31 May 2012

Status
Approved

Revision
V13r00

Document File Name
Personal Health Devices Transcoding_WP_V13

Document Owner
Medical Devices WG

E-mail Address
med-main@bluetooth.org

Bluetooth®
SPECIAL INTEREST GROUP

PERSONAL HEALTH DEVICES TRANSCODING WHITE PAPER

ABSTRACT: This document is informative. It aims to facilitate the task of implementing a transcoder from GATT based specifications designed for *Bluetooth* LE devices to a format compatible with IEEE 11073-20601. It provides recommendations and examples describing how a transcoding process can be done.

Revision History		
Revision	**Date**	**Description**
V10r00	10 May 2011	Approved by the Bluetooth SIG Board of Directors.
V11r00	25 October 2011	Incorporated Heart Rate transcoding section 3.2. Incorporated fixes and clarifications throughout document based on feedback. Approved by the Bluetooth SIG Board of Directors.
V12r00	15 November 2011	Added Blood Pressure transcoding section 3.3. Minor editorial and formatting fixes in other sections. Incorporated fixes and clarifications based on feedback. Approved by the Bluetooth SIG Board of Directors.
V13r00	31 May 2012	Added Glucose Meter transcoding section 3.4. Changed Thermometer transcoding section 3.1.2. Added Metric-Id to identify thermometer location. Changed section 2.2.4, item Date-and-Time. Incorporated fixes and clarifications based on feedback.

Contributors	
Name	**Company**
Raúl González Gómez	AT4 wireless
Robert D. Hughes	Intel Corporation
Doug Bogia	Intel Corporation
Krishna Shingala	MindTree
Laurie Kermes	Nonin Medical
Mateus Lima	Signove
Raul Herbster	Signove
Elvis Pfützenreuter	Signove
Len Ott	Socket Mobile
John R. Barr	Motorola Mobility, Inc.
Guillaume Schatz	Polar
Raymond Strickland	Roche

Contents

1. Overview

This document describes how data sent by a *Bluetooth*® low energy (LE) Sensor and received by a Collector can be transcoded at the Collector into an IEEE Std. 11073-20601a [1] compatible nomenclature and model, thus enabling compatibility with the ISO/IEEE Std. 11073-104xx family of standards (both hereafter known as 11073). Services and profiles for *Bluetooth* LE implementations have been developed using the GATT based profile architecture designed to support low power and low cost device implementation. Data values used in these profiles are defined as characteristics associated with a UUID accessible via the Bluetooth SIG Assigned Numbers [4].

This document covers how characteristic values can be mapped or transcoded in a consistent way to 11073 nomenclature/object/attribute equivalents. This data compatibility will enable data from *Bluetooth* LE devices to be used in the broader health ecosystem such that the transcoded measurement data will look the same as data from a *Bluetooth* Health Device Profile device or a USB Personal Healthcare Device Class device.

This data compatibility will enable data to be useable and consumable by a variety of healthcare-related organizations including the Continua Health Alliance and standards organizations related to health records such as HL7. All mandatory (as well as some optional) attributes defined for each specialization in 11073 are supported by *Bluetooth* profiles defined within this document, but support for optional attributes is not specifically required for data compatibility.

For the profiles encompassed by the material in this document, all characteristics and fields that are relevant to 11073 have been defined with the intent that they can be transcoded at the Collector without any loss of precision.

In order to enable such a process for a particular device, the Collector device implementing transcoding software is required to follow the general requirements in Section 2 and the device-specific requirements in Section 3. Section 3 will be expanded as new *Bluetooth* profiles become available. Section 3.3 provides an end-to-end example describing how data can be mapped from a LE Health Thermometer [3] to a Collector implementing a Transcoder.

While it is beyond the scope of this document to mandate or specify a specific *method* for transcoding *Bluetooth* characteristics into 11073, this white paper provides requirements and guidelines to enable implementations to do so. This document does not discuss 11073 concepts and details; rather, it focuses on how data from *Bluetooth* sensors can be transcoded for use in the 11073 domain.

Although some areas of this document summarize requirements of 11073 documents as a useful reference, refer to the 11073 standards to ensure the most accurate information regarding 11073 specifications' requirements. Similarly, refer to the relevant *Bluetooth* profile specifications with their associated service specifications and characteristic(s) as the official sources for *Bluetooth*-related requirements.

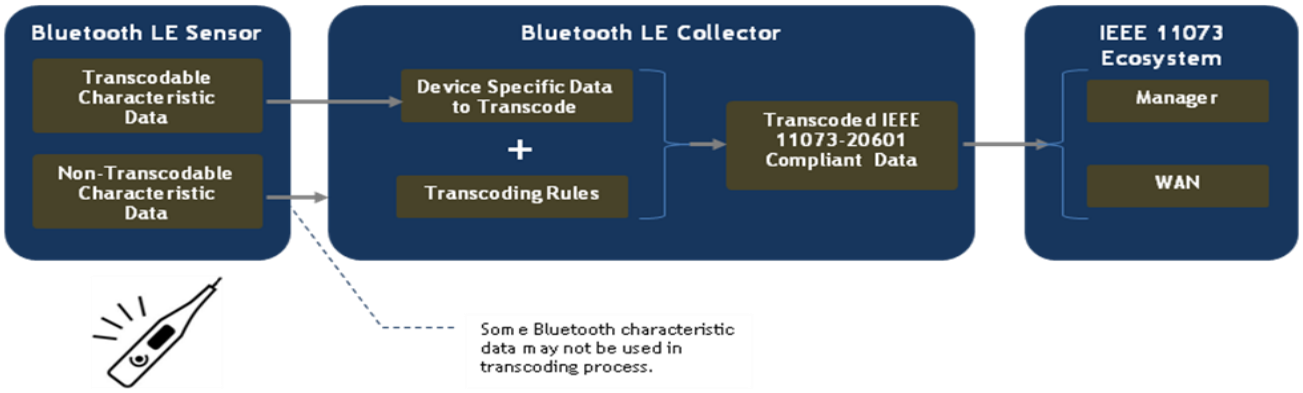

Figure 1: Bluetooth LE Sensor to 11073 Data Flow

2. General Data Requirements

This section describes the transcoding of general data from a *Bluetooth* sensor device for compatibility with the 11073 ecosystem, and general data mapping requirements that are common to all devices addressed by this white paper. Device-specific data requirements are described in Section 3.

2.1 COMMON MDS CLASS REQUIREMENTS

This sub-section describes general Medical Device System (MDS) class requirements from 11073-20601. 11073 attributes not mentioned in this section likely refer to *Bluetooth* device-specific data requirements and are shown in Section 3.

The Device Information Service (DIS) [2] is a general *Bluetooth* service designed to describe characteristics that are often common between different sensors. The DIS includes characteristics that contain information such as the manufacturer name, model number, hardware revision, firmware revision, and software revision among others. Many characteristics in the DIS are used within the 11073 and Continua Health Alliance infrastructure.

11073 Attribute	*Bluetooth* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	None	N/A	N/A	HANDLE[1]	INT-U16
System-Model	Model Number String, Manufacturer Name String	DIS	UTF-8 String, UTF-8 String	SystemModel[2]	(OCTET STRING, OCTET STRING)
System-Id	System ID	DIS	EUI-64	OCTET STRING	OCTET STRING
Attribute-Value-Map	None	N/A	N/A	AttrValMap[3]	List of (INT-U16, INT-U16)
Production-Specification	Serial Number String, Hardware Revision String, Software Revision String, Firmware Revision String[4]	DIS	UTF-8 String	ProductionSpec	List of (INT-U16, INT-U16, OCTET STRING)
Date-and-Time	Date Time	Various[5]	Aggregate	AbsoluteTime	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Mds-Time-Info	None	N/A	N/A	MdsTimeInfo	(BITS-16, INT-U16, INT-U32, INT-U16, INT-U16, INT-U32)
Relative-Time	None	N/A	N/A	RelativeTime	INT-U32

11073 Attribute	*Bluetooth* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
HiRes-Relative-Time	None	N/A	N/A	HighResRelativeTime	OCTET STRING (SIZE(8))
Date-and-Time-Adjustment	None	N/A	N/A	AbsoluteTimeAdjust	OCTET STRING (SIZE(6))
Power-Status	Drafting	Drafting	N/A	PowerStatus	BITS-16
Battery-Level	Battery Level	Battery	UINT8	INT-U16[6]	INT-U16
Remaining-Battery-Time	None	N/A	N/A	BatMeasure	(FLOAT-Type, INT-U16)
Reg-Cert-Data-List	IEEE 11073-20601 Regulatory Certification Data List	DIS	Aggregate	RegCertDataList	List of ((INT-U8, INT-U8), ANY)
System-Type-Spec-List	None	N/A	N/A	TypeVerList[7]	List of (INT-U16, INT-U16)
Confirm-Timeout	None	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))

Table 1: Common MDS Class Requirements

Notes:

1. Always set to 0.
2. SystemModel is described as SEQUENCE (Manufacturer Name, Model Number).
3. This 11073 attribute is transmission-related only. As this document is only relevant for nomenclature and model compatibility and does not mandate any way to reach the 11073 domain, this attribute is implementation specific.
4. Each field is mapped into a unique entry of ProductionSpec list; for example, Serial Number String is mapped as (0x0001, INT-U16, serial_number_value). See Section 2.2.4.
5. See item 2.2.4, “Date and Time”.
6. Both Battery Level characteristic and Battery-Level attribute express battery level as an integer percentage (0 to 100). Characteristic is transcoded simply by promoting value from UINT8 to INT-U16 with no scaling.
7. For each profile addressed by this white paper and implemented by the device, an entry has to be added to the TypeVerList. For example, if a device implements the Health Thermometer Profile [3], the following entry is required by 11073-20601 to be added to the TypeVerList:

   0x10 0x08 type = MDC_DEV_SPEC_PROFILE_TEMP
   0x00 0x01 version = version 1 of the specialization

2.2 TRANSCODING *BLUETOOTH* CHARACTERISTICS TO 11073 ATTRIBUTES

2.2.1 32-BIT FLOATING POINT DATA TYPE (FLOAT-TYPE)

The following information is defined in ISO/IEEE Std. 11073-20601™-2008 [1].

The FLOAT-Type data type is defined to represent numeric values that are not integer in type. The FLOAT-Type is defined as a 32-bit value with a 24-bit mantissa and an 8-bit exponent. See Annex F.6 of [1] for a thorough definition of the FLOAT-Type. This data type is defined as follows:

	Exponent	**Mantissa**
Size	1 octet	3 octets

The 32 bits contain an 8-bit signed exponent to base 10, followed by a 24-bit signed integer (mantissa).

Special values are assigned to express the following:

- + INFINITY [exponent 0, mantissa +(2^23 –2) → 0x007FFFFE]
- NaN (not a number) [exponent 0, mantissa +(2^23 –1) → 0x007FFFFF]
- NRes (not at this resolution) [exponent 0, mantissa –(2^23) → 0x00800000]
- Reserved for future use [exponent 0, mantissa –(2^23–1) → 0x00800001]
- – INFINITY [exponent 0, mantissa –(2^23 –2) → 0x00800002]

NaN is used to report an invalid result from a computation step or to indicate missing data due to the hardware's inability to provide a valid measurement, perhaps from sensor disturbances.

NRes is used to report that the value cannot be represented with the available range and resolution, possibly resulting from an overflow or underflow situation.

Example

Example for temperature measurement in 11073-20601 FLOAT-Type format:

Consider a temperature measurement of 36.4 **degrees** Celsius with precision of 0.1 **degrees** Celsius. The FLOAT-Type representation is a 32-bit value consisting of an exponent of an 8-bit signed integer followed by a mantissa of a 24-bit signed integer; here, the exponent is -1 (0xFF) and the mantissa is 364 (0x00016C). Therefore, the FLOAT-Type representation of 36.4 is 0xFF00016C.

2.2.2 16-BIT FLOATING POINT DATA TYPE (SFLOAT-TYPE)

The following information is defined in ISO/IEEE Std. 11073-20601™-2008 [1].

The SFLOAT-Type data type (a shortened version of FLOAT-Type) is defined to represent numeric values that are not integer in type. The SFLOAT-Type is defined as a 16-bit value with 12-bit mantissa and 4-bit exponent. See Annex F.8 of [1] for a thorough definition of the SFLOAT-Type. This data type is defined as follows:

	Exponent	**Mantissa**
Size	4 bit	12 bit

The 16–bit value contains a 4-bit exponent to base 10, followed by a 12-bit mantissa. Each is in twos-complement form.

Special values are assigned to express the following:

- NaN [exponent 0, mantissa +(2^11 –1) → 0x07FF]
- NRes [exponent 0, mantissa –(2^11) → 0x0800]
- + INFINITY [exponent 0, mantissa +(2^11 –2) → 0x07FE]
- – INFINITY [exponent 0, mantissa –(2^11 –2) → 0x0802]
- Reserved for future use [exponent 0, mantissa –(2^11 –1) → 0x0801]

Example

Example for blood pressure measurement in 11073-20601 SFLOAT-Type format:

Consider a systolic blood pressure measurement of 114 mmHg with a precision of 1 mmHg. The SFLOAT-Type representation is a 16-bit value consisting of an exponent of a 4-bit signed integer followed

by a mantissa of a12-bit signed integer; here, the exponent is 0 (0x0) and the mantissa is 114 (0x072). Therefore, the SFLOAT-Type representation of 114 is 0x0072.

2.2.3 STRING CONVERSION

11073 variable-length string type is encoded with a field length of 2 octets followed by the specific OCTET STRING data array. 11073 strings must be even length (16-bit aligned). For optimized data exchange over *Bluetooth* LE, no such requirement exists. *Bluetooth* characteristic strings can be odd or even length, and the length of the string can be deciphered from the data. To transcode an odd length string, append a zero (0x00) byte to the end of the string, and increment the string length field.

Bluetooth characteristic strings are encoded as UTF-8, whereas 11073 strings are encoded as ASCII printable characters (a UTF-8 subset). The transcoder shall convert non-ASCII characters in characteristic strings to ASCII in order to satisfy 11073 standards. The converted string may have a different length than the original UTF-8 string. The conversion should be done such that it maintains human readability; e.g., accented characters should be converted to non-accented equivalents.

2.2.4 MDS ATTRIBUTE CONVERSION

System-Model

The System-Model 11073 attribute consists of a sequence that contains manufacturer name and model number, respectively. Its content is vendor-decided, and represented as an OCTET STRING. Therefore, it must follow the string conversion rules as described in Section 2.2.3.

All fields of the System-Model attribute are derived from the characteristics in the DIS [2]. The *Bluetooth* “Manufacturer Name String” and “Model Number String” characteristics of the DIS map to the Manufacturer Name and Model Number field of the System Model 11073 Attribute.

System-Id

The System-Id 11073 attribute has the same constraints as defined by the “System ID” *Bluetooth* characteristic (an EUI-64, which consists of a 24-bit Organizationally Unique Identifier followed by a 40-bit manufacturer-defined identifier). It is mapped directly from the DIS “System ID” *Bluetooth* characteristic value. For more information, see the DIS [2].

Production-Specification

The 11073 attribute consists of a ProdSpecEntry list. Each entry may describe specific information such as serial number, hardware revision, software revision, protocol revision, firmware revision, and part numbers. Additionally, each entry in the list contains a Spec Type defining which type of specification it refers to, a vendor-specified component ID, and a vendor-specified ASCII printable string, mapped directly from the DIS characteristics as follows.

ProdSpecEntry Spec Type	*Bluetooth* Characteristic	11073 Spec Type Value
Unspecified	N/A	0x0000
Serial Number	Serial Number String (DIS)	0x0001
Part Number	N/A	0x0002
Hardware Revision	Hardware Revision String (DIS)	0x0003
Software Revision	Software Revision String (DIS)	0x0004
Firmware Revision	Firmware Revision String (DIS)	0x0005
Protocol Revision	N/A	0x0006
GMDN (Global Medical Device Nomenclature)	N/A	0x0007

Table 2: Production Specification

The conversion is done as follows for each *Bluetooth* characteristic (Serial Number String, Hardware Revision String, Software Revision String, and Firmware Revision String):

1. Create a new ProdSpecEntry.
2. Set the first field (spec_type) according to table above.

3. Set the second field to the vendor-specified component ID.
4. Set the third field to the corresponding *Bluetooth* characteristic value.

Date-and-Time

The transcoder shall provide this attribute if measurements have timestamps (ref: [1] Section 6.3.2.3). The attribute represents the current time in sensor's clock.

If the sensor reports a timestamp in any message, the sensor implementation shall include the Date Time characteristic, which is transcoded to the Date-and-Time attribute as specified in Section 2.2.6.

2.2.5 IEEE 11073-20601 REGULATORY CERTIFICATION DATA LIST

Health and Medical Devices may claim adherence to various regulatory and/or certification compliance items as an informative statement.

The IEEE 11073-20601 Regulatory Certification Data List enables a device to list the compliance items identifying the authorizing body and its data. IEEE, Continua, and the FDA are a few examples of authorizing bodies.

The IEEE 11073-20601 Regulatory Certification Data List is defined as an opaque structure in the DIS by a regulatory body. The endianness of this data structure is as defined in the associated regulatory specification. For example, if the regulatory body is Continua, the "RegCertDataList" data structure will be in big-endian format.

Following is an example of this structure based on Continua Design Guidelines 2010 (v1.5) [7].

In this example, the *IEEE 11073-20601 Regulatory Certification Data List* characteristic is required to have the following format:

Field Name	Offset	Size	Data Type	Definition / Notes
Regulatory Certification Data List	0			
Count	0	2 octet	INT-U16	
Length	2	2 octet	INT-U16	
Authorization Body	4	1 octet	INT-U8	Code assigned by IEEE 11073-20601 identifying the authorizing body
Authorization Body Structure Type	5	1 octet	INT-U8	Identifies the data structure
Authorization Body Structure Length	6	2 octet	INT-U16	Defines authorization body data length
Authorizing Body Data	8	variable length	Opaque structure	Format defined by Authorizing Body (Continua)
Major IG version	8	1 octet	INT-U8	
Minor IG version	9	1 octet	INT-U8	
Certified device class list	10			
Count (c)	10	2 octet	INT-U16	Number of device classes
Length	12	2 octet	INT-U16	c * sizeof(INT-U16)
Certified device class entry	12+n*2	2 octet	INT-U16	May be several of these entries (i.e., c > 1), where *n* is the index of device entry [1..c].
Continua Regulatory Structure	14+c*2	2 octet	INT-U16	
Structure length	16+c*2	2 octet	INT-U16	
Regulation Bit Field Type	18+c*2	2 octet	BITS-16	

Table 3: Format Example for IEEE 11073-20601 Regulatory Certification Data List Characteristic

Regulatory Certification Data List and *Certified device class list* are types based on SEQUENCE OF, which specifies a list header with both *count* and *length*, even when the entry size is perfectly known. This allows for robust decoding, allowing the decoder to skip the sequence when it does not know the entry type.

Table 4 is an example showing the contents of this structure based on Continua Design Guidelines 2010 (v1.5) [7]. The minor and major Interoperability Guidelines (IG) version represents the current Continua Guideline being followed, and will be updated as new guidelines are adopted and followed by this document.

In this example, the Continua Health Alliance is the regulatory body and the device includes only one device specialization – the IEEE 11073-10408 Thermometer device specialization [5]. As a result, the *IEEE 11073-20601 Regulatory Certification Data List* characteristic is required to have the following format and values:

Data	Description
0x00 0x02	RegCertDataList.count = 2
0x00 0x12	RegCertDataList.length = 18
0x02 0x01	RegCertDataList[0] auth-body = auth-body-continua = 2 RegCertDataList[0].auth-body-struc-type = continua-version-struct = 1 (*ContinuaBodyStruct*)
0x00 0x08	RegCertDataList[0]. auth-body-data.length = 8
0x01 0x05	RegCertDataList[0]. auth-body-data: ContinuaBodyStruct.major-IG-version = 1 ContinuaBodyStruct.minor-IG-version = 5
0x00 0x01	CertifiedDeviceClassList.count = 1
0x00 0x02	CertifiedDeviceClassList.length = 2
0x80 0x08	CertifiedDeviceClassList[0] = 0x8008 Based on Continua 2010 (v1.5) guidelines for a Low Power Wireless PAN Thermometer:Transport Code (TCode) = 4 (Low Power Wireless PAN) MDC_DEV_SPEC_PROFILE_TEMP = 0x10 0x08 = 4104_{10} CertifiedDeviceClass = 4104-4096+4*8192 = 32776_{10} -> 0x8008
0x02 0x02	RegCertDataList[1] auth-body = auth-body-continua = 2 RegCertDataList[1].auth-body-struc-type = continua-reg-struct = 2 (*ContinuaRegStruct*)
0x00 0x02	RegCertDataList[1]. auth-body-data.length = 2
0x00 0x00	This is a regulated device

Table 4: IEEE 11073-20601 Regulatory Certification Data List Characteristic Example

In this example, the total length of the structure is 22 octets.

2.2.6 TRANSCODING TIME STAMP TO IEEE 11073-20601 ABSOLUTETIME

For the *Bluetooth* profiles addressed by this white paper, the measurement time stamp follows the format of the *Bluetooth* Date Time characteristic.

This format is encoded according to the Date Time characteristic definition accessible via the Bluetooth SIG Assigned Numbers [4]. Each field is an 8-bit integer, except for "year", which is a 16-bit integer. For example, the time stamp for 18th December 2010, 15:23:06 is encoded as 0x07DA 0x0C 0x12 0x0F 0x17 0x06.

IEEE 11073-20601 AbsoluteTime data format is encoded using binary coded decimal (i.e., 4-bit nibbles) and every field has 8 bits. For example, the time stamp for 18th December 2010, 15:23:06:73 is encoded as 0x20 0x10 0x12 0x18 0x15 0x23 0x06 0x73.

AbsoluteTime specifies time with a resolution of 1/100 of a second and the *Bluetooth* Date Time characteristic has a resolution of one second, so the sec-fractions field must be set to zero in the transcoding process.

Because AbsoluteTime is encoded as a Binary Coded Decimal (BCD) format, a conversion between formats is needed; however, this conversion does not result in a loss of precision.

Table 5 shows mapping from the *Bluetooth* Time Stamp fields to the AbsoluteTime type fields:

Bluetooth Field Name (Date Time)	AbsoluteTime Field Name
Year	century[1]
	year[2]
Month	month
Day	day
Hours	hour
Minutes	minute
Seconds	second
N/A	sec-fractions

Table 5: Bluetooth Date Time Characteristic to IEEE 11073-20601 AbsoluteTime Conversion

Notes:

1. This field is set to the two most significant digits of the Bluetooth Year field.
2. This field is set to the two least significant digits of the Bluetooth Year field.

2.3 ATTRIBUTE-VALUE-MAP

Each Metric object has a conditional Attribute-Value-Map attribute. It is mandatory if the agent uses fixed-format data update messages. Fixed-format event reports refer to those attributes.

As this document is only relevant for nomenclature and model compatibility and does not mandate any way to reach the 11073 domain, this attribute is implementation-specific.

2.4 PM-SEGMENT-ENTRY-MAP

Each PM-Segment contains a mandatory PM-Segment-Entry-Map attribute that defines the (fixed) format of each segment's entry. All entries of a given segment have the same format.

As this document is only relevant for nomenclature and model compatibility and does not mandate any way to reach the 11073 domain, this attribute is implementation-specific.

3. Device Specific Data Requirements

This section describes the mapping of device specific data from a *Bluetooth* environment to an 11073 environment. This section will be expanded for various *Bluetooth* profiles and services in the future as they become available.

3.1 HEALTH THERMOMETER

This sub-section defines transcoding the thermometer device specific data into IEEE 11073-10408 Thermometer device specialization [5] class attributes.

3.1.1 DEVICE-SPECIFIC MDS CLASS REQUIREMENTS

In addition to the MDS class requirements shown in Section 2.1, Table 6 shows incremental MDS class requirements specific to this device.

11073 Attribute	*Bluetooth* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
System-Type	N/A	N/A	N/A	TYPE[1]	(INT-U16, INT-U16)
Dev-Configuration-Id	N/A	N/A	N/A	ConfigId[2]	INT-U16
System-Type-Spec-List	N/A	N/A	N/A	TypeVerList[3]	List of (INT-U16, INT-U16)
Confirm-Timeout	N/A	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))

Table 6: Device-specific MDS Class Requirements

Notes:

1. Value not present since System-Type-Spec-List exists.
2. Set to any value in range of 0x4000 to 0x7FFF (Extended Configuration). This is to assure that the Manager requests the configuration and does not assume it, as it could for a Standard Configuration.
3. Since the “Health Thermometer Service“ [6] is a “Primary Service”, an entry is required to be added to the TypeVerList as follows:

   0x10 0x08 type = MDC_DEV_SPEC_PROFILE_TEMP
   0x00 0x01 version = version 1 of the 11073 device specialization

3.1.2 11073 NUMERIC CLASS REQUIREMENTS

This section describes the 11073 numeric class requirements. It is restricted to those 11073 attributes that are used on the Thermometer device. All unmentioned attributes defined in [1] are not applicable to a Thermometer device.

11073 Attribute	*Bluetooth* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Metric-Id	Temperature Measurement	Health Thermometer	Aggregate	OID-Type[4]	INT-U16
Unit-Code	Temperature Type and Temperature Measurement	Health Thermometer	Aggregate	OID-Type[5]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[6]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Temperature Measurement	Health Thermometer	Aggregate	AbsoluteTime[7]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Simple-Nu-Observed-Value	Temperature Measurement	Health Thermometer	Aggregate	SimpleNuObsValue[8]	FLOAT-Type

Table 7: 11073 Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Set to {MDC_PART_SCADA, MDC_TEMP_BODY}.
3. When the Measurement Interval characteristic is not present or when it is present and its value is zero (aperiodic mode) then this is set to 0xF040 (mss-avail-intermittent, mss-avail-stored-data, mss-upd-aperiodic, mss-msmt-aperiodic, mss-acc-agent-initiated). When the Measurement Interval characteristic is present and its value is non-zero (periodic mode) this is set to 0x4040 (mss-avail-stored-data, mss- acc-agent-initiated).
4. The value is inferred based on Temperature Type field of Temperature Measurement, when it is present (presence indicated by bit 2 of Flags field). If not present, it is inferred on Temperature Type characteristic (which is optional). If both are absent, Metric-Id shall be set to MDC_TEMP_BODY.

11073 Temperature Metric-Id code	*Bluetooth* Temperature Type Description	*Bluetooth* Value
MDC_TEMP_AXILLA	Armpit	0x01
MDC_TEMP_BODY	Body (general)	0x02
MDC_TEMP_EAR	Ear (usually ear lobe)	0x03
MDC_TEMP_FINGER	Finger	0x04
MDC_TEMP_GIT	Gastro-intestinal Tract	0x05
MDC_TEMP_ORAL	Mouth	0x06
MDC_TEMP_RECT	Rectum	0x07
MDC_TEMP_TOE	Toe	0x08

11073 Temperature Metric-Id code	*Bluetooth* Temperature Type Description	*Bluetooth* Value
MDC_TEMP_TYMP	Tympanum (ear drum)	0x09
	Reserved	All other values

Table 8: Temperature Metric-Id Description Conversion

5. This value is mapped from Bit 0 of the least significant octet of the Temperature Measurement characteristic. The mapping is as follows in Table 9:

11073 Temperature Unit Value	*Bluetooth* Temperature Unit Value	Temperature Unit Description
MDC_DIM_DEGC	0	Celsius
MDC_DIM_FAHR	1	Fahrenheit

Table 9: Temperature Type Description Conversion

6. See Section 2.3.
7. When supported, this value is derived from the Time Stamp field of Temperature Measurement characteristic. See Section 2.2.6.
8. This value is derived from the Temperature Measurement Value field of the Temperature Measurement characteristic.

3.2 HEART RATE SENSOR

This section defines transcoding the heart rate sensor device specific data into IEEE 11073-10406 Basic Electrocardiograph (ECG) device specialization [8] class attributes.

3.2.1 DEVICE-SPECIFIC MDS CLASS REQUIREMENTS

In addition to the MDS class requirements shown in Section 2.1, Table 10 shows incremental MDS class requirements specific to this device.

11073 Attribute	*BLUETOOTH* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
System-Type	N/A	N/A	N/A	TYPE[1]	(INT-U16, INT-U16)
Dev-Configuration-Id	N/A	N/A	N/A	ConfigId[2]	INT-U16
System-Type-Spec-List	N/A	N/A	N/A	TypeVerList[4]	List of (INT-U16, INT-U16)
Confirm-Timeout	N/A	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))
Tick-resolution[3]	N/A	N/A	N/A	FLOAT-Type	FLOAT-Type

Table 10: Device-specific MDS Class Requirements

Notes:

1. Value not present since System-Type-Spec-List exists.
2. Set to any value in range of 0x4000 to 0x7FFF (Extended Configuration). This is to assure that the Manager requests the configuration and does not assume it, as it could for a Standard Configuration.
3. Defined as 2^(-10)=1/1024. Matches [9] RR-Interval unit, and can be represented exactly by FLOAT-Type.

4. Since the “Heart Rate Service” [9] is a “Primary Service”, the following entries are required to be added to the TypeVerList:

   0x10 0x06 Specialization value = MDC_DEV_SPEC_PROFILE_ECG
   0x00 0x01 version = version 1 of the 11073 device specialization

   0x10 0x8D Profile value = MDC_DEV_SUB_SPEC_PROFILE_HR
   0x00 0x01 version = version 1 of the 11073 device specialization

3.2.2 11073 NUMERIC CLASS REQUIREMENTS

This section describes the 11073 numeric class requirements. It is restricted to those 11073 attributes that are used on the Basic Electrocardiograph device and the Cardiovascular Fitness and Activity Monitor device. All unmentioned attributes defined in [1] are not applicable to those devices.

3.2.2.1 HEART RATE MEASUREMENT

11073 Attribute	*Bluetooth* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	N/A	N/A	N/A	OID-Type[4]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[5]	List of (INT-U16, INT-U16)
Simple-Nu-Observed-Value	Heart Rate Measurement (Heart Rate Measurement Value field)	Heart Rate	Aggregate	SimpleNuObs Value[6]	FLOAT-Type

Table 11: 11073 Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value is set to {MDC_PART_SCADA | MDC_ECG_HEART_RATE_INSTANT}.
3. Set to 0x4040 (mss-avail-stored-data, mss-acc-agent-initiated).
4. Unit is MDC_DIM_BEAT_PER_MIN.
5. See Section 2.3.
6. This value is derived from the Heart Rate Measurement Value field of the Heart Rate Measurement characteristic, which is either an 8-bit or 16-bit unsigned integer depending upon bit 0 of the Flags field. This value is converted to FLOAT-Type for transcoding using an exponent of 0.

3.2.2.2 RR-INTERVAL

Optionally transcoded when Heart Rate Measurement contains RR-Interval data.

11073 Attribute	*Bluetooth* Equivalent Characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	N/A	N/A	N/A	OID-Type[4]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[5]	List of (INT-U16, INT-U16)
Compound-Simple-Nu-Observed-Value	Heart Rate Measurement (RR-Interval field)	Heart Rate	Aggregate	Compound-SimpleNuObsValue[6]	List of FLOAT-Type

Table 12: 11073 Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_SCADA | MDC_ECG_TIME_PD_RR_GL.
3. Set to 0x5440 (mss-avail-stored-data, mss-acc-agent-initiated, mss-msmt-btb-metric, mss-msmt-aperiodic.
4. Unit is MDC_DIM_TICK (1/1024s).
5. See Section 2.3.
6. One or more values are derived from the variable-size RR-Interval field of Heart Rate Measurement characteristic.

3.3 BLOOD PRESSURE MONITOR

This section defines transcoding the blood pressure monitor device specific data into IEEE 11073-10407 Blood Pressure Monitor device specialization [11] class attributes.

3.3.1 DEVICE-SPECIFIC MDS CLASS REQUIREMENTS

In addition to the MDS class requirements shown in Section 2.1, Table 13 shows incremental MDS class requirements specific to this device.

11073 Attribute	*Bluetooth* equivalent characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
System-Type	N/A	N/A	N/A	TYPE[1]	(INT-U16, INT-U16)
Dev-Configuration-Id	N/A	N/A	N/A	ConfigId[2]	INT-U16
System-Type-Spec-List	N/A	N/A	N/A	TypeVerList[3]	List of (INT-U16, INT-U16)
Confirm-Timeout[4]	N/A	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))

Table 13: Device-specific MDS Class Requirements

Notes:

1. Value not present since System-Type-Spec-List exists.
2. Set to any value in range of 0x4000 to 0x7FFF (Extended Configuration). This is to assure that the Manager requests the configuration and does not assume it, as it could for a Standard Configuration.
3. Since the "Blood Pressure Service" [10] is a "Primary Service," the following entries are required to be added to the TypeVerList:

   0x10 0x07 Specialization value = MDC_DEV_SPEC_PROFILE_BP
   0x00 0x01 version = version 1 of the 11073 device specialization

4. This is optional and informs the manager how much time the transcoding agent will wait for confirmation of event reports before transitioning to Unassociated state.

3.3.2 11073 NUMERIC CLASS REQUIREMENTS

This section describes the 11073 numeric class requirements. It is restricted to those 11073 attributes that are used on the Blood Pressure Monitor device. All unmentioned attributes defined in [1] are not applicable to those devices.

3.3.2.1 SYSTOLIC/DIASTOLIC/MAP COMPOUND NUMERIC OBJECT (BLOOD PRESSURE)

11073 Attribute	*Bluetooth* equivalent characteristic	*Bluetooth* Service	*Bluetooth* Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Metric-Structure-Small	N/A	N/A	N/A	MetricStructure Small[4]	(INT-U8, INT-U8)
Metric-Id-List	N/A	N/A	N/A	MetricIdList[5]	List of (INT-U16)
Unit-Code	Blood Pressure Measurement	Blood Pressure	Aggregate	OID-Type[6]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[7]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Blood Pressure Measurement	Blood Pressure	Aggregate	AbsoluteTime[8]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Compound-Basic-Nu-Observed-Value	Blood Pressure Measurement	Blood Pressure	Aggregate	Compound-BasicNuObserved Value[9]	List of SFLOAT-Type

Table 14: 11073 Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value is set to {MDC_PART_SCADA | MDC_PRESS_BLD_NONINV}.
3. Set to 0xF040 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated).

4. Set to (0x03, 0x03) {ms-struct-compound-fix, 3}.
5. List set to {MDC_PRESS_BLD_NONINV_SYS, MDC_PRESS_BLD_NONINV_DIA, MDC_PRESS_BLD_NONINV_MEAN}.
6. This value is mapped from Bit 0 of the least significant octet of the Blood Pressure Measurement characteristic. The mapping is as follows in Table 15:

11073 Pressure Unit Value	Bluetooth Pressure Unit Value	Unit Description
MDC_DIM_MMHG	0	mmHg
MDC_DIM_KILO_PASCAL	1	kPa

Table 15: Pressure Type Description Conversion

7. See Section 2.3.
8. When supported, this value is derived from the Time Stamp field of Blood Pressure Measurement characteristic. See Section 2.2.6.
9. This is a list of exactly three values derived from Blood Pressure Measurement characteristic: Systolic pressure, Diastolic Pressure and MAP. If any of measures is temporarily unavailable, this condition is signaled with NaN special value.

3.3.2.2 PULSE RATE

Optionally transcoded when Blood Pressure Measurement contains Pulse Rate data.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (asn.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	N/A	N/A	N/A	OID-Type[4]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[5]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Blood Pressure Measurement (Time Stamp field)	Blood Pressure	Aggregate	AbsoluteTime[6]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Basic-Nu-Observed-Value	Blood Pressure Measurement (Pulse Rate field)	Blood Pressure	Aggregate	BasicNuObserved Value[7]	SFLOAT-Type

Table 16: 11073 Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_SCADA | MDC_PULS_RATE_NON_INV.
3. Set to 0xf040 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated).
4. Unit is MDC_DIM_BEAT_PER_MIN.

5. See Section 2.3.
6. When supported, this value is derived from the Time Stamp field of Blood Pressure Measurement characteristic. See Section 2.2.6.
7. This value is derived from Pulse Rate field of Blood Pressure Measurement characteristic, if available.

3.3.2.3 USER ID

The User ID data field in Blood Pressure Measurement characteristic value may be transcoded at the event report level by using multiple-person event reports e.g. ScanReportInfoMPFixed. Such reports transmit a Person-ID along with each event, which can accommodate the User ID value.

Person-ID is of INT-U16 type, while User ID from the characteristic is one octet, unsigned. When the User ID is in range 0x00 to 0xFE, conversion shall keep the numeric value. For example, User ID 0x02 would be transcoded to Person-ID 0x0002, and 0x85 would be transcoded to 0x0085.

When the User ID is 0xFF (unknown or guest user) it shall be transcoded to 0xFFFF (unknown-person-id).

If User-ID is absent in the characteristic value, the transcoder shall use the simple event report types (e.g. ScanReportInfoFixed). That is, absence of User-ID is **not** transcoded to unknown-person-id.

3.3.2.4 MEASUREMENT STATUS

The Measurement Status data field in the Blood Pressure Measurement characteristic is not transcodable.

3.3.2.5 BLOOD PRESSURE FEATURE

The Blood Pressure Feature characteristic of the Blood Pressure Service is not transcodable.

3.4 GLUCOSE METER

This section defines transcoding the glucose meter device-specific data into IEEE 11073-10417 Glucose Meter device specialization [12] class attributes.

The 10417 specialization specifies two ways to transmit data to manager: event reports or having a PM-Store. They are mutually exclusive; that is, if the configuration has a PM-Store, it shall not send measurement events.

The transcoding implementation must choose which method it will use to expose measurement data.

Using a PM-Store is strongly recommended by [12] section 6.9.1 because it better handles the (rather typical) situation of a device storing tens or hundreds of measurements. Moreover, RACP (Record Access Control Point) maps closely to PM-Store procedures.

3.4.1 DEVICE-SPECIFIC MDS CLASS REQUIREMENTS

In addition to the MDS class requirements shown in Section 2.1, Table 17 shows incremental MDS class requirements specific to this device.

Since the Bluetooth LE glucose sensor [14] does not provide a method to set date and time of its internal clock, the Set-Time action should not be supported for the MDS object.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
System-Type	N/A	N/A	N/A	TYPE[1]	(INT-U16, INT-U16)
Dev-Configuration-Id	N/A	N/A	N/A	ConfigId[2]	INT-U16
System-Type-Spec-List	N/A	N/A	N/A	TypeVerList[3]	List of (INT-U16, INT-U16)

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Confirm-Timeout[4]	N/A	N/A	N/A	RelativeTime	OCTET STRING (SIZE(6))

Table 17: Device-specific MDS Class Requirements

Notes:

1. Value is not present since System-Type-Spec-List exists.
2. Set to any value in the range of 0x4000 to 0x7FFF (Extended Configuration). This is to assure that the Manager requests the configuration and does not assume it, as it could for a Standard Configuration.
3. Since the "Glucose Service" [13] is a "Primary Service", the following entries are required to be added to the TypeVerList:

   0x10 0x11 Specialization value = MDC_DEV_SPEC_PROFILE_GLUCOSE
   0x00 0x02 version = version 2 of the 11073 device specialization

4. This is optional and informs the manager how much time the transcoding agent will wait for confirmation of event reports before transitioning to an Unassociated state.

3.4.2 11073 NUMERIC CLASS REQUIREMENTS

This section describes the 11073 numeric class requirements. It is restricted to those 11073 attributes that are used on the Glucose Sensor device. All unmentioned attributes defined in [1] are not applicable to the Glucose Sensor device.

3.4.2.1 BLOOD GLUCOSE OBJECTS

Each 11073 object has a static TYPE related to the type of blood sample (capillary plasma, control solution etc.).

A generic transcoder that wishes to use the same extended configuration for every transcoded device will need to present several blood glucose objects in its configuration – one per blood sample TYPE, making a total of ten (10) objects.

On the other hand, a transcoder that needs to support only a certain device model or a set of models, or creates a different extended configuration for each transcoded device, may present only the necessary objects. There will be at least two objects: the accepted blood type and the control solution.

In store-and-forward transcoding, the transcoder also possesses type information that makes it possible to enumerate only the necessary objects for optimized exchange.

Table 18 is the template structure for each of the ten blood glucose objects. Each object has a different TYPE.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	Glucose Measurement	Glucose	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	Glucose Measurement	Glucose	Aggregate	OID-Type[4]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[5]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[6]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Basic-Nu-Observed-Value	Glucose Measurement	Glucose	Aggregate	BasicNuObserved Value[7]	SFLOAT-Type

Table 18: 11073 Blood Glucose Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Each object has a different TYPE, of the ten types specified in [12]. They are related to measurements according to Table 19:

Type Nibble in Glucose Measurement	11073 Type
0x1 (capillary wholeblood)	{MDC_PART_SCADA | MDC_CONC_GLU_CAPILLARY_WHOLEBLOOD}
0x2 (capillary plasma)	{MDC_PART_SCADA | MDC_CONC_GLU_CAPILLARY_PLASMA}
0x3 (venous wholeblood)	{MDC_PART_SCADA | MDC_CONC_GLU_VENOUS_WHOLEBLOOD}
0x4 (venous plasma)	{MDC_PART_SCADA | MDC_CONC_GLU_VENOUS_PLASMA}
0x5 (arterial wholeblood)	{MDC_PART_SCADA | MDC_CONC_GLU_ARTERIAL_WHOLEBLOOD}
0x6 (arterial plasma)	{MDC_PART_SCADA | MDC_CONC_GLU_ARTERIAL_PLASMA}
0x7 (undetermined wholeblood)	{MDC_PART_SCADA | MDC_CONC_GLU_UNDETERMINED_WHOLEBLOOD}
0x8 (undetermined plasma)	{MDC_PART_SCADA | MDC_CONC_GLU_UNDETERMINED_PLASMA}
0x9 (interstitial fluid – ISF)	{MDC_PART_SCADA | MDC_CONC_GLU_ISF}
0xA (control solution)	{MDC_PART_SCADA | MDC_CONC_GLU_CONTROL}
0xF (type not available)	N/A

Table 19: Object Type relationship with measurement type

3. Set to 0xF040 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated).

4. This value is mapped from Bit 1 of the Flags field of the Glucose Measurement characteristic. The mapping is as follows:

11073 Glucose Unit Value	Bluetooth Glucose Unit Value	Unit Description
MDC_DIM_MILLI_G_PER_DL	0	mg/dL
MDC_DIM_MDC_DIM_MILLI_MOLE_PER_L	1	mmol/L

Table 20: Glucose: Glucose Unit Conversion

5. See Section 2.3.
6. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to the user at measurement time. See also Section 2.2.6.
7. This value is derived from Glucose Concentration value of Glucose Measurement characteristic. If measurement value is above device capabilities, value shall be +INFINITY. Symmetrically, if measurement is below device capabilities, value shall be –INFINITY. Both conditions can be detected in Sensor Status Annunciation Value Field of Glucose Measurement Characteristic, bits 5 and 6 respectively. Device may report a NaN or NRes value as well to signal some problem during measurement, which shall be transcoded as is.

The Sequence Number of Glucose Measurement Characteristic is not transcodable.

3.4.2.2 HEMOGLOBIN BOUND TO GLUCOSE A1C FORM

Optionally transcoded when Glucose Measurement Context characteristic is present and the Flags field of that characteristic signals (in bit 6) that HbA1c value is present.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	N/A	N/A	N/A	OID-Type[4]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[5]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[6]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Basic-Nu-Observed-Value	Glucose Measurement Context	Glucose	Aggregate	BasicNuObserved Value[7]	SFLOAT-Type

Table 21: 11073 HbA1c Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_SCADA | MDC_CONC_HBA1C.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).

4. Unit is MDC_DIM_PERCENT.
5. See Section 2.3.
6. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to the user at measurement time. See also Section 2.2.6.
7. This value is derived from HbA1c value of Glucose Measurement Context characteristic.

3.4.2.3 CONTEXT EXERCISE

Optionally transcoded when Glucose Measurement Context characteristic is present and the Flags field of that characteristic signals (in bit 3) that exercise value is present.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth data type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Unit-Code	N/A	N/A	N/A	OID-Type[4]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[5]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[6]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Measure-Active-Period	Glucose Measurement Context	Glucose	Aggregate	FLOAT-Type[7]	FLOAT-Type
Basic-Nu-Observed-Value	Glucose Measurement Context	Glucose	Aggregate	BasicNuObserved Value[8]	SFLOAT-Type

Table 22: 11073 Context Exercise Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_CTXT_GLU_EXERCISE.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. Unit is MDC_DIM_PERCENT.
5. See Section 2.3.
6. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.

7. This value is derived from Exercise Duration value of Glucose Measurement Context characteristic. Value is converted to FLOAT with an exponent of 0.
8. This value is derived from Exercise Intensity value of Glucose Measurement Context characteristic. Value is converted to SFLOAT with an exponent of 0.

3.4.2.4 CONTEXT MEDICATION

Optionally transcoded when Glucose Measurement Context characteristic is present and the Flags field of that characteristic signals (in bit 4) that medication value is present.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Metric-Id	Glucose Measurement Context	Glucose	Aggregate	OID-Type[4]	INT-U16
Unit-Code	Glucose Measurement Context	Glucose	Aggregate	OID-Type[5]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[6]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[7]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Basic-Nu-Observed-Value	Glucose Measurement Context	Glucose	Aggregate	BasicNuObserved Value[8]	SFLOAT-Type

Table 23: 11073 Context Medication Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value is set to MDC_PART_PHD_DM | MDC_CTXT_MEDICATION.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. Value is set based on Medication ID Value field of Glucose Measurement Context characteristic, accordingly to the following table:

Medication ID	11073 Value
0x1 (rapid action insulin)	MDC_CTXT_MEDICATION_RAPIDACTING
0x2 (short acting insulin)	MDC_CTXT_MEDICATION_SHORTACTING
0x3 (intermediate acting insulin)	MDC_CTXT_MEDICATION_INTERMEDIATEACTING
0x4 (long acting insulin)	MDC_CTXT_MEDICATION_LONGACTING
0x5 (pre-mixed insulin)	MDC_CTXT_MEDICATION_PREMIX

Table 24: Medication ID and Metric ID relationship

5. Unit is set based on Medication Units Flag (bit 5 of Flags of the Glucose Measurement Context characteristic), accordingly to the following table:

Medication Units Flag	11073 Value	Unit
0x0	MDC_DIM_MILLI_G	mg
0x1	MDC_DIM_MILLI_L	ml

Table 25: Medication Units Flag and Unit relationship

6. See Section 2.3.
7. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
8. This value is derived from Medication Value of Glucose Measurement Context characteristic.

3.4.2.5 CONTEXT CARBOHYDRATES

Optionally transcoded when Glucose Measurement Context characteristic is present and the Flags field of that characteristic signals (in bit 0) that carbohydrates value is present.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Metric-Id	Glucose Measurement Context	Glucose	Aggregate	OID-Type[4]	INT-U16
Unit-Code	N/A	N/A	N/A	OID-Type[5]	INT-U16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[6]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[7]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Basic-Nu-Observed-Value	Glucose Measurement Context	Glucose	Aggregate	BasicNuObserved Value[8]	SFLOAT-Type

Table 26: 11073 Context Carbohydrates Numeric Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_CTXT_GLU_CARB.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. Value is set based on Carbohydrate ID Value field of Glucose Measurement Context characteristic, accordingly to the following table:

Carbohydrate ID	11073 Value
0x1 (breakfast)	MDC_CTXT_GLU_CARB_BREAKFAST
0x2 (lunch)	MDC_CTXT_GLU_CARB_LUNCH

Carbohydrate ID	11073 Value
0x3 (dinner)	MDC_CTXT_GLU_CARB_DINNER
0x4 (snack)	MDC_CTXT_GLU_CARB_SNACK
0x5 (drink)	MDC_CTXT_GLU_CARB_DRINK
0x6 (supper)	MDC_CTXT_GLU_CARB_SUPPER
0x7 (brunch)	MDC_CTXT_GLU_CARB_BRUNCH

Table 27: Carbohydrate ID and Metric Id relationship

5. Unit is MDC_DIM_X_G.
6. See Section 2.3.
7. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
8. This value is derived from Carbohydrate Value of Glucose Measurement Context characteristic.

3.4.3 ENUMERATION OBJECTS

This section describes the 11073 enumeration class requirements. It is restricted to those 11073 attributes that are used on the Glucose Sensor device. All unmentioned attributes defined in [1] are not applicable to those devices.

3.4.3.1 DEVICE AND SENSOR ANNUNCIATION

Optionally transcoded when Glucose Measurement characteristic contains the Sensor Status Annunciation Value field.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[4]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[5]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Enum-Observed-Value-Basic-Bit-Str	Glucose Measurement	Glucose	Aggregate	BITS-16[6]	BITS-16

Table 28: 11073 Device and Sensor Annunciation Enumeration Class Requirements

Notes

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_MDC_GLU_METER_DEV_STATUS
3. Set to 0xF040 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated).
4. See Section 2.3.

5. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
6. This value is derived from Sensor Status Annunciation Value Field of the Glucose Measurement characteristic. Bits with value 1 mean that the respective failure mode has happened at measurement time. The bitmap is composed accordingly Table 29:

11073 Device Or Sensor Condition Bit (GlucoseDevStat) Bit 0 is MSB	Bluetooth Sensor Status Annunciation Bit Bit 0 is LSB
device-battery-low (0)	0 (device battery low)
sensor-malfunction (1)	1 (sensor malfunction)
sensor-sample-size-insufficient (2)	2 (sample size insufficient, not enough blood or control solution)
sensor-strip-insertion (3)	3 (strip insertion error)
sensor-strip-type-incorrect (4)	4 (strip type is incorrect)
sensor-result-too-high (5)	5 (sensor result higher than device can process)
sensor-result-too-low (6)	6 (sensor result lower than device can process)
sensor-temp-too-high (7)	7 (ambient temperature too high for a valid test/result)
sensor-temp-too-low (8)	8 (ambient temperature too low for a valid test/result)
sensor-read-interrupt (9)	9 (reading was interrupted and/or strip was pulled too soon)
device-gen-fault (10)	10 (general device fault)
No correspondence	11 (time fault)

Table 29: Relationship between Sensor Status Annunciation and 11073 sensor condition

For example, if the battery was low and on top of that the user pulled the strip too soon, the sensor status value would be 0x0201 (bits 0 and 9 LSB set), and the 11073 transcoded value would be 0x8040 (bits 0 and 9 MSB set).

3.4.3.2 CONTEXT MEAL

Optionally transcoded when Glucose Measurement Context characteristic is present and the respective Flags field indicates that Meal Value field is present.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[4]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[5]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Enum-Observed-Value-Simple-OID	Glucose Measurement Context	Glucose	Aggregate	OID-Type[6]	INT-U16

Table 30: 11073 Context Meal Enumeration Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_CTXT_GLU_MEAL.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. See Section 2.3.
5. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
6. This value is derived from Meal Value Field of the Glucose Measurement Context characteristic, accordingly to the table below:

11073 Nomenclature Value	Bluetooth Meal Value
MDC_CTXT_GLU_MEAL_PREPRANDIAL	0x1 (preprandial – before meal)
MDC_CTXT_GLU_MEAL_POSTPRANDIAL	0x2 (postprandial – after meal)
MDC_CTXT_GLU_MEAL_FASTING	0x3 (fasting)
MDC_CTXT_GLU_MEAL_CASUAL	0x4 (casual – snacks, drinks etc.)
MDC_CTXT_GLU_MEAL_BEDTIME	0x5 (bedtime)

Table 31: Relationship between Bluetooth Meal value and context meal enumeration

3.4.3.3 CONTEXT SAMPLE LOCATION

Optionally transcoded when Glucose Measurement characteristic contains the Type/Sample Location Value field, and the Sample nibble value of this field is different from 0xF (0xF means sample location not available).

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[4]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[5]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Enum-Observed-Value-Simple-OID	Glucose Measurement	Glucose	Aggregate	OID-Type[6]	INT-U16

Table 32: 11073 Context Sample Location Enumeration Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_CTXT_GLU_SAMPLELOCATION
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. See section 2.3.
5. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
6. This value is derived from Sample Location nibble in Type/Sample Location Value Field of the Glucose Measurement characteristic, accordingly to the table below:

11073 Nomenclature Value	Bluetooth Sample Location Nibble Value
MDC_CTXT_GLU_SAMPLELOCATION_FINGER	0x1 (finger)
MDC_CTXT_GLU_SAMPLELOCATION_AST	0x2 (alternate site test)
MDC_CTXT_GLU_SAMPLELOCATION_EARLOBE	0x3 (earlobe)
MDC_CTXT_GLU_SAMPLELOCATION_CTRLSOLUTION	0x4 (control solution)

Table 33: Relationship between Bluetooth Sample Location and location nomenclature

3.4.3.4 CONTEXT TESTER

Optionally transcoded when Glucose Measurement Context characteristic is present, the respective Flags field indicates that Tester/Health field is present, and Tester nibble value of this field is different from 0xF (0xF means health value not available).

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[4]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[5]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Enum-Observed-Value-Simple-OID	Glucose Measurement Context	Glucose	Aggregate	OID-Type[6]	INT-U16

Table 34: 11073 Context Tester Enumeration Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_CTXT_GLU_TESTER.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. See section 2.3.
5. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
6. This value is derived from Tester nibble in Tester/Health Value Field of the Glucose Measurement Context characteristic, accordingly to the table below:

11073 Nomenclature Value	Bluetooth Tester Nibble Value
MDC_CTXT_GLU_TESTER_SELF	0x1 (self)
MDC_CTXT_GLU_TESTER_HCP	0x2 (health care professional)
MDC_CTXT_GLU_TESTER_LAB	0x3 (lab test)

Table 35: Relationship between Bluetooth Tester value and tester nomenclature value

3.4.3.5 CONTEXT HEALTH

Optionally transcoded when Glucose Measurement Context characteristic is present, the respective Flags field indicates that Tester/Health field is present, and Health nibble value of this field is different from 0xF (0xF means health value not available).

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
Type	N/A	N/A	N/A	TYPE[2]	(INT-U16, INT-U16)
Metric-Spec-Small	N/A	N/A	N/A	MetricSpecSmall[3]	BITS-16
Attribute-Value-Map	N/A	N/A	N/A	AttrValMap[4]	List of (INT-U16, INT-U16)
Absolute-Time-Stamp	Glucose Measurement	Glucose	Aggregate	AbsoluteTime[5]	(INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8, INT-U8)
Enum-Observed-Value-Simple-OID	Glucose Measurement Context	Glucose	Aggregate	OID-Type[6]	INT-U16

Table 36: 11073 Context Health Enumeration Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Value set to MDC_PART_PHD_DM | MDC_CTXT_GLU_HEALTH.
3. Set to 0xF048 (mss-avail-intermittent | mss-avail-stored-data | mss-upd-aperiodic | mss-msmt-aperiodic | mss-acc-agent-initiated | mss-cat-manual).
4. See section 2.3.
5. This value is derived from the mandatory Base Time field of Glucose Measurement characteristic (which is mandatory), added by the optional Time Offset field of the same characteristic. When measurements are sent via RACP, the first measurement always includes the Time Offset field. Time Offset is a signed integer number of minutes added to Base Time to determine the face time shown to user at measurement time. See also Section 2.2.6.
6. Derived from Health nibble in Tester/Health Value Field of the Glucose Measurement Context characteristic, accordingly to the table below:

11073 Nomenclature Value	Bluetooth Health Nibble Value
MDC_CTXT_GLU_HEALTH_MINOR	0x1 (minor health issues)
MDC_CTXT_GLU_HEALTH_MAJOR	0x2 (major health issues)
MDC_CTXT_GLU_HEALTH_MENSES	0x3 (menses)
MDC_CTXT_GLU_HEALTH_STRESS	0x4 (under stress)
MDC_CTXT_GLU_HEALTH_NONE	0x5 (no health issues)

Table 37: Relationship between Bluetooth Health nibble in Tester/Health Value Field and Context Health nomenclature

3.4.4 PM-STORE

Bluetooth LE glucose sensors implement the RACP (Record Access Control Point) feature, which allows for selective retrieval and deletion of measurement records. RACP is analogous to PM-Store in this regard.

This section discusses the layout of PM-Store objects in the 11073 configuration, and how to adapt PM-Store to RACP for glucose.

3.4.4.1 MAPPING GET-SEGMENT-INFO TO RACP IN LIVE TRANSCODING

In spite of being analogous, PM-Store *Get-Segment-Info* cannot be mapped directly to RACP operations, because of Time Offset.

Each Bluetooth LE measurement contains a Base Time and an optional Time Offset. Time Offset is always present in the first transmitted measurement. The combination of the two is the user-facing time. The 11073 measurement objects contain only an Absolute-Time attribute filled with user-facing time. In the other hand, each PM-Segment contains a Segment-Date-and-Time-Adjustment attribute.

In order not to lose Time Offset information in the transcoding process, every continuous block of measurements (ordered by sequence number) with the same Time Offset shall go to a separate PM-Segment. The transcoder needs to fetch all measurements beforehand to analyze data and create the necessary segments.

3.4.4.2 MAPPING CLEAR-SEGMENTS TO RACP IN LIVE TRANSCODING

The method *Clear-Segments* can be mapped to RACP *Delete-Stored-Records* operation directly, and the clear operation can be transcoded live.

RACP has some filtering options that map to *SegmSelection*. But support to every filter is optional, and cannot be discovered except by trying the command and detecting if it fails. Note that *Pm-Store-Capab* must be filled beforehand, and cannot change.

Because of this, a generic transcoder that does not implement model-specific behavior should support *Clear-Segments* with segment selection *all-segments* choice only.

A transcoder that supports specific glucose sensor models, and knows their RACP capabilities, may implement other choices of segment selection.

In selection *all-segments*, the segments should be removed completely, not only emptied.

3.4.4.3 PM-STORE ATTRIBUTES

The transcoder shall present exactly one PM-Store in its configuration. This section specifies how the PM-Store atributes are filled.

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Handle	N/A	N/A	N/A	HANDLE[1]	INT-U16
PM-Store-Capab	N/A	N/A	N/A	PmStoreCapab[2]	INT-U16
Store-Sample-Algorithm	N/A	N/A	N/A	StoSampleAlg[3]	INT-U16
Store-Capacity-Count	N/A	N/A	N/A	INT-U32[4]	INT-U32
Store-Usage-Count	N/A	N/A	N/A	INT-U32[5]	INT-U32
Operational-State	N/A	N/A	N/A	OperationalState[6]	INT-U16
PM-Store-Label	N/A	N/A	N/A	Octet string[7]	Octet string
Number-Of-Segments	N/A	N/A	N/A	INT-U16[8]	INT-U16
Clear-Timeout	N/A	N/A	N/A	RelativeTime[9]	INT-U32

Table 38: Device-specific MDS Class Requirements

Notes:

1. Each object is required to have a unique non-zero ID assigned by the implementation.
2. Set to 0x8840 (pmsc-var-no-of-segm, pmsc-epi-seg-entries, pmsc-clear-segm-remove).
3. Set to 0.

4. Set to 655350. This value is big enough to fit all segment entries, in every combination of measurements and contexts.
5. Set to actual number of entries in all segments.
6. Set to 0x01 (OS_ENABLED).
7. Set to “Transcoded offline RACP records”.
8. Set to actual number of segments. See the following sections to determine the number of segments.
9. Defined by implementation.

3.4.4.4 PM-SEGMENTS

There shall be at least one instantiated PM-Segment type per numeric and enumeration object. This could be as many as nineteen (19) segment instantiations (up to ten glucose objects, one per blood type, plus nine context objects).

Actual segment count may be less than 19 (may be even 0) because there may be no stored entries for some (or most) segment types.

The segment count may be higher than 19 due to time offset changes. Objects of each class shall be ordered by Bluetooth LE Sequence Number and then grouped by Bluetooth LE Time Offset. Each grouped block shall be put into a separate PM-Segment.

This grouping scheme avoids the loss of Time Offset information in the transcoding process, but it implies recovering all records in Bluetooth LE device via RACP beforehand. Manager methods on PM-Segments cannot be translated to live (real-time) RACP requests under this scheme.

Note that Time Offset field is not required to be present in every Bluetooth LE measurement; it may be present only when Time Offset changes. Bluetooth LE measurements without Time Offset inherit the offset of the last measurement with a Time Offset field, and are grouped under the same PM-Segment.

3.4.4.5 PM-SEGMENTS COMMON ATTRIBUTES

The following PM-Segment atributes are found in every segment and shall be filled as specified:

11073 Attribute	Bluetooth Equivalent Characteristic	Bluetooth Service	Bluetooth Data Type	11073 Attribute Type (ASN.1)	11073 Data Type (informative)
Instance-Number	N/A	N/A	N/A	InstNumber[1]	INT-U16
PM-Segment-Entry-Map	N/A	N/A	N/A	PmSegment EntryMap[2]	PmSegment EntryMap
Operational-State	N/A	N/A	N/A	OperationalState[3]	INT-U16
Segment-Label	N/A	N/A	N/A	Octet string[4]	Octet string
Segment-Start-Abs-Time	N/A	N/A	N/A	AbsoluteTime[5]	Array of 8 x INT-U8
Segment-End-Abs-Time	N/A	N/A	N/A	AbsoluteTime[6]	Array of 8 x INT-U8
Segment-Date-and-Time-Adjustment	N/A	N/A	N/A	AbsoluteTime Adjust[7]	Array of 6 x INT-U8 (containing a INT-I48)
Segment-Usage-Count	N/A	N/A	N/A	INT-U32[8]	INT-U32
Confirm-Timeout	N/A	N/A	N/A	RelativeTime[9]	INT-U32
Transfer-Timeout	N/A	N/A	N/A	RelativeTime[10]	INT-U32

Table 39: Device-specific MDS Class Requirements

Notes:

1. Each segment must be given an instance number unique within the PM-Store.
2. See item 2.4. Each segment shall have a map congruent to the numeric or enumeration object related to the entries.
3. Set to 0x01 (OS_ENABLED).
4. Set to “Transcoded segment <type> <timeadjust>” where <type> is a human-readable name of the type of entry stored in this segment, and <timeadjust> is the value of original Bluetooth LE Time Offset (positive or negative).
5. Set to user-facing time of oldest entry.
6. Set to user-facing time of newest entry;
7. Value of Bluetooth LE Time Offset multiplied by 6000 (since Time Offset is in minutes and AbsoluteTimeAdjust is in hundredths of a second).
8. Set to actual number of entries.
9. Set to zero (transcoded PM-Store does not support *Clear-Segments*).
10. Set to a reasonable value, enough to transfer all entries (timeout unit is 125 microseconds).

4. End-To-End Example

This section provides an example of an end-to-end communication between a *Bluetooth* low energy Health Thermometer and a Collector (e.g., phone) implementing an 11073 Manager and a transcoder. This section also describes how *Bluetooth* characteristic data can be mapped to 11073 nomenclature and modeling. This example illustrates the steps required to use the mappings so that the transcoder can generate 11073 APDUs based on the received data.

Hypothetical Health Thermometer data is used in Section 4.1 as an input. Section 4.3 discusses how this data could be mapped into 11073 objects.

4.1 HEALTH THERMOMETER DATA

Table 40 and Table 41 describe the Health Thermometer data being sent to the Collector, which implements a Transcoder. The Health Thermometer *Bluetooth* Address is 00:23:6C:AF:BD:F4.

Health Thermometer Service Data

This data refers to a previous time-stamped measurement taken from a general body location.

Bluetooth Characteristic	*Bluetooth* Value
Date Time	18th December 2010 15:23:06 encoded as 0x07DA 0x0C 0x12 0x0F 0x17 0x06
Temperature Measurement	37.0 degrees Celsius with Timestamp of 18th December 2010 15:00:00: 0x02 0xFF 0x00 0x01 0x72 0x07 0xDA 0x0C 0x12 0x0F 0x00 0x00
Temperature Type	Body: 0x02
Intermediate Temperature	Not transcoded to 11073.
Measurement Interval	Not transcoded to 11073.
Valid Range Descriptor	Not transcoded to 11073.

Table 40: Health Thermometer Service Data

Device Information Service Data

This data refers to a Health Thermometer that has its System ID filled in based on its *Bluetooth* address as described in the characteristic definition accessible via the Bluetooth SIG Assigned Numbers [4].

Bluetooth Characteristic	*Bluetooth* Value
System ID	0x00 0x23 0x6C 0xFF 0xFE 0xAF 0xBD 0xF4
Model Number String	“TS-1017”
Manufacturer Name String	“ACME”
Serial Number String	“237495-3282-A”
Firmware Revision String	“1.23”
Hardware Revision String	“1.0”
Software Revision String	“1.2”
IEEE 11073-20601 Regulatory Certification Data List	For an example on how to populate this structure, refer to Section 2.2.5.

Table 41: Device Information Service data

4.2 HEALTH THERMOMETER SERVICE RECORD

Table 42 shows the Health Thermometer Service record and the attributes contained on the server.

Bluetooth Attribute	*Bluetooth* Attribute Value	Description
Primary Service (0x2800)	**0x180A**	**Device Information Service**
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A23}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "System ID"
System ID (0x2A23)	{0x00236CFFFEAFBDF4}	System ID
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A29}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "Manufacturer Name String"
Manufacturer Name String (0x2A29)	{0x41, 0x43, 0x4D, 0x45}	Manufacturer Name String = UTF-8 String "ACME"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A24}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "Model Number String"
Model Number String (0x2A24)	{0x54, 0x53, 0x2D, 0x31, 0x30, 0x31, 0x37}	Model Number String = UTF-8 String "TS-1017"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A25}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "Serial Number String"
Serial Number String (0x2A25)	{0x32, 0x33, 0x37, 0x34, 0x39, 0x35, 0x2D, 0x33, 0x32, 0x38, 0x32, 0x2D, 0x41}	Serial Number String = UTF-8 String "237495-3282-A"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A26}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "Firmware Revision String"
Firmware Revision String (0x2A26)	{0x31, 0x2E, 0x32, 0x33}	Firmware Revision String = UTF-8 String "1.23"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A27}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "Hardware Revision"
Hardware Revision String (0x2A27)	{0x31, 0x2E, 0x30}	Hardware Revision String = UTF-8 String "1.0"
Characteristic (0x2803)	{0x02, 0xhhhh, 0x2A28}	Characteristic Value is Read, Value Handle reference is 0xhhhh, and Characteristic is "Software Revision String"
Serial Number String (0x2A28)	{0x31, 0x2E, 0x32}	Software Revision String = UTF-8 String "1.2"
Primary Service (0x2800)	**0x1809**	**Health Thermometer Service**
Characteristic (0x2803)	0x20, 0xhhhh, 0x2A1C	Characteristic Value is Indicated, Value Handle reference is 0xhhhh and Characteristic is "Temperature Measurement"
Temperature Measurement (0x2A1C)	{0x02, 0xFF000172, 0x07DA 0x0C 0x12 0x0F 0x17 0x06}	Timestamp [18th December 2010 15:23:06] Temperature Measurement of 37.0 degrees in Celsius

Table 42: Health Thermometer Service Record

4.3 11073 OBJECTS

The following tables describe how data are represented as 11073 objects.

MDS Object

11073 Attribute	*Bluetooth* Equivalent Characteristic	11073 Value	Reference
Handle	None	0	
System-Model	Model Number String	"TS-1017" [1]	Section 2.2.4 – System Model
	Manufacturer Name String	"ACME"	
System-Id	System ID	0x00 0x23 0x6C 0xFF 0xFE 0xAF 0xBD 0xF4	Section 2.2.4 – System Id
Dev-Configuration-Id	None	MDC_TEMP_BODY = 0x4002	Section 3.1.1
Production-Specification	Serial Number String	"237495-3282-A" [1]	Section 2.2.4 – Prod. Specification
	Hardware Revision String	"1.0" [1]	
	Software Revision String	"1.2" [1]	
	Firmware Revision String	"1.23"	
Date-and-Time	Date Time	0x20 0x10 0x0C 0x12 0x0F 0x17 0x06 0x00	Section 2.2.6
Reg-Cert-Data-List	IEEE 11073-20601 Regulatory Certification Data List	For an example on how to populate this structure, refer to Section 2.2.5.	Section 2.2.4 – Reg-Cert-Data-List
System-Type-Spec-List	None	{MDC_DEV_SPEC_PROFILE_TEMP, 1}	

Table 43: MDS Object for Health Thermometer

Notes:

1. Because this is an odd-sized string, a zero (0x00) byte must be appended to its end and its length field must be incremented. See Section 2.2.3 for more information.

Numeric Object

11073 Attribute	*Bluetooth* Equivalent Characteristic	Value	Reference
Handle	None	1	
Type	None	MDC_PART_SCADA	Section 3.1.1
	None	MDC_TEMP_BODY	
Metric-Spec-Small	None		
Measurement-Status	None	relevant data, 0x10	
Metric-Id	None	MDC_TEMP_BODY = 0x4002	Section 3.1.2
Metric-Id-List	None		
Metric-Id-Partition	None		
Unit-Code	Temperature Measurement	MDC_DIM_DEGC	Section 3.1.2
Attribute-Value-Map	None		
Absolute-Time-Stamp	Temperature Measurement	0x20 0x10 0x0C 0x12 0x0F 0x00 0x00	Section 3.1.2
Measure-Active-Period	None		
Simple-Nu-Observed-Value	None	0xFF000172 (37.0)	Section 3.1.2 and Annex F.8 of [1]
Compound-Simple-Nu-Observed-Value	None		
Basic-Nu-Observed-Value	None		
Compound-Basic-Nu-Observed-Value	None		
Nu-Observed-Value	Temperature Measurement		
Compound-Nu-Observed-Value	None		
Accuracy	None		

Table 44: Numeric Object for Health Thermometer

5. Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
APDU	Application Protocol Data Unit
ASCII	American Standard Code for Information Interchange as defined in ISO/IEC 646 (1991)
ASN.1	Abstract Syntax Notation One
BCD	Binary-Coded Decimal
DIS	Device Information Service
EUI	Extended Unique Identifier
FDA	Food and Drug Administration
GATT	Generic Attribute Profile
IEEE	Worldwide technical society which generated the IEEE 11073 series standards, www.ieee.org
ISO	International Organization for Standardization
LE	Low Energy
MDS	Medical Device System
RACP	Record Access Control Point
USB	Universal Serial Bus
UTF-8	Unicode Transformation Format-8
UUID	Universally Unique Identifier

Table 45: Acronyms and Abbreviations

6. References

[1] ISO/IEEE Std 11073-20601™- 2008 Health Informatics - Personal Health Device Communication - Application Profile - Optimized Exchange Protocol - version 1.0.This also includes ISO/IEEE Std 11073-20601a™-2010 – Amendment 1

[2] Device Information Service V10r00

[3] Health Thermometer Profile V10r00

[4] Characteristic descriptions are accessible via the Bluetooth SIG Assigned Numbers.

[5] ISO/IEEE Std 11073-10408-2008 Standard for Health informatics - Personal health device communication - Device specialization - Thermometer

[6] Health Thermometer Service V10r00

[7] Continua Design Guidelines 2010 (version 1.5)

[8] IEEE Std 11073-10406-2011™ Standard for Health informatics - Personal health device communication - Device specialization - Basic Electrocardiograph (ECG) (1 to 3-lead ECG)

[9] Heart Rate Service V10r00

[10] Blood Pressure Service V10r00

[11] IEEE Std 11073-10407-2008™ Standard for Health informatics – Personal health device communication – Device specialization – Blood Pressure Monitor

[12] IEEE Std 11073-10417-2011™ Standard for Health Informatics – Personal health device communication – Device specialization – Glucose meter

[13] Glucose Service V10r00

[14] Glucose Profile V10r00

[15] Bluetooth Core Specification v4.0 or later