Industrial Measurement Device Service

***Bluetooth*® Service Specification**

- **Version:** v1.0
- **Version Date:** 2024-10-15
- **Prepared By:** Automation Working Group

Abstract:

The Industrial Measurement Device Service (IMDS) provides a standard way for wireless industrial measurement devices such as smart tool holders, smart clamping chucks, and other measurement devices to communicate real-time and historical measurement data with any Bluetooth-enabled machine tool control system.

Version History

Version Number	Date (yyyy-mm-dd)	Comments
v1.0	2024-10-15	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Benjamin Wieder	WTO Werkzeug-Einrichtungen GmbH
Matthias Kirn	WTO Werkzeug-Einrichtungen GmbH
Philipp Sommer	HAINBUCH GMBH SPANNENDE TECHNIK
Katrin Schindler	Hoffmann SE
Victor Zhodzishsky	Infineon Technologies AG

Contents

1 Introduction

The machine tools used in smart machine factories today (e.g., milling centers and turning centers or combinations of both) are highly automated and offer a variety of services to tool operators, the factory control system, and maintenance services.

Tool operators and service technicians need to monitor and control the condition of these tools' internal components and peripheral devices to optimize processes, determine predictive maintenance services, and identify possible defects.

Tool operators and service technicians can control the condition of the internal components of these machine tools, but only if the tool is physically connected by cable to the tool's computer-based control system.

IMDS allows peripheral devices like tool holders, clamping chucks, and measurement devices to report their state and other measurements like acceleration, force, or temperature to a Bluetooth-enabled machine tool control system by using a wireless data connection.

1.1 Language

1.1.1 Language conventions

In the development of a specification, the Bluetooth SIG has established the following conventions for use of the terms *"shall"*, *"shall not"*, *"should"*, *"should not"*, *"may"*, *"must"*, and *"can"*. In this Bluetooth specification, the terms in Table 1.1 have the specific meanings given in that table, irrespective of other meanings that exist.

Term	Definition
shall	—used to express what is required by the specification and is to be implemented exactly as written without deviation
shall not	—used to express what is forbidden by the specification
should	—used to express what is recommended by the specification without forbidding anything
should not	—used to indicate that something is discouraged but not forbidden by the specification
may	—used to indicate something that is permissible within the limits of the specification
must	—used to indicate either: 1. an indisputable statement of fact that is always true regardless of the circumstances 2. an implication or natural consequence if a separately-stated requirement is followed
can	—used to express a statement of possibility or capability

Table 1.1: Language conventions terms and definitions

1.1.1.1 Implementation alternatives

When specification content indicates that there are multiple alternatives to satisfy specification requirements, if one alternative is explained or illustrated in an example it is not intended to limit other alternatives that the specification requirements permit.

1.1.1.2 Discrepancies

It is the goal of Bluetooth SIG that specifications are clear, unambiguous, and do not contain discrepancies. However, members can report any perceived discrepancy by filing an erratum and can request a test case waiver as appropriate.

1.1.2 Reserved for Future Use

Where a field in a packet, Protocol Data Unit (PDU), or other data structure is described as "Reserved for Future Use" (irrespective of whether in uppercase or lowercase), the device creating the structure shall set its value to zero unless otherwise specified. Any device receiving or interpreting the structure shall ignore that field; in particular, it shall not reject the structure because of the value of the field.

Where a field, parameter, or other variable object can take a range of values, and some values are described as "Reserved for Future Use," a device sending the object shall not set the object to those values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous; however, this does not apply in a context where the object is described as being ignored or it is specified to ignore unrecognized values.

When a field value is a bit field, unassigned bits can be marked as Reserved for Future Use and shall be set to 0. Implementations that receive a message that contains a Reserved for Future Use bit that is set to 1 shall process the message as if that bit was set to 0, except where specified otherwise.

The acronym RFU is equivalent to Reserved for Future Use.

1.1.3 Prohibited

When a field value is an enumeration, unassigned values can be marked as “Prohibited.” These values shall never be used by an implementation, and any message received that includes a Prohibited value shall be ignored and shall not be processed and shall not be responded to.

Where a field, parameter, or other variable object can take a range of values, and some values are described as “Prohibited,” devices shall not set the object to any of those Prohibited values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous.

“Prohibited” is never abbreviated.

1.2 Table requirements

Requirements in this section are defined as "Mandatory" (M), "Optional" (O), "Excluded" (X), “Not Applicable” (N/A), or "Conditional" (C.n). Conditional statements (C.n) are listed directly below the table in which they appear.

1.3 Conformance

Each capability of this specification shall be supported in the specified manner. This specification may provide options for design flexibility, because, for example, some products do not implement every portion of the specification. For each implementation option that is supported, it shall be supported as specified.

1.4 Byte transmission order

All characteristics used with this service shall be transmitted with the least significant octet (LSO) first (i.e., little endian). Where the format is described in tables in this document, the LSO is the first octet in the topmost field of the table; the most significant octet (MSO) is the last octet in the bottommost field of

the table. Where characteristics are defined in the GATT Specification Supplement (GSS), see Section 2.2 in GSS [1].

1.5 Terminology

Table 1.2 lists terms defined in this document.

Term	Definition
Industrial Measurement Device	The Industrial Measurement Device (IMD) is a measurement device that is mounted in or near a machine tool and is the target device of IMDS.
Life Cycle	The Life Cycle Data characteristic provides values describing the life cycle data defined by the manufacturer. The life cycle is the interval between manufacturing and scrapping a device.
Service Cycle	The Service Cycle Data characteristic provides values describing the service cycle of the IMD. A service cycle is the interval between two maintenance operations.
Work Cycle	A work cycle is the process of enclosing a single IMD operation (e.g., drilling one hole into a workpiece).
Yellow Limit	Manufacturer-defined high Yellow Limit and low Yellow Limit indicate that the IMD is not operating under optimal manufacturer-defined conditions. User-defined high Yellow Limit and low Yellow Limit values indicate that the process is being performed outside of the optimal user-defined conditions. The color coding is inherited by commonly used stack lights (see also [5]).
Red Limit	Manufacturer-defined high Red Limit and low Red Limit indicate that the IMD is operating under harmful manufacturer-defined conditions. User-defined high Red Limit and low Red Limit values indicate that the process is being performed outside of the user-defined conditions. The color coding is inherited by commonly used stack lights (see also [5]).
Green Zone	The Green Zone describes the operating area within which, in the context of the specific measured physical property, the IMD is functioning under optimal conditions. The color coding is inherited by commonly used stack lights (see also [5]).
Yellow Zone	The Yellow Zone (Warning Zone) describes the operating area within which the IMD is not operating under optimal conditions. The color coding is inherited by commonly used stack lights (see also [5]).

Term	Definition
Red Zone	The Red Zone (Error Zone) describes the operating area within which the IMD is operating under harmful conditions. The color coding is inherited by commonly used stack lights (see [5]).
Target Value	The target value is the measured value defined by the user that should be reached in the measuring process.
Custom Condition	A Custom Condition is a manufacturer-defined condition when measurement is performed. The condition is either automatically enabled or enabled offline and depends on the IMD internal design and functionality.

Table 1.2: Terminology

2 Industrial Measurement Device Service

2.1 Service dependencies

IMDS is not dependent upon any other services.

2.2 Bluetooth Core Specification release compatibility

This specification is compatible with any Bluetooth Core Specification that includes the Generic Attribute Profile (GATT) [4].

2.3 Transport dependencies

There are no transport-related dependencies.

2.4 Attribute Profile error codes

IMDS references the Common Profile and Service Error Codes that are defined in Part B, Section 1 in the Core Specification Supplement (CSS) [2]. IMDS also defines the Attribute Protocol Application Error codes listed in Table 2.1.

Name	Error Code	Description
Invalid Time	0x80	Client requested an operation specifying a time in the past.
Time is Not Set	0x81	Client requested an operation that needs to be executed at a specific time and IMD time is not set.

Table 2.1: Attribute Protocol Application error codes defined by IMDS

2.5 GATT sub-procedure requirements

Requirements in this section represent a minimum set of requirements for a server. Other GATT sub-procedures may be used if supported by both client and server.

Table 2.2 summarizes additional GATT sub-procedure requirements beyond those required by all GATT Servers, as defined in Table 4.1 in Volume 3, Part G, Section 4.2 in the Bluetooth Core Specification [4].

GATT Sub-Procedure	Requirements
Characteristic Value Notification	M
Characteristic Value Indication	C.1
Read Characteristic Descriptors	M
Write Characteristic Descriptors	C.2
Write Characteristic Value	C.3

Table 2.2: GATT sub-procedure requirements

C.1: Mandatory if any of the service characteristics (see Section 3) supports indications, otherwise Excluded.

C.2: Mandatory if any of the service characteristic descriptors is writable, otherwise Excluded.

C.3: Mandatory if the service contains writable characteristics, otherwise Excluded.

2.6 Declaration

IMDS should be instantiated as a «Primary Service».

The service UUID (Universal Unique Identifier) shall be set to «Industrial Measurement Device Service» as defined in the Bluetooth Assigned Numbers [3].

A device should include only one instance of IMDS.

2.7 Behavior

IMDS provides real-time and historical measurement data from an industrial device used in the machine tool environment. IMDS is used to observe the device's state, and the measurement characteristics shall provide immediate feedback to users.

3 Service characteristics

This section defines the characteristic and descriptor requirements. Where a characteristic can be indicated or notified, a Client Characteristic Configuration descriptor must be included in that characteristic as required in Volume 3, Part G, Section 3.3.1.1 in [4].

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
IMD Measurement (see Section 3.1)	M	Read, Notify	Write	Write with Authorization, C.1
IMD Status (see Section 3.2)	O	Notify	-	None
IMDS Descriptor Value Changed (see Section 3.3)	C.2	Indicate	-	None
First Use Date (see Section 3.4)	O	Read	Write	Write with Authorization, C.1
Life Cycle Data (see Section 3.5)	O	Read	Write	Write with Authorization, C.1
Work Cycle Data (see Section 3.6)	O	Read	Write, Notify	Write with Authorization, C.1
Service Cycle Data (see Section 3.7)	O	Read, Write	-	Write with Authorization, C.1
IMD Control (see Section 3.8)	O	Write	-	Write with Authorization, C.1
IMD Historical Data (see Section 3.9)	O	Notify	-	None
Record Access Control Point (see Section 3.10)	O	Write, Indicate	-	None

Table 3.1: Characteristics for use with IMDS

C.1: If the characteristic is writable and Write requires authorization, then encryption is required. The authorization method is implementation specific.

C.2: Mandatory if any of the IMD Measurement characteristic descriptors defined in Table 3.2 is writable, otherwise Excluded.

If the appropriate authorized level has not been established, then the IMD Server may reject an attempt to write to the characteristics with the security permissions set to Write with Authorization by returning an error response with the error code set to Insufficient Authorization.

3.1 IMD Measurement characteristic

IMDS shall expose at least one IMD Measurement characteristic.

Permitted types of IMD Measurement characteristics are defined in the IMDS section in [3]. For example, an IMD Measurement characteristic can represent a temperature or an acceleration.

The IMD Server may expose multiple instances of the same IMD Measurement characteristic (i.e., characteristics with the same UUID). If more than one instance of the IMD Measurement characteristic is present, then each instance shall have a Measurement Description descriptor (see Section 3.1.2.1), and each Measurement Description descriptor shall have a unique combination of the Sampling Function (see Section 3.1.2.1.2) and the Description (see Section 3.1.2.1.5) to allow the instances to be differentiated by the client. The Measurement Description descriptor allows an IMD to report measurements in more than one way (e.g., a device may report maximum and minimum values measured during the work cycle).

For example, a force measurement sensor may provide both instantaneous readings and work cycle maximum readings. In this example, the instantaneous and work cycle maximum measurements are identified by the Measurement Description descriptor associated with each instance of the IMD Measurement characteristic, so that the client can distinguish between the maximum since the beginning of the work cycle force measurement and the instantaneous force measurement.

3.1.1 Characteristic format

Each permitted type of the IMD Measurement characteristic is defined in the IMDS section in [3]. The UUID values are also defined in [3].

The formats of various IMD Measurement characteristics are defined in [1].

3.1.2 Characteristic descriptors

There shall be no more than one instance of each descriptor associated with each instance of an IMD Measurement characteristic.

All descriptor values shall be persistent for bonded clients. If an IMD Measurement characteristic descriptor defined in Table 3.2 is changed locally or is successfully updated by the client, then the IMDS Descriptor Value Changed characteristic shall be used to alert all bonded client(s) except for the client that changed the descriptor value (see Section 3.3). The IMD Server may indicate the change to the client that performed the change.

IMD Measurement characteristic descriptor requirements are defined in Table 3.2.

Characteristic Descriptor	Requirement	Mandatory Permissions	Optional Permissions	Security Permissions
Measurement Description	C.1	Read	None	None
Characteristic User Description	O	Read	Write	Write with Authorization, C.4
Manufacturer Limits	O	Read	None	None

Characteristic Descriptor	Requirement	Mandatory Permissions	Optional Permissions	Security Permissions
Process Tolerances	O	Read	Write	Write with Authorization, C.4
Trigger Settings	C.2	Read	Write	Write with Authorization, C.4
Valid Range	O	Read	None	None
Characteristic Extended Properties	C.3	Read	None	None

Table 3.2: IMD Measurement characteristic descriptors

C.1: Mandatory if multiple instances of an IMD Measurement characteristic with the same UUID are supported, otherwise Optional.

C.2: Mandatory if the device does not have a Custom Condition (see Section 3.1.2.5), otherwise Excluded.

C.3: Mandatory if a Characteristic User Description descriptor is present and is writable, otherwise Excluded.

C.4: If the descriptor is writable and Write requires authorization, then encryption is required. The authorization method is implementation-specific.

The requirements in Table 3.2 are per individual characteristic. If the IMDS supports several IMD Measurement characteristics, each characteristic may include different descriptors.

If an appropriate authorized level has not been established, then the IMD Server may reject an attempt to write the descriptor with security permissions set to Write with Authorization by returning an error response with the error code set to Insufficient Authorization.

3.1.2.1 Measurement Description descriptor

The Measurement Description descriptor characteristic describes the associated IMD Measurement characteristic by providing additional information pertaining to the measured value.

The descriptor UUID shall be set to «IMD Measurement Description» as defined in [3].

The definition of the IMD Measurement Description descriptor is shown in Table 3.3.

If there is more than one IMD Measurement characteristic with the same UUID, the values corresponding to the Sampling Function (see Section 3.1.2.1.2) and the Description (see Section 3.1.2.1.5) shall uniquely identify the characteristic.

Field Name	Format	Size (Octets)	Exponent Base	Exponent	Unit	Resolution	Field Requirement
Flags	boolean[16]	2	N/A	N/A	unitless	N/A	M
Sampling Function	uint8	1	N/A	N/A	unitless	N/A	C.1
Measurement Period	uint24	3	10	0	second	1 millisecond	C.1
Internal Update Interval	uint24	3	10	0	second	1 millisecond	C.1
Description	uint16	2	N/A	N/A	unitless	N/A	C.1
Resolution	variable	variable	N/A	N/A	variable	variable	C.1
Relative Uncertainty	uint8	1	10	-1	percentage	0.1 percent	C.1, C.2
Absolute Uncertainty	variable	variable	N/A	N/A	variable	variable	C.1, C.2

Table 3.3: Measurement Description descriptor structure

C.1: At least one of the fields listed in Table 3.3 shall be present.

C.2: Either the Relative Uncertainty field or the Absolute Uncertainty field may be present, but not both.

3.1.2.1.1 Flags

The Flags field provides additional information about the fields of the Measurement Description descriptor. Possible values of the bitfield are defined in Table 3.4.

Bit	Description
0	If set, then the Sampling Function field is present
1	If set, then the Measurement Period field is present
2	If set, then the Internal Update Interval field is present
3	If set, then the Description field is present
4	If set, then the Resolution field is present
5	If set, then the Relative Uncertainty field is present
6	If set, then the Absolute Uncertainty field is present
7 to 15	RFU

Table 3.4: Flags field of the Measurement Description descriptor

3.1.2.1.2 Sampling Function

The Sampling Function field specifies the average operation or type of sampling function applied to the value of the IMD Measurement characteristic. For example, this field of the descriptor can identify whether the measurement provided by the IMD Server is an arithmetic mean value since the beginning of the work cycle or an instantaneous value. The assigned numbers for this field are listed in Table 3.5.

The absence of the Sampling Function field from the Measurement Description descriptor as well as the absence of the Measurement Description descriptor signifies that the IMDS publishes the instantaneous measurement.

Value	Description
0x00	Unspecified
0x01	Instantaneous
0x02	Arithmetic mean since the beginning of the work cycle
0x03	Root Mean Square (RMS) value since the beginning of the work cycle
0x04	Maximum value in the current work cycle
0x05	Minimum value in the current work cycle
0x06	Moving average
0x07 to 0xFF	RFU

Table 3.5: Sampling Function field of the Measurement Description descriptor

3.1.2.1.3 Measurement Period

The Measurement Period field specifies the measurement period in milliseconds over which the measurement is taken. For example, the Measurement Period field can specify the length of the period used to obtain an average reading. Possible values for the Measurement Period field are defined in Table 3.6.

The Measurement Period field may be excluded from the Measurement Description descriptor for cases where a value for the measurement period is not available or is not applicable.

Value	Description
0x000000	Used when Sampling Function is set to 0x01 (Instantaneous)
0x000001 to 0xFFFFFF	Time period in milliseconds

Table 3.6: Measurement Period field of the Measurement Description descriptor

If the Measurement Description descriptor or the Measurement Period field is not present, then the measurement period is unspecified.

3.1.2.1.4 Internal Update Interval

The value of the IMD characteristic is internally refreshed by the IMD Server at the frequency indicated in the Internal Update Interval field (e.g., a temperature value is internally updated every 2 seconds). This field allows the client to determine the interval (in milliseconds) between updates. Possible values for the Internal Update Interval field are defined in Table 3.7.

Value	Description
0x000000	RFU
0x000001 to 0xFFFFFF	Time interval in milliseconds

Table 3.7: Internal Update Interval field of the Measurement Description descriptor

If the Measurement Description descriptor or the Internal Update Interval field is not present, then the internal update interval is unspecified.

3.1.2.1.5 Description

The Description field specifies additional information about how the IMD Measurement characteristic is designed to be used (e.g., to differentiate internal versus external temperature measurements).

The assigned numbers for this field are enumerated in the Bluetooth SIG GATT Characteristic Presentation Format Description table in [3]. The implementation shall select the value that provides the closest match to the IMD Measurement characteristic description. For example, a caliper can measure the outside and the inside of an object. The IMDS will contain two IMD Measurement characteristics. The Description field of the Measurement Description descriptor for the first characteristic will be set to 0x010C (outside). For the second characteristic, the Description field of the Measurement Description descriptor will be set to 0x010B (inside).

The absence of the Description field in the Measurement Description descriptor corresponds to a value of 0 (unknown).

3.1.2.1.6 Resolution

The Resolution field specifies the smallest increment of the IMD Measurement characteristic that can be measured by the IMD. The format of the Resolution field matches the format of the IMD Measurement characteristic, but only positive values are allowed. Zero indicates unknown resolution and negative values are Prohibited.

3.1.2.1.7 Relative Uncertainty and Absolute Uncertainty

This field shows the measurement uncertainty of the IMD Measurement characteristic. As specified in Table 3.3, either the Relative Uncertainty field or the Absolute Uncertainty field may be present, but not both.

The Relative Uncertainty field is expressed as a percentage as defined in Table 3.8. For example, if the measured value is 500 mm/s^2 and the value of the Relative Uncertainty field is 10 (i.e., 1%), then the actual value is between 495 and 505 mm/s^2.

Value	Description
0x00 to 0xFF	The maximum error from the actual value, expressed as a difference relative to the measured value in 0.1% increments Represents values from 0.0% to 25.5%

Table 3.8: Relative Uncertainty level of the Measurement Description descriptor

If the Absolute Uncertainty field is present, then the size and format of this value is the same as the size and format of the IMD Measurement characteristic. For example, if an IMD sensor measures acceleration as uint32 in mm/s^2 (millimeters per seconds squared), then the absolute value is expressed as uint32 with a resolution of mm/s^2. If the measured value is 500 mm/s^2 and the uncertainty value is 10 mm/s^2, then the actual value is between 490 and 510 mm/s^2.

If the Manufacturer Limits descriptor (see Section 3.1.2.3) is present, and if the reported IMD Measurement characteristic is in the Red Zone, then the Uncertainty level may be incorrect.

3.1.2.2 Characteristic User Description descriptor

This optional descriptor may be used to associate a user textual description with a specific IMD Measurement characteristic to allow a human-readable label to be associated with the measurement. The format of the Characteristic User Description descriptor is defined in Volume 3, Part G, Section 3.3.3.2 in [4].

The User Description descriptor may be writable. If so, then the IMD Measurement characteristic shall have the Extended Properties property, the Characteristic Extended Properties descriptor shall be present, and the Writable Auxiliaries bit of the Characteristic Extended Properties descriptor shall be set to 1.

3.1.2.3 Manufacturer Limits descriptor

3.1.2.3.1 Definition

The structure of the Manufacturer Limits descriptor is defined in Table 3.9.

The descriptor UUID shall be set to «IMD Manufacturer Limits» as defined in [3].

The format of each field matches the format of the IMD Measurement characteristic.

Field	Data Type	Field Requirement	Description
Low Red Limit	variable	M	Contains the manufacturer-defined low limit for the error operation. When the device reports measurements below the Low Red Limit, the values may be outside the uncertainty limits defined in the IMD Measurement descriptor. The device could be damaged if operating in that condition.

Field	Data Type	Field Requirement	Description
Low Yellow Limit	variable	M	Contains the manufacturer-defined low limit for the warning operation. When the device reports measurements below the Low Yellow Limit, the values may be outside the uncertainty limits defined in the IMD Measurement descriptor.
High Yellow Limit	variable	M	Contains the manufacturer-defined high limit for the warning operation. When the device reports measurements above the High Yellow Limit, the values may be outside the uncertainty limits defined in the IMD Measurement descriptor.
High Red Limit	variable	M	Contains the manufacturer-defined high limit for the error operation. When the device reports measurements above the High Red Limit, the values may be outside the uncertainty limits defined in the IMD Measurement descriptor. The device could be damaged if operating in that condition.

Table 3.9: Manufacturer Limits descriptor

3.1.2.4 Process Tolerances descriptor

The Process Tolerances descriptor contains user-defined values such as usage limits.

The descriptor UUID shall be set to «IMD Process Tolerances» as defined in [3].

3.1.2.4.1 Definition

The structure of the descriptor is shown in Table 3.10. The initial values of the fields shall be identical to the Manufacturer Limits descriptor and the Target Value shall be set to zero.

The values used for the Process Tolerances descriptor may not exceed the corresponding Manufacturer Limits descriptor range:

- The Low Red Limit of the Process Tolerance descriptor shall be greater than or equal to the Low Red Limit of the Manufacturer Limits descriptor (see Section 3.1.2.3).
- The Low Yellow Limit of the Process Tolerance descriptor shall be greater than or equal to the Low Yellow Limit of the Manufacturer Limits descriptor.
- The High Yellow Limit of the Process Tolerance descriptor shall be less than or equal to the High Yellow Limit of the Manufacturer Limits descriptor.
- The High Red Limit of the Process Tolerance descriptor shall be less than or equal to the High Red Limit of the Manufacturer Limits descriptor.

If a client attempts to write an invalid value, then the IMD Server shall reject the request by returning an error response with the error code set to Value Not Allowed (see Volume 3, Part F, Section 3.4.1.1 in [4]).

Only one Process Tolerances descriptor declaration shall exist in a characteristic definition. The Process Tolerances descriptor value shall be persistent, meaning that the value written by a client shall be used until modified by the same client or a different client.

Field	Data Type	Field Requirement	Description
Flags	boolean[8]	M	See Section 3.1.2.4.2.
Target Value	variable	C.1	Contains the user-defined target value for the current process.
Low Red Process Tolerance	variable	C.2	Contains the user-defined low tolerance for the error operation. This value is absolute or relative to the target value, depending on the Flags field.
Low Yellow Process Tolerance	variable	C.2	Contains the user-defined low tolerance for the warning operation. This value is absolute or relative to the target value, depending on the Flags field.
High Yellow Process Tolerance	variable	C.2	Contains the user-defined high tolerance for the warning operation. This value is absolute or relative to the target value, depending on the Flags field.
High Red Process Tolerance	variable	C.2	Contains the user-defined high tolerance for the error operation. This value is absolute or relative to the target value, depending on the Flags field.

Table 3.10: Process Tolerances descriptor

C.1: Mandatory during the Read operation and Optional during the Write operation, according to the Flags field. Also Mandatory when the Flags field indicates a change from absolute to relative tolerances in a Write operation.

C.2: Mandatory during the Read operation and Optional during the Write operation, according to the Flags field. Also Mandatory when the Flags field indicates a change from absolute to relative tolerances or from relative to absolute tolerances in a Write operation.

The format of the Target Value field shall match the format of the IMD Measurement characteristic.

When the Flags field indicates that the tolerance values are absolute, then the format of each tolerance field shall match the format of the IMD Measurement characteristic.

When the Flags field indicates that the tolerance values are relative, then the format of each tolerance field shall match the format of the IMD measurement characteristic except that the values are never negative.

The absolute tolerance value can be calculated from the Target Value and the relative tolerance value using the following formulas:

- Low Red Process Tolerance(absolute) = Target Value – Low Red Process Tolerance(relative)
- Low Yellow Process Tolerance(absolute) = Target Value – Low Yellow Process Tolerance(relative)
- High Yellow Process Tolerance(absolute) = Target Value + High Yellow Process Tolerance(relative)
- High Red Process Tolerance(absolute) = Target Value + High Red Process Tolerance(relative)

3.1.2.4.2 Flags

During Read operations, the Flags field defines whether tolerance values are absolute or relative to the target value.

Bit	Definition
0	If not set, then tolerance values are absolute. If set, then tolerance values are relative to the target value.
1 to 7	RFU

Table 3.11: Flags field of the Process Tolerances descriptor during Read operations

During Write operations, the Flags field additionally identifies the presence of other fields in the descriptor.

Bit	Definition
0	If not set, then tolerance values are absolute. If set, then tolerance values are relative to the target value.
1	Target Value field is present.
2	Low Red Process Tolerance field is present.
3	Low Yellow Process Tolerance field is present.
4	High Yellow Process Tolerance field is present.
5	High Red Process Tolerance field is present.
6 to 7	RFU

Table 3.12: Flags field of the Process Tolerances descriptor during Write operations

3.1.2.5 Trigger Settings descriptor and Custom Condition

If the IMD uses the Custom Condition to send measurement results and when the Custom Condition is triggered, then the value of the characteristic shall be notified (assuming that the notifications are enabled via the Client Characteristic Configuration descriptor). If the IMD uses the Custom Condition to notify the IMD Measurement characteristic, then the Trigger Settings descriptor shall not be present for that characteristic.

Zero or one instances of the Trigger Settings descriptor shall be defined for each IMD Measurement characteristic.

3.1.2.5.1 Trigger Settings descriptor

The Trigger Settings descriptor is a writable descriptor that is used to define the way the IMD Server sends notifications to connected clients.

The descriptor UUID shall be set to «IMD Trigger Settings» as defined in [3].

The format of the Trigger Settings descriptor is defined in Table 3.13.

Field	Data Type	Field Requirement
Time Condition	uint32	M
Delta Condition	variable	M

Table 3.13: Trigger Settings descriptor format

If the Time Condition field of the Trigger Settings descriptor is set to zero, then the IMD Server shall not send notifications periodically.

If the Time Condition field of the Trigger Settings descriptor is set to a non-zero value, then the IMD Server shall send notifications to a connected client periodically. The interval between notifications should be equal to the value of the Time Condition field of the Trigger Settings descriptor in milliseconds. Meanwhile, the IMD Server might not be able to select the time interval exactly as requested by the client and the resulting time interval might differ from the time interval requested by the client. When the Trigger Settings Descriptor is read, the IMD Server shall return its estimate of the interval it is using for notifications.

If the client attempts to write a Time Condition that is lower than the supported interval of the IMD Server, then the IMD Server may either send notifications with the shortest possible time interval or respond with a Write Request Rejected error code (see Part B, Section 2.4 in [2]).

The default value for the Time Condition shall be zero.

If the Delta Condition field of the Trigger Settings descriptor is set to a non-zero value, then the IMDS shall send a notification to a connected client when the difference between the measured value and the previously notified value exceeds the value represented by the Delta Condition field. If the value of the Delta Condition field is set to zero, then the device shall not send notifications on the value changed event.

The default value for the Delta Condition shall be zero.

Only one Trigger Settings descriptor declaration shall exist in a characteristic definition. The Trigger Settings descriptor value shall be persistent, meaning that the value written by a client shall be used until modified by the same client or a different client.

To disable IMD Measurement characteristic notifications, the client uses the Client Characteristic Configuration descriptor. When notifications are enabled, but both Time Condition and Delta Condition fields of the Trigger Settings descriptor are set to zero, the behavior of IMDS is device-specific. For example, some devices may not send notifications at all, while others may send a notification after every measurement.

When a combination of Time Condition and Delta Condition is selected, then the IMD Server sends a notification before the interval expires, or when the measured value has changed more than the specified Delta Condition. This causes a notification and a reset of the time counter of the periodic notifications (see Figure 3.1).

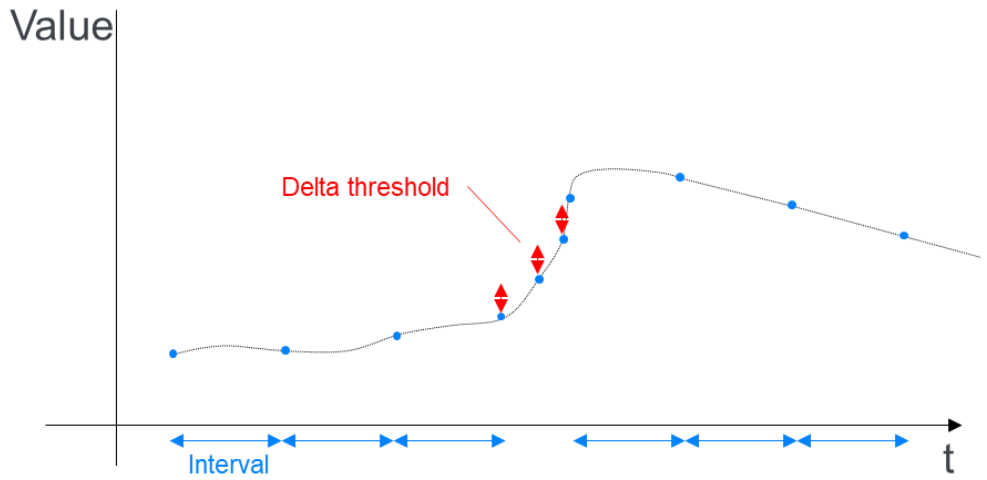

Figure 3.1: Trigger Settings example

The format of the Delta Condition field of the IMD Trigger Settings descriptor matches the format of the IMD Measurement characteristic. Only positive values are allowed.

Setting the Trigger Settings may cause a work cycle to be started on the IMD. The IMD Server may require the current time to be set before starting the first work cycle. If the current time in IMD is not set, then the IMD Server may reject the command to set Trigger Settings by returning an error response with the error code set to Invalid Time (see Section 2.4).

3.1.2.6 Valid Range descriptor

The IMD can define the Valid Range descriptor as specified in GSS [1]. For an IMD, the Valid Range descriptor defines the lower inclusive value and the upper inclusive value that can be provided by the IMD Server. These values might be lower or higher than the manufacturer-defined limits. A client may use the Valid Range descriptor to display the full range of values in case the device is operating below or above the manufacturer-defined limits.

3.1.3 Characteristic behavior

All IMD Measurement characteristics have identical behavioral options as described in this subsection.

The client can read the IMD Measurement characteristic value at any time. When read, the IMD Server shall report the most recent measurement value. When the measurement value is not available (e.g., when no measurement has been performed after the IMD Server was restarted), the IMD Server may reject the read request with an error response with the error code set to Read Not Permitted (see Volume 3, Part F, Section 3.4.1.1 in [4]).

If the IMD Measurement characteristic supports notifications, and if the client configures the IMD Measurement characteristic for notifications via the Client Characteristic Configuration descriptor, and if a measurement is available, then IMDS shall notify the characteristic while in a connection in accordance with the trigger conditions specified by the Trigger Settings descriptor for that characteristic as described in Section 3.1.2.5.

Some IMDs may support writing to the IMD Measurement characteristic. Behavior of the IMD when the value is written is implementation specific. For example, some devices may use writing of the IMD Measurement characteristic for calibration.

If the IMD Client attempts to write to the IMD Measurement characteristic a value that is not an acceptable value (e.g., outside the Valid Range descriptor), the IMD Server shall reject the request with a Value Not Allowed error code (see Volume 3, Part F, Section 3.4.1.1 in [4]).

3.2 IMD Status

The status of an IMD Measurement characteristic is considered changed if a limit specified in either the Manufacturer Limits descriptor or in the Process Tolerances descriptor has been reached. If there is at least one IMD Measurement characteristic the status of which can change, the IMD Server may expose a single instance of the IMD Status characteristic.

3.2.1 Characteristic format

The format of the IMD Status characteristic when notified is specified in Table 3.14.

Field	Data Type	Size (in octets)	Description
Status	boolean[16]	2	Measured value status (see Section 3.2.1.1)
UUID	uint16	2	UUID of the IMD Measurement characteristic that caused the status change (see Section 3.1)
Sampling Function	uint8	1	Sampling function of the IMD Measurement characteristic that caused the status change (see Section 3.1.2.1.2)
Description	uint16	2	Description of the IMD Measurement characteristic that caused the status change (see Section 3.1.2.1.5)

Table 3.14: IMD Status characteristic format when notified

3.2.1.1 Status field

The Status field identifies the status of the IMD when the measurement was taken as defined in Table 3.15. The status values are represented in a bit field.

Bit	Description
0	User-defined low Red Limit of the process tolerance value exceeded
1	User-defined low Yellow Limit of the process tolerance value exceeded
2	User-defined high Yellow Limit of the process tolerance value exceeded
3	User-defined high Red Limit of the process tolerance value exceeded
4	Manufacturer-defined low Red Limit exceeded
5	Manufacturer-defined low Yellow Limit exceeded
6	Manufacturer-defined high Yellow Limit exceeded

Bit	Description
7	Manufacturer-defined high Red Limit exceeded
8 to 15	RFU

Table 3.15: Status field definition

The Measured Value Status depends on the values set in the Process Tolerances descriptor (see Section 3.1.2.4) and the Manufacturer Limits descriptor (see Section 3.1.2.3).

Figure 3.2 shows the relation between the target value, manufacturer-defined limits, and user-defined process tolerances.

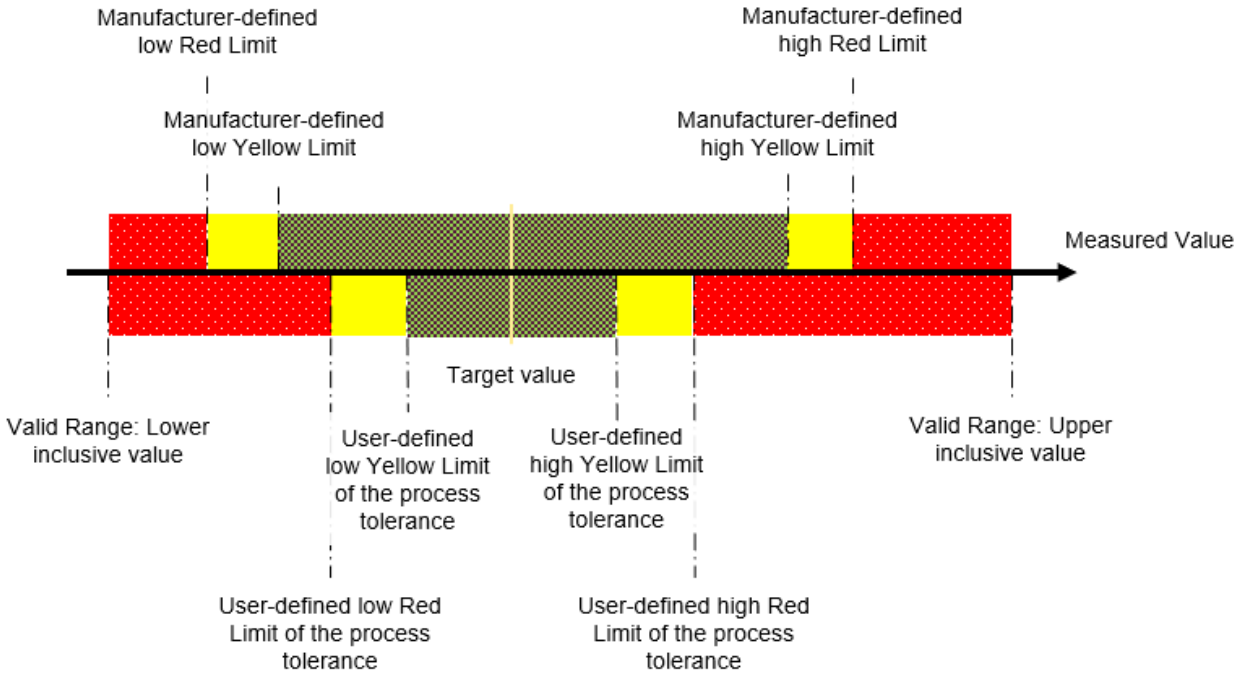

Figure 3.2: Target value, limits, tolerances, and valid range

If a Red Limit is exceeded, then both the Yellow Exceeded bit and the Red Limit exceeded bit will be set. This applies to the high and low limits and to the Manufacture-defined and User-defined limits. For example, if the measurement value exceeds the Manufacturer-defined high Red Limit, then the Status field of the IMDS Status characteristic will be set to 0xC6, which corresponds to the sum of bits 0x02 (User-defined high Yellow Limit of the process tolerance value exceeded), 0x04 (User-defined high Red Limit of the process tolerance value exceeded), 0x40 (Manufacturer-defined high Yellow Limit exceeded), and 0x80 (Manufacturer-defined high Red Limit exceeded).

The green, yellow, and red zones as defined are shown in Figure 3.2. The values of the Valid Range descriptor mark the very top and bottom edges of the red zones.

User-defined process tolerance values and corresponding operating zones are shown below the black line in Figure 3.2. The manufacturer-defined limits and operating zones are shown above the black line.

3.2.2 Characteristic behavior

If notifications are enabled via the Client Characteristic Configuration descriptor, then the IMDS shall send a notification whenever a measurement results in a status change.

When notified, the IMD Server shall set the bits in the Status field corresponding to the latest measured value and the values of the manufacturer-defined limits and user-defined process tolerance values. For example, if the measured value is larger than the manufacturer-defined high Red Limit, then four bits shall be set: the user-defined high Yellow Process Tolerance value exceeded, the user-defined high Red Process Tolerance value exceeded, the manufacturer-defined high Yellow Limit exceeded, and the manufacturer-defined high Red Limit exceeded.

The rules defined by the Trigger Settings descriptors of IMD Measurement characteristics (see Section 3.1.2.5.1) are used to send notifications of the IMD Status characteristic, meaning that if the Time Condition or the Delta Condition of the Trigger Settings descriptor for the IMD Measurement characteristic is satisfied, and if the status changes, then the IMD Status characteristic shall be notified. Therefore, if notifications for an IMD Measurement characteristic are enabled, and if the measurement results in a status change, then the notifications for the IMD Status characteristics and for the IMD Measurement characteristic shall be sent one after another.

If the IMD Measurement Description descriptor for the IMD Measurement characteristic that resulted in the status change is not present, or if the Sampling Function field in the IMD Measurement Description descriptor is not present, then the IMD Server shall set the Sampling Function field in the notification to the default value (0x01 Instantaneous).

If the IMD Measurement Description descriptor for the IMD Measurement characteristic that resulted in the status change is not present, or if the Description field in the corresponding IMD Measurement Description descriptor is not present, then the IMD Server shall set the Description field in the notification to the default value (0x0000 unknown).

3.3 IMDS Descriptor Value Changed characteristic

The IMDS Descriptor Value Changed characteristic is present if any of the IMD Measurement characteristics descriptors defined in Table 3.2 can be changed by the IMD Server or if any of the descriptors is writable.

3.3.1 Characteristic format

The structure of this characteristic is defined in Table 3.16.

Field	Data Type	Size (in octets)	Description
Descriptor Handle	uint16	2	The GATT handle of the descriptor that has been changed.

Table 3.16: IMDS Descriptor Value Changed characteristic

3.3.2 Characteristic behavior

If the IMD Client has registered to receive indications via the Client Characteristic Configuration descriptor, then whenever any of the IMD Measurement characteristic descriptor values changes (except for the Client Characteristic Configuration descriptor), the IMD Server shall send the IMDS Descriptor Value Changed characteristic indication.

If an IMD Measurement characteristic descriptor value changes while a bonded client is not connected, then the IMD Server shall send an indication on a reconnection to the bonded client. If more than one of the IMD Measurement characteristic descriptors have changed, then the Descriptor Handle shall be set to 0.

3.4 First Use Date characteristic

The First Use Date characteristic is set by the device when the first work cycle is started.

3.4.1 Characteristic format

The structure of this characteristic is defined in Table 3.17.

Field	Data Type	Size (in octets)	Description
First Use Date	uint16	2	Date of first use in days since 2000-01-01 0x0000 shall be the default, meaning the value is not yet set

Table 3.17: First Use Date characteristic

3.4.2 Characteristic behavior

When written, the IMD Server shall store the First Use Date value in persistent storage. The IMD Server shall return the written value when read by a client.

The current time of the IMD Server shall be set before executing the first work cycle. If the IMD Server starts executing a work cycle while the First Use Date value is zero, the IMD Server shall set the First Use Date characteristic value to the current time of the IMD Server.

The IMD Device may have internal logic for writing the First Use Date value and may make the First Use Date characteristic read-only.

3.5 Life Cycle Data characteristic

The Life Cycle Data characteristic provides values describing the life cycle data defined by the manufacturer. The Use Time Yellow Limit defines how long the IMD may be in use (runtime) until it should be serviced. The Use Time Red Limit indicates the maximum expected use time.

3.5.1 Characteristic format

When written, the structure of this characteristic is defined in Table 3.18.

Field	Data Type	Size (in octets)	Description
Manufacturing Date	uint16	2	Date of manufacturing in days since 2000-01-01
Use Time Yellow Limit	uint24	3	Yellow Limit value in hours
Use Count Yellow Limit	uint24	3	Yellow Limit value in number of work cycles
Use Time Red Limit	uint24	3	Red Limit value in hours
Use Count Red Limit	uint24	3	Red Limit value in number of work cycles

Table 3.18: Life Cycle Data characteristic format when written

When read, the structure of this characteristic is defined in Table 3.19.

Field	Data Type	Size (in octets)	Description
Flags	boolean[16]	2	See Section 3.5.1.1
Life Cycle Status	uint8	1	Indicates if the device is in the green, yellow, or red zone (see Section 1.5) 0x00 = Green Zone 0x01 = Yellow Zone 0x02 = Red Zone 0x03 – 0xFF = RFU Present if bit 0 of the Flags field is set to 1
Manufacturing Date	uint16	2	Date of manufacturing in days since 2000-01-01 Present if bit 1 of the Flags field is set to 1
Use Time Yellow Limit	uint24	3	Yellow Limit value in hours Present if bit 2 of the Flags field is set to 1
Use Count Yellow Limit	uint24	3	Yellow Limit value in number of work cycles Present if bit 3 of the Flags field is set to 1
Use Time Red Limit	uint24	3	Red Limit value in hours Present if bit 2 of the Flags field is set to 1

Field	Data Type	Size (in octets)	Description
Use Count Red Limit	uint24	3	Red Limit value in number of work cycles Present if bit 3 of the Flags field is set to 1
Use Time In Yellow Zone	uint24	3	Use time while in Yellow Zone in hours Present if bit 4 of the Flags field is set to 1
Use Count In Yellow Zone	uint24	3	Use count while in Yellow Zone in number of work cycles Present if bit 5 of the Flags field is set to 1
Use Time In Red Zone	uint24	3	Use time while in while in Red Zone in hours Present if bit 4 of the Flags field is set to 1
Use Count In Red Zone	uint24	3	Use count while in Red Zone in number of work cycles Present if bit 5 of the Flags field is set to 1
Work Cycle Counter	uint24	3	A counter indicating number of times IMD executed a work cycle during Life Cycle Present if bit 6 of the Flags field is set to 1
Actual Use Time	uint24	3	Actual time the IMD was executing work cycles in hours Present if bit 7 of the Flags field is set to 1

Table 3.19: Life Cycle Data characteristic format when read

3.5.1.1 Flags

The IMD may not support all of the counters defined in Table 3.19. The Flags field provides additional information about the fields present in the Life Cycle Data characteristic when read. Possible values of the bitfield are defined in Table 3.20.

Bit	Description
0	If set, then the Life Cycle Status field is present.
1	If set, then the Manufacturing Date field is present
2	If set, then the Use Time Yellow Limit and the Use Time Red Limit fields are present.
3	If set, then the Use Count Yellow Limit and the Use Count Red Limit fields are present.
4	If set, then the Use Time In Yellow Zone and the Use Count In Red Zone fields are present.

Bit	Description
5	If set, then the Use Count In Yellow Zone and the Use Time In Red Zone fields are present.
6	If set, then the Work Cycle Counter field is present.
7	If set, then the Actual Use Time field is present.
8 to 15	RFU

Table 3.20: Flags field of the Life Cycle Data characteristic

3.5.2 Characteristic behavior

When read, the IMD Server shall return the Life Cycle Status, the Manufacturing Date, the Use Time in Yellow Zone and in Red Zone Limits, the Use Count in Yellow Zone and in Red Zone Limits, the Use Times in Yellow Zone and in Red Zone, the Use Counts in Yellow Zone and in Red Zone, the Work Cycle Counter, and the Actual Use Time. If the IMD does not support one or more of the counts, then the IMD Server may set the corresponding bit in the Flags field to zero and exclude corresponding fields from the read response.

When written, the IMD Server shall update the Manufacturing Date, the Use Time in Yellow Zone and in Red Zone Limits, and the Use Count in Yellow Zone and in Red Zone Limits, as set by the client.

The IMD Server may reject the client's write request in a certain device state by replying with an error response with the error code set to Value Not Allowed (see Volume 3, Part F, Section 3.4.1.1 in [4]). For example, the IMD may accept writing of the Life Cycle Data characteristic only once during manufacturing and reject attempts to change the value.

3.6 Work Cycle Data characteristic

The Work Cycle Data characteristic provides information about the work cycle of the IMD. It may also be used to start or stop a work cycle.

3.6.1 Characteristic Format

The characteristic format when read or notified is defined in Table 3.21.

Field	Data Type	Size (Octets)	Description
Work Cycle Index	uint24	3	Index of the current Work Cycle
Start Time	Elapsed Time (see [1])	9	The time when the Work Cycle started
Status	uint8	1	Status of the Work Cycle (see Section 3.6.1.1)

Table 3.21: Work Cycle Data characteristic format when read

The characteristic format when written is defined in Table 3.22.

Field	Data Type	Size (Octets)	Description
Operation Request Code	uint8	1	Operation requested by the client (see Section 3.6.1.2)

Table 3.22: Work Cycle Data characteristic format when written

3.6.1.1 Status field

Possible values of the Status field are defined in Table 3.23.

Value	Description
0x00	Unknown
0x01	Work Cycle is in progress
0x02	Work Cycle is completed
0x03 to 0xFF	RFU

Table 3.23: Status field definition

3.6.1.2 Operation Request Code field

Possible values of the Operation Request Code field are defined in Table 3.24.

Value	Description
0x00	Start new Work Cycle
0x01	Stop current Work Cycle
0x02 to 0xFF	RFU

Table 3.24: Operation Request Code field definition

3.6.2 Characteristic behavior

When read, the IMD Server shall return the current Work Cycle index, the time when the current Work Cycle was started, and the Status. If the IMD Server does not know its status (for example, after the restart), then it shall return Status set to Unknown (0x00), and the values for Work Cycle Index and Start Time shall be set to zero.

If the IMD Work Cycle Data characteristic supports notifications, and if the client configures the characteristic for notifications via the Client Characteristic Configuration descriptor, then IMDS shall notify the characteristic while in a connection if there are any changes to Work Cycle Index, Start Time, or Status.

When written, the IMD Server shall start a new Work Cycle, or stop the current Work Cycle, as specified in the Operation Request Code. Actual actions that the IMD performs on the start or stop are implementation-specific.

If the IMD Work Cycle Data write indicates an unsupported Operation Request Code, then the IMD Server shall return an error response with the error code set to Write Request Rejected (see Part B, Section 2.4 in [2]).

The IMD Server may reject the client's write request in a certain device state by replying with an error response with the error code set to Value Not Allowed (see Volume 3, Part F, Section 3.4.1.1 in [4]). The reason for accepting or rejecting the write request is manufacturer-specific. For example, the device may reject the Start command when the current state is in progress or reject the Stop command when the current state is completed.

If the IMD Server supports IMD Historical Record, when a Work Cycle is completed, including the successful processing of the Stop command, then the IMD Server shall create a new IMD Historical Record with the type 0x01 = Work Cycle Data Record (see Section 3.9.2).

3.7 Service Cycle Data characteristic

The Service Cycle Data characteristic provides values describing the service cycle of the IMD.

3.7.1 Characteristic Format

When written, the characteristic format is defined in Table 3.25.

Field	Data Type	Size (octets)	Description
Next Service Date	uint16	2	The field identifies the date in number of days since 2000-01-01 when the next service should be performed. Value of 0 specifies that the value is not set.
Max Use Time	uint24	3	The field specifies the amount of time in hours the IMD can be used until the next service should be performed. Value of 0 specifies that the value is not set.
Max Work Cycles Count	uint24	3	The field specifies the number of work cycles the IMD can execute until the next service should be performed. Value of 0 specifies that the value is not set.

Table 3.25: Service Cycle Data characteristic format when written

When read, the characteristic format is defined in Table 3.26.

Field	Data Type	Size (octets)	Description
Flags	boolean[16]	2	See Section 3.7.1.1

Field	Data Type	Size (octets)	Description
Service Cycle Status	uint8	1	The field describes whether the device requires service or not (see Section 3.7.1.2). Present if bit 0 of the Flags field is set to 1
Next Service Date	uint16	2	The date by which the next service of the device should be performed, in number of days since 2000-01-01. Present if bit 1 of the Flags field is set to 1
Max Use Time	uint24	3	The field specifies the amount of time in hours the IMD can be used until the next service should be performed. Present if bit 2 of the Flags field is set to 1
Max Work Cycles Count	uint24	3	The field specifies the upper limit for the number of work cycles the IMD can execute until the next service should be performed. Present if bit 3 of the Flags field is set to 1
Actual Use Time	uint24	3	The Actual Use Time field indicates the actual use time of the IMD in hours since the service has been performed. The value is set to zero when the service is performed. Present if bit 4 of the Flags field is set to 1
Work Cycle Counter	uint24	3	A counter indicating the number of times the IMD executed a work cycle since the last service. The value of the counter is set to zero when the service is performed. Present if bit 5 of the Flags field is set to 1

Table 3.26: Service Cycle Data characteristic format when read

Some devices may not be capable of supporting all service cycle data (e.g., devices with single measurements do not count Use Time). The Flags field defines which of those fields are present.

3.7.1.1 Flags

The IMD may not support all of the counters defined in Table 3.26. The Flags field provides information about which fields of the Service Cycle Data characteristic are present when read by the client. Possible values of the bitfield are defined in Table 3.27.

Bit	Description
0	If set, then the Service Cycle Status field is present.
1	If set, then the Next Service Date field is present.
2	If set, then the Max Use Time field is present.
3	If set, then the Max Work Cycles Count field is present.
4	If set, then the Actual Use Time field is present.
5	If set, then the Work Cycle Counter field is present.
6 to 15	RFU

Table 3.27: Flags field of the Service Cycle Data characteristic

3.7.1.2 Service Cycle Status

The Service Cycle Status field describes whether the device requires service or not.

Value	Description
0x00	No service needed; the device operates under good conditions.
0x01	Maintenance service is recommended because the device operates under suboptimal but still acceptable conditions.
0x02	Maintenance service is required because the device operates under suboptimal conditions.
0x03 to 0xFF	RFU

Table 3.28: Service Cycle Status field definition

3.7.2 Characteristic behavior

The characteristic provides information about a device's service cycle. Depending on the value of the Flags field, only certain values are present.

The Service Cycle Data characteristic exposes the recorded Service Cycle data of the IMD. The Work Cycle counter is the number of times the IMD executed a work cycle. The Actual Use Time is the time in hours when the IMD is in a work cycle. The Service Cycle Status indicates the status as Green Zone, Yellow Zone, or Red Zone depending on limits defined in the Life Cycle Data in Section 3.5.

If the write request specifies a non-zero Next Service Date field, and if the IMD Server finds the value of the Next Service Date field unacceptable (e.g., if the time is in the past), then the IMD Server may return an error response with the error code set to Invalid Time (see Section 2.4).

If the IMD Server does not accept the write request parameters for any other reason, then the IMD Server shall return an error response with the error code set to Value Not Allowed (see Volume 3, Part F, Section 3.4.1.1 in [4]).

If the value of the Next Service Date field is acceptable, then the Next Service Date of the IMD Server is updated to the given next service date. The Max Use Time, the Max Work Cycles Count, the Actual Use Time, and the Work Cycle Counter shall be reset to zero in case they are used.

When the IMD Service Cycle Data characteristic is written, and if the IMD Historical Data characteristic is supported, then the IMDS shall create an IMD Historical Data record of type 0x00 = IMD Service Cycle Data Record (see Section 3.9.2).

3.8 IMD Control characteristic

The IMD Control characteristic allows a client to request actions to be performed by the device.

3.8.1 Characteristic format

The format of the characteristic when written is defined in Table 3.29.

Field	Data Type	Size (in octets)	Description
Op Code	uint8	1	Operation requested by the client
Parameters	struct	variable	Parameters of the requested operation

Table 3.29: IMD Control characteristic write format

3.8.1.1 Op Code field

The values of the IMD Control characteristic Op Code field are defined in Table 3.30.

Value	Description
0x00	Request to start measurement
0x01	Request to abort current operation
0x02 to 0x7F	RFU
0x80 to 0xFF	Manufacturer-specific

Table 3.30: Values of the Op Code field

3.8.2 Characteristic behavior

If the IMD Server receives an Op Code that it does not understand or does not support, then the IMD Server shall return an error response with the error code set to Request Not Supported (see Volume 3, Part F, Section 3.4.1.1 in [4]).

3.8.2.1 Request to Start Measurement

If the IMD Control characteristic value is written with the Op Code set to 0, then the IMD Server shall perform a measurement. Parameters are defined in Table 3.31.

Parameter	Requirement	Size (octets)	Description
UUID	M	2	UUID of the IMD Measurement characteristic to be measured
Sampling Function	M	1	The Sampling Function field of the Measurement Description descriptor of the IMD Measurement characteristic (see Section 3.1.2.1.2)
Description	M	2	The Description field of the Measurement Description descriptor of the IMD Measurement characteristic (see Section 3.1.2.1.5)
Delay	O	4	Wait for Delay milliseconds before starting the measurement. If 0, then start with no delay.

Table 3.31: Perform Measurement parameters format

The IMD Server may require the current time to be set before starting the first work cycle. If the current time in IMD is not set, then the IMD Server may reject the command to start measurement by returning an error response with the error code set to Invalid Time (see Section 2.4).

If the IMD Client uses the default value for the Sampling Function (Instantaneous), then the IMD Server shall start the measurement for the IMD Measurement characteristic with the specified Sampling Function, or for the characteristic that does not include the Measurement Description descriptor or that includes the Measurement Description descriptor with no Sampling Function.

If the IMD Client uses the default value for the Description (Unknown), then the IMD Server shall start the measurement for the IMD Measurement characteristic with the specified Description, or for the characteristic that does not include the Measurement Description descriptor or that includes the Measurement Description descriptor with no Description.

If the UUID, Sampling Function, and Description combination does not match any of the instances of the IMD Measurement characteristic and corresponding descriptors, then the IMD Server shall return an error response with the error code set to Value Not Allowed (see Volume 3, Part F, Section 3.4.1.1 in [4]).

If the Delay field is not present, then the IMD Server shall start performing the measurement immediately. If the device is busy performing another operation, then the IMD Server shall return an error response with the error code set to Procedure Already In Progress (see Part B, Section 2.2 in [2]).

If the Delay field is present and the value of the Delay field is not zero, then the IMD Server shall wait for the specified number of milliseconds before starting the measurement. While waiting, if the IMD Server receives another request to start measurement, then the IMD Server shall accept the request, modifying the wait time as specified in the new request.

If the length of the Parameters field is not correct according to sizes specified in Table 3.31, then the IMD Server shall return an error response with the error code set to Invalid Attribute Value Length (see Volume 3, Part F, Section 3.4.1.1 in [4]).

Upon completion of the measurement, the IMDS will notify the client as defined in Section 3.1.3.

3.8.2.2 Request to Abort Current Operation

If the IMD Control characteristic value is written with the Op Code set to a value of 1, then the IMD Server shall treat it as a command to abort the current operation. There are no parameters associated with this request.

The IMD Server shall abort a Start Measurement request that has not been started (i.e., an operation that has been requested with non-zero Delay parameter and for which the delay has not expired).

If the IMD Server cannot perform the Abort operation for any reason, then the IMD Server shall return an error response with the error code set to Request Not Supported (see Volume 3, Part F, Section 3.4.1.1 in [4]).

3.9 IMD Historical Data characteristic

The IMD Historical Data characteristic provides a mechanism for transferring a set of IMD Historical Records (see Section 3.9.2) to the IMD Client.

Several IMD Historical Records should be combined in one transmission to optimize over-the-air performance using a series of notifications.

3.9.1 Characteristic format

When notified, the structure of the IMD Historical Data characteristic is defined in Table 3.32.

Field	Data Type	Size (in octets)	Description
IMD Historical Record 1	struct	variable	First record (see Section 3.9.2.5.1)
…	-	-	-
IMD Historical Record N	struct	variable	Nth record

Table 3.32: IMD Historical Data characteristic

3.9.2 IMD Historical Record

Several types of IMD Historical Records that may be reported by the IMD Server are defined in this section.

The IMD Server shall keep a circular buffer with IMD Historical Records, deleting the older records when newer records become available. The buffer shall have space for at least one IMD Historical Record. The number of IMD Historical Records supported by the IMD Server is manufacturer specific.

The first four fields should be the same for all record types and should include the Segmentation Header, the Record Sequence Number, the Record Timestamp, and the Record Type.

A 1-byte segmentation scheme is used to support IMD Historical Records that do not fit in a single Attribute Protocol Maximum Transmission Unit (MTU).

3.9.2.1 IMD Historical Record format

The first segment of the IMD Historical Record has the format defined in Table 3.33. If the IMD Historical Record does not fit into a single MTU, the format of the subsequent segments of the IMD Historical Record is depicted in Table 3.34.

Field	Requirement	Data Type	Size (in octets)	Description
Segmentation Header	M	struct	1	See Section 3.9.2.2
Record Sequence Number	M	uint24	3	See Section 3.9.2.3
Record Timestamp	M	Elapsed Time (see [1])	9	Timestamp in Elapsed Time format (see Section 3.9.2.4)
Record Type	M	uint8	1	Type of Record 0x00 = IMD Service Cycle Data Record 0x01 = IMD Work Cycle Data Record 0x02-0xFF are RFU
Historical Record Body Segment	M	struct	variable	A segment of a Historical Record Body structure as defined in Section 3.9.2.5 Segmentation is defined in Section 3.9.3.1.

Table 3.33: Format of the first segment of the IMD Historical Record

Field	Requirement	Data Type	Size (in octets)	Description
Segmentation Header	M	struct	1	See Section 3.9.2.2
Historical Record Body Segment	M	struct	variable	A segment of a Historical Record Body structure as defined in Section 3.9.2.5 Segmentation is defined in Section 3.9.3.1.

Table 3.34: Format of additional segments of the IMD Historical Record

3.9.2.2 Segmentation Header

Table 3.35 defines the values of the Segmentation Header field. The server shall use these values to provide information to the client about the segments it needs to concatenate to get a complete IMD Historical Record.

Bit number	Definition
0	**First Segment:** The characteristic contains the first segment of content that should be concatenated by the client. 0 = False 1 = True
1	**Last Segment:** The characteristic contains the last segment of content that should be concatenated by the client. 0 = False 1 = True
2-7	Rolling Segment Counter: 0 to 63 If the Rolling Segment Counter is equal to 63, then the counter shall roll over to 0 when it is next incremented.

Table 3.35: Segmentation Header structure

3.9.2.3 Record Sequence Number field

The value of the Sequence Number starts at 0x000000 and increments by 1 with each record. After the Sequence Number reaches 0xFFFFFF, it shall roll over to 0x000000 the next time it is incremented. Refer to Section 3.10.2.6.1 for requirements related to the handling of a rollover when using the Record Access Control Point (RACP).

3.9.2.4 Record Timestamp field

The Record Timestamp field contains the time when the record was saved in the Elapsed Time characteristic format (see [1]). The meaning of the field may be different for different record types. For the record type 0x00 IMD Service Cycle Data Record, the Timestamp field shall be set to the time when the service was completed. For the record type 0x01 IMD Work Cycle Data Record, the Timestamp field shall be set to the time when the work cycle was started.

3.9.2.5 Historical Record Segment Body

The Segmentation Header field, the Record Sequence Number field, the Record Timestamp field, and the Record Type field are used in both the IMD Service Data Record and the IMD Work Cycle Data Record.

3.9.2.5.1 IMD Service Data Record Segment Body

The IMD Server shall create a single IMD Service Data Record for each service event. The structure of the IMD Service Data Record field is defined in Table 3.36.

Field	Data Type	Size (in Octets)	Description
Service Cycle Index	uint16	2	Service Cycle 0 is started at First Use Date (see Section 3.4) and increases by a value of 1 after the service has been performed.

Field	Data Type	Size (in Octets)	Description
Actual Use Time	uint24	3	This field specifies the amount of time in hours the IMD executed Work Cycles during the Service Cycle.
Work Cycle Counter	uint24	3	This field specifies the number of Work Cycles the IMD executed during the Service Cycle.
Date Of Expected Next Service	uint16	2	This field specifies the date when the next service is expected in number of days since 2000-01-01.

Table 3.36: IMD Service Data Record Segment Body field

3.9.2.5.2 IMD Work Cycle Record Segment Body

The IMD Work Cycle Record Segment Body field provides information about the Work Cycles executed on the IMD. The IMD Server shall create an IMD Measurement Data Record for each IMD Measurement characteristic for each work cycle executed by the IMD.

Field	Data Type	Size (in octets)	Description
Work Cycle Index	uint24	3	Work Cycle Index starts at a value of 0 and increases by a value of 1 after a work cycle is completed.
Work Cycle Duration	uint24	3	Active time since the start of the work cycle in ms.
Number of Entries	uint8	1	Number of IMD Measurements in the IMD Work Cycle Record
IMD Measurement UUID N	uint16	2	UUID of the Nth IMD Measurement
Sampling Function N	uint8	1	Sampling Function applied to the measured value (see Section 3.1.2.1.2)
Description N	uint16	2	Description of the measured value (see Section 3.1.2.1.5)
Measured Value Status N	boolean[16]	2	Measured value status (see Section 3.2.1.1)

Field	Data Type	Size (in octets)	Description
IMD Measurement Size N	uint8	1	Size in bytes of the Nth measured value
IMD Measurement N	variable	variable	Measured value for specified UUID, Sampling function, and Description

Table 3.37: IMD Work Cycle Record format

The IMD Work Cycle Record may contain several measurements. The IMD manufacturer selects which measurement data is included in the Historical Data Record for a work cycle. For example, a record may contain the Minimum and Maximum measurements recorded during the work cycle. If so, then the Historical Data Record contains two values. The IMD Measurement UUID and the Description for both values are the same, but the Sampling function is 0x04 and 0x05 for the minimum and maximum values.

3.9.3 Characteristic behavior

When the IMD Historical Data characteristic is enabled for notifications and the RACP is enabled for indications, the IMD Server shall notify the IMD Historical Data characteristic when it receives a RACP request that results in a non-empty set of IMD Historical Data Records to be reported.

The number of IMD Historical Data Record fields that are included in a notification depends on the number of the IMD Historical Records requested by the client and on the MTU. The IMD Server should increase message efficiency by filling each message with as many full IMD Measurement Data Records as possible, not exceeding ATT_MTU -4.

If the number of IMD Historical Data Records presented on the IMD and as requested by the client using the Record Access Control Point exceeds the number of records that can fit into a single notification, then multiple notifications shall be sent to transfer each of the records.

3.9.3.1 Segmentation

If the total size of a generated IMD Historical Record is greater than ATT_MTU -4, then multiple segments shall be sent to transfer a single Historical Record (the data to be transported).

When the IMD Historical Data characteristic is configured for notifications via the Client Characteristic Configuration descriptor and when a Historical Record is available, then data shall be notified per segment as described in the following steps.

Each of the IMD Historical Records needs to be transported in one or more segments. The number of the segments shall be calculated by dividing the size of the IMD Historical Record by ATT_MTU -4, rounded up to the nearest integer. N represents this number in the following steps.

For every IMD Historical Record that needs to be sent:

1. If N=1, then the First Segment and the Last Segment bits of the Segmentation Header field shall both be set to a value of 1.
2. If N>1, then the First Segment bit of the Segmentation Header field shall be set to a value of 1 and the Last Segment bit shall be set to a value of 0.
3. The Rolling Segment Counter shall be set to the current value maintained by the server.

4. Up to ATT_MTU -4 octets of the data to be transported shall be used to fill the characteristic message.
5. The IMD Historical Data characteristic shall be notified.
6. If the value of the Last Segment bit is 1, then continue to the next IMD Historical Record; otherwise continue to step 7.
7. The First Segment bit of the Segmentation Header field shall be set to a value of 0.
8. If the number of octets remaining to be sent can be sent in one more notification of the IMD Historical Data characteristic, then the Last Segment bit of the Segmentation Header field shall be set to a value of 1; otherwise it shall remain set to a value of 0.
9. The value of the Rolling Segment Counter shall be incremented by 1.
10. The next octets of the contents of the data to be sent shall be used to fill the characteristic message, in order, consisting of up to ATT_MTU -4 octets.
11. Repeat from step 4.

When the service is initialized (e.g., upon connection to the client), then the Rolling Segment Counter may be set to any number in the range of 0 to 63. The Rolling Segment Counter should then be incremented continuously without being reset each time the above procedure is used. When the Rolling Segment Counter reaches 63, then the Rolling Segment Counter shall roll over to 0 the next time it is incremented. By not resetting this counter, the Rolling Segment Counter has different values for consecutive records sent. Therefore, data can be structured on the client side without reading the Record Sequence numbers explicitly.

3.10 Record Access Control Point

The RACP as defined in GSS [1] shall be used with IMDS.

For IMDS to operate, the client configures the RACP characteristic for indications (i.e., via the Client Characteristic Configuration descriptor).

Within the context of IMDS, a record refers to the IMD Historical Record defined in Section 3.9.2. One or more IMD Historical Records may be transferred between the server and a client by sending notifications of the IMD Historical Data characteristic. The IMD implementing the server shall store IMD Historical Records for retrieval by an authorized client.

3.10.1 RACP procedure requirements

Table 3.38 and Table 3.39 show the requirements for the RACP procedures (Op Codes, Operators, and Operands) in the context of IMDS.

Op Code	Op Code Requirement	Operator	Operator Requirement	Operand			Operand Requirement
				Record Type (see Table 3.41)	**Filter Type (see Table 3.42)**	**Filter Parameters (see Table 3.40)**	
Delete Stored Records (0x02)	O	All records	M	<Record Type>	No Operand Used		N/A
		Less than or equal to	M	<Record Type>	Sequence Number	<maximum filter value>	M
				<Record Type>	Timestamp	<maximum filter value>	O
		Greater than or equal to	O	<Record Type>	Sequence Number	<minimum filter value>	M
				<Record Type>	Timestamp	<minimum filter value>	O
		Within range of (inclusive)	O	<Record Type>	Sequence Number	<minimum filter value>, <maximum filter value>	M
				<Record Type>	Timestamp	<minimum filter value>, <maximum filter value>	O
		First record	O	<Record Type>	No Operand Used		N/A
		Last record	O	<Record Type>	No Operand Used		N/A
Abort Operation (0x03)	M	Null (0x00)	M	No Operand Used			N/A
Report Number of Stored Records (0x04)	M	All records	M	<Record Type	No Operand Used		N/A
		Less than or equal to	O	<Record Type>	Sequence Number	<maximum filter value>	M
				<Record Type>	Timestamp	<maximum filter value>	O
		Greater than or equal to	M	<Record Type>	Sequence Number	<minimum filter value>	M
				<Record Type>	Timestamp	<minimum filter value>	O
		Within range of (inclusive)	O	<Record Type>	Sequence Number	<minimum filter value>, <maximum filter value>	M
				<Record Type>	Timestamp	<minimum filter value>, <maximum filter value>	O
		First record	O	<Record Type>	No Operand Used		N/A
		Last record	O	<Record Type>	No Operand Used		N/A

Op Code	Op Code Requirement	Operator	Operator Requirement	Operand			Operand Requirement
				Record Type (see Table 3.41)	**Filter Type (see Table 3.42)**	**Filter Parameters (see Table 3.40)**	
Combined Report (0x07)	M	All records	M	<Record Type>	No Operand Used		N/A
		Less than or equal to	M	<Record Type>	Sequence Number	<maximum filter value>	M
				<Record Type>	Timestamp	<maximum filter value>	O
		Greater than or equal to	M	<Record Type>	Sequence Number	<minimum filter value>	M
				<Record Type>	Timestamp	<minimum filter value>	O
		Within range of (inclusive)	M	<Record Type>	Sequence Number	<minimum filter value>, <maximum filter value>	M
				<Record Type>	Timestamp	<minimum filter value>, <maximum filter value>	O
		First record	M	<Record Type>	No Operand Used		N/A
		Last record	M	<Record Type>	No Operand Used		N/A

Table 3.38: RACP procedure requirements – requests

Notes:

1. If an optional Operator is not supported, then the corresponding Operand requirements are not applicable.
2. Support of a given Operand for one Op Code and Operator combination does not mean that other Op Code and Operator combinations support that Operand.
3. Support of a given Operator for one Op Code does not mean that other Op Codes support that Operator.
4. Where a Filter Type and filter parameters are used, the byte order for the Operand is specified in Section 3.10.2.2.

Responses					
Op Code	**Op Code Requirement**	**Operator**	**Operator Requirement**	**Operand**	**Operand Requirement**
Number of Stored Records Response (0x05)	M	Null (0x00)	M	uint32 containing the number of records based on the Operator and any Operand parameters(s) in the requested procedure	M
Response Code (0x06)	M	Null (0x00)	M	Request Op Code, Response Code value	M
Combined Report Response (0x08)	M	Null (0x00)	M	uint32 containing the number of records that were sent based on the Operator and any Operand parameter(s) in the requested procedure	M

Table 3.39: RACP procedure requirements – responses

Table 3.40 shows the relationships between RACP Operators and Filter Type and Filter Parameters Description of the Operands.

Procedure Operator	Filter Type and Filter Parameters Description
Null	Operand used only in the case of the Number of Stored Records Response and Response Code responses as shown in Table 3.39.
All records	No Operand used.
Less than or equal to	Operand represents Filter Type (see Table 3.42) and maximum field value.
Greater than or equal to	Operand represents the Filter Type and minimum field value.
Within range of (inclusive)	Operand represents the Filter Type and minimum field value, maximum field value pair.
First record	No Operand used.
Last record	No Operand used.

Table 3.40: RACP procedure Operator and Operand relationships

When using the “within range of” Operator, the minimum value of the range shall be less than or equal to the maximum value of the range regardless of the Filter Type used in the Operand.

The type of Record Type filter operand is uint8. Table 3.41 shows the Record Type that can be filtered for IMDS.

Value	Description
0x00	IMD Service Cycle Data Record
0x01	IMD Work Cycle Data Record
0x02 to 0xFF	RFU

Table 3.41: RACP Record Type values with IMDS

IMDS may not be able to process both IMD Service Cycle Data Records and IMD Work Cycle Data Records at the same time.

Table 3.42 shows the supported Filter Types that apply to three of the Operators listed in Table 3.40 (i.e., less than or equal to, greater than or equal to, and within range of). Within the Operand, the Filter Type specifies the field of the record upon which the filtering is based (see Section 3.10.2.2).

The type of Filter Types operand is uint8. Table 3.42 shows RACP Operands to enable customized filtering of IMD Historical Records.

Operand Filter Type Value	Filter Type Description for IMDS
0x00	RFU
0x01	Sequence Number
0x02	Timestamp
0x03 to 0xFF	RFU

Table 3.42: RACP Filter Types for use with IMDS

3.10.2 RACP behavioral description

The RACP shall be used to control notifications of IMD Historical Records with the IMD Historical Data characteristic. Procedures are triggered by a write to this characteristic that includes an Op Code specifying the operation (see Table 3.38 and Table 3.39) and an Operator and Operand that are valid within the context of that Op Code (see Table 3.40, Table 3.41, and Table 3.42). The RACP characteristic shall be indicated as described in Sections 3.10.2.1 to 3.10.2.6.

3.10.2.1 Record Type

The Record Type field defines which record type is requested from the RACP. The possible values are defined in Table 3.41.

3.10.2.2 Filter Types

A Filter Type field is defined to enable the flexibility to filter based on different criteria (e.g., Sequence Number or Timestamp).

Some Procedure Operators (i.e., less than or equal to, greater than or equal to, and within range of) require a Filter Type as part of the Operand. The Filter Type is used to specify the field in the IMD Historical Data record that performs the filtering. When used, the Filter Type field shall precede the applicable filter parameter(s) within the Operand. For example, when used with the “within range of”

Operator, the Operand has the format <Filter Type><minimum><maximum>, where Filter Type is the LSO of the Operand. See Table 3.42 for a list of valid Filter Type values.

When using the Sequence Number Filter Type, the format of the Operand is the Sequence Number Filter Type value followed by the applicable Sequence Number value or value pair, depending on the Operator. The format of the sequence number values used for filtering shall match the format of the Record Sequence Number field of the IMD Historical Record as defined in Section 3.9.2.3.

Filter parameters for Filter Type Time are 6-byte time values as specified in Elapsed Time in GSS [1]. Although the Timestamp may be used for filtering, using the Sequence Number is beneficial, considering the possibility of time faults that can cause discontinuities in the timeline of the server.

3.10.2.3 Delete Stored Records procedure

If the IMD Server supports the Delete Stored Records procedure, when the Delete Stored Records Op Code is written to the RACP, then the IMD Server shall delete the specified IMD Historical Records based on Operator and Operand values. The deletion of records from the database is permanent. Refer to Table 3.40 for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used.

The server shall indicate the RACP using the Response Code Op Code with a Response Code value of Success if the records are successfully deleted from the database.

If the server does not locate any records matching the filter criteria of the request, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to No Records Found.

If the operation results in an error condition, then this shall be indicated using the Response Code Op Code and the appropriate Response Code value in the Operand for the error condition.

IMD should keep valid records on the device if more than one client can bond with the server.

3.10.2.4 Abort Operation procedure

When the Abort Operation Op Code is written to the RACP, the server shall stop any RACP procedures in progress and shall stop sending any further data.

After the server stops all RACP procedures, the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Success.

If the operation results in an error condition, this shall be indicated using the Response Code Op Code and the appropriate Response Code value in the Operand for the error condition.

3.10.2.5 Report Number of Stored Records procedure

When the Report Number of Stored Records Op Code is written to the RACP, the server shall calculate and respond with a record count in uint32 format based on filter criteria, the Operator, and Operand values. Refer to Table 3.40 for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used. The record count reported in the response is calculated based on the current state of the database and may change between connections or after records are cleared. The response is indicated using the Number of Stored Records Response Op Code. If the operation results in an error condition, this shall be indicated using the Response Code Op Code and the appropriate Response Code value in the Operand for the error condition.

If the server does not locate any records matching the filter criteria of the request, then the server shall indicate the RACP with a Number of Stored Records Response Op Code and the Operand set to 0x0000. The No Records Found Response Code value shall not be used in this situation.

If the operation results in an error condition, this shall be indicated using the Response Code Op Code and the appropriate Response Code value in the Operand for the error condition.

3.10.2.6 Combined Report procedure

When the Combined Report Op Code is written to the RACP, the server shall notify the requested set of IMD Historical Records based on the filter criteria specified in the Operator and Operand. Refer to Table 3.40 for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used. When the server sends records to the client, it shall retain the original log entries.

The transfer of an IMD Historical Record shall include a notification of the IMD Historical Records within the IMD Historical Data characteristic (see Section 3.9.2).

If a new IMD Historical Record becomes available during the transfer (i.e., after the Combined Report procedure is initiated), then the server may include this new record in the record transfer.

When all IMD Historical Records for a given request have been notified by the IMD Server, the IMD Server shall indicate the RACP with a Combined Report Response Op Code, an Operator of Null, and an Operand equal to the number of records sent.

If the server does not locate any IMD Historical Records matching the filter criteria of the request, then the server shall indicate the RACP with a Response Code Op Code and a Response Code Value in the Operand set to No Records Found (see RACP in GSS [1]).

When the server is reporting stored IMD Historical Records, the server shall report records by order of first in, first out, with the oldest Sequence Number reported first.

If the operation results in an error condition, then this shall be indicated using a Response Op Code, an Operator of Null, a Request Op Code of Combined Report, and the appropriate Response Code value in the Operand for the error condition (see RACP in GSS [1]).

If the IMD Server interrupts its data transfer before completion for any reason other than the Abort Operation procedure (see Section 3.10.2.4), then the server shall indicate the RACP with a Response Code Op Code, an Operator of Null, a Request Op Code of Combined Report, and a Response Code value in the Operand set to Procedure Not Completed.

3.10.2.6.1 Discontinuity of time and sequence numbers

Because the timestamp can have discontinuities, the Sequence Number Filter Type should be used to request stored records along with the “greater than or equal to” Operator. However, it is also possible that the Sequence Number is discontinuous because of a prior Delete Records procedure or because the Sequence Number rolls over from 0xFFFFFF to 0x000000. To provide seamless operation when transmitting records, the server shall handle Sequence Number rollovers during an RACP request in a way that does not require client intervention.

If the client requests all records by Sequence Number using the “greater than or equal to” Operator, then the server shall transmit all records up to and including the record with a Sequence Number of 0xFFFFFF. The server shall continue to transmit any records starting at 0x000000 until there is a gap

between Sequence Numbers, indicating that the most recent IMD Historical Data Record has been reached and transmitted to the client.

3.10.3 General error handling procedures

Aside from error handling procedures that are specific to certain Op Codes, the following general procedures apply:

- If the Op Code that was written to the RACP characteristic is unsupported by the server, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Op Code Not Supported (see RACP in GSS [1]).
- If the Operator that was written to the RACP characteristic is invalid, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Invalid Operator.
- If the Operator that was written to the RACP characteristic is not supported by the server, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Operator Not Supported.
- If the Operand that was written to the RACP characteristic is invalid, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Invalid Operand.
- If the Record Type or the Filter Type within an Operand that was written to the RACP characteristic is not supported by the server, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Operand Not Supported.
- If the server is unable to complete a procedure for any reason not stated here, then the server shall indicate the RACP with a Response Code Op Code and a Response Code value in the Operand set to Procedure Not Completed.
- If a request with an Op Code other than Abort Operation is written to the RACP while the server is performing a previously triggered RACP operation (i.e., resulting from invalid client behavior), then the server shall return an error response with the error code set to Procedure Already In Progress.
- If the Op Code that was written to the RACP characteristic requests notifications of the IMD Historical Data characteristic and the Client Characteristic Configuration descriptor of the IMD Historical Data characteristic is not configured for notifications, then the server shall return an error response with the error code set to Client Characteristic Configuration Descriptor Improperly Configured.
- If an Op Code is written to the RACP characteristic and the Client Characteristic Configuration descriptor of the RACP characteristic is not configured for indications, then the server shall return an error response with the error code set to Client Characteristic Configuration Descriptor Improperly Configured.

4 SDP interoperability

If IMDS is exposed over BR/EDR, then it shall have the following SDP record:

Item	Definition	Type	Value	Status
Service Class ID List	-	-	-	M
Service Class #0	-	UUID	«Industrial Measurement Device Service»	M
Protocol Descriptor List	-	-	-	M
Protocol #0	-	UUID	L2CAP	M
Parameter #0 for Protocol #0	PSM	uint16	PSM = ATT	M
Protocol #1	-	UUID	ATT	M
BrowseGroupList	-	-	PublicBrowseRoot*	M

Table 4.1: SDP Record

* PublicBrowseRoot shall be present; however, other browse UUIDs may also be included in the list.

5 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
ATT	Attribute Protocol
GATT	GATT Generic Attribute Profile
IMD	Industrial Measurement Device
IMDS	Industrial Measurement Device Service
L2CAP	Logical Link Control and Adaptation Protocol
LSO	Least Significant Octet
MSO	Most Significant Octet
MTU	Maximum Transmission Unit
PSM	Protocol/ Service Multiplexer
RACP	Record Access Control Point
RMS	Root Mean Square
UUID	UUID Universally Unique Identifier

Table 5.1: Acronyms and abbreviations

6 References

[1] GATT Specification Supplement, https://www.bluetooth.com/specifications/gss/

[2] Core Specification Supplement, Version 11 or later

[3] Bluetooth Assigned Numbers, https://www.bluetooth.com/specifications/assigned-numbers

[4] Bluetooth Core Specification, Version 5.4 or later

[5] IEC 60073:2002 Basic and safety principles for man-machine interface, marking and identification - Coding principles for indicators and actuators, https://webstore.iec.ch/publication/587