INTERNATIONAL STANDARD **ISO/IEEE 11073-10415**

First edition
2010-05-01

Health informatics — Personal health device communication —

Part 10415: Device specialization — Weighing scale

Informatique de santé — Communication entre dispositifs de santé personnels —

Partie 10415: Spécialisation des dispositifs — Plateau de balance

Reference number
ISO/IEEE 11073-10415:2010(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.

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

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 11073-10415 was prepared by the 11073 Committee of the Engineering in Medicine and Biology Society of the IEEE (as IEEE Std 11073-10415-2008). 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 document uses the optimized framework created in IEEE Std 11073-20601[a] and describes a specific, interoperable communication approach for weighing scales. 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 10415: Device specialization — Weighing scale

IMPORTANT NOTICE: This standard is not intended to assure 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.

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

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, and set top boxes). Interoperability is the key to growing the potential market for these devices and to enabling people to be better informed participants in the management of their health.

1.3 Context

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

This document, IEEE Std 11073-10415, defines the device specialization for the weighing scale, being a specific agent type, and it 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, which in turn draws information from both ISO/IEEE 11073-10201:2004 [B4][1] and ISO/IEEE 11073-20101:2004 [B5]. The medical device encoding rules (MDER) used within this standard are fully described in IEEE Std 11073-20601.

This standard reproduces relevant portions of the nomenclature found in ISO/IEEE 11073-10101:2004 [B3] and adds new nomenclature codes for the purposes of this standard. Between this standard and IEEE Std 11073-20601, all required nomenclature codes for implementation are documented.

NOTE— In this standard, IEEE Std 11073-104zz is used to refer to the collection of device specialization standards that utilize IEEE Std 11073-20601, where zz can be any number from 01 to 99, inclusive.[2]

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 Protocol.[3, 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* [B2] 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, it 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 term used to refer to a physical apparatus implementing either an agent or a 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]The numbers in brackets correspond to those of the bibliography in Annex A.
[2] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.
[3]The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.
[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 agent systems. Some examples of managers include a cellular phone, health appliance, set top box, or a computer system.

3.1.7. mass: An intrinsic property of matter that can be measured using the effect of the gravitational field on an object.

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

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

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

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

3.1.12. weight: The force that results from the exertion of gravity on an object. The weight is directly proportional to the mass of the object**.** However, in the health care domain the term body weight is typically used to denote the body mass of a person. This notation applies also to this standard.

3.2 Acronyms and abbreviations

APDU	application protocol data unit
ASN.1	Abstract Syntax Notation One
BMI	body mass index
DIM	domain information model
EUI-64	extended unique identifier (64 bits)
ICS	implementation conformance statement
MDC	medical device communication
MDER	medical device encoding rules
MDS	medical device system
MOC	managed object class
PHD	personal health device
RT-SA	real-time sample array
VMO	virtual medical object
VMS	virtual medical system

4. Introduction to ISO/IEEE 11073 personal health devices

4.1 General

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

IEEE Std 11073-20601 supports the modeling and implementation of an extensive set of personal health devices. This standard defines aspects of the weighing scale device. It describes all aspects necessary to implement the application layer services and data exchange protocol between an ISO/IEEE 11073 PHD weighing scale agent and a manager. This standard defines a subset of the objects and functionality contained in IEEE Std 11073-20601 and extends and adds definitions where appropriate. All new definitions are given in Annex B in Abstract Syntax Notation One (ASN.1) [B6]. Nomenclature codes referenced in this standard, which are not defined in ISO/IEEE 11073-20601, 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, 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 for a detailed description of the modeling constructs.

4.2.2 Domain information model

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

4.2.3 Service model

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

4.2.4 Communication model

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

4.2.5 Implementing the models

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

5. Weighing scale device concepts and modalities

5.1 General

This clause presents the general concepts of weighing scale devices. In the context of personal health devices in this family of standards, a weighing scale is a device that measures the body weight of a person and, optionally, determines other physiological quantities (e.g., the body mass index or the height of a person). Weighing scale devices considered in this standard are typically placed on the floor with a person stepping on the device to perform a weight measurement, with the result being converted into mass internally of the device.

In the personal health context, the body weight of a person is typically not measured more frequently than twice a day.

Weighing scale devices may use a variety of techniques for measuring body weight. One typical method is to place several strain-gauge load cells under the measurement plane to convert deformation into weight.

5.2 Body weight

The primary data type of a weighing scale device is body weight. It has measurement units of kilograms (kg) or pounds (lb).

5.3 Body height

If body mass index reporting is supported, then body height is required. Body height denotes the actual height of the person using a weighing scale device. It has measurement units of centimeters (cm) or inches (in). This observation is typically entered manually.

5.4 Body mass index

The body mass index (BMI) is a measure for indicating an overweight or underweight condition of a person and is defined as the individual's body weight, in kilograms, divided by the square of height, in meters (see Garrow and Webster [B1]):

$$\text{BMI} = \frac{\text{body weight [kg]}}{\text{body height squared [m}^2\text{]}}$$

BMI is not measured directly but is derived from body weight and body height. In the case where pounds and inches are used as measurement units instead of kilograms and meters, the BMI may be calculated as follows:

$$\text{BMI} = 703 \times \frac{\text{body weight [lb]}}{\text{body height squared [in}^2\text{]}}$$

Using the value 703 as a conversion factor gives a relative error with respect to using kilograms and meters of less than 0.01%.

6. Weighing scale domain information model

6.1 Overview

This clause describes the domain information model of the weighing scale.

6.2 Class extensions

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

6.3 Object instance diagram

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

The objects of the DIM, as shown in Figure 1, are described in 6.4 to 6.12. This includes the 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), the enumeration objects (see 6.8), the PM-store objects (see 6.9), and the scanner objects (see 6.10). See 6.11 for rules for extending the weighing scale information model beyond elements as described in this standard. Each clause that describes an object of the weighing scale contains the following information:

— The nomenclature code used to identify the class of the object. One example of where this code is used is the configuration event, where the object class is reported for each object. This allows the manager to determine whether the class of the object being specified is a numeric, real-time sample array, enumeration, scanner, or PM-store class.

— The attributes of the object. Each object has attributes that represent and convey information on the physical device and its data sources. Each object has a Handle attribute that identifies the object instance within an agent. Attribute values are accessed and modified using methods such as GET and SET. Attributes 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.

— 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 is referenced, then it contains the conditions), R — Attribute is Recommended, NR — Attribute is Not Recommended, and O — Attribute is Optional. Mandatory attributes shall be implemented by the agent. Conditional attributes shall be implemented if the condition applies and may be implemented otherwise. Recommended attributes should be implemented by the agent. Not recommended attributes should not be implemented by the agent. Optional attributes may be implemented 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.

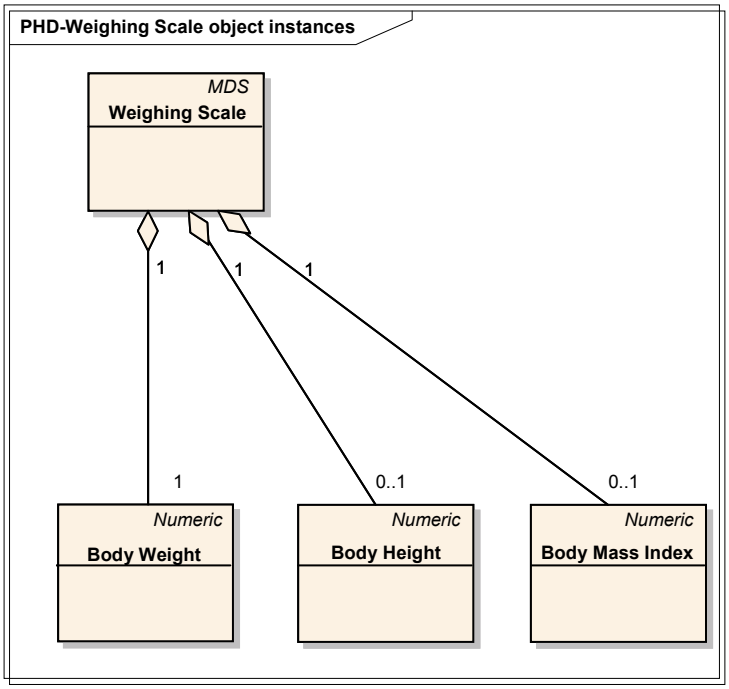

Figure 1—Weighing scale—domain information model

6.4 Types of configuration

6.4.1 General

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

6.4.2 Standard configuration

Standard configurations are defined in the IEEE 11073-104zz specializations (such as this standard) and are assigned a well-known identifier (Dev-Configuration-Id). The usage of a standard configuration is negotiated at association time between the agent and the manager. If the manager acknowledges that it recognizes and wants to operate using the configuration, then the agent can send 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 the objects, attributes, and values that are used in a configuration and assigns a configuration identifier. When the agent associates with a manager, it negotiates an acceptable configuration. Typically, the manager does not recognize the agent's configuration on the first connection, so the manager responds that the agent must send its configuration information as a configuration event report. If, however, the manager already recognizes the configuration, either because it was preloaded in some way or the agent had previously associated with the manager, then the manager responds that the configuration is known and no further configuration information needs to be sent.

6.5 Medical device system object

6.5.1 MDS object attributes

Table 1 summarizes the attributes of the weighing scale 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.	C
System-Model	{"Manufacturer","Model"}.	M
System-Id	Extended unique identifier (64 bits) (EUI-64).	M
Dev-Configuration-Id	Standard config: 0x05DC (1500). Extended configs: 0x4000–0x7FFF.	M
Attribute-Value-Map	See IEEE Std 11073-20601.	C
Production-Specification	See IEEE Std 11073-20601.	O
Mds-Time-Info	See IEEE Std 11073-20601.	C
Date-and-Time	See IEEE Std 11073-20601.	C
Relative-Time	See IEEE Std 11073-20601.	C
HiRes-Relative-Time	See IEEE Std 11073-20601.	C
Date-and-Time-Adjustment	See IEEE Std 11073-20601.	C
Power-Status	*onBattery* or *onMains.*	R
Battery-Level	See IEEE Std 11073-20601.	R
Remaining-Battery-Time	See IEEE Std 11073-20601.	R
Reg-Cert-Data-List	See IEEE Std 11073-20601.	O
System-Type-Spec-List	{MDC_DEV_SPEC_PROFILE_SCALE, 1}.	M
Confirm-Timeout	See IEEE Std 11073-20601.	O

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

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

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

The Dev-Configuration-Id attribute holds a locally unique 16-bit identifier that identifies the device configuration. For a weighing scale agent with extended configuration, this identifier is chosen in the range of extended-config-start to extended-config-end (see IEEE Std 11073-20601) 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) 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); the Parameters (action-info-args) column defines the associated ASN.1 data structure (see IEEE Std 11073-20601 for ASN.1 definitions) to use in the action message for the action-info-args field of the request; and the Results (action-info-args) column defines the structure to use in the action-info-args of the response.

Table 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	—

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

Agents following only this device specialization and no others shall send event reports (see 6.5.3) using agent-initiated measurement data transmission. Agents following this device specialization as well as others shall send event reports in the appropriate fashion. During the association procedure (see 8.3), DataReqModeCapab shall be set to the appropriate value for the event report style. Implementation of the MDS-Data-Request method/action is not required in this standard and is not shown in Table 2.

6.5.3 MDS object events

Table 3 defines the events that can be sent by the weighing scale MDS object.

Table 3—Weighing scale MDS object events

Service	Subservice type name	Mode	Subservice type (event-type)	Parameters (event-info)	Results (event-reply-info)
EVENT REPORT	MDS-Configuration-Event	Confirmed	MDC_NOTI_CONFIG	ConfigReport	ConfigReportRsp
	MDS-Dynamic-Data-Update-Var	Confirmed	MDC_NOTI_SCAN_REPORT_VAR	ScanReportInfoVar	—
	MDS-Dynamic-Data-Update-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_FIXED	ScanReportInfoFixed	—
	MDS-Dynamic-Data-Update-MP-Var	Confirmed	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	—
	MDS-Dynamic-Data-Update-MP-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	—

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

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

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

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

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

NOTE— IEEE Std 11073-20601 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 weighing scale 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 weighing scale 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 weighing scale agent; in which case, the manager shall send the “Remote Operation Invoke | Get” message (see roiv-cmip-get in IEEE Std 11073-20601) with the reserved MDS handle value of 0. The weighing scale agent shall report its MDS object attributes to the manager using the “Remote Operation Response | Get” message (see rors-cmip-get in IEEE Std 11073-20601). See Table 4 for a summary of the GET service including some message fields.

Table 4—Weighing scale MDS object GET service

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

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

6.5.4.2 SET service

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

6.6 Numeric objects

6.6.1 General

The weighing scale DIM (see Figure 1) contains one required numeric object for body weight and two optional numeric objects for body height and body mass index. These are described in 6.6.2 to 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 Body weight

Table 5 summarizes the attributes of the body weight numeric object. The nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. The body weight numeric object shall be supported by a weighing scale agent.

Table 5—Body weight numeric object attributes

Attribute name	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x05DC)	
	Value	**Qual.**	**Value**	**Qual.**
Handle	See IEEE Std 11073-20601.	M	1	M
Type	MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL.	M	MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL.	M
Supplemental-Types	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated.	M	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated.	M
Metric-Structure-Small	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Measurement-Status	See IEEE Std 11073-20601.	R	Attribute not initially present. If present follow IEEE Std 11073-20601.	O
Metric-Id	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Metric-Id-List	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Metric-Id-Partition	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Unit-Code	MDC_DIM_KILO_G or MDC_DIM_LB.	M	MDC_DIM_KILO_G.	M
Attribute-Value-Map	See IEEE Std 11073-20601.	C	MDC_ATTR_NU_VAL_OBS_SIMP, then MDC_ATTR_TIME_STAMP_ABS.	M
Source-Handle-Reference	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Label-String	See IEEE Std 11073-20601.	O	Attribute not initially present. If present follow IEEE Std 11073-20601.	O
Unit-LabelString	See IEEE Std 11073-20601.	O	Attribute not initially present. If present follow IEEE Std 11073-20601.	O

	Extended configuration		Standard configuration (Dev-Configuration-Id = 0x05DC)	
Absolute-Time-Stamp	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601apply.	C
Relative-Time-Stamp	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
HiRes-Time-Stamp	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
Measure-Active-Period	See IEEE Std 11073-20601.	NR	Attribute not initially present. If present follow IEEE Std 11073-20601.	NR
Simple-Nu-Observed-Value	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601. If fixed format is used and the standard configuration is not adjusted, this attribute is mandatory; otherwise, the conditions from IEEE Std 11073-20601 apply.	C
Compound-Simple-Nu-Observed-Value	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
Basic-Nu-Observed-Value	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
Compound-Basic-Nu-Observed-Value	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
Nu-Observed-Value	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
Compound-Nu-Observed-Value	See IEEE Std 11073-20601.	C	Attribute not initially present. If present follow IEEE Std 11073-20601.	C
Accuracy	See IEEE Std 11073-20601.	R	Attribute not initially present. If present follow IEEE Std 11073-20601.	R

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

For a weighing scale agent with standard configuration, the AttrValMap structure (see IEEE Std 11073-20601) of the Attribute-Value-Map attribute shall contain the attribute ID and attribute length information of the Simple-Nu-Observed-Value and Absolute-Time-Stamp attribute in the same order as indicated in Table 5.

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

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

6.6.3 Body height

Table 6 summarizes the attributes of the body height numeric object. The nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. The body height numeric object may be supported by a weighing scale agent with extended configuration. If the body mass index numeric object is supported, the body height numeric object shall be present. It shall not be present in the standard configuration.

Table 6—Body height numeric object attributes

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

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

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

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

6.6.4 Body mass index

Table 7 summarizes the attributes of the body mass index numeric object. The nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. The body mass index numeric object may be supported by a weighing scale agent with extended configuration and shall not be present in the standard configuration.

Table 7—Body mass index numeric object attributes

Attribute	Extended configuration	
Name	**Value**	**Qual.**
Handle	See IEEE Std 11073-20601.	M
Type	MDC_PART_SCADA | MDC_RATIO_MASS_BODY_LEN_SQ.	M
Supplemental-Types	See IEEE Std 11073-20601.	NR
Metric-Spec-Small	mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-calculation.	M
Metric-Structure-Small	See IEEE Std 11073-20601.	NR
Measurement-Status	See IEEE Std 11073-20601.	R
Metric-Id	See IEEE Std 11073-20601.	NR
Metric-Id-List	See IEEE Std 11073-20601.	NR
Metric-Id-Partition	See IEEE Std 11073-20601.	NR
Unit-Code	MDC_DIM_KG_PER_M_SQ.	M
Attribute-Value-Map	See IEEE Std 11073-20601.	C
Source-Handle-Reference	The value of this attribute shall be set to the Handle value of the associated body weight object.	M
Label-String	See IEEE Std 11073-20601.	O
Unit-LabelString	See IEEE Std 11073-20601.	O
Absolute-Time-Stamp	See IEEE Std 11073-20601.	C
Relative-Time-Stamp	See IEEE Std 11073-20601.	C
HiRes-Time-Stamp	See IEEE Std 11073-20601.	C
Measure-Active-Period	See IEEE Std 11073-20601.	NR
Simple-Nu-Observed-Value	See IEEE Std 11073-20601.	C
Compound-Simple-Nu-Observed-Value	See IEEE Std 11073-20601.	C
Basic-Nu-Observed-Value	See IEEE Std 11073-20601.	C
Compound-Basic-Nu-Observed-Value	See IEEE Std 11073-20601.	C
Nu-Observed-Value	See IEEE Std 11073-20601.	C
Compound-Nu-Observed-Value	See IEEE Std 11073-20601.	C
Accuracy	See IEEE Std 11073-20601.	R

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

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

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

6.7 Real-time sample array objects

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

6.8 Enumeration objects

Enumeration objects are not required by this standard.

6.9 PM-store objects

PM-store objects are not required by this standard.

6.10 Scanner objects

Scanner objects are not required by this standard.

6.11 Class extension objects

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

6.12 Weighing scale information model extensibility rules

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

A weighing scale 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).

7. Weighing scale service model

7.1 General

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

7.2 Object access services

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

— GET service: used by the manager to retrieve the values of the agent MDS object attributes. The list of weighing scale 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. There are no settable attributes defined for a weighing scale agent according to this standard.

— Event report service: used by the agent to send configuration reports and measurement data to the manager. The list of event reports for the weighing scale device specialization is given in 6.5.3.

— Action service: used by the manager to invoke actions (or methods) supported by the agent. An example is Set-Time action, which is used to set a real-time clock with the absolute time at the agent.

Table 8 summarizes the object access services described in this standard.

Table 8—Weighing scale 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.
EVENT REPORT	MDS-Configuration-Event	Confirmed	MDC_NOTI_CONFIG	ConfigReport	ConfigReportRsp	Configuration Report to inform manager of the configuration of the agent.
	MDS-Scan-Report-Var	Confirmed	MDC_NOTI_SCAN_REPORT_VAR	ScanReportInfoVar	—	Data Report to provide dynamic data to manager for some or all of the agent's objects in variable format.
	MDS-Scan-Report-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_FIXED	ScanReportInfoFixed	—	Data Report to provide dynamic data to manager for some or all of the agent's objects in fixed format.
	MDS-Scan-Report MP-Var	Confirmed	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	—	This is the same as MDS-Dynamic-Data-Update-Var but allows inclusion of data from multiple people.
	MDS-Scan-Report-MP-Fixed	Confirmed	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	—	This is the same as MDS-Dynamic-Data-Update-Fixed but allows inclusion of data from multiple people.
ACTION	Set-Time	Confirmed	MDC_ACT_SET_TIME	SetTimeInvoke	—	Manager method to invoke the agent to set time to requested value.

7.3 Object access event report services

The event report service (see Table 8) 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.

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

— Event reports shall be used in confirmed mode.

— Agent-initiated mode shall be supported for measurement data transmission.

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

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

8. Weighing scale communication model

8.1 Overview

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

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

8.2 Communications characteristics

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

For a weighing scale agent implementing no other device specialization except this standard, the maximum size of an APDU sent shall be not larger than N_{tx}. For this standard, it is N_{tx} = 896 octets. An agent according to this definition shall be capable of receiving an APDU up to the size of at least N_{rx}. For this standard, it is N_{rx} = 224 octets.

For a weighing scale agent implementing multiple functions according to multiple device specializations, the maximum size of an APDU sent shall not be larger than $N_{tx,i}$, if this agent implements another device specialization i with $N_{tx,i} > N_{tx}$. Otherwise, the maximum size of an APDU sent shall not be larger than N_{tx}. An agent according to this definition shall be capable of receiving an APDU up to the size of at least $N_{rx,i}$ octets if this agent implements another device specialization i with $N_{rx,i} > N_{tx}$. Otherwise, the agent shall be capable of receiving an APDU up to the size of at least N_{rx}.

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

8.3 Association procedure

8.3.1 General

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

8.3.2 Agent procedure—association request

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

- The version of the association procedure used by the agent shall be set to assoc-version1 (i.e., *assoc-version* = 0x80000000).
- The DataProtoList structure element of the data protocol identifier shall be set to data-proto-id-20601 (i.e., *data-proto-id* = 0x5079).
- The *data-proto-info* field shall contain a PhdAssociationInformation structure that shall contain the following parameter values:
    1) The version of the data exchange protocol shall be set to protocol-version1 (i.e., *protocol-version* = 0x80000000).
    2) At least the MDER shall be supported (i.e., *encoding-rules* = 0x8000).
    3) The version of the nomenclature used shall be set to nom-version1 (i.e., *nomenclature-version* = 0x80000000).
    4) The field *functional-units* may have the test association bits set but shall not have any other bits set.
    5) The field *system-type* shall be set to sys-type-agent (i.e., *system-type* = 0x00800000).
    6) The *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 weighing scale with which it is associating and, optionally, to implement a simple access restriction policy.
    7) The *dev-config-id* field shall be set to the value of the Dev-Configuration-Id attribute of the MDS object of the agent.
    8) If the agent supports only the weighing scale specialization, then the field indicating the data request modes (*data-req-mode-capab*) supported by the weighing scale agent shall be set to *data-req-supp-init-agent*.
    9) If the agent supports only the weighing scale specialization, then the *data-req-init-manager-count* field shall be set to 0 and the *data-req-init-agent-count* field shall be set to 1.

8.3.3 Manager procedure—association response

In the association response message sent by the manager:

- The *result* field shall be set to an appropriate response from those defined in IEEE Std 11073-20601. For example, if all other conditions of the association protocol are satisfied, accepted is returned when the manager recognizes the *dev-config-id* of the agent and accepted-unknown-config otherwise.
- In the DataProtoList structure element, the data protocol identifier shall be set to data-proto-id-20601 (i.e., *data-proto-id* = 0x5079).

— The *data-proto-info* field shall be filled in with a PhdAssociationInformation structure that shall contain the following parameter values:

1) The version of the data exchange protocol shall be set to protocol-version1 (i.e., *protocol-version* = 0x80000000).
2) The manager shall respond with a single selected encoding rule that is supported by both agent and manager. The manager shall support at least the MDER.
3) The version of the nomenclature used shall be set to nom-version1 (i.e., *nomenclature-version* = 0x80000000).
4) The field *functional-units* shall have all bits reset except for those relating to a test association.
5) The field *system-type* shall be set to sys-type-manager (i.e., *system-type* = 0x80000000).
6) The *system-id* field shall contain the unique system ID of the manager device, which shall be a valid EUI-64 type identifier.
7) The field *dev-config-id* shall be manager-config-response (0).
8) The field *data-req-mode-capab* shall be 0.
9) The field *data-req-init-*-count* shall be 0.

8.4 Configuring procedure

8.4.1 General

The agent enters the Configuring state if it receives an association response of accepted-unknown-config. In this case, the configuration procedure as specified in IEEE Std 11073-20601 shall be followed. Subclause 8.4.2 specifies the configuration notification and response messages for a weighing scale agent with standard configuration ID 0x05DC. Normally, a manager would already know the standard configuration. However, standard configuration devices are required to send their configuration, if requested. This covers a case where an agent associates with a manager that does not have preconfigured knowledge of the standard configuration (e.g., due to a version mismatch between agent and manager).

8.4.2 Weighing scale—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). The ConfigReport structure is used for the *event-info* field (see Table 3). For a weighing scale agent with standard configuration ID 0x05DC, the format and contents of the configuration notification message are as follows:

```
0xE7 0x00                 APDU CHOICE Type (PrstApdu)
0x00 0x44                 CHOICE.length = 68
0x00 0x42                 OCTET STRING.length = 66
0xXX 0xXX                 invoke-id (differentiates this message from any other outstanding)
0x01 0x01                 CHOICE (Remote Operation Invoke | Confirmed Event Report)
0x00 0x3C                 CHOICE.length = 60
0x00 0x00                 obj-handle = 0 (MDS object)
0xXX 0xXX 0xXX 0xXX       event-time (set to 0xFFFFFFFF if RelativeTime is not supported)
0x0D 0x1C                 event-type = MDC_NOTI_CONFIG
0x00 0x32                 event-info.length = 50 (start of ConfigReport)
0x05 0xDC                 config-report-id (Dev-Configuration-Id value)
```

0x00 0x01	config-obj-list.count = 1 Measurement object will be “announced”
0x00 0x2C	config-obj-list.length = 44
0x00 0x06	obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01	obj-handle = 1 (→ 1st Measurement is body weight)
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 0xE1 0x40	MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0xF0 0x40	intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x09 0x96	attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02	attribute-value.length = 2
0x06 0xC3	MDC_DIM_KILO_G
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 0x56 0x00 0x04	MDC_ATTR_NU_VAL_OBS_SIMP | value length = 4
0x09 0x90 0x00 0x08	MDC_ATTR_TIME_STAMP_ABS | value length = 8

Note, at the locations of the message, where the content is not fixed, the value “0xXX” denotes a placeholder and depends on the implementation or on the preceding messaging of the agent.

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). As a response to the standard configuration notification message in 8.4.2.1, the format and contents of the manager’s configuration notification response message are as follows:

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x16	CHOICE.length = 22
0x00 0x14	OCTET STRING.length = 20
0xXX 0xXX	invoke-id (differentiates this message from any other outstanding)
0x02 0x01	CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E	CHOICE.length = 14
0x00 0x00	obj-handle = 0 (MDS object)
0xXX 0xXX 0xXX 0xXX	currentTime
0x0D 0x1C	event-type = MDC_NOTI_CONFIG
0x00 0x04	event-reply-info.length = 4
0x05 0xDC	ConfigReportRsp.config-report-id = 0x05DC
0x00 0x00	ConfigReportRsp.config-result = accepted-config.

Again, the value “0xXX” denotes a placeholder and refers to a fixed location, varying content parts of the message.

8.5 Operating procedure

8.5.1 General

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

8.5.2 GET weighing scale 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 weighing scale agent shall respond with a rors-cmip-get service message in which the attribute-list contains a list with the values 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, then the weighing scale agent shall respond with a rors-cmip-get service message in which the attribute-list contains a list of the values of the requested attributes of the MDS object that are implemented. If this capability is not implemented, the weighing scale agent shall respond with a "Remote Operation Error Result" (roer) service message (see IEEE Std 11073-20601) with the error-value field set to no-such-action (9).

8.5.3 Measurement data transmission

See Table 3 for a summary of the event report services available for measurement data transfer.

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

8.6 Time synchronization

Time synchronization between a weighing scale agent and a manager may be used 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 weighing scale 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 weighing scale 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 weighing scale 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 a body weight measurement of 55 kg. 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. If body height is supported, it shall be simulated as 172 cm.

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.

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

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

10.2 Conformance specification

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

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

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

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

Since this standard is used in conjunction with IEEE Std 11073-20601, the ICS should be created for this standard first. The ICS created for IEEE Std 11073-20601 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 and IEEE 11073-104zz documents. All mandatory features defined in the object definition tables and in the ICS tables are implemented. Furthermore, any conditional, recommended, or optional features that are implemented shall follow the requirements in IEEE Std 11073-20601 and IEEE 11073-104zz documents.

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

Conformance level 2 meets conformance level 1 but also uses or adds extensions in at least one of the information, service, communication, or nomenclature models. These extensions shall conform to 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

10.4.1 General format

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

Each ICS table has the following columns:

Index	Feature	Reference	Req/Status	Support	Comment

The table column headings have the following meaning:

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

— Comment: contains any additional information on the feature. This column is to be filled out by the implementer.

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

10.4.2 General implementation conformance statement

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

Table 9 shows the general ICSs.

Table 9—IEEE 11073-10415 general ICSs' table

Index[a]	Feature	Reference	Req./Status	Support	Comment
GEN 11073-10415-1	Implementation Description	—	Identification of the device/application. Description of functionality.		
GEN 11073-10415-2	Standards followed and their revisions	(standard documents)	(set of existing revisions)	(set of supported revision)	
GEN 11073-10415-3	Nomenclature document used and revision	(standard documents)	(set of existing revisions)	(set of supported revisions)	
GEN 11073-10415-4	Conformance Adherence - Level 1 -	See 10.3.2	Base conformance declaration that device meets the following IEEE Std 11073-10415 conformance requirements: a) All mandatory requirements shall be implemented. b) If implemented, conditional, recommended, and optional requirements shall conform to standard.	Yes/No (No is not expected as No implies that the implementation is non-conformant)	
GEN 11073-10415-5	Conformance Adherence - Level 2 -	See 10.3.3	In addition to GEN 11073-10415-4, if the device implements extensions and/or additions, they shall conform to nomenclature codes from ASN.1 and/or 10101 framework. These extensions should also be defined in ICS tables pointing towards their reference.	Yes/No	
GEN 11073-10415-6	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.		

Index[a]	Feature	Reference	Req./Status	Support	Comment
GEN 11073-10415-7	Nomenclature document used and revision	(standard documents)	(set of existing revisions)	(set of supported revision)	
GEN 11073-10415-8	Data Structure Encoding	—	—	description of encoding method(s) for ASN.1 data structures	
GEN 11073-10415-9	Use of Private Objects	—	Does the implementation use objects that are not defined in the DIM?	Yes/No (If yes: explain in Table 10)	
GEN 11073-10415-10	Use of Private Nomenclature Extensions	—	Does the implementation use private extensions to the nomenclature (i.e. 0xF000-0xFFFF codes from ISO/IEEE 11073-10101)? Private Nomenclature extensions are only allowed if the standard nomenclature does not include the specific terms required by the application.	Yes/No (If yes: explain in Table 13)	
GEN 11073-10415-11	11073-20601 Conformance		Provide the conformance report required by the IEEE Std 11073-20601.		

[a]The prefix GEN11073-10415- is used for the index in the general ICSs table.

10.4.3 DIM MOC implementation conformance statement

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 10.

Table 10—Template for DIM MOC ICS table

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

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

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

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

10.4.4 MOC attribute implementation conformance statement

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 11 is a template only.

Table 11—Template for MOC attribute ICS table

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

All private attributes shall 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; as well as any other information.

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

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

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

10.4.5 MOC notification implementation conformance statement

The MOC notification ICS specifies all implemented notifications (typically in the form of the event report service) that are emitted by the agent. Table 12 provides a template for use. One table has to be provided for each object that supports special object notifications. One row of the table shall be used for each notification.

Table 12—Template for MOC notification ICS table

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

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

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

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

10.4.6 MOC nomenclature conformance statement

The MOC nomenclature ICS specifies all nonstandard nomenclature codes that are utilized by the agent. Table 13 provides a template for use. One row of the table is to be used for each nomenclature element.

Table 13—Template for MOC nomenclature ICS table

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

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

Annex A

(informative)

Bibliography

[B1] Garrow, J.S., and Webster, J. Quetelet's index (W/H^2) as a measure of fatness. *International Journal of Obesity,* vol. 9, pp. 147–153, 1985.

[B2] IEEE 100™, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition. New York, Institute of Electrical and Electronic Engineers, Inc.[5,6]

[B3] ISO/IEEE 11073-10101:2004, Health informatics — Point-of-care medical device communication — Part 10101: Nomenclature.[7]

[B4] ISO/IEEE 11073-10201:2004, Health informatics — Point-of-care medical device communication — Part 10201: Domain information model.

[B5] ISO/IEEE 11073-20101:2004, Health informatics — Point-of-care medical device communication — Part 20101: Application profiles — Base standard.

[B6] ITU-T Rec. X.680-2002, Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation.[8]

[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]The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[7] ISO/IEEE publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEEE publications are also available in the United States from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

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

Annex B

(normative)

Any additional ASN.1 definitions

No additional ASN.1 definitions are defined.

Annex C

(normative)

Allocation of identifiers

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

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

```
/*****************************************************************************
* From Medical supervisory control and data acquisition (MDC_PART_SCADA)
*****************************************************************************/
#define MDC_LEN_BODY_ACTUAL                 57668 /*                         */
#define MDC_RATIO_MASS_BODY_LEN_SQ          57680 /*                         */

/*****************************************************************************
* From Dimensions (MDC_PART_DIM)
*****************************************************************************/
#define MDC_DIM_CENTI_M                     1297  /*  cm                     */
#define MDC_DIM_INCH                        1376  /*  in                     */
#define MDC_DIM_LB                          1760  /*  lb                     */
#define MDC_DIM_KG_PER_M_SQ                 1952  /*  kg m-2                 */
```

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 weighing scale agent intends to connect it to a manager for the first time. The weighing scale is capable of performing body weight and BMI measurements. Thus, it operates as an extended configuration. It is assumed that the person's height has already been manually entered into the weighing scale.

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

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

c) Subsequently, the manager requests the MDS object attributes of the agent by sending a data message with the "Remote Operation Invoke | Get" command. Note that the manager may request the MDS object attributes as soon as the agent enters the Associated state, including the Configuring and Operating substates. 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.4.2 and E.4.3 for the corresponding PDU examples.

d) As a next step, the user of the agent device takes a single measurement. The measurement data is 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. In that particular example, a set of two measurements is transmitted, with the first one being a stored, not yet transmitted, measurement.

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

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

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

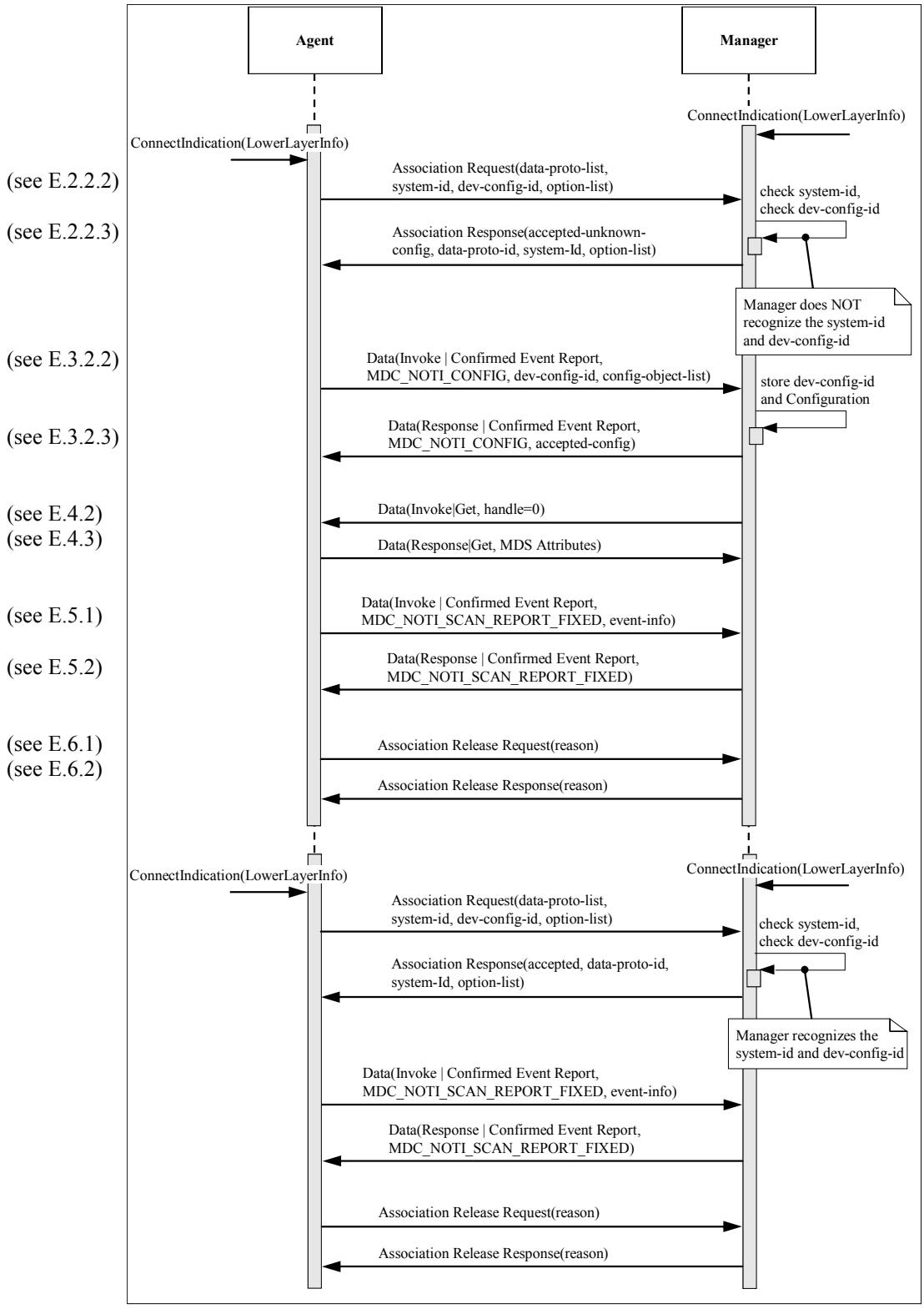

Figure D.1—Sequence diagram for weighing scale example use case

Annex E

(informative)

Protocol data unit examples

E.1 General

This annex shows MDER-encoded binary examples of messages exchanged between a weighing scale agent and manager. Three different scenarios containing the association and configuration information exchanges are presented in E.2 and E.2.3. 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 illustrates the agent presenting the same extended configuration to the manager, and the manager does have the configuration from the previously transferred configuration exchange. Finally, the agent presents a standard configuration to the manager, and the manager has the configuration because the manager has been preprogrammed with this configuration.

E.2 Association information exchange

E.2.1 General

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

E.2.2 Extended configuration

E.2.2.1 General

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

E.2.2.2 Association request

The weighing scale 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 0x88
0x40 0x00                   dev-config-id – extended configuration
0x00 0x01                   data-req-mode-flags
0x01 0x00                   data-req-init-agent-count, data-req-init-manager-count
0x00 0x00 0x00 0x00         optionList.count = 0 | optionList.length = 0
```

E.2.2.3 Association response

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

```
0xE3 0x00                   APDU CHOICE Type (AareApdu)
0x00 0x2C                   CHOICE.length = 44
0x00 0x03                   result = accepted-unknown-config
0x50 0x79                   data-proto-id = 20601
0x00 0x26                   data-proto-info length = 38
0x80 0x00 0x00 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

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

E.2.3.2 Association request

The weighing scale 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 0x88
0x40 0x00                   dev-config-id – extended configuration
0x00 0x01                   data-req-mode-flags
0x01 0x00                   data-req-init-agent-count, data-req-init-manager-count
0x00 0x00 0x00 0x00         optionList.count = 0 | optionList.length = 0
```

E.2.3.3 Association response

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

```
0xE3 0x00                   APDU CHOICE Type (AareApdu)
0x00 0x2C                   CHOICE.length = 44
0x00 0x00                   result = accepted
0x50 0x79                   data-proto-id = 20601
0x00 0x26                   data-proto-info length = 38
0x80 0x00 0x00 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 weighing scale agent sends the following message to the manager. The agent intends to associate using a standard configuration. The agent is willing to enter into a test association as defined in Clause 10.

```
0xE2 0x00                   APDU CHOICE Type (AarqApdu)
0x00 0x32                   CHOICE.length = 50
0x80 0x00 0x00 0x00         assoc-version
0x00 0x01 0x00 0x2A         data-proto-list.count = 1 | length = 42
0x50 0x79                   data-proto-id = 20601
0x00 0x26                   data-proto-info length = 38
0x80 0x00 0x00 0x00         protocolVersion
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 0x88
0x05 0xDC                   dev-config-id – standard configuration
0x00 0x01                   data-req-mode-flags
0x01 0x00                   data-req-init-agent-count, data-req-manager-count
0x00 0x00 0x00 0x00         optionList.count = 0 | optionList.length = 0
```

E.2.4.3 Association response

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

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

E.3 Configuration information exchange

E.3.1 General

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

E.3.2 Extended configuration

E.3.2.1 General

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

E.3.2.2 Remote operation invoke event report configuration

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

```
0xE7 0x00                   APDU CHOICE Type (PrstApdu)
0x00 0xA2                   CHOICE.length = 162
0x00 0xA0                   OCTET STRING.length = 160
0x12 0x35                   invoke-id = 0x1235 (start of DataApdu. MDER encoded.)
0x01 0x01                   CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x9A                   CHOICE.length = 154
0x00 0x00                   obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF         event-time = 0xFFFFFFFF
```

```
0x0D 0x1C                 event-type = MDC_NOTI_CONFIG
0x00 0x90                 event-info.length = 144 (start of ConfigReport)
0x40 0x00                 config-report-id
0x00 0x03                 config-obj-list.count = 3 Measurement objects will be “announced”
0x00 0x8A                 config-obj-list.length = 138
0x00 0x06                 obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01                 obj-handle = 1     (→ 1st Measurement is body weight)
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 0xE1 0x40       MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL
0x0A 0x46                 attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02                 attribute-value.length = 2
0xF0 0x40                  intermittent, stored data, upd & msmt aperiodic, agent init, measured
0x09 0x96                 attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02                 attribute-value.length = 2
0x06 0xC3                 MDC_DIM_KILO_G
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 0x56 0x00 0x04       MDC_ATTR_NU_VAL_OBS_SIMP | value length = 4
0x09 0x90 0x00 0x08       MDC_ATTR_TIME_STAMP_ABS | value length = 8
0x00 0x06                 obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x02                 obj-handle = 2     (→2nd Measurement is body height)
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 0xE1 0x44       MDC_PART_SCADA | MDC_LEN_BODY_ACTUAL
0x0A 0x46                 attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02                 attribute-value.length = 2
0xF0 0x48                  intermittent, stored data, upd & msmt aperiodic, agent init, manual
0x09 0x96                 attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02                 attribute-value.length = 2
0x05 0x11                 MDC_DIM_CENTI_M
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 0x56 0x00 0x04       MDC_ATTR_NU_VAL_OBS_SIMP, 4
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 body mass index)
0x00 0x05                 attributes.count = 5
0x00 0x2A                 attributes.length = 42
0x09 0x2F                 attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04                 attribute-value.length = 4
0x00 0x02 0xE1 0x50       MDC_PART_SCADA | MDC_RATIO_MASS_BODY_LEN_SQ
0x0A 0x46                 attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02                 attribute-value.length = 2
```

0xF0 0x42	intermittent, stored data, upd & msmt aperiodic, agent init, calculated
0x09 0x96	attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02	attribute-value.length = 2
0x07 0xA0	MDC_DIM_KG_PER_M_SQ
0x0A 0x47	attribute-id = MDC_ATTR_SOURCE_HANDLE_REF
0x00 0x02	attribute-value.length = 2
0x00 0x01	reference handle = 1
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 0x56 0x00 0x04	MDC_ATTR_NU_VAL_OBS_SIMP, 4
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 this 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 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
0x40 0x00	ConfigReportRsp.config-report-id = 0x4000
0x00 0x00	ConfigReportRsp.config-result = accepted-config

E.3.3 Known configuration

E.3.3.1 General

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

E.3.3.2 Remote operation invoke event report configuration

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

E.3.3.3 Remote operation response event report configuration

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

E.3.4 Standard configuration

E.3.4.1 General

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

E.3.4.2 Remote operation invoke event report configuration

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

E.3.4.3 Remote operation response event report configuration

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

E.4 GET MDS attributes service

E.4.1 General

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

E.4.2 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 0x34	invoke-id = 0x1234 (differentiates this message from any other outstanding, choice is implementation specific)
0x01 0x03	CHOICE (Remote Operation Invoke | Get)
0x00 0x06	CHOICE.length = 6
0x00 0x00	handle = 0 (MDS object)
0x00 0x00	attribute-id-list.count = 0 (all attributes)
0x00 0x00	attribute-id-list.length = 0

E.4.3 Get response with all MDS attributes

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

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x6E	CHOICE.length = 110
0x00 0x6C	OCTET STRING.length = 108
0x12 0x34	invoke-id = 0x1234 (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 0x0F                  type = MDC_DEV_SPEC_PROFILE_SCALE
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 0x54 0x68        string length = 12 | “TheScaleABC\0”
0x65 0x53 0x63 0x61
0x6C 0x65 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 0x88  octet string length = 8 | EUI-64
0x0a 0x44                  attribute-id = MDC_ATTR_DEV_CONFIG_ID
0x00 0x02                  attribute-value.length = 2
0x40 0x00                  dev-config-id = 16384 (extended-config-start)
0x09 0x2D                  attribute-id = MDC_ATTR_ID_PROD_SPECN
0x00 0x12                  attribute-value.length = 18
0x00 0x01                              ProductionSpec.count = 1
0x00 0x0E                              ProductionSpec.length = 14
0x00 0x01                              ProdSpecEntry.spec-type = 1 (serial-number)
0x00 0x00                              ProdSpecEntry.component-id = 0
0x00 0x08 0x44 0x45        string length = 8 | prodSpecEntry.prod-spec = “DE124567”
0x31 0x32 0x34 0x35
0x36 0x37
0x09 0x87                  attribute-id =MDC_ATTR_TIME_ABS
0x00 0x08                  attribute-value.length = 8
0x20 0x07 0x02 0x01        Absolute-Time-Stamp = 2007-02-01T12:05:0000
0x12 0x05 0x00 0x00
```

E.5 Data reporting

E.5.1 Confirmed measurement data transmission

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

```
0xE7 0x00                  APDU CHOICE Type (PrstApdu)
0x00 0x5A                  CHOICE.length = 90
0x00 0x58                  OCTET STRING.length = 88
0x12 0x36                  invoke-id = 0x1236
0x01 0x01                  CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x52                  CHOICE.length = 82
0x00 0x00                  obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00        event-time = 0
0x0D 0x1D                  event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x48                  event-info.length = 72
0xF0 0x00                  ScanReportInfoFixed.data-req-id = 0xF000
0x00 0x00                  ScanReportInfoFixed.scan-report-no = 0
0x00 0x04                  ScanReportInfoFixed.obs-scan-fixed.count = 4
```

```
0x00 0x40                 ScanReportInfoFixed.obs-scan-fixed.length = 64
0x00 0x01                 ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00 0x0C                 ScanReportInfoFixed.obs-scan-fixed.value[0]. obs-val-data.length = 12
0xFF 0x00 0x02 0xFA       Simple-Nu-Observed-Value = 76.2 (kg)
0x20 0x07 0x12 0x06       Absolute-Time-Stamp = 2007-12-06T12:10:0000
0x12 0x10 0x00 0x00
0x00 0x03                 ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 3
0x00 0x0C                 ScanReportInfoFixed.obs-scan-fixed.value[1]. obs-val-data.length = 12
0xFF 0x00 0x00 0xF3       Simple-Nu-Observed-Value = 24.3 (kg/m2)
0x20 0x07 0x12 0x06       Absolute-Time-Stamp = 2007-12-06T12:10:0000
0x12 0x10 0x00 0x00
0x00 0x01                 ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00 0x0C                 ScanReportInfoFixed.obs-scan-fixed.value[0]. obs-val-data.length = 12
0xFF 0x00 0x02 0xF8       Simple-Nu-Observed-Value = 76.0 (kg)
0x20 0x07 0x12 0x06       Absolute-Time-Stamp = 2007-12-06T20:05:0000
0x20 0x05 0x00 0x00
0x00 0x03                 ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 3
0x00 0x0C                 ScanReportInfoFixed.obs-scan-fixed.value[1]. obs-val-data.length = 12
0xFF 0x00 0x00 0xF2       Simple-Nu-Observed-Value = 24.2 (kg/m2)
0x20 0x07 0x12 0x06       Absolute-Time-Stamp = 2007-12-06T20:05:0000
0x20 0x05 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 0x36                 invoke-id = 0x1236 (mirrored from invocation)
0x02 0x01                 CHOICE(Remote Operation Response | Confirmed Event Report)
0x00 0x0A                 CHOICE.length = 10
0x00 0x00                 obj-handle = 0 (MDS object)
0x00 0x00 0x00 0x00       currentTime = 0
0x0D 0x1D                 event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x00                 event-reply-info.length = 0
```

E.6 Disassociation

E.6.1 Association release request

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

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

E.6.2 Association release response

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

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

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

Keywords: medical device communication, personal health devices, weighing scale

ICS 35.240.80
ISBN 978-0-7381-5818-1

Price based on 41 pages