4.2 Pulse oximeter—standard configuration

#### 8.4.2.1 Agent procedure

The agent performs the configuration procedure using a “Remote Operation Invoke | Confirmed Event Report” message with an MDC_NOTI_CONFIG event to send its configuration to the manager (see IEEE Std 11073-20601-2008). The ConfigReport structure is used for the event-info field (see Table 3). For a pulse oximetry agent with standard configuration ID 0x0190, the format and contents of the configuration notification message are as follows:

| | |
|---|---|
| `0xE7 0x00` | APDU CHOICE Type (PrstApdu) |
| `0x00 0x68` | CHOICE.length = 104 |

```
0x00 0x66               OCTET STRING.length = 102
0x12 0x35               invoke-id (differentiates this from other outstanding messages)
0x01 0x01               CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x60               CHOICE.length = 96
0x00 0x00               obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF     event-time = 0xFFFFFFFF
0x0D 0x1C               event-type = MDC_NOTI_CONFIG
0x00 0x56               event-info.length = 86 (start of ConfigReport)
0x01 0x90               config-report-id
0x00 0x02               config-obj-list.count = 2 Measurement objects will be "announced"
0x00 0x50               config-obj-list.length = 80
0x00 0x06               obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01               obj-handle = 1     (→ 1st Measurement is SpO2)
0x00 0x04               attributes.count = 4
0x00 0x20               attributes.length = 32
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x02 0x4B 0xB8     MDC_PART_SCADA | MDC_PULS_OXIM_SAT_O2
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0x40 0x40               avail-stored-data, acc-agent-init, measured
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x02 0x20               MDC_DIM_PERCENT
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x08               attribute-value.length = 8
0x00 0x01               AttrValMap.count = 1
0x00 0x04               AttrValMap.length = 4
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x00 0x06               obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x0A               obj-handle = 10   (→2nd Measurement is pulse rate)
0x00 0x04               attributes.count = 4
0x00 0x20               attributes.length = 32
0x09 0x2F               attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04               attribute-value.length = 4
0x00 0x02 0x48 0x1A     MDC_PART_SCADA | MDC_PULS_OXIM_PULS_RATE
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0x40 0x40               avail-stored-data, acc-agent-init, measured
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x0A 0xA0               MDC_DIM_BEAT_PER_MIN
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x08               attribute-value.length = 8
0x00 0x01               AttrValMap.count = 1
0x00 0x04               AttrValMap.length = 4
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC, 2
```

#### 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 3). An example configuration notification response message (corresponding to the configuration notification request message described in 8.4.2.1) can be seen as follows:

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x16                CHOICE.length = 22
0x00 0x14                OCTET STRING.length = 20
0x12 0x35                invoke-id = 0x1235 (mirrored from invocation)
0x02 0x01                CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E                CHOICE.length = 14
0x00 0x00                obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00      currentTime = 0
0x0D 0x1C                event-type = MDC_NOTI_CONFIG
0x00 0x04                event-reply-info.length = 4
0x01 0x90                ConfigReportRsp.config-report-id = 0x190
0x00 0x00                ConfigReportRsp.config-result = accepted-config
```

## 8.5 Operating procedure

### 8.5.1 General

Measurement data and status information are communicated from the pulse oximetry agent during the Operating state. If not stated otherwise, the operating procedure for a pulse oximetry agent of this standard shall be as specified in IEEE Std 11073-20601-2008.

### 8.5.2 GET pulse oximeter MDS attributes

See Table 4 for a summary of the GET service.

If the manager leaves the attribute-id-list field in the roiv-cmip-get service message empty, the pulse oximetry agent shall respond with a rors-cmip-get service message in which the attribute-list contains a list of all implemented attributes of the MDS object.

If the manager requests specific MDS object attributes, indicated by the elements in attribute-id-list, and the agent supports this capability, the pulse oximetry agent shall respond with a rors-cmip-get service message in which the attribute-list contains a list of the requested attributes of the MDS object that are implemented. It is not required for a pulse oximetry agent to support this capability. If this capability is not implemented, the pulse oximetry agent shall respond with a “Remote Operation Error Result” (roer) service message (see IEEE Std 11073-20601-2008) with the error-value field set to no-such-action(9).

### 8.5.3 Measurement data transmission

See Table 3, Table 17, and Table 20 for a summary of the EVENT REPORT services available for measurement data transfer.

Measurement data transfer for a pulse oximetry agent of this standard may be initiated by either the agent or the manager (see agent- and manager-initiated measurement data transmission in IEEE Std 11073-20601-2008). To limit the amount of data being transported within an APDU, the pulse oximetry 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 may be sent either using multiple event reports or by incorporating a persistent store facility. If multiple oximetry 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 oximetry 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 pulse oximetry agent and a manager to coordinate the clocks used when reporting physiological events. Note that the mechanism for synchronizing an agent to a manager is outside the scope of this standard. If time synchronization is used, then this shall be reported in the Mds-Time-Info attribute of the MDS object.

## 9. Test associations

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

### 9.1 Behavior with standard configuration

In order to facilitate automated standardized test processes, a pulse oximeter that presents the standard configuration ID and enters into a test association should be able to simulate the arrival of measurement data from the device sensors. It should not be necessary for an operator to stimulate the sensors in order for the measurement data to be generated.

After the agent enters the Operating state, it simulates the reception of an event from the sensors representing an $SpO_2$ measurement of 123% and a pulse rate of 456 beats per minute. To the extent possible, this measurement is seen only by those components of the agent that understand the test association. When the event is propagated into a numeric object, the test-data bit of the measurement-status attribute shall be set if the measurement-status attribute is supported. An agent is not required to use the measurement-status attribute if it would not normally do so outside of a test association.

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

### 9.2 Behavior with extended configurations

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

## 10. Conformance

### 10.1 Applicability

This standard shall be used in conjunction with IEEE Std 11073-20601-2008.

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.

Since this standard is used in conjunction with IEEE Std 11073-20601-2008, the ICS should be created for this standard first. The ICS created for IEEE Std 11073-20601-2008 may then refer to the ICS for this standard where applicable.

## 10.3 Levels of conformance

### 10.3.1 General

This standard defines the following levels of conformance:

### 10.3.2 Conformance Level 1: base conformance

The application uses elements of the information, service, and communication models (object hierarchy, actions, event reports, and data type definitions) and the nomenclature scheme defined in IEEE Std 11073-20601-2008 and the ISO/IEEE P11073-104zz standards. All mandatory features defined in the object definition tables and in the ICS tables are implemented. Further, any conditional, recommended, or optional features that are implemented shall follow the requirements in IEEE Std 11073-20601-2008 and the ISO/IEEE P11073-104zz standards.

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

Conformance Level 2 meets conformance level 1, but also uses or adds extensions in at least one of the information, service, communication, or nomenclature models. These extensions shall conform to nomenclature codes from ASN.1 and/or within the ISO/IEEE 11073-10101 framework (0xF000 – 0xFFFF). These extensions should be defined in ICS tables pointing towards 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 specific ICS tables in the form given by the templates in 10.4.2 through 10.4.6.

Each ICS table has the following columns:

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

The table column headings have the following meanings:

— Index, which is an ID (e.g., a tag) of a specific feature.

— Feature, which briefly describes the characteristic for which a conformance statement is being made.

— Reference, which is to the clause or subclause within this standard or an external source for the definition of the feature (may be empty).

— Req./Status, which specifies the conformance requirement (e.g., mandatory or recommended). In some cases, this standard does not specify conformance requirements, but requests that the status of a particular feature be provided.

— Support, which 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, which contains any additional information on the feature. This column is to be filled out by the implementer.

### 10.4.2 General ICS

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

Table 24 shows the general ICS:

**Table 24—IEEE 11073-10404 general ICS**

| Index | Feature | Reference | Req./Status | Support | Comment |
|---|---|---|---|---|---|
| GEN 11073-10404-1 | Implementation Description | — | Identification of the device/application. Description of functionality. | | |
| GEN 11073-10404-2 | Standards Followed and Their Revisions | (standard documents) | (set of existing revisions) | (set of supported revision) | |
| GEN 11073-10404-3 | Nomenclature Document Used and Revision | (standard documents) | (set of existing revisions) | (set of supported revisions) | |

### Table 24 —IEEE 11073-10404 general ICS

| Index | Feature | Reference | Req./Status | Support | Comment |
|---|---|---|---|---|---|
| GEN 11073-10404-4 | Conformance Adherence<br>- Level 1 - | See 10.3.2 | Base conformance declaration that device meets the following IEEE 11073-10404 conformance requirements:<br>A) All mandatory requirements shall be implemented.<br>B) If implemented, conditional, recommended, and optional requirements shall conform to this standard. | Yes/No<br>(No is not expected as no implies that the implementation is NON-conformant) | |
| GEN 11073-10404-5 | Conformance Adherence<br>- Level 2 - | See 10.3.3 | In addition to GEN 11073-10404-4, 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 towards their reference. | Yes/No | |
| GEN 11073-10404-6 | Object Containment Tree | See 6.3 | Provide object containment diagram showing relations between object instances used by the application. A conforming implementation uses only object relations as defined in the DIM. | | |
| GEN 11073-10404-7 | Nomenclature Document Used and Revision | (standard documents) | (set of existing revisions) | (set of supported revision) | |
| GEN 11073-10404-8 | Data Structure Encoding | — | — | description of encoding method(s) for ASN.1 data structures | |
| GEN 11073-10404-9 | Use of Private Objects | — | Does the implementation use objects that are not defined in the DIM? | Yes/No<br>[If yes: explain in Table 25] | |
| GEN 11073-10404-10 | Use of Private Nomenclature Extensions | — | Does the implementation use private extensions to the nomenclature (i.e., 0xF000–0xFFFF codes from ISO/IEEE 11073-10101:2004 [B2])?<br>Private nomenclature extensions are allowed *only* if the standard nomenclature does not include the specific terms required by the application. | Yes/No<br><br>[If yes: explain in the Table 28] | |

**Table 24 —IEEE 11073-10404 general ICS**

| Index | Feature | Reference | Req./Status | Support | Comment |
|---|---|---|---|---|---|
| GEN 11073-10404-11 | IEEE 11073-20601 Conformance | | Provide the conformance report required by IEEE Std 11073-20601-2008. | | |

### 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 25.

**Table 25 —Template for DIM MOC ICS**

| Index | Feature | Reference | Req./Status | Support | Comment |
|---|---|---|---|---|---|
| MOC-*n* | Object description | Reference to the clause in this standard or other location where the object is defined. | Implemented | Specify restrictions, e.g., maximum 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 be simply a unique number (1..*m*).

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

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

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

### 10.4.4 MOC attribute ICS

For each supported object as defined in the DIM MOC ICS, a MOC attribute ICS has to be provided that defines which attributes are used/supported by the implementation, including any inherited attributes. Table 26 is a template only.

**Table 26 —Template for MOC attribute ICS**

| Index | Feature | Reference | Req./Status | Support | Comment |
|---|---|---|---|---|---|
| ATTR-*n*-*x* | Attribute Name.<br>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/<br>C = Conditional/<br>R = Recommended/<br>O = Optional<br>[as per definition in attribute definition tables] | Implemented?<br>Yes/No<br>Static/Dynamic<br>Specify restrictions, e.g., value ranges.<br>Describe how attribute is accessed (e.g., Get, Set, sent in config event report, sent in a data event report)<br>Describe any specific restrictions. | |

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

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

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

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

### 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 27 provides a template for use. One table has to be provided for each object that supports special object notifications.

**Table 27 —Template for MOC notification ICS**

| Index | Feature | Reference | Req./Status | Support | Comment |
|---|---|---|---|---|---|
| NOTI-*n*-*x* | Notification Name and Notification ID | Reference to the clause in this 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 MOC 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 28 provides a template for use. One row of the table is to be used for each nomenclature element.

**Table 28 —Template for MOC nomenclature ICS**

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

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

# Annex A

## (informative)

## Bibliography

[B1] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition, New York, Institute of Electrical and Electronic Engineers, Inc.[5]

[B2] ISO/IEEE 11073-10101:2004, Health informatics—Point-of-care medical device communication—Part 10101: Nomenclature. [6]

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

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

[B5] Draft Guidance for Industry and FDA Staff – Pulse Oximeters – Premarket Notification Submissions [510(k)s], July 19, 2007.

[B6] ISO 9919 (2005), Medical electrical equipment – Particular requirements for the basic safety and essential performance of pulse oximeter equipment for medical use.

[B7] ITU-T Rec. X.680 (Jul. 2002), Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.[7]

[5] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[6] ISO publications are available from the ISO Central Secretariat, 1, ch. de la Voie-Creuse, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[7] ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

# Annex B

# (normative)

# Additional ASN.1 definitions

## B.1 Numeric extensions for thresholds

The threshold extensions to the $SpO_2$ and pulse rate objects require the following three ASN.1 structure definitions:

```
--
-- Alert-Op-State attribute defines the current limit alert state
-- NOTE--The bits refer to the limit alarm only, not to the global alert state of the metric
--
CurLimAlStat ::= BITS-16 {
                    lim-alert-off(0), -- if this bit is set, all alerts (both high and low) are off
                    lim-low-off(1), -- low-limit violation detection is off
                    lim-high-off(2) -- high-limit violation detection is off
}

--
-- Current-Limits attribute
--
CurLimAlVal ::= SEQUENCE {
                    lower FLOAT-Type,
                    upper FLOAT-Type
}

--
-- Alert-Op-Text-String attribute assigns individual labels to upper and lower alarm limit
--
AlertOpTextString ::= SEQUENCE {
                    lower-text OCTET STRING,
                    upper-text OCTET STRING
}
```

## B.2 Pulsatile characteristic bit mapping

The Pulsatile Characteristic object requires the following ASN.1 structure definition:

```
PulseOxPulsChar ::= BITS-16 {
                    pulse-qual-nominal(0),
                    pulse-qual-marginal(1),
                    pulse-qual-minimal(2),
                    pulse-qual-unacceptable(3)
}
```

## B.3 Device and sensor status bit mapping

The threshold extensions to the numeric class require the following four ASN.1 structure definition:

```
PulseOxDevStat ::= BITS-16 {
                    sensor-disconnected(0),
                    sensor-malfunction(1),
                    sensor-displaced(2),
                    sensor-unsupported(3),
                    sensor-off(4),
                    sensor-interference(5),
                    signal-searching(6),
                    signal-pulse-questionable(7),
                    signal-non-pulsatile(8),
                    signal-erratic(9),
                    signal-low-perfusion(10),
                    signal-poor(11),
                    signal-inadequate(12),
                    signal-processing-irregularity(13),
                    device-equipment-malfunction(14),
                    device-extended-update(15)
}
```

# Annex C

# (normative)

# Allocation of identifiers

## C.1 General

This annex contains the nomenclature codes used in this standard and not found in IEEE Std 11073-20601-2008. For codes not contained in this annex, the normative definition is found in IEEE Std 11073-20601-2008.

## C.2 Definitions of terms and codes

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

```
/*******************************************************************************
* From Object Infrastructure (MDC_PART_OBJ)
********************************************************************************/
#define MDC_ATTR_AL_OP_STAT                 2310 /*                          */
#define MDC_ATTR_LIMIT_CURR                 2356 /*                          */
#define MDC_ATTR_AL_OP_TEXT_STRING          2478 /*                          */
#define MDC_ATTR_AL_COND                    2476 /*                          */


/*******************************************************************************
* From Medical supervisory control and data acquisition (MDC_PART_SCADA)
********************************************************************************/
#define MDC_PULS_OXIM_PULS_RATE             18458 /*                         */
#define MDC_SAT_O2_QUAL                     19248 /*                         */
#define MDC_PULS_OXIM_PERF_REL              19376 /*                         */
#define MDC_PULS_OXIM_PLETH                 19380 /*                         */
#define MDC_PULS_OXIM_SAT_O2                19384 /*                         */
#define MDC_PULS_OXIM_PULS_CHAR             19512 /*                         */
#define MDC_PULS_OXIM_DEV_STATUS            19532 /*                         */

#define MDC_TRIG                            53250 /*                         */
#define MDC_TRIG_BEAT                       53251 /*                         */
#define MDC_TRIG_BEAT_MAX_INRUSH            53259 /*                         */
#define MDC_METRIC_NOS                      61439 /*                         */

#define MDC_MODALITY_FAST                   19508 /*                         */
#define MDC_MODALITY_SLOW                   19512 /*                         */
#define MDC_MODALITY_SPOT                   19516 /*                         */
```

## C.3 Systematic derivations of terms and codes

| Systematic name | Common term | Acronym | Description/ Definition | Reference ID | Code |
|---|---|---|---|---|---|
| PatternEvent \| Rhythm \| Artifact \| CVS | Pulse characteristic status | | Object containing status flags representing several characteristics of the pulse event | MDC_PULS_OXIM_PULS_CHAR | 19512 |
| Status \| value \| FunctionalStatus \| Device | Device Status | | Object containing sensor or pulse oximeter-specific status flags | MDC_PULS_OXIM_DEV_STATUS | 19532 |

# Annex D

# (informative)

# Message sequence examples

Figure D.1 shows a sequence diagram of the messaging procedure corresponding to the following use case. The user of a pulse oximetry agent device intends to connect it to a manager device for the first time. The pulse oximeter is capable of performing $SpO_2$ and pulse rate measurements. Thus, it operates as an extended configuration.

a) When the user connects the pulse oximeter, the manager does not yet know the agent’s configuration and sends a response to the agent’s Association Request with the result accepted-unknown-config. See E.2.2.2 and E.2.2.3 for the corresponding PDU examples.

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

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

d) As a next step, the user of the agent device takes a single measurement. The measurement data are transmitted to the manager using a confirmed event report. After having successfully received the measurement data, the manager sends a confirmation to the agent. See E.5.1 and E.5.2 for the corresponding PDU examples.

e) The user ends the measurement session (e.g., by pushing the appropriate button on the device or by simply not using the device for a duration longer than a certain time period). As a consequence, the agent disassociates from the manager by sending an Association Release Request. The manager responds with an Association Release Response. See E.7.1 and E.7.2 for the corresponding PDU examples.

f) When the agent requests to associate to the manager for the next measurement session (e.g., the next day), the manager responds that it has accepted the configuration, as it already knows the agent’s configuration from the previous measurement session. Both devices transition directly to the Operating state.

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

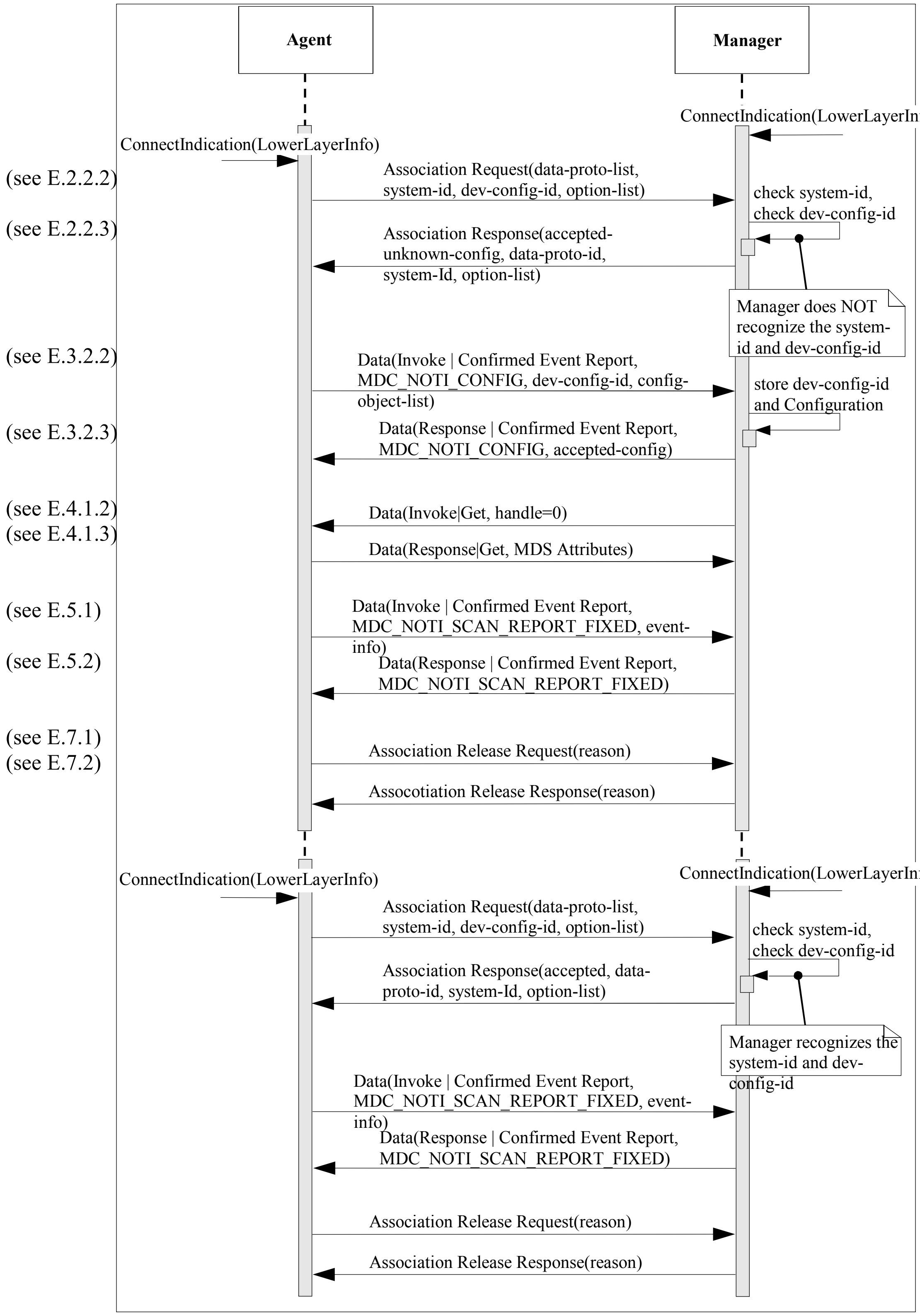
Agent
Manager
ConnectIndication(LowerLayerInfo)
ConnectIndication(LowerLayerInfo)
(see E.2.2.2)
Association Request(data-proto-list, system-id, dev-config-id, option-list)
check system-id, check dev-config-id
(see E.2.2.3)
Association Response(accepted-unknown-config, data-proto-id, system-Id, option-list)
Manager does NOT recognize the system-id and dev-config-id
(see E.3.2.2)
Data(Invoke | Confirmed Event Report, MDC_NOTI_CONFIG, dev-config-id, config-object-list)
store dev-config-id and Configuration
(see E.3.2.3)
Data(Response | Confirmed Event Report, MDC_NOTI_CONFIG, accepted-config)
(see E.4.1.2)
Data(Invoke|Get, handle=0)
(see E.4.1.3)
Data(Response|Get, MDS Attributes)
(see E.5.1)
Data(Invoke | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED, event-info)
(see E.5.2)
Data(Response | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED)
(see E.7.1)
(see E.7.2)
Association Release Request(reason)
Assocotiation Release Response(reason)
ConnectIndication(LowerLayerInfo)
ConnectIndication(LowerLayerInfo)
Association Request(data-proto-list, system-id, dev-config-id, option-list)
check system-id, check dev-config-id
Association Response(accepted, data-proto-id, system-Id, option-list)
Manager recognizes the system-id and dev-config-id
Data(Invoke | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED, event-info)
Data(Response | Confirmed Event Report, MDC_NOTI_SCAN_REPORT_FIXED)
Association Release Request(reason)
Association Release Response(reason)

**Figure D.1—Sequence diagram for pulse oximeter example use case**

# Annex E

(informative)

# PDU examples

## E.1 General

This annex shows binary examples of messages exchanged between a pulse oximeter agent and manager. Three different scenarios containing the association and configuration information exchanges are presented in E.2 and E.3. The first scenario illustrates the case when the agent intends to operate using an extended configuration. The manager does not have the configuration declared by the agent from a prior association. The second scenario illustrates the agent presenting the same extended configuration to the manager, and the manager does have the configuration from the previously transferred configuration exchange. In the third scenario, the agent presents a standard configuration to the manager, and the manager has the configuration because the manager has been preprogrammed with this configuration.

## E.2 Association information exchange

### E.2.1 General

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

### E.2.2 Extended configuration

#### E.2.2.1 General

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

#### E.2.2.2 Association request

The pulse oximetry agent sends the following message to the manager. The agent intends to associate using an extended configuration.

```
0xE2 0x00                APDU CHOICE Type (AarqApdu)
0x00 0x32                CHOICE.length = 50
0x80 0x00 0x00 0x00      assoc-version
0x00 0x01 0x00 0x2A      data-proto-list.count = 1 | length = 42
0x50 0x79                data-proto-id = 20601
0x00 0x26                data-proto-info length = 38
0x80 0x00 0x00 0x00      protocolVersion
0xA0 0x00                encoding rules = MDER or PER
0x80 0x00 0x00 0x00      nomenclatureVersion
0x00 0x00 0x00 0x00      functionalUnits – no test association capabilities
0x00 0x80 0x00 0x00      systemType = sys-type-agent
0x00 0x08                system-id length = 8 and value (manufacturer- and device- specific)
0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x04
0x40 0x00                dev-config-id – extended configuration
0x00 0x01                data-req-mode-flags
```

```
0x01 0x00               data-req-init-agent-count, data-req-init-manager-count
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

### E.2.2.3 Association response

A manager responds to the agent that it can associate but does not have the pulse oximeter extended configuration (i.e., the agent needs to send its configuration).

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

## E.2.3 Previously known extended configuration

### E.2.3.1 General

In this exchange, the agent sends an Association Request intending to use an extended configuration during measurement transfer, and the manager has this configuration from the previously transferred configuration exchange.

### E.2.3.2 Association request

The pulse oximetry agent sends the following message to the manager. The agent intends to associate using an extended configuration.

```
0xE2 0x00               APDU CHOICE Type (AarqApdu)
0x00 0x32               CHOICE.length = 50
0x80 0x00 0x00 0x00     assoc-version
0x00 0x01 0x00 0x2A     data-proto-list.count = 1 | length = 42
0x50 0x79               data-proto-id = 20601
0x00 0x26               data-proto-info length = 38
0x80 0x00 0x00 0x00     protocolVersion
0xA0 0x00               encoding rules = MDER or PER
0x80 0x00 0x00 0x00     nomenclatureVersion
0x00 0x00 0x00 0x00     functionalUnits, no test association capabilities
0x00 0x80 0x00 0x00     systemType = sys-type-agent
0x00 0x08               system-id length = 8 and value (manufacturer- and device- specific)
0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x04
0x40 0x00               dev-config-id – extended configuration
0x00 0x01               data-req-mode-flags
```

```
0x01 0x00               data-req-init-agent-count, data-req-init-manager-count
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

### E.2.3.3 Association response

A manager responds to the agent that it can associate with the agent and that it recognizes, accepts, and has the pulse oximeter’s extended configuration (i.e., the agent does not need to send its configuration).

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

## E.2.4 Standard configuration

### E.2.4.1 General

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

### E.2.4.2 Association request

The pulse oximetry agent sends the following message to the manager. The agent intends to associate using a standard configuration. The agent is willing to enter into a test association as defined in Clause 9.

```
0xE2 0x00               APDU CHOICE Type (AarqApdu)
0x00 0x32               CHOICE.length = 50
0x80 0x00 0x00 0x00     assoc-version
0x00 0x01 0x00 0x2A     data-proto-list.count = 1 | length = 42
0x50 0x79               data-proto-id = 20601
0x00 0x26               data-proto-info length = 38
0x80 0x00 0x00 0x00     protocolVersion
0xA0 0x00               encoding rules = MDER or PER
0x80 0x00 0x00 0x00     nomenclatureVersion
0x40 0x00 0x00 0x00     functionalUnits, has test association capabilities
0x00 0x80 0x00 0x00     systemType = sys-type-agent
0x00 0x08               system-id length = 8 and value (manufacturer- and device- specific)
0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x04
0x01 0x90               dev-config-id – standard configuration
0x00 0x01               data-req-mode-flags
```

```
0x01 0x00               data-req-init-agent-count, data-req-manager-count
0x00 0x00 0x00 0x00     optionList.count = 0 | optionList.length = 0
```

### E.2.4.3 Association response

A manager responds to the agent that it can associate with the agent and that it recognizes, accepts, and has the pulse oximeter standard configuration (i.e., the agent does not need to send its configuration). The manager does not start a test association.

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

## E.3 Configuration information exchange

### E.3.1 General

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

### E.3.2 Extended configuration

#### E.3.2.1 General

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

#### E.3.2.2 Remote operation invoke event report configuration

The pulse oximeter agent sends the description of its extended configuration by sending a Confirmed Event Report message of type MDC_NOTI_CONFIG.

```
0xE7 0x00               APDU CHOICE Type (PrstApdu)
0x00 0xA8               CHOICE.length = 168
0x00 0xA6               OCTET STRING.length = 166
0x12 0x36               invoke-id (differentiates this from other outstanding messages)
0x01 0x01               CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0xA0               CHOICE.length = 160
```

```
0x00 0x00              obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF    event-time = 0xFFFFFFFF
0x0D 0x1C              event-type = MDC_NOTI_CONFIG
0x00 0x96              event-info.length = 150 (start of ConfigReport)
0x40 0x00              config-report-id
0x00 0x03              config-obj-list.count = 3 Measurement objects will be “announced”
0x00 0x90              config-obj-list.length = 144
0x00 0x06              obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01              obj-handle = 1    (→ 1st Measurement is SpO2)
0x00 0x04              attributes.count = 4
0x00 0x24              attributes.length = 36
0x09 0x2F              attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04              attribute-value.length = 4
0x00 0x02 0x4B 0xB8    MDC_PART_SCADA | MDC_PULS_OXIM_SAT_O2
0x0A 0x46              attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02              attribute-value.length = 2
0x40 0xC0              avail-stored-data, acc-manager-init, acc-agent-init, measured
0x09 0x96              attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02              attribute-value.length = 2
0x02 0x20              MDC_DIM_PERCENT
0x0A 0x55              attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C              attribute-value.length = 12
0x00 0x02              AttrValMap.count = 2
0x00 0x08              AttrValMap.length = 8
0x0A 0x4C 0x00 0x02    MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x09 0x90 0x00 0x08    MDC_ATTR_TIME_STAMP_ABS | value length = 8
0x00 0x06              obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x0A              obj-handle = 10   (→2nd Measurement is pulse rate)
0x00 0x04              attributes.count = 4
0x00 0x24              attributes.length = 36
0x09 0x2F              attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04              attribute-value.length = 4
0x00 0x02 0x48 0x1A    MDC_PART_SCADA | MDC_PULS_OXIM_PULS_RATE
0x0A 0x46              attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02              attribute-value.length = 2
0x40 0xC0              avail-stored-data, acc-manager-init, acc-agent-init, measured
0x09 0x96              attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02              attribute-value.length = 2
0x0A 0xA0              MDC_DIM_BEAT_PER_MIN
0x0A 0x55              attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C              attribute-value.length = 12
0x00 0x02              AttrValMap.count = 2
0x00 0x08              AttrValMap.length = 8
0x0A 0x4C 0x00 0x02    MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x09 0x90 0x00 0x08    MDC_ATTR_TIME_STAMP_ABS, 8
0x00 0x06              obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x03              obj-handle = 3    (→3rd Measurement is SpO2 – fast response)
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 0x4B 0xB8    MDC_PART_SCADA | MDC_PULS_OXIM_SAT_O2
```

```
0x0A 0x46               attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02               attribute-value.length = 2
0x40 0xC0               avail-stored-data, acc-manager-init, acc-agent-init, measured
0x09 0x96               attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02               attribute-value.length = 2
0x02 0x20               MDC_DIM_PERCENT
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 0x02 0x4C 0x34      MDC_PART_SCADA | MDC_MODALITY_FAST
0x0A 0x55               attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C               attribute-value.length = 12
0x00 0x02               AttrValMap.count = 2
0x00 0x08               AttrValMap.length = 8
0x0A 0x4C 0x00 0x02     MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x09 0x90 0x00 0x08     MDC_ATTR_TIME_STAMP_ABS, 8
```

#### E.3.2.3 Remote operation response event report configuration

The manager responds that it can utilize the agent's configuration. The manager does by sending the Confirmed Event Report response with a config-result of accepted-config.

```
0xE7 0x00               APDU CHOICE Type (PrstApdu)
0x00 0x16               CHOICE.length = 22
0x00 0x14               OCTET STRING.length = 20
0x12 0x36               invoke-id (mirrored from invocation)
0x02 0x01               CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E               CHOICE.length = 14
0x00 0x00               obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00     currentTime = 0
0x0D 0x1C               event-type = MDC_NOTI_CONFIG
0x00 0x04               event-reply-info.length = 4
0x40 0x00               ConfigReportRsp.config-report-id = 0x4000
0x00 0x00               ConfigReportRsp.config-result = accepted-config
```

### E.3.3 Known configuration

#### E.3.3.1 General

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

#### E.3.3.2 Remote operation invoke event report configuration

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

#### E.3.3.3 Remote operation response event report configuration

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

### E.3.4 Standard configuration

#### E.3.4.1 General

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

#### E.3.4.2 Remote operation invoke event report configuration

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

#### E.3.4.3 Remote operation response event report configuration

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

## E.4 GET MDS attributes service

#### E.4.1.1 General

The GET MDS attributes is invoked at any time when an agent is in the Associated state.

#### E.4.1.2 Get all medical device system attributes request

The manager queries the agent for its MDS object attributes.

```
0xE7 0x00                 APDU CHOICE Type (PrstApdu)
0x00 0x0E                 CHOICE.length = 14
0x00 0x0C                 OCTET STRING.length = 12
0x12 0x37                 invoke-id (differentiates this from other outstanding messages)
0x01 0x03                 CHOICE (Remote Operation Invoke | Get)
0x00 0x06                 CHOICE.length = 6
0x00 0x00                 handle = 0 (MDS object)
0x00 0x00                 attribute-id-list.count = 0 (all attributes)
0x00 0x00                 attribute-id-list.length = 0
```

#### E.4.1.3 Get response with all MDS attributes

The pulse oximetry agent responds to the manager with its attributes. Further, some optional fields are communicated as well.

```
0xE7 0x00                 APDU CHOICE Type (PrstApdu)
0x00 0x6E                 CHOICE.length = 110
0x00 0x6C                 OCTET STRING.length = 108
0x12 0x37                 invoke-id (mirrored from request)
0x02 0x03                 CHOICE (Remote Operation Response | Get)
```

```
0x00 0x66                CHOICE.length = 102
0x00 0x00                handle = 0 (MDS object)
0x00 0x06                attribute-list.count = 6
0x00 0x60                attribute-list.length = 96
0x0A 0x5A                attribute-id = MDC_ATTR_SYS_TYPE_SPEC_LIST
0x00 0x08                attribute-value.length = 8
0x00 0x01                TypeVerList count = 1
0x00 0x04                TypeVerList length = 4
0x10 0x04                type = MDC_DEV_SPEC_PROFILE_PULS_OXIM
0x00 0x01                version = version 1 of the specialization
0x09 0x28                attribute-id = MDC_ATTR_ID_MODEL
0x00 0x1A                attribute-value.length = 26
0x00 0x0A 0x54 0x68      string length = 10 | “TheCompany”
0x65 0x43 0x6F 0x6D
0x70 0x61 0x6E 0x79
0x00 0x0C 0x4F 0x78      string length = 12 | “OximeterABC\0”
0x69 0x6D 0x65 0x74
0x65 0x72 0x41 0x42 0x43 0x00
0x09 0x84                attribute-id = MDC_ATTR_SYS_ID
0x00 0x0A                attribute-value.length = 10
0x00 0x08 0x11 0x22 0x33 0x44
0x55 0x66 0x77 0x04      octet string length = 8 | EUI-64
0x0A 0x44                attribute-id = MDC_ATTR_DEV_CONFIG_ID
0x00 0x02                attribute-value.length = 2
0x40 0x00                dev-config-id = 16384 (extended-config-start)
0x09 0x2D                attribute-id = MDC_ATTR_ID_PROD_SPECN
0x00 0x12                attribute-value.length = 18
0x00 0x01                            ProductionSpec.count = 1
0x00 0x0E                            ProductionSpec.length = 14
0x00 0x01                            ProdSpecEntry.spec-type = 1 (serial-number)
0x00 0x00                            ProdSpecEntry.component-id = 0
0x00 0x08 0x44 0x45      string length = 8 | prodSpecEntry.prod-spec = “DE124567”
0x31 0x32 0x34 0x35
0x36 0x37
0x09 0x87                attribute-id =MDC_ATTR_TIME_ABS
0x00 0x08                attribute-value.length = 8
0x20 0x07 0x02 0x01      Absolute-Time-Stamp = 2007-02-01T12:05:0000
0x12 0x05 0x00 0x00
```

## E.5 Data reporting

### E.5.1 Confirmed measurement data transmission

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

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x36                CHOICE.length = 54
0x00 0x34                OCTET STRING.length = 52
0x12 0x38                invoke-id (differentiates this from other outstanding messages)
0x01 0x01                CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x2E                CHOICE.length = 46
0x00 0x00                obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF      event-time = 0xFFFFFFFF
```

```
0x0D 0x1D              event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x24              event-info.length = 36
0xF0 0x00              ScanReportInfoFixed.data-req-id = 0xF000
0x00 0x00              ScanReportInfoFixed.scan-report-no = 0
0x00 0x02              ScanReportInfoFixed.obs-scan-fixed.count = 2
0x00 0x1C              ScanReportInfoFixed.obs-scan-fixed.length = 28
0x00 0x01              ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00 0x0A              ScanReportInfoFixed.obs-scan-fixed.value[0]. obs-val-data.length = 10
0x00 0x62              Basic-Nu-Observed-Value = 98 (%)
0x20 0x07 0x12 0x06    Absolute-Time-Stamp = 2007-12-06T12:10:0000
0x12 0x10 0x00 0x00
0x00 0x0A              ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 10
0x00 0x0A              ScanReportInfoFixed.obs-scan-fixed.value[1]. obs-val-data.length = 10
0x00 0x48              Simple-Nu-Observed-Value = 72 (beats/min)
0x20 0x07 0x12 0x06    Absolute-Time-Stamp = 2007-12-06T12:10:0000
0x12 0x10 0x00 0x00
```

### E.5.2 Response to confirmed measurement data transmission

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

```
0xE7 0x00              APDU CHOICE Type (PrstApdu)
0x00 0x12              CHOICE.length = 18
0x00 0x10              OCTET STRING.length = 16
0x12 0x38              invoke-id (mirrored from invocation)
0x02 0x01              CHOICE(Remote Operation Response | Confirmed Event Report)
0x00 0x0A              CHOICE.length = 10
0x00 0x00              obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00    currentTime = 0
0x0D 0x1D              event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x00              event-reply-info.length = 0
```

## E.6 Scanner example

### E.6.1 General

The following example illustrates how a periodic configurable scanner object can be described and transmitted.

For brevity, the assumption is made that the scanner object with a handle of 40 observes and reports the information from four objects:

- An SpO2 object, with a handle of 1, which reports its $SpO_2$ in the BasicNuObsVal form, followed by the current threshold status and ending with the absolute time.
- A Pulse Rate object, with a handle of 11, which reports its pulse rate in the BasicNuObsVal form.
- An SpO2 object, with a handle of 2, which has previously been described as having the fast-response modality, and reports its $SpO_2$ in the BasicNuObsVal form.
- A Pulse Rate object, with a handle of 12, which has previously been described as having the fast-response modality, and reports its pulse rate in the BasicNuObsVal form.

The whole Configuration Report Request will not be presented, only the part that pertains to the scanner object description.

### E.6.2 Portion of configuration report describing a periodic configurable scanner

```
0x0D 0x1C                event-type = MDC_NOTI_CONFIG
0x00 0x94                event-info.length = 148 (for example)
0x40 0x00                config-report-id
0x00 0x0n                config-obj-list.count = n Measurement objects will be "announced"
0x00 0x8E                config-obj-list.length = 142 (for example)
...
0x00 0x13                obj-class = MDC_MOC_SCAN_PERI
0x00 0x28                obj-handle = 40   (→ measurement i is a Periodic Scanner)
0x00 0x04                attributes.count = 4
0x00 0x50                attributes.length = 80
0x09 0x53                attribute-id = MDC_ATTR_OP_STAT
0x00 0x02                attribute-value.length = 2
0x00 0x00                disabled
0x0A 0x54                attribute-id = MDC_ATTR_SCAN_REP_PD_MIN
0x00 0x04                attribute-value.length = 4
0x00 0x00 0x0A 0x6B      RelativeTime of 333.33 (333.375 exact) msec
0x09 0x13                attribute-id = MDC_ATTR_CONFIRM_MODE
0x00 0x02                attribute-value.length = 2
0x00 0x00                unconfirmed mode
0x0A 0x53                attribute-id = MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP
0x00 0x36                attribute-value.length = 54
0x00 0x04                HandleAttrValMap.count = 4
0x00 0x32                HandleAttrValMap.length = 50
0x00 0x01                HandleAttrValMap 1: SpO2
0x00 0x03                HandleAttrValMap.count = 3
0x00 0x10                HandleAttrValMap.length = 16
0x0A 0x4C 0x00 0x02      MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x09 0xAC 0x00 0x02      MDC_ATTR_AL_COND,2
0x09 0x47 0x00 0x02      MDC_ATTR_MSMT_STAT,2

0x09 0x90 0x00 0x08      MDC_ATTR_TIME_STAMP_ABS, 8
0x00 0x0B                HandleAttrValMap 2: Pulse Rate
0x00 0x01                HandleAttrValMap.count = 1
0x00 0x04                HandleAttrValMap.length = 4
0x0A 0x4C 0x00 0x02      MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x00 0x02                HandleAttrValMap 3: SpO2 – fast-response
0x00 0x01                HandleAttrValMap.count = 1
0x00 0x04                HandleAttrValMap.length = 4
0x0A 0x4C 0x00 0x02      MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x00 0x0C                HandleAttrValMap 4: Pulse Rate – fast-response
0x00 0x01                HandleAttrValMap.count = 1
0x00 0x04                HandleAttrValMap.length = 4
0x0A 0x4C 0x00 0x02      MDC_ATTR_NU_VAL_OBS_BASIC, 2
```

### E.6.3 Measurement data transfer of a periodic configurable scanner

```
0xE7 0x00                APDU CHOICE Type (PrstApdu)
0x00 0x2C                CHOICE.length = 44
0x00 0x2A                OCTET STRING.length = 42
0x12 0x39                invoke-id (differentiates this from other outstanding messages)
```

```
0x01 0x00                CHOICE(Remote Operation Invoke | Unconfirmed Event Report)
0x00 0x24                CHOICE.length = 36
0x00 0x28                obj-handle = 40 (scanner object)
0xFF 0xFF 0xFF 0xFF      event-time = 0xFFFFFFFF
0x0D 0x2A                event-type = MDC_NOTI_BUF_SCAN_REPORT_GROUPED
0x00 0x1A                event.length = 26
0xF0 0x00                data-request-id = 0xF000 (agent-initiated)
0x00 0x01                scan-report-no = 1
0x00 0x14                grouped-length = 20
0x00 0x62                Handle 1: SpO2 98%
0x00 0x00                Handle 1: SpO2 limit status : within thresholds
0x00 0x00                Handle 1: Measurement status
0x20 0x08 0x07 0x03      Absolute-Time-Stamp = 2008-07-03T12:10:0000
0x12 0x10 0x00 0x00
0x00 0x49                Handle 2: Pulse rate = 73 beats per minute
0x00 0x62                Handle 11: SpO2 fast-response = 98%
0x00 0x48                Handle 12: Pulse rate fast-response = 72 beats per minute
```

## E.7 Disassociation

### E.7.1 Association release request

The pulse oximetry agent sends the following message to the manager:

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

### E.7.2 Association release response

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

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

## IEEE Notice to Users

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to: Secretary, IEEE-SA Standards Board, 445 Hoes Lane, Piscataway, NJ 08854, USA.

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

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

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

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

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

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

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

**Participants:** The list of IEEE participants can be accessed at the following URL:
http://standards.ieee.org/downloads/11073/11073-10415/11073-10415-2010_wg_participants.pdf.

***IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.***

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

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

**Keywords:** medical device communication, personal health devices, pulse oximeter

**ICS 35.240.80**
**ISBN 978-0-7381-5879-2**

Price based on 69 pages