Wearable Exposure Notification Service

***Bluetooth*® Specification**

- **Revision Date:** 2020-12-03
- **Group Prepared By:** Exposure Notification Working Group

This document is a working draft of a potential Bluetooth specification. It is subject to change and Bluetooth SIG is not required to finalize or adopt it. Although the Bluetooth SIG Board of Directors has decided to publicly release this working draft, other draft specifications, including prior or subsequent versions of this document, are Bluetooth SIG Confidential Information, unless they have been approved for public release by the Bluetooth SIG Board of Directors. Bluetooth SIG Members must comply with the Bluetooth SIG Confidentiality Policy and not disclose any document that is Bluetooth SIG confidential to anyone that is not a member of the Bluetooth SIG.

Abstract:

This specification defines a standardized method to enable a non-Internet-connected wearable device to operate in a manner complementary with one or more deployed Client-based Exposure Notification Systems (ENSs), therefore enabling significantly more individuals to participate in an ENS. The methods defined in this specification are applicable for the containment of a wide variety of infections, including SARS-CoV-2.

Use of this specification is your acknowledgement that you agree to and will comply with the following notices and disclaimers. You are advised to seek appropriate legal, engineering, and other professional advice regarding the use, interpretation, and effect of this specification.

Use of Bluetooth specifications by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG and its members, including those agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Any use of this specification by a member that is not in compliance with the applicable membership and other related agreements is prohibited and, among other things, may result in (i) termination of the applicable agreements and (ii) liability for infringement of the intellectual property rights of Bluetooth SIG and its members. This specification may provide options, because, for example, some products do not implement every portion of the specification. Each option identified in the specification is intended to be within the bounds of the Scope as defined in the Bluetooth Patent/Copyright License Agreement ("PCLA"). Also, the identification of options for implementing a portion of the specification is intended to provide design flexibility without establishing, for purposes of the PCLA, that any of these options is a "technically reasonable non-infringing alternative."

Use of this specification by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement of the intellectual property rights of Bluetooth SIG and its members. The furnishing of this specification does not grant any license to any intellectual property of Bluetooth SIG or its members. THIS SPECIFICATION IS PROVIDED "AS IS" AND BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES AND DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR THAT THE CONTENT OF THIS SPECIFICATION IS FREE OF ERRORS. For the avoidance of doubt, Bluetooth SIG has not made any search or investigation as to third parties that may claim rights in or to any specifications or any intellectual property that may be required to implement any specifications and it disclaims any obligation or duty to do so.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES DISCLAIM ALL LIABILITY ARISING OUT OF OR RELATING TO USE OF THIS SPECIFICATION AND ANY INFORMATION CONTAINED IN THIS SPECIFICATION, INCLUDING LOST REVENUE, PROFITS, DATA OR PROGRAMS, OR BUSINESS INTERRUPTION, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, AND EVEN IF BLUETOOTH SIG, ITS MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF THE DAMAGES.

Products equipped with Bluetooth wireless technology ("Bluetooth Products") and their combination, operation, use, implementation, and distribution may be subject to regulatory controls under the laws and regulations of numerous countries that regulate products that use wireless non-licensed spectrum. Examples include airline regulations, telecommunications regulations, technology transfer controls, and health and safety regulations. You are solely responsible for complying with all applicable laws and regulations and for obtaining any and all required authorizations, permits, or licenses in connection with your use of this specification and development, manufacture, and distribution of Bluetooth Products. Nothing in this specification provides any information or assistance in connection with complying with applicable laws or regulations or obtaining required authorizations, permits, or licenses.

Bluetooth SIG is not required to adopt any specification or portion thereof. If this specification is not the final version adopted by Bluetooth SIG's Board of Directors, it may not be adopted. Any specification adopted by Bluetooth SIG's Board of Directors may be withdrawn, replaced, or modified at any time. Bluetooth SIG reserves the right to change or alter final specifications in accordance with its membership and operating agreements.

Contents

1 Introduction

Exposure Notification Systems (ENSs) are being deployed around the world to help slow down and potentially contain SARS-CoV-2 outbreaks by detecting and notifying individuals about potential exposure. Within this specification, an ENS is defined as a system that notifies individuals if they have been in proximity of someone who was later diagnosed with an infectious disease such as SARS-CoV-2.

An ENS relies on Internet-connected devices that are carried by its users (a carried device) and that exchange (transmit and receive) ENS broadcast messages over Bluetooth Low Energy (LE). A carried device uses temporary keys (which generally rotate once per day) to generate Proximity Identifiers (PIs) (which generally change several times per hour) and includes these in transmitted broadcast messages. Carried devices also receive PIs in broadcast messages along with Received Signal Strength Indication (RSSI), which are used by the device to identify proximity events. The carried device periodically communicates with a cloud-based key server to determine whether any of its received PIs are amongst those associated with infected individuals. If there is a match, the individual is notified to self-isolate and test for the infectious disease. If an individual is positively diagnosed with the disease, the individual may authorize the carried device to transfer the temporary keys used to generate the recently broadcast PIs to a cloud-based key server. This will enable other individuals using the same system of ENS devices to be notified of potential exposure events.

Most ENSs are reliant on individuals owning and carrying compatible smartphones; however, a large part of the global population does not own a compatible smartphone and would be unable to participate in an ENS. This affects specific groups in society such as children and elderly people. Even individuals who own a compatible smartphone may not always carry or have access to it (e.g., during classroom sessions or when participating in team sports) and may be unable to participate during those times. Some countries or regions may issue wearable devices (e.g., wrist-worn or lanyard-attached) to everyone as an alternative to individuals using privately owned smartphones as part of an ENS.

The **Wearable Exposure Notification Service (WENS)** enables a wide variety of non-internet-connected wearable devices to participate in an ENS (see example in Figure 1.1). An internet-connected Client such as a smartphone (which is likely taking part in an ENS) uses the WENS to configure the wearable (Server) to participate in an ENS. Once configured, the Server advertises ENS data as required by the ENS, scans for ENS data from other devices using the same ENS, and stores the data in the form of ENS records. The WENS further enables an authorized Client to retrieve the stored ENS data from the Server at a later convenient time and perform exposure or risk calculations on the Server’s behalf. Because multiple types of ENSs have been deployed around the world, the WENS was designed to provide generic tools for a wide variety of ENSs to use. Wearables may also support multiple ENSs by supporting multiple instances of this service. Although this specification enables a Server to participate in an ENS, the requirements for ENS-specific operation are left to each supported ENS to define.

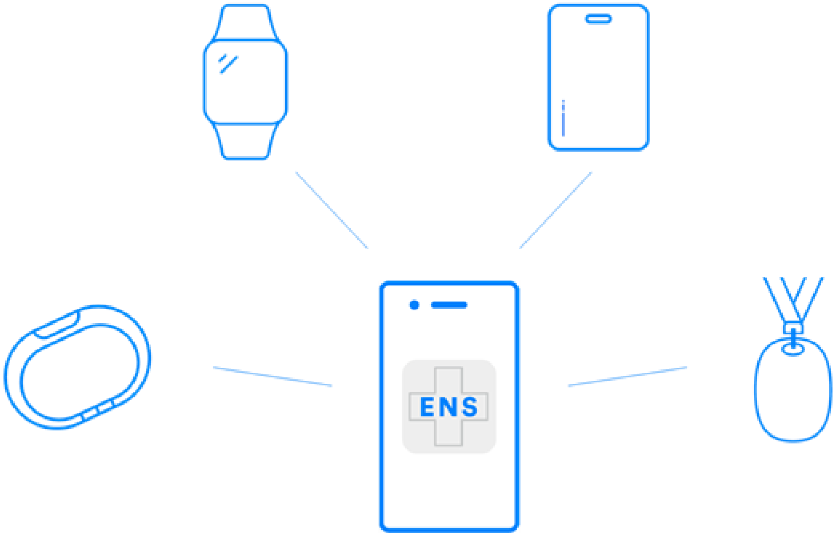


Figure 1.1: Examples of wearable devices using the WENS

1.1 Conformance

If conformance to this specification is claimed, all capabilities indicated as mandatory for this specification shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated.

1.2 Service dependencies

This service is dependent upon the Device Time Service (DTS) [8] to enable a Client to synchronize the Server’s clock to Coordinated Universal Time (UTC+0).

1.3 Bluetooth Core specification release compatibility

This specification is compatible with any Bluetooth Core Specification [1] that includes the Generic Attribute Profile (GATT) and the Bluetooth LE Controller portions.

To enable increasing the speed of data transfer, the Server should support the following Core Specification features:

- The 2Mbps PHY feature of Bluetooth Core Specification Version 5.0 and later [1]
- The LE Data Packet Length Extension feature of Bluetooth Core Specification Version 4.2 and later [1]

1.4 GATT sub-procedure requirements

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

Table 1.1 summarizes additional GATT sub-procedures required beyond those required by all GATT Servers.

GATT Sub-Procedure	**Server Requirements**
Write Characteristic Values	M
Notifications	M
Indications	M
Write Long Characteristic Values	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M
Exchange Maximum Transmission Unit (MTU)	M

Table 1.1: Additional GATT sub-procedure requirements

1.5 Transport dependencies

The ENS functionality (i.e., advertising and scanning) requires use of the Bluetooth LE transport. GATT transactions required by this specification shall be implemented using the Bluetooth LE transport and may additionally be implemented using the Basic Rate/Enhanced Data Rate (BR/EDR) transport.

1.6 Application error codes

This service does not define any Attribute Protocol Application error codes that are not already defined in Part B, Section 1 of the Core Specification Supplement (CSS) [3].

1.7 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 specification, 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 GSS Section 2.2 [4].

1.8 Language

1.8.1 Language conventions

The Bluetooth SIG has established the following conventions for use of the words ***shall***, ***must***, ***will***, ***should***, ***may***, ***can***, ***is***, and ***note*** in the development of specifications:

shall	<u>is required to</u> – used to define requirements.
must	is used to express: a natural consequence of a previously stated mandatory requirement. OR an indisputable statement of fact (one that is always true regardless of the circumstances).
will	<u>it is true that</u> – only used in statements of fact.

should	is recommended that – used to indicate that among several possibilities one is recommended as particularly suitable, but not required.
may	is permitted to – used to allow options.
can	is able to – used to relate statements in a causal manner.
is	is defined as – used to further explain elements that are previously required or allowed.
note	Used to indicate text that is included for informational purposes only and is not required in order to implement the specification. Each note is clearly designated as a “Note” and set off in a separate paragraph.

For clarity of the definition of those terms, see Core Specification Volume 1, Part E, Section 1.

1.8.2 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.

1.8.3 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.

2 Service declaration

2.1 Declaration

The Server shall support the Generic Access Profile (GAP) Peripheral role.

The service Universally Unique Identifier (UUID) shall be set to «Wearable Exposure Notification Service» as defined in [2].

This service shall be instantiated as a GATT Primary Service.

One instance of this service shall be exposed on a device for each supported ENS. If a device supports multiple ENSs, then multiple instances of this service shall be declared.

2.2 Service overview

This section provides an overview of the WENS.

This service uses two different types of advertisements:

- A 16-bit Service UUIDs AD Type, which is used for identification of a WENS device for initial connection
- ENS-specific service data, which is used once the device has been configured for use with a specific ENS and started

This service contains the following characteristics:

- ENS Log, which is used to transfer one or more logged data records using segmentation and reassembly to accommodate a wide range of ATT_MTU
- WEN Features, which contains the supported features of a device
- ENS Identifier, which identifies the ENS in use by the service and the supported ENS version
- ENS Settings, which allows customization of some ENS-specific settings
- Temporary Key List, which contains the temporary keys to be used in a sequence along with the time at which the Server will start to use each key
- ENS Advertisement List, which contains the advertising packets to be used in a sequence along with the time at which the Server will start to use each advertising packet
- Record Access Control Point (RACP), which is used for transferring logged information such as ENS data records or ENS-related events from a Server to a Client
- WEN Status, which is used for performing actions such as pausing or resuming an ENS or clearing all ENS data

This service also requires the DTS to:

- Expose the Server’s time and time status via the Device Time characteristic
- Expose the accuracy and maximum drift of the Server’s clock

- Allow a Client to manage updates to the device time of the Server
- Enable the Server to log and communicate changes to the device time, including time faults

This service also requires the Device Information Service (DIS) to expose device data.

3 Advertising and scanning

This service describes two types of advertisements: a 16-bit Service UUIDs AD Type (see Section 3.1) and ENS-specific service data (see Section 3.2).

3.1 Service UUIDs AD Type

While in a GAP Discoverable Mode for initial connection to a Client, the Server shall include the UUID «Wearable Exposure Notification Service» defined in [2] in either the «Incomplete List of 16-bit Service UUIDs» AD Type or «Complete List of 16-bit Service UUIDs» AD Type so long as there is sufficient space in the advertising packet. This enhances the user experience by enabling a Client to identify support of the WENS before initiating a connection to the Server.

3.1.1 Advertising behavior

For advertising and scanning parameters for connection establishment both before and after initial connection, refer to Volume 3, Part C, Section 9.3.11 of the Bluetooth Core Specification [1].

As required in Part A, Section 1.3.1 of the CSS [3], the Flags AD type must be included in the advertising packet when any of the Flag bits are non-zero and the advertising packet is connectable; otherwise, the Flags AD type is optional.

3.1.2 Bond management

Requirements relating to the retention and deletion of WENS data upon the creation and deletion of a bond are defined in the following subsections. Notwithstanding the requirements of these subsections, the Clear All ENS Data procedure in Section 4.9.2.3 or the RACP Delete Stored Records procedure (see Section 4.8.3.3) may be used to delete such data.

3.1.2.1 Devices that support a single bond

This subsection specifies requirements for devices that support bonding with only one Client.

The Server shall delete all ENS Records, the Temporary Key List, the ENS Advertisement List, all other logged records, and all WENS settings, upon deletion of the single bond.

Therefore, when the Server bonds with a new Client, any such data pertaining to a previously bonded Client is deleted.

3.1.2.2 Devices that support multiple bonds

This subsection specifies requirements for Servers that support bonding with multiple Clients concurrently.

When at least one Client is bonded and the Server bonds with a new Client that uses the same ENS, the ENS Records, the Temporary Key List, the ENS Advertisement List, all other logged records, and all WENS settings already established, shall be retained.

When a bond is deleted but at least one other Client remains bonded, the Server shall not delete such data.

However, upon deletion of the last bond, the Server shall delete all ENS Records, the Temporary Key List, the ENS Advertisement List, all other logged records, and all WENS settings.

3.2 ENS-specific advertising data

The format of ENS-specific advertising data is specified by each ENS. This type of advertisement is used after the Server has been configured by a Client and is in a mode of advertising ENS-specific advertising data. The role of this service specification in this respect is therefore limited to enabling and disabling the transmission and reception of the ENS-defined data. In addition, the settable Maximum Advertisement Duration value described in Section 4.5 enables this service to set a limit on the maximum duration for which an ENS advertisement may be used.

3.2.1 Advertising and scanning behavior

Once the device is bonded with a Client, settings are configured by the Client as needed (see Section 4.5), and the ENS is started by the Client, the Server shall advertise using the advertising packet format and interval as defined by that ENS and scan for advertising packets of the same ENS type using a scan interval and scan window as defined by that ENS. If the specific advertising and scan parameters are not defined for the Server by the ENS, suggested values are provided (see Section 4.5).

ENSs vary and the WENS enables three possible methods for a Server to be configured to advertise according to the requirements of the ENS. The first method makes use of the ENS Advertisement List characteristic (see Section 4.7); the other two methods make use of the Temporary Key List characteristic (see Section 4.6). The method used is determined by the requirements of the ENS.

- Method 1: The Client routinely writes a sequence of ENS advertising packets (and a rotation schedule) to the ENS Advertisement List characteristic for the Server to broadcast.
    - This method does not require the Server to generate its own keys or have a full understanding of how to generate ENS-compliant advertising packets.
    - See Section 3.2.1.1 for additional requirements that shall be applied when this method is used.
- Method 2: The Client routinely writes a sequence of temporary keys (and a rotation schedule) to the Temporary Key List characteristic, which the Server uses to generate ENS-compliant advertising packets.
    - This method does not require the Server to generate its own keys, but the Server does need to have an understanding of how to generate ENS-compliant advertising packets.
- Method 3: The Server can self-generate its own sequence of temporary keys (and a rotation schedule), which the Server uses to generate ENS-compliant advertising packets.
    - This method requires the Server to generate its own keys and have an understanding of how to generate ENS-compliant advertising packets.

The rotation schedule is as specified by the ENS, but PIs typically rotate several times an hour and temporary keys typically rotate once a day. The ENS-specific behavior is left to each supported ENS to define.

Regardless of the method used to configure the Server to advertise ENS-specific advertising data, it is important that the contents of the advertising are frequently changed to protect the privacy of the user. If the events described elsewhere in this specification that would normally cause the contents of the ENS advertisement to be updated do not occur for a period such that the Maximum Advertisement Duration expires, the Server shall cease sending ENS-specific advertising data. The Server may resume sending ENS-specific advertising data if new advertising data becomes available, e.g., through the use of Method 1, 2, or 3.

3.2.1.1 Additional requirements for Method 1

This subsection applies if the Server supports the ENS Advertisement List characteristic described in Section 4.7.

If supported, this characteristic enables a Client to write a schedule of pre-determined ENS advertising packets for the Server to advertise in the form of Timestamp-ENS Advertisement pairs. The purpose of this subsection is to define how the timing information provided by a Client shall then be used by the Server.

The Server shall start the use of each ENS advertisement only once the time represented by the associated timestamp has been reached and shall switch to the next ENS advertisement in sequence at the appropriate time.

However, this specification does not require that the ENS advertisement value rotations shall occur at the precise instant indicated by the nominal value included in a timestamp-value pair.

If the Server reaches the last ENS advertisement in its list, the Server shall continue to use the ENS advertisement only up until the end of life of the ENS advertisement which shall be a time equal to the value of the timestamp of the timestamp-value pair plus the value of the Maximum Advertisement Duration field of the ENS Settings characteristic.

3.2.2 Requirements for rotating Random Private Addresses (RPA)

Whenever an ENS advertisement is rotated, the Resolvable Private Address (RPA) shall also be rotated at the same advertising packet. Observing this requirement shall take precedence over any other inputs to the ENS advertisement timing calculations.

4 Service characteristics

This section defines requirements related to GATT characteristics and descriptors used by this service. Unless otherwise specified, only one instance of each characteristic in Table 4.1 is permitted within this service.

Where a characteristic can be indicated or notified, a Client Characteristic Configuration descriptor must be included in that characteristic as required by the Bluetooth Core Specification [1].

4.1 Summary

The characteristic requirements for this service are shown in Table 4.1.

All supported characteristics shall be set to Security Mode 1 and Security Level 2 with LE Secure Connections pairing or higher.

Characteristic Name	Requirement	Mandatory Properties	Optional Properties
ENS Log (see Section 4.2)	M	Notify	–
WEN Features (see Section 4.3)	M	Indicate, Read	–
ENS Identifier (see Section 4.4)	M	Indicate, Read	–
ENS Settings (see Section 4.5)	M	Indicate, Read, Write	–
Temporary Key List (see Section 4.6)	C.1	Read, Write (C.2)	–
ENS Advertisement List (see Section 4.7)	C.1	Write	–
Record Access Control Point (see Section 4.8)	M	Indicate, Write	–
WEN Status (see Section 4.9)	M	Indicate, Write	–

Table 4.1: Service characteristics summary

C.1: At least one, the Temporary Key List characteristic or the ENS Advertisement List characteristic, shall be supported.

C.2: The write property is Mandatory if the ENS Advertisement List characteristic is not supported and self-generation of temporary keys is not supported; otherwise excluded.

Properties not listed as Mandatory or Optional are excluded for this version of the service.

4.2 ENS Log

4.2.1 Description

The ENS Log characteristic provides a mechanism for transferring a set of ENS Records to the Client.

Segmentation is supported to allow a large amount of data to be transferred using a series of notifications. The ENS Log characteristic allows the Server to transfer a part of an ENS Record, one full ENS Record, or multiple ENS Records to a Client.

4.2.2 ENS Record definition

While scanning, the Server receives ENS advertising packets from devices advertising with the same ENS UUID as the Server is advertising (see Section 3.2.1), which it then stores as ENS Records. A Client can retrieve ENS Records using RACP procedures discussed in Section 4.8. The structure of an ENS Record is shown in Table 4.2.

Field	Data Type	Size (in octets)	Description
Sequence Number	uint24	3	See Section 4.2.2.1.
Timestamp	uint32	4	Identical to the format of the Base_Time field of Device Time characteristic defined in the DTS [8]. See Section 4.2.2.2.
Length	uint16	2	Length of the rest of the record in bytes.
LTV Structure	struct	variable (equal to the value of the Length field)	One or more LTV Types specified in Section 4.2.2.3.

Table 4.2: ENS Record structure

4.2.2.1 Sequence Number field

The value of the Sequence Number starts at 0x000000 and increments by 1 with each ENS Record. Once the Sequence Number reaches 0xFFFFFF, it shall roll over to 0x000000 the next time it is incremented. Refer to Section 4.8.3.4.1 for requirements related to the handling of a rollover when using the RACP. Refer to the Events Length Type Value (LTV) (see Section 4.2.2.3.1) for event logging requirements.

4.2.2.2 Timestamp field

The Timestamp field corresponds to the Base_Time field of the Device Time characteristic defined in the DTS [8] and uses the Epoch Year 2000 feature of the DTS. Therefore, this field represents the time in seconds since the epoch start time of 00:00:00 on January 1, 2000 (i.e., the first second of January 1, 2000) and is aligned to Coordinated Universal Time (UTC+0) as required by the DTS.

The time on the Server may either be updated via the Device Time Server role of the Device Time Profile [9], or via a user interface on the Server.

4.2.2.3 LTV Structure field

The LTV Structure field is made up of one or more concatenated LTV structures. LTV structures follow the format shown in Table 4.3. It is left to the ENS and implementation as to which LTV structures are used (if any), and the order of the LTVs.

Length **(1 octet)**	**Type** **(1 octet)**	**Value** **(variable; format varies by Type)**
The size of the Value field in octets	The value of the Type indicating what the Value field represents	Value sized in accordance with the Length and Type fields

Table 4.3: LTV Structure

The LTV Types defined for use with this service are shown in Table 4.4.

LTV Name	**Length**	**Type**	**Value Format**	**Notes**
ENS-specific Data	variable; specified by the ENS	0x00	variable; specified by the ENS	–
Events	0x01	0x01	uint8	See Section 4.2.2.3.1.
RSSI	0x01	0x02	sint8	Identical to the definition in the HCI_Read_RSSI event defined in Volume 4, Part E, Section 7.5.4 of the Bluetooth Core Specification [1]. This is in units of dBm measured at the antenna port.
Tx Power Level	0x01	0x03	sint8	The value received in the Tx Power Level AD Type when used. This is in units of dBm measured at the antenna port.
Pathloss	0x01	0x04	uint8	Equal to the Tx Power Level minus RSSI. See Note 1.
Distance Estimation	0x01	0x05	uint8	This is the estimated distance between two devices. Base unit is meters. Resolution per bit is 0.1 meter. Maximum value is 25.4 meters. 0xFF = Larger than 0xFE.
Temperature of Wearable User	0x02	0x06	sint16	Base unit is degrees Celsius. Resolution per bit is 0.1 degree Celsius.

LTV Name	Length	Type	Value Format	Notes
Pulse Rate of Wearable User	0x01	0x07	uint8	Base unit is beats per minute. Resolution per bit is 1 beat per minute.
Vendor-specific	variable	0x08	uint16, struct	16-bit Company ID from Assigned Values [2] followed by vendor-specific data.
RFU	–	0x09 to 0xFF	–	Others can be specified as needed.

Table 4.4: LTV Types

Note 1: Due to fading and varying antenna, circuit, and chip characteristics, the resulting pathloss value will have uncertainty. Some of the uncertainty (for example, due to fading) can be reduced if multiple packets are received from the same device.

4.2.2.3.1 Events LTV

The Events LTV shall be supported by Servers. The Server shall support at least one of the events defined in Table 4.5. When an event is supported, the requirements in the following subsections apply.

The Events LTV contains a 1-octet value that represents an event that is to be logged. However, the Events LTV shall not be used to log events related to the device time; for such cases, the Time Change Logging feature of the DTS should be used.

The values of this field are defined in Table 4.5.

Value	Definition
0	ENS Started
1	ENS Paused by Server
2	ENS Paused by Client
3	ENS Resumed by Server
4	ENS Resumed by Client
5	Sequence Number Rollover Occurred
6 to 255	RFU

Table 4.5: Events LTV field

4.2.2.3.1.1 Start, Pause, and Resume events

If the Server supports the ability to self-pause the ENS and self-resume the ENS (as indicated by the Self-Pause and Self-Resume Supported bit of the WEN Features characteristic defined in Section 4.3 being set to 1), the Server shall log the corresponding event.

When starting, pausing, or resuming are initiated by the Client via the Pause ENS or Resume ENS procedures (see Section 4.9.2.1 and Section 4.9.2.2), the Server shall log the corresponding event.

4.2.2.3.1.2 Sequence Number Rollover event

When the value of the Sequence Number rolls over from 0xFFFFFF to 0x000000 as described in Section 4.2.2.1, the Server shall log the corresponding event.

4.2.3 Definition

The structure of this characteristic is defined in Table 4.6.

Field	Data Type	Size	Description
Segmentation	struct	2 bits	Bits used for segmentation and reassembly (see Section 4.2.3.1).
Flags	struct	6 bits	See Section 4.2.3.2.
ENS Payload	struct	variable	A sequence of complete or segmented ENS Records (see Section 4.2.2 and Section 4.2.3.3).

Table 4.6: ENS Log characteristic

4.2.3.1 Segmentation field

The bits of the Segmentation field are defined in Table 4.7.

Value	Description
0b00	ENS Payload field contains complete (unsegmented) ENS Records
0b01	ENS Payload field contains the first segment of an ENS Record
0b10	ENS Payload field contains a continuation segment of an ENS Record
0b11	ENS Payload field contains the last segment of an ENS Record

Table 4.7: Segmentation field

Section 4.2.4 describes how the Segmentation field is used when the size of an ENS Record to be sent exceeds the capacity of a single ATT_HANDLE_VALUE_NTF PDU (i.e., when the value to be notified, including the header, would exceed ATT_MTU-3 octets in length).

4.2.3.2 Flags field

The bits of the Flags field are defined in Table 4.8.

Bit	Definition
2 to 7	RFU

Table 4.8: Flags field

4.2.3.3 ENS Payload field

The ENS Payload field of the ENS Log characteristic may contain part of an ENS Record, one full ENS Record, or multiple ENS Records according to the requirements in Section 4.2.4.

4.2.4 Behavior

When notifying ENS Records within the ENS Log characteristic using the RACP, the Server should avoid segmenting ENS Records when possible. When notifying a segmented ENS Record, the notification of the first segment of the ENS Record shall be followed by a notification for each subsequent segment in order and ending with the last segment of the ENS Record. The first segment sent shall contain the LSO of the ENS Record and the last segment sent shall contain the MSO of the ENS Record.

To maximize efficiency when notifying a large number of ENS Records using the RACP, the Server should negotiate a large ATT_MTU size so that one or more full ENS Records can be transferred in each notification. If that is not possible, the segmentation and reassembly of each record shall be performed as described below.

If the size of the ENS Record to be sent exceeds the capacity of a single ATT_HANDLE_VALUE_NTF PDU (i.e., when the value to be notified, including the header, would exceed ATT_MTU-3 octets in length), multiple notifications shall be sent to transfer each of the ENS Records.

When the transfer of large amounts of data is required, the Server should use a small connection interval where possible.

When notifying ENS Records within the ENS Log characteristic, three scenarios are possible:

- One ENS Record fits into a single notification
- Several ENS Records fit into a single notification
- One ENS Record is spread across multiple notifications

When one or more ENS Records completely fit into the ENS Payload field of a single notification, the value of the Segmentation field shall be set to 0b00.

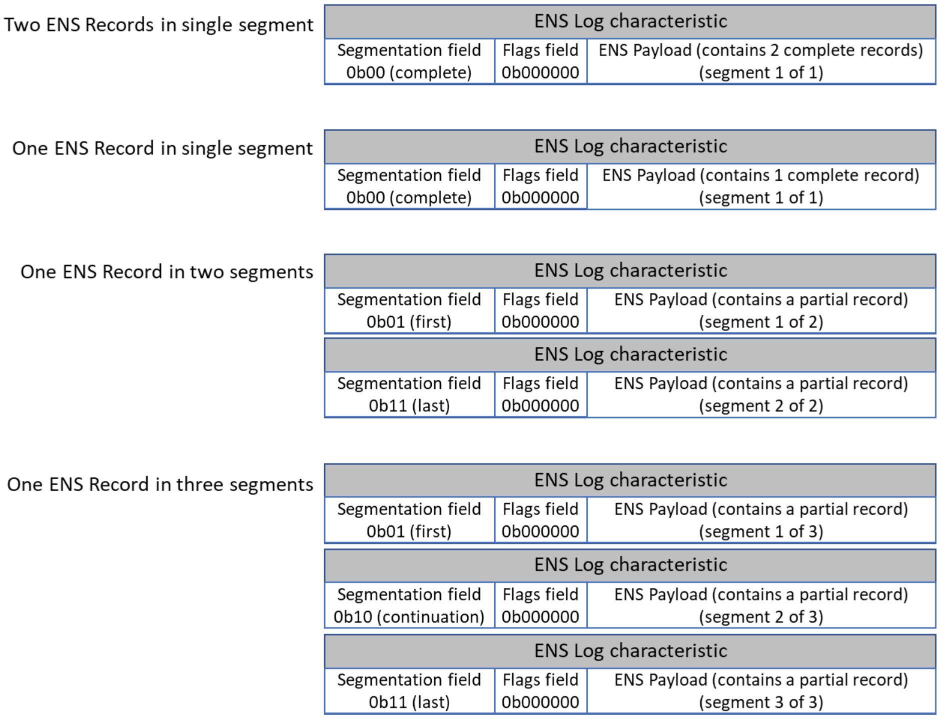
When the value of the Segmentation field is set to 0b00 and the Server has additional ENS Records to send, the next ENS Record in sequence shall be included in the ENS Payload field and sent in the same notification, if it fits entirely within the notification. Multiple ENS Records may be included in a single notification in this manner until there is insufficient space to add any more. When the space remaining in the ENS Payload field is insufficient to include the next ENS Record in sequence in its entirety, it shall not be included; the next ENS Record in sequence shall then be sent in the next notification.

When the Segmentation field value equals 0b01, the ENS Payload field shall contain only the first segment of an ENS Record and this segment shall contain the maximum possible number of octets.

If the ATT_MTU is not large enough for an ENS Record to fit within a single notification, then the first segment shall have the value of the Segmentation field set to 0b01, the continuation segment(s) (if any) shall have the value of the Segmentation field set to 0b10, and the last segment shall have the value of the Segmentation field set to 0b11.

When the Segmentation field is set to 0b10 (representing a continuation segment), this segment shall contain the maximum possible number of octets.

Figure 4.1 shows examples of ENS Log notifications where multiple ENS Records fit in a single segment, one ENS Record fits in a single segment, and ENS Records that needed to be segmented across multiple notifications.



Two ENS Records in single segment

ENS Log characteristic		
Segmentation field 0b00 (complete)	Flags field 0b000000	ENS Payload (contains 2 complete records) (segment 1 of 1)

One ENS Record in single segment

ENS Log characteristic		
Segmentation field 0b00 (complete)	Flags field 0b000000	ENS Payload (contains 1 complete record) (segment 1 of 1)

One ENS Record in two segments

ENS Log characteristic		
Segmentation field 0b01 (first)	Flags field 0b000000	ENS Payload (contains a partial record) (segment 1 of 2)

ENS Log characteristic		
Segmentation field 0b11 (last)	Flags field 0b000000	ENS Payload (contains a partial record) (segment 2 of 2)

One ENS Record in three segments

ENS Log characteristic		
Segmentation field 0b01 (first)	Flags field 0b000000	ENS Payload (contains a partial record) (segment 1 of 3)

ENS Log characteristic		
Segmentation field 0b10 (continuation)	Flags field 0b000000	ENS Payload (contains a partial record) (segment 2 of 3)

ENS Log characteristic		
Segmentation field 0b11 (last)	Flags field 0b000000	ENS Payload (contains a partial record) (segment 3 of 3)

Figure 4.1: Examples of ENS Log notifications with segmented and unsegmented ENS Records

4.3 WEN Features

4.3.1 Description

The WEN Features characteristic contains information about the supported features of the Server.

4.3.2 Definition

The structure of this characteristic is defined in Table 4.9.

Field	Data Type	Size (in octets)	Description
WEN Features	16-bit	2	See Section 4.3.2.1.
Approximate Total Storage Capacity	uint16	2	Approximate total number of ENS Records the device is capable of storing in multiples of 100 records. A value of 0xFFFF represents greater than 6,553,400 records. A value of 0x0000 is prohibited.

Table 4.9: WEN Features characteristic

4.3.2.1 WEN Features field

The bits of the WEN Features field are defined in Table 4.10.

Bit	Definition
0	Multiple Bonds Supported: 0 = False 1 = True
1	Self-Pause and Self-Resume Supported: 0 = False 1 = True
2	Self-Generation of Temporary Keys Supported: 0 = False 1 = True
3 to 15	RFU

Table 4.10: WEN Features field

4.3.2.2 Approximate Total Storage Capacity field

The Approximate Total Storage Capacity field represents the approximate total number of ENS Records the device is capable of storing in multiples of 100 records. This is a static value.

4.3.3 Behavior

If the Server supports multiple bonds, the Multiple Bonds Supported bit shall be set to 1; otherwise, it shall be set to 0.

If the Server supports the ability to pause itself (e.g., if the Server supports the ability to sense whether it is body worn, or the lack of movement over a period of time), the Self-Pause and Self-Resume Supported bit shall be set to 1; otherwise, it shall be set to 0.

If the Server supports the ability to self-generate its own temporary keys, the Self-Generation of Temporary Keys Supported bit shall be set to 1; otherwise, it shall be set to 0.

The value of the WEN Features characteristic may either be static for the lifetime of the device or static only during a connection. If enabled for indications, any change to the value of this characteristic shall be indicated by the Server to bonded Clients.

4.4 ENS Identifier

4.4.1 Description

The ENS Identifier characteristic represents the ENS in use with this service.

4.4.2 Definition

The structure of this characteristic is defined in Table 4.11.

Field	Data Type	Size (in octets)	Description
ENS UUID	uint16	2	See Section 4.4.2.1.
ENS Version	utf8	variable	See Section 4.4.2.2.

Table 4.11: ENS Identifier characteristic

4.4.2.1 ENS UUID field

The value of the ENS UUID field represents the ENS UUID associated with an instance of this service. Table 4.12 shows ENS UUIDs that may be used with this service, and the value of the ENS UUID field shall be a value from this table. To request that additional ENS UUIDs be added, please contact the Exposure Notification Working Group using the email address on the cover page of this specification.

This table will be moved to the Bluetooth SIG Assigned Values [2] before final publication.

UUID	Exposure Notification System	Notes
0xFD6F	Apple/Google protocol	ENS developed by Apple/Google [5]
0xFD64	ROBERT protocol	StopCovid ENS developed by INRIA and based on the ROBERT protocol [6]

Table 4.12: ENSs supported by the WENS

4.4.2.2 ENS Version field

The ENS Version field represents the version number of the ENS associated with an instance of this service and is specified by the ENS associated with an instance of this service.

4.4.3 Behavior

When read, the ENS Identifier characteristic returns the value of the ENS UUID field and the ENS Version field.

The value of the ENS Identifier characteristic may either be static for the lifetime of the device or static only during a connection. If enabled for indications, any change to the value of this characteristic shall be indicated by the Server to bonded Clients.

4.5 ENS Settings

4.5.1 Description

The ENS Settings characteristic represents the device settings specific to the ENS.

4.5.2 Definition

The structure of this characteristic is defined in Table 4.13.

The Server shall ultimately meet the requirements of the ENS when selecting ENS settings; however, when there are no such requirements for an ENS, values shown in the table as “suggested” may be used. When selecting values, the Server should consider the impact on memory and power consumption.

Field	**Data Type**	**Size (in octets)**	**Description**
Data Retention Policy	uint8	1	Data retention in days after which the ENS data shall be purged from memory. A value of 0 is Prohibited. The suggested value is 14 days.
Temporary Key Length	uint8	1	Size of the Temporary Key field of the Temporary Key List characteristic in octets (see Section 4.6). The maximum value is 16 octets. A value of 0 shall be used when the Server does not support the Temporary Key List characteristic. The suggested value is 16 octets.
Maximum Key Duration	uint16	2	The maximum time in minutes that a temporary key may be used (measured from the timestamp). A value of 0 is Prohibited. The suggested value is 1440 minutes (i.e., 24 hours).
ENS Advertisement Length	uint8	1	Size of the ENS Advertisement field of the ENS Advertisement Record in octets (see Section 4.7.2.3). The maximum value is 29 octets. A value of 0 shall be used when the Server does not support the ENS Advertisement List characteristic. The suggested value is 29 octets.
Maximum Advertisement Duration	uint8	1	The maximum time in minutes that an ENS advertisement may be used (measured from the timestamp). The suggested value is 10 minutes.
Scan On Time	uint8	1	Duration in which the Server scans for ENS advertisements. This value represents a multiple of 50 ms in the range 50 ms to 12.8 seconds (0x01 to 0xFF). A value of 0 is Prohibited. The suggested value is 300 ms (0x06).
Scan Off Time	uint16	2	Duration between scans for ENS advertisements.

Field	Data Type	Size (in octets)	Description
			This value represents a multiple of 1 second in the range 1 second to 65,535 seconds (0x0001 to 0xFFFF). A value of 0 is Prohibited. The suggested value is 60 seconds (0x003C).
Minimum Advertising Interval	uint16	2	Advertising interval when the Server is broadcasting ENS packets. This value represents a multiple of 0.625 ms in the range 20 ms to 10,485.759375 seconds (0x0020 to 0x4000). All other values are Prohibited. The suggested value is 200 ms (0x0140).
Maximum Advertising Interval	uint16	2	Advertising interval when the Server is broadcasting ENS packets. This value represents a multiple of 0.625 ms in the range 20 ms to 10,485.759375 seconds (0x0020 to 0x4000). All other values are Prohibited. This value shall be greater than or equal to the value of the Minimum Advertising Interval field. The suggested value is 270 ms (0x01B0).
Self-Pause and Self-Resume	uint8	1	0x00: Disabled 0x01: Enabled 0x02 to 0xFF: RFU The suggested value is Disabled (0x00)

Table 4.13: ENS Settings characteristic

4.5.3 Behavior

The Server shall expose the current values for each of the fields of this characteristic for the Client to read from and write to.

The value of the Data Retention Policy field represents the number of days after which the ENS data shall be purged from memory by the Server (typically 14 days or as otherwise specified by the ENS). The data to be purged includes temporary keys (see Section 4.6), the advertisement list (see Section 4.7), and the contents of the Events LTV (see Section 4.2.2.3.1).

The value of the Temporary Key Length field represents the size of the Temporary Key field of the Temporary Key List characteristic in octets (typically 16 bytes or as otherwise specified by the ENS).

The value of the Maximum Key Duration field represents the maximum duration in minutes that each temporary key is valid (typically 1440 minutes (i.e., 24 hours) or as otherwise specified by the ENS) before rolling to the next temporary key. The value of the Maximum Key Duration field shall be greater than the value of the Maximum Advertisement Duration field. If the Client attempts to write a value to the ENS Settings characteristic in which the value of the Maximum Key Duration field is less than the value of the Maximum Advertisement Duration field, the Server shall reject the write request by responding with the Attribute Protocol (ATT) error response “Write Request Rejected” (0xFC).

The value of the ENS Advertisement Length field represents the size of the ENS Advertisement field of the ENS Advertisement Record in octets (as specified by the ENS).

The Maximum Advertisement Duration field represents the maximum duration in minutes that each ENS advertisement is valid (typically 10 minutes or as otherwise specified by the ENS) before rolling to the next ENS advertisement. The value of the Maximum Advertisement Duration field shall be less than the value of the Maximum Key Duration field. If the Client attempts to write a value to the ENS Settings characteristic in which the value of the Maximum Advertisement Duration field is greater than the value of the Maximum Key Duration field, the Server shall reject the write request by responding with the ATT error response “Write Request Rejected” (0xFC).

The Scan On Time field represents the duration in which the Server scans for ENS advertisements (typically 300 ms or as otherwise specified by the ENS) before the next Scan Off Time. During the Scan On Time, the Server should scan as close to 100 percent duty cycle as possible or as otherwise specified by the ENS. If the Server is unable to scan at 100 percent duty cycle, the Server should increase the Scan On Time appropriately. The Scan Off Time field represents the duration between scans for ENS advertisements (typically 60 seconds or as otherwise specified by the ENS) before the next Scan On Time. The recommendations for Scan On Time and Scan Off Time are intended to provide balance between power consumption of the Server and the number of records needing to be stored.

Unless otherwise written to by the Client, the values of the advertising parameters (Minimum Advertising Interval and Maximum Advertising Interval) shall be as specified by the ENS.

If the value of the Self-Pause and Self-Resume Supported bit of the WEN Features characteristic is set to True, the value of the Self-Pause and Self-Resume field represents whether the Client allows the Server to enter a mode where it may pause and resume (likely due to power conservation features within the Server device) autonomously or whether such a mode is controlled only by the Client. If the value of the Self-Pause and Self-Resume Supported bit of the WEN Features characteristic is set to False, this field has no meaning and shall be set to 0x00 (Disabled).

If the Server is unable to accept a new value that the Client is attempting to write, the Server shall reject the write request and respond with an ATT error response “Write Request Rejected” (0xFC).

If enabled for indications, any change to the value of the ENS Settings characteristic shall be indicated to all bonded Clients except for the Client that made the change.

4.6 Temporary Key List

4.6.1 Description

The Temporary Key List characteristic represents a list of timestamp and temporary key pairs. If supported, this enables a Client to read temporary keys that have recently been used, and may be written to by the Client to provide a schedule of temporary keys for the Server to use when generating PIs.

4.6.2 Definition

The structure defined in Table 4.14 may be concatenated up to N times to transfer N timestamp and temporary key pairs as a single characteristic value where the value of N may range from 1 to 30.

Field	Data Type	Size (in octets)	Description
Timestamp	uint32	4	This field contains a 32-bit timestamp value representing the time when the corresponding temporary key was first used or will be used. This value corresponds to the Base_Time field of the Device Time characteristic defined in the DTS [8] and uses the Epoch Year 2000 feature of the DTS.
Temporary Key	variable	variable (equal to the value of the Temporary Key Length field of the ENS Settings characteristic)	This field contains an integer representing the value of the temporary key.

Table 4.14: Temporary Key List characteristic

4.6.3 Behavior

As with data records, timestamp and temporary key value pairs (timestamp-value pair) that have been used, and that are older than the time specified in the Data Retention Policy field of the ENS Settings characteristic, shall be purged from memory.

4.6.3.1 Behavior when read

When reading the Temporary Key List characteristic, the Server shall return the list of timestamp-value pairs within the data retention policy that were used, including the timestamp and temporary key in use. Temporary keys that will be valid in the future (i.e., those not yet used) shall not be returned because of privacy reasons.

Once the current temporary key has been read, it shall be replaced by one of two methods:

- If the Temporary Key List characteristic is writable, the Client writes a new value for the temporary key as described in Section 4.6.3.2.
- If the Temporary Key List characteristic is not writable, the Server generates a new value for the temporary key.

However, if the current temporary key has been read but has not been replaced, the Server shall stop advertising until the next temporary key which has not been read yet is scheduled to be used. If no such keys exist, the Server shall stop advertising until new keys are written by the Client or generated by the Server.

If the Server has a user interface, it should provide a mode of operation to allow the user to authorize the value of this characteristic to be read for a limited time.

4.6.3.2 Behavior when written

When writing to the Temporary Key List characteristic, the Client may include a new value to replace the temporary key currently in use (if the temporary key currently in use had been read as described in Section 4.6.3.1) and values for future temporary keys. When writing a new value for the current temporary key, the Client will include either an identical timestamp as the current temporary key being replaced or a timestamp equal to the current time. In either case, the Server shall only use the new temporary key as the current temporary key until the next temporary key is scheduled to be used or the Maximum Key Duration expires, whichever occurs first.

If the Server receives a timestamp-value pair with a timestamp that is older than the timestamp of the current temporary key, the Server shall ignore the timestamp-value pair and shall not store that timestamp-value pair.

If the Server receives a timestamp-value pair with a timestamp equal to the timestamp of a temporary key already in memory, the Server shall replace the value of that key.

If the Server receives a timestamp-value pair with a timestamp equal to the timestamp of the current temporary key already in memory, the Server shall immediately begin using the new temporary key to generate new PIs.

Once the Server receives new timestamp-value pairs, the Server shall use these new values along with any other timestamp-value pairs in memory that are still valid to generate new PIs and advertisements in accordance with the requirements of the ENS. The Server shall start the use of each temporary key only after the time represented by the associated timestamp, and shall switch to the next temporary key in sequence at the appropriate time. If the Server reaches the last temporary key in its list, the Server shall continue to use it only up until the end of life of the temporary key, which is equal to the value of the starting timestamp plus the value of the Maximum Key Duration field of the ENS Settings characteristic.

4.7 ENS Advertisement List

4.7.1 Description

The ENS Advertisement List characteristic provides a mechanism for transferring a list of ENS advertisements and their associated timestamps to the Server. If supported, this enables a Client to provide a schedule of pre-determined ENS advertising packets for the Server to advertise.

Segmentation is supported to allow a large amount of data to be transferred using a series of write requests. The ENS Advertisement List characteristic allows the Server to receive a part of an ENS Advertisement Record, one full ENS Advertisement Record, or multiple ENS Advertisement Records, from a Client.

4.7.2 Definition

The structure of this characteristic is defined in Table 4.15.

Field	Data Type	Size	Description
Segmentation	struct	2 bits	Bits used for segmentation and reassembly (see Section 4.7.2.1).
Flags	struct	6 bits	See Section 4.7.2.2.
Advertising Data Payload	struct	variable	A sequence of complete or segmented ENS Advertisement Records (see Section 4.7.2.3).

Table 4.15: ENS Advertisement List characteristic

4.7.2.1 Segmentation field

The bits of the Segmentation field are defined in Table 4.16.

Value	Description
0b00	Advertising Data Payload field contains the complete (unsegmented) ENS Advertisement Record
0b01	Advertising Data Payload field contains the first segment of an ENS Advertisement Record
0b10	RFU
0b11	Advertising Data Payload field contains the last segment of an ENS Advertisement Record

Table 4.16: Segmentation field

Section 4.7.3 describes how the Segmentation field is used when the size of an ENS Advertisement Record to be sent exceeds the capacity of a single write request (i.e., when the value to be written, including the header, would exceed ATT_MTU-3 octets in length).

4.7.2.2 Flags field

The bits of the Flags field are defined in Table 4.17.

Bit	Definition
2 to 7	RFU

Table 4.17: Flags field

4.7.2.3 Advertising Data Payload field

The Advertising Data Payload field of the ENS Advertisement List characteristic may contain part of an ENS Advertisement Record, one full ENS Advertisement Record, or multiple ENS Advertisement Record according to the requirements in Section 4.7.3. The structure of an ENS Advertisement Record is defined in Section 4.7.2.3.1.

4.7.2.3.1 ENS Advertisement Record

The ENS Advertisement Record structure is defined in Table 4.18 and may be concatenated up to N times to transfer N timestamp and ENS advertisement pairs along with an advertisement number for each pair.

Field	Data Type	Size (in octets)	Description
Index	uint16	2	This field contains a 16-bit index counter value that ranges from 0x0000 to 0xFFFF.
Timestamp	uint32	4	This field contains a 32-bit timestamp value representing the time when the corresponding ENS advertisement was first used or will be used. This value corresponds to the Base_Time field of the Device Time characteristic defined in the DTS [8] and uses the Epoch Year 2000 feature of the DTS.
ENS Advertisement	variable	variable (determined by the value of the ENS Advertisement Length field of the ENS Settings characteristic)	This field contains an ENS advertisement value.

Table 4.18: ENS Advertisement Record structure

4.7.3 Behavior

When writing to the ENS Advertisement List characteristic, the Client will typically only include future values for ENS advertisements with the exception that it may also include a new value for the current ENS advertisement. When writing a new value for the current ENS advertisement, the Client will either include an identical timestamp as the current ENS advertisement being replaced or a timestamp equal to the current time. In either case, the Server shall only use the new ENS advertisement as the current ENS advertisement until the next ENS advertisement is scheduled to be used or the Maximum Advertisement Duration expires, whichever occurs first.

The ENS Advertisement Record will also include an Index field, which will start at 0 and will increase by 1 with each ENS Advertisement Record. The Index value is reset to 0 for each subsequent data transfer session.

If the Server receives a timestamp and ENS advertisement value pair (timestamp-value pair) with a timestamp that is older than the timestamp of the current ENS advertisement, the Server shall ignore the timestamp-value pair and shall not store that timestamp-value pair.

If the Server receives a timestamp-value pair with a timestamp equal to the timestamp of an ENS advertisement already in memory, the Server shall replace the value of that ENS advertisement.

If the Server receives a timestamp-value pair with a timestamp equal to the timestamp of the current ENS advertisement already in memory, the Server shall begin using the new ENS advertisement value as described below.

Once the Server receives new timestamp-value pairs, the Server shall use these new values along with any other timestamp-value pairs in memory that are still valid to generate advertisements in accordance with the requirements of the ENS.

The Clear ENS Advertisement List procedure described in Section 4.9.2.4 may be used to clear the list prior to writing a new list.

When writing to the ENS Advertisement List characteristic, the Client will avoid segmenting ENS Advertisement Records when possible. When a segmented ENS Advertisement Record is written, the Client will write the first segment of the ENS Advertisement Record followed by writing subsequent segments in order and ending with the last segment of the ENS Advertisement Record. The first segment sent will contain the LSO of the ENS Advertisement Record and the last segment sent will contain the MSO of the ENS Advertisement Record.

To maximize efficiency when writing a large number of ENS Advertisement Records, the Server should negotiate a large ATT_MTU size so one or more full ENS Advertisement Records can be transferred with each write operation. If that is not possible, the segmentation and reassembly of each record will be performed as described below.

If the size of the ENS Advertisement Record to be sent exceeds the capacity of a single write request (i.e., when the value to be written, including the header, would exceed ATT_MTU-3 octets in length), multiple write operations will be required to transfer each of the ENS Advertisement Records.

When a Client writes ENS Advertisement Records within the ENS Advertisement List characteristic, three scenarios are possible:

- One ENS Advertisement Record fits into a single write operation

- Several ENS Advertisement Records fit into a single write operation
- One ENS Advertisement Record is spread across two write operations (e.g., if the ATT_MTU is 23 octets and the ENS Advertisement Record is 29 octets in size, an ENS Advertisement Record can be transmitted in two segments consisting of a first segment and a last segment)

When one or more ENS Advertisement Records completely fit into the Advertising Data Payload field of a single write operation, the value of the Segmentation field will be set to 0b00.

When the value of the Segmentation field is set to 0b00 and the Client has additional ENS Advertisement Records to send, the next ENS Advertisement Record in sequence will be included in the Advertising Data Payload field and sent in the same write operation, if it fits entirely within the write. Multiple ENS Advertisement Records may be included in a single write operation in this manner until there is insufficient space to add any more. When the space remaining in the Advertising Data Payload field is insufficient to include the next ENS Advertisement Record in sequence in its entirety, it will not be included by the Client; the next ENS Advertisement Record in sequence will then be sent in the next write operation.

When the Segmentation field value equals 0b01, the Advertising Data Payload field shall contain only the first segment of an ENS Advertisement Record.

If the ATT_MTU is not large enough for an ENS Advertisement Record to fit within a single write, then the first segment will have the value of the Segmentation field set to 0b01, and the last segment will have the value of the Segmentation field set to 0b11.

Figure 4.2 shows an example series of ENS Advertisement List write operations where the ENS Advertisement Records fit in a single write operation.

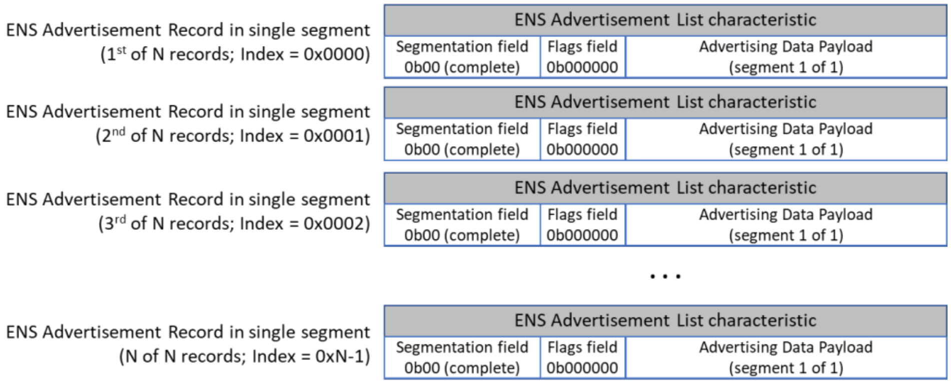

ENS Advertisement Record in single segment
(1st of N records; Index = 0x0000)
ENS Advertisement List characteristic
Segmentation field
0b00 (complete)
Flags field
0b000000
Advertising Data Payload
(segment 1 of 1)
ENS Advertisement Record in single segment
(2nd of N records; Index = 0x0001)
ENS Advertisement List characteristic
Segmentation field
0b00 (complete)
Flags field
0b000000
Advertising Data Payload
(segment 1 of 1)
ENS Advertisement Record in single segment
(3rd of N records; Index = 0x0002)
ENS Advertisement List characteristic
Segmentation field
0b00 (complete)
Flags field
0b000000
Advertising Data Payload
(segment 1 of 1)
. . .
ENS Advertisement Record in single segment
(N of N records; Index = 0xN-1)
ENS Advertisement List characteristic
Segmentation field
0b00 (complete)
Flags field
0b000000
Advertising Data Payload
(segment 1 of 1)

Figure 4.2: Example of ENS Advertisement List writes with unsegmented ENS Advertisement Records

Figure 4.3 shows an example series of ENS Advertisement List write operations where the ENS Advertisement Records require two write operations and segmentation.

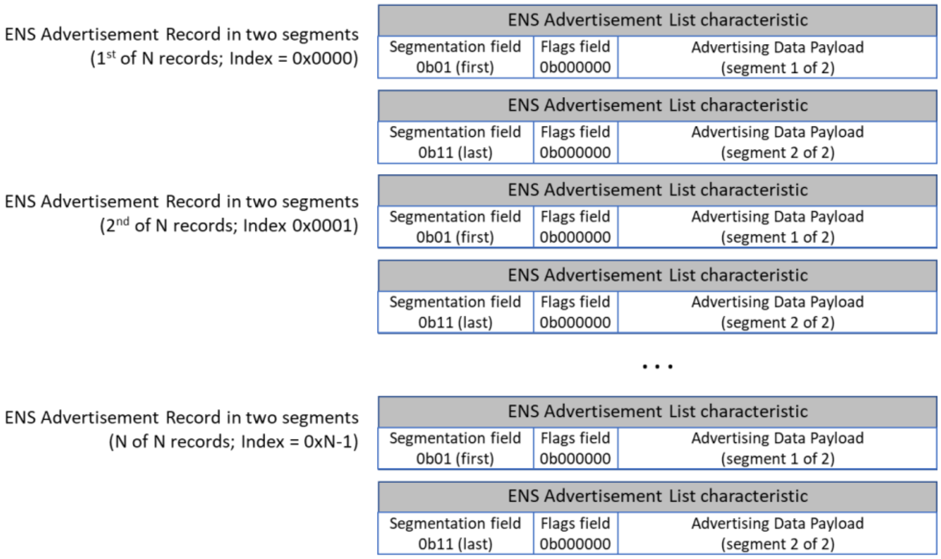

ENS Advertisement Record in two segments
(1st of N records; Index = 0x0000)
ENS Advertisement List characteristic
Segmentation field 0b01 (first)
Flags field 0b000000
Advertising Data Payload (segment 1 of 2)
ENS Advertisement List characteristic
Segmentation field 0b11 (last)
Flags field 0b000000
Advertising Data Payload (segment 2 of 2)
ENS Advertisement Record in two segments
(2nd of N records; Index 0x0001)
ENS Advertisement List characteristic
Segmentation field 0b01 (first)
Flags field 0b000000
Advertising Data Payload (segment 1 of 2)
ENS Advertisement List characteristic
Segmentation field 0b11 (last)
Flags field 0b000000
Advertising Data Payload (segment 2 of 2)
. . .
ENS Advertisement Record in two segments
(N of N records; Index = 0xN-1)
ENS Advertisement List characteristic
Segmentation field 0b01 (first)
Flags field 0b000000
Advertising Data Payload (segment 1 of 2)
ENS Advertisement List characteristic
Segmentation field 0b11 (last)
Flags field 0b000000
Advertising Data Payload (segment 2 of 2)

Figure 4.3: Example of ENS Advertisement List writes with segmented ENS Advertisement Records

The Server shall support the capacity to store at least 7 days of ENS Advertisement Records. For example, if an ENS specifies that PIs roll every 10 minutes, this would require the storage capacity for 144 ENS Advertisement Records per day or 1008 ENS Advertisement Records for a week. There is no requirement to store ENS Advertisement Records with expired timestamps.

If the Server receives a Segmentation field with the value set to the RFU value of 0b10, the Server shall reject the write request, respond with an ATT error response “Write Request Rejected” (0xFC), and ignore the ENS Advertisement Record containing the rejected segment.

If the Server is unable to accept a new value that the Client is attempting to write because the device memory is full, the Server shall reject the write request and respond with an ATT error response “Write Request Rejected” (0xFC).

4.8 Record Access Control Point

The RACP as defined in the GSS [4] shall be used with this service.

For convenience, a replication of details for the RACP from the GSS is provided in Appendix B (see Section B.1) but will be deleted before final publication.

For this service to operate, the Client must configure the RACP characteristic for indications (i.e., via the Client Characteristic Configuration descriptor).

4.8.1 Record definition

Within the context of this service, a record refers to the ENS Record defined in Section 4.2.2. One or more ENS Records may be transferred between the Server and a Client by sending notifications of the ENS Log characteristic. ENS Records may be segmented across multiple notifications when required, as described in Section 4.2.4. The device implementing the Server shall store ENS Records for retrieval by an authorized Client, but no longer than the data retention period specified by the Client in the Data Retention Policy field of the ENS Settings characteristic (see Section 4.5) or as otherwise specified by the ENS.

4.8.2 RACP procedure requirements

Table 4.19 and Table 4.20 show the requirements for the RACP procedures (Opcodes, Operators, and Operands) in the context of this service:

Opcode	Opcode Requirement	Operator	Operator Requirement	Operand		Operand Requirement
				Filter Type (see Table 4.22)	Filter Parameters (see Table 4.21)	
Report Stored Records (0x01)	Excluded	N/A	N/A	N/A		N/A
Delete Stored Records (0x02)	M	All records	M	No Operand Used		N/A
		Less than or equal to	M	Sequence Number	<maximum filter value>	M
				Timestamp	<maximum filter value>	O
		Greater than or equal to	O	Sequence Number	<minimum filter value>	M
				Timestamp	<minimum filter value>	O
		Within range of (inclusive)	O	Sequence Number	<minimum filter value>, <maximum filter value>	M
				Timestamp	<minimum filter value>, <maximum filter value>	O
		First record	O	No Operand Used		N/A
		Last record	O	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	No Operand Used		N/A
		Less than or equal to	O	Sequence Number	<maximum filter value>	M
				Timestamp	<maximum filter value>	O
		Greater than or equal to	M	Sequence Number	<minimum filter value>	M
				Timestamp	<minimum filter value>	O
		Within range of (inclusive)	O	Sequence Number	<minimum filter value>, <maximum filter value>	M

Opcode	Opcode Requirement	Operator	Operator Requirement	Operand		Operand Requirement
				Filter Type (see Table 4.22)	**Filter Parameters** (see Table 4.21)	
				Timestamp	<minimum filter value>, <maximum filter value>	O
		First record	O	No Operand Used		N/A
		Last record	O	No Operand Used		N/A
Combined Report (0x07)	M	All records	M	No Operand Used		N/A
		Less than or equal to	O	Sequence Number	<maximum filter value>	M
				Timestamp	<maximum filter value>	O
		Greater than or equal to	M	Sequence Number	<minimum filter value>	M
				Timestamp	<minimum filter value>	O
		Within range of (inclusive)	M	Sequence Number	<minimum filter value>, <maximum filter value>	M
				Timestamp	<minimum filter value>, <maximum filter value>	O
		First record	M	No Operand Used		N/A
		Last record	M	No Operand Used		N/A

Table 4.19: 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 Opcode and Operator combination does not imply support of that Operand for other Opcode and Operator combinations.
3. Support of a given Operator for one Opcode does not imply support of that Operator for other Opcodes.
4. Where a Filter Type and filter parameters are used, the byte order for the Operand is specified in Section 4.8.3.1.

Responses					
Opcode	**Opcode 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 Opcode, 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 4.20: RACP procedure requirements – responses

Table 4.21 shows the relationships between RACP Operators and Operands.

Procedure Operator	Operand Description
Null	Operand used only in the case of the Number of Stored Records Response and Response Code responses as shown in Table 4.20.
All records	No Operand used.
Less than or equal to	Operand represents Filter Type (see Table 4.22) and maximum field value.
Greater than or equal to	Operand represents Filter Type (see Table 4.22) and minimum field value.
Within range of (inclusive)	Operand represents Filter Type (see Table 4.22) and minimum field value, maximum field value pair.
First record	No Operand used.
Last record	No Operand used.

Table 4.21: 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.

Table 4.22 shows the supported Filter Types that apply to three of the Operators listed in Table 4.21 (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 ENS Record upon which the filtering is based. See Section 4.8.3.1 for further information.

RACP Operands to enable customized filtering of ENS Records are shown in Table 4.22.

Operand Filter Type Value	Filter Type Description for this Service
0x01	Sequence Number
0x02	Timestamp

Table 4.22: RACP Filter Types for use with this service

4.8.3 RACP behavioral description

The RACP shall be used to control notifications of ENS Records with the ENS Log characteristic. Procedures are triggered by a write to this characteristic that includes an Opcode specifying the operation (see Table 4.19 and Table 4.20) and an Operator and Operand that are valid within the context of that Opcode (see Table 4.21). The RACP characteristic shall be indicated as described in the following subsections.

4.8.3.1 Filter Types

A Filter Type field is defined to enable the flexibility to filter based on different criteria (e.g., Sequence Number, 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. This is used to specify the field in the ENS Log characteristic that is used to perform 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 4.22 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 upon the Operator.

When using the Timestamp Filter Type, the format of the Operand is the Timestamp Filter Type value followed by the applicable Timestamp value or value pair depending upon the Operator. While the Timestamp may be used for filtering, it is not as robust of a solution as the Sequence Number due to the possibility of time faults that can cause discontinuities in the timeline of the Server.

4.8.3.2 Report Number of Stored Records procedure

When the Report Number of Stored Records Opcode 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 4.21 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 Opcode. If the operation results in an error condition, this shall be indicated using the Response Code Opcode 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, the Server shall indicate the RACP with a Number of Stored Records Response Opcode and the Operand set to 0x0000.

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

4.8.3.3 Delete Stored Records procedure

When the Delete Stored Records Opcode is written to the RACP, the Server may delete the specified ENS Records based on Operator and Operand values. The deletion of records from the ENS Record database is permanent. Refer to Table 4.21 for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used.

The Server shall indicate this characteristic with a Response Code value of Success if the records were successfully deleted from the ENS Record database.

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

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

It is generally unnecessary to delete records after they are successfully transferred because they will be automatically deleted by the device at a later time based on the data retention policy. Leaving valid records on the device is useful if more than one Client can bond with the Server.

4.8.3.4 Combined Report procedure

When the Combined Report Opcode is written to the RACP, the Server shall notify the requested set of ENS Records based on the filter criteria specified in the Operator and Operand. Refer to Table 4.21 for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used. A record transfer is defined within this specification as transfer of a “copy” of the records and not a “move” of the records because the Server retains the original log entries.

The transfer of an ENS Record shall include a notification of the ENS Records (which may be segmented) within the ENS Log characteristic (see Section 4.2).

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

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

If the Server does not locate any ENS Records matching the filter criteria of the request, the Server shall indicate the RACP with a Combined Report Response Opcode, an Operator of Null, and a Response Code value in the Operand set to 0.

When the Server is reporting stored ENS Records, the Server shall report records by order of first in, first out (FIFO), with the oldest Sequence Number reported first.

If the operation results in an error condition, this shall be indicated using a Response Opcode, an Operator of Null, a Request Opcode of Combined Report, and the appropriate Response Code value in the Operand for the error condition.

If the Server interrupts its data transfer before completion for any reason other than the Abort Operation procedure (see Section 4.8.3.5), the Server shall indicate the RACP with a Response Code Opcode, an Operator of Null, a Request Opcode of Combined Report, and a Response Code value in the Operand set to Procedure Not Completed.

4.8.3.4.1 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 and 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, the Server shall transmit all records up to and including the record with a Sequence Number of 0xFFFFFF and then continue to transmit any records starting at 0x000000 and continuing until there is a gap between Sequence Numbers, indicating that the most recent ENS Record has been reached and transmitted to the Client.

4.8.3.5 Abort Operation procedure

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

Once all RACP procedures have been stopped, the Server shall indicate the RACP with a Response Code Opcode and 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 Opcode and the appropriate Response Code value in the Operand for the error condition.

4.8.4 General error handling procedures

Other than error handling procedures that are specific to certain Opcodes, the following apply.

If the Opcode that was written to the RACP characteristic is unsupported by the Server, the Server shall indicate the RACP with a Response Code Opcode and Response Code value in the Operand set to Opcode Not Supported.

If the Operator that was written to the RACP characteristic is invalid, the Server shall indicate the RACP with a Response Code Opcode and 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, the Server shall indicate the RACP with a Response Code Opcode and Response Code value in the Operand set to Operator Not Supported.

If the Operand that was written to the RACP characteristic is invalid, the Server shall indicate the RACP with a Response Code Opcode and Response Code value in the Operand set to Invalid Operand.

If the Filter Type within an Operand that was written to the RACP characteristic is not supported by the Server, the Server shall indicate the RACP with a Response Code Opcode and 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, the Server shall indicate the RACP with a Response Code Opcode and Response Code value in the Operand set to Procedure Not Completed.

If a request with an Opcode 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), the Server shall return an ATT error response set to “Procedure Already In Progress”.

If the Opcode that was written to the RACP characteristic requests notifications of ENS Records and the Client Characteristic Configuration descriptor is not configured for notifications, the Server shall return an ATT error response set to “Client Characteristic Configuration Descriptor Improperly Configured”.

4.8.5 Procedure timeout

In the context of the RACP characteristic, a procedure is started when the Client writes a particular Opcode to the RACP to perform some desired action, and ends when the Client sends the Handle Value Confirmation to acknowledge the RACP indication sent by the Server at the end of the procedure with the Opcode set to Response Code.

An RACP procedure is not considered started and not queued in the Server when a write to a control point results in an ATT error response.

An RACP procedure may consist of multiple characteristic notifications followed by an indication of the RACP characteristic.

After the Combined Report procedure described in Section 4.8.3.4 has been initiated and a notification has been received, the procedure is considered to have timed out if 30 seconds have elapsed since the most recent notification of the ENS Log characteristic. The timer shall be stopped when an RACP indication is received and the Opcode is set to Response Code. All other RACP procedures are considered to have timed out if an RACP indication is not received within 30 seconds from the start of the procedure.

If the link is lost while a control point operation is in progress, then the operation shall be considered to have timed out. If a control point operation times out, then no new control point operation shall be permitted until a new link is established.

If a timeout occurs, the Server shall stop sending any further indications and notifications related to the procedure and consider the procedure to have failed.

4.9 WEN Status

For this service to operate, the Client must configure the WEN Status characteristic for indications (i.e., via the Client Characteristic Configuration descriptor) at the first connection. The WEN Status characteristic behaves in a similar way as a control point except that it can be read by a Client. The structure of this characteristic is defined in Table 4.23.

Field	Data Type	Size (in octets)	Description
Opcode	uint8	1	See Section 4.9.1.1.
Parameter	struct	0 to 18	See Section 4.9.1.1.

Table 4.23: WEN Status characteristic structure

4.9.1 WEN Status procedure requirements

The procedures defined in this section are initiated when a Client uses the Write Characteristic Value procedure to write an Opcode (and applicable parameter, if any) defined in Table 4.24 to the Server.

The Server shall support all procedures defined in this section with the exception that the Clear ENS Advertisement List procedure is conditional upon support for the ENS Advertisement List characteristic.

4.9.1.1 Opcode and Parameter fields

The Opcode and Parameter fields are defined in Table 4.24.

Opcode Value	Definition	Parameter	Parameter Type	Description
0x00	RFU	N/A	N/A	N/A
0x01	Pause ENS (see Section 4.9.2.1)	N/A	N/A	Initiate the procedure to pause the ENS. The response to this control point is Opcode 0x20 followed by the appropriate Response Code value. Applicable Response Code values: Success, Operation Failed
0x02	Resume ENS (see Section 4.9.2.2)	N/A	N/A	Initiate the procedure to resume the ENS. The response to this control point is Opcode 0x20 followed by the appropriate Response Code value. Applicable Response Code values: Success, Operation Failed
0x03	Clear All ENS Data (see Section 4.9.2.3)	N/A	N/A	Clear logged data and settings of this instance of the WENS. The response to this control point is Opcode 0x20 followed by the appropriate Response Code value. Applicable Response Code values: Success, Operation Failed
0x04	Clear ENS Advertisement List (see Section 4.9.2.4)	N/A	N/A	Clear the ENS Advertisement List for this instance of the WENS. The response to this control point is Opcode 0x20 followed by the appropriate Response Code value. Applicable Response Code values: Success, Operation Failed
0x05 to 0x1F	RFU	N/A	N/A	N/A

Opcode Value	**Definition**	**Parameter**	**Parameter Type**	**Description**
0x20	Response Code	Request Opcode, Response Code value, Response Parameter	N/A	See Section 4.9.1.2.
0x21 to 0xFF	RFU	N/A	N/A	N/A

Table 4.24: WEN Status Opcode and Parameter fields

4.9.1.2 Response Code values

The Response Code values associated with the WEN Status are defined in Table 4.25.

Response Code Value	**Definition**	**Response Parameter**	**Description**
0x00	Success	uint8	Normal response for successful operation. Response parameter includes the current value of the ENS State as defined in Table 4.26.
0x01	Opcode Not Supported	None	Response if unsupported Opcode is received.
0x02	Invalid Operand	None	Response if parameter received does not meet the requirements of the service.
0x03	Operation Failed	uint8	Response if the requested procedure failed. Response parameter includes the current value of the ENS State as defined in Table 4.26.
0x04 to 0xFF	RFU	N/A	N/A

Table 4.25: WEN Status Response Code values

4.9.2 WEN Status behavioral description

The WEN Status is used by a Client to control certain behaviors of the Server. Procedures are triggered by a Write to this characteristic value that includes an Opcode specifying the operation (see Section 4.9.1), which may be followed by a parameter that is valid within the context of that Opcode. The WEN Status characteristic shall be indicated as described in the following subsections.

When read, the WEN Status characteristic returns the current value of the ENS State as defined in Table 4.26.

Field	Data Type	Size (in octets)	Description
ENS State	uint8	1	Current ENS State: 0x00: RFU 0x01: Stopped 0x02: Running 0x03 to 0xFF: RFU

Table 4.26: Values of WEN Status characteristic when read

4.9.2.1 Pause ENS

When the Pause ENS Opcode is written to the WEN Status characteristic, the Server shall pause the ENS so that it is not advertising ENS advertising packets or scanning for ENS advertising packets. The response shall be indicated when the value is applied using the Response Code Opcode, the Request Opcode along with “Success” or other appropriate Response Code value, and a Response Parameter as specified in Table 4.25. Refer to the Events LTV (see Section 4.2.2.3.1) for event logging requirements.

If the operation results in an error condition, see Section 4.9.2.4.

If there are no bonds associated with an instance of the service, or if the last bond associated with an instance of the service has been deleted, the ENS shall automatically be paused.

This procedure is used when the ENS is operating but not in use, when switching to another ENS device, or when switching to a different ENS.

4.9.2.2 Resume ENS

When the Resume ENS Opcode is written to the WEN Status characteristic, the Server shall start or resume the ENS so that it advertises and scans according to the requirements of the ENS. The response shall be indicated when the value is applied using the Response Code Opcode, the Request Opcode along with “Success” or other appropriate Response Code value, and a Response Parameter as specified in Table 4.25. Refer to the Events LTV (see Section 4.2.2.3.1) for event logging requirements.

If the operation results in an error condition, see Section 4.9.2.4.

By default, the ENS shall be paused.

This procedure is used to initially activate the ENS, when the ENS is paused but ready to be operational again, when switching back from another ENS device, or when switching back from another ENS.

4.9.2.3 Clear All ENS Data

When the Clear All ENS Data Opcode is written to the WEN Status characteristic, the Server shall clear all ENS Records, all logged events, all temporary keys, and all ENS Settings associated with an instance of this service and pause operation of the ENS. The response shall be indicated when the value is applied using the Response Code Opcode, the Request Opcode along with “Success” or other appropriate Response Code value, and a Response Parameter as specified in Table 4.25.

If the operation results in an error condition, see Section 4.9.2.4.

This procedure enables the Client to remove any user-specific logged data or settings associated with an instance of the service.

4.9.2.4 Clear ENS Advertisement List

When the Clear ENS Advertisement List Opcode is written to the WEN Status characteristic, the Server shall clear all ENS Advertisement List characteristic. The response shall be indicated when the value is applied using the Response Code Opcode, the Request Opcode along with “Success” or other appropriate Response Code value, and a Response Parameter as specified in Table 4.25.

If the operation results in an error condition, see Section 4.9.2.4.

This procedure is provided to enable the Client to remove any user-specific logged data or settings associated with an instance of the service.

4.9.3 General Error Handling procedures

Other than error handling procedures that are specific to certain Opcodes, the following apply.

If an Opcode is written to the WEN Status characteristic that is unsupported by the Server, the Server, after sending a Write Response, shall indicate the WEN Status with a Response Code Opcode, the Request Opcode, and the Response Code value set to Opcode Not Supported.

If a parameter is written to the WEN Status characteristic that is invalid (e.g., the Client writes the Update Sensor Location Opcode with a sensor location that is not valid in the context of the Server, or out of the supported range of the Server), the Server, after sending a Write Response, shall indicate the WEN Status with a Response Code Opcode, the Request Opcode, and the Response Code value set to Invalid Parameter.

If an Opcode is written to the WEN Status characteristic while the Server is performing a previously triggered WEN Status operation (e.g., resulting from invalid Client behavior), the Server shall return an ATT error response set to “Procedure Already In Progress”.

If an Opcode is written to the WEN Status characteristic and the Client Characteristic Configuration descriptor of the WEN Status is not configured for indications, the Server shall return an ATT error code set to “Client Characteristic Configuration Descriptor Improperly Configured”.

4.9.4 Procedure Timeout

In the context of the WEN Status characteristic, a procedure is started when a Client writes to the WEN Status characteristic and ends when the Client sends the Handle Value Confirmation to acknowledge the WEN Status indication sent by the Server at the end of the procedure, with the Opcode set to Response Code.

In the context of the WEN Status characteristic, a procedure is not considered started and not queued in the Server when a write to the WEN Status results in an ATT error response code.

A WEN Status procedure is not considered started and not queued in the Server when a write operation results in an ATT error response.

A WEN Status procedure is considered to have timed out if an indication of the WEN Status characteristic is not received within 30 seconds from the start of the write operation.

If the link is lost while a write operation is in progress, then the procedure shall be considered to have timed out. If a procedure times out, then no new write operation shall be permitted until a new link is established.

If a timeout occurs, the Server shall stop sending any further indications and notifications related to the procedure and consider the procedure to have failed.

5 Additional GATT services

This section defines requirements related to additional GATT services published by the Bluetooth SIG.

- Incremental DTS [8] requirements or recommendations:
  - Access to DTS characteristics shall have the same security level requirements as access to WENS characteristics or higher.
  - The Server shall enable a bonded Client to read the current value of the clock and update it to the current time using the DTS.
  - The Epoch Year 2000 feature of the DTS shall be supported.
  - A clock used for the timing of ENS advertisements should have an accuracy of better than or equal to 500 ppm.
  - The Time Change Logging feature of the DTS should be supported.
  - Clocks that have not been synchronized with an external time source by either the Server itself or through use of the DTS shall count upwards from 0 (representing Jan 1, 2000, 00:00:00 (UTC+0)).
  - Upon reset or loss of power, the clock resets to a value of 0 (representing Jan 1, 2000, 00:00:00 (UTC+0)).
- The DIS [10] shall be used to enable a Client to check common device data on the Server. As such, the following DIS characteristics are mandatory and the characteristics shall be readable with authentication:
  - Manufacturer Name String
  - Model Number String
  - Hardware Revision String
  - Firmware Revision String
- The Battery Service [7] should be used to enable a Client to check battery status on the Server.
- The Bond Management Service [11] should be used to enable a Client to delete bonds from the Server.

6 SDP interoperability

If this service 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	«Wearable Exposure Notification 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
Parameter #0 for Protocol #1	GATT Start Handle	Uint16	First handle of this service in the GATT database	M
Parameter #1 for Protocol #1	GATT End Handle	Uint16	Last handle of this service in the GATT database	M
BrowseGroupList			PublicBrowseRoot*	M

Table 6.1: SDP Record

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

7 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
ATT	Attribute Protocol
ATT_MTU	Attribute Protocol Maximum Transmission Unit
BR/EDR	Basic Rate / Enhanced Data Rate
CSS	Core Specification Supplement
DIS	Device Information Service
DTS	Device Time Service
ENS	Exposure Notification System
FRD	Functional Requirements Document
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GSS	GATT Specification Supplement
LE	Low Energy
LSO	least significant octet
LTV	Length Type Value
MSO	most significant octet
MTU	Maximum Transmission Unit
PDU	Protocol Data Unit
PI	Proximity Identifier
PSM	Protocol/Service Multiplexer
RACP	Record Access Control Point
RFU	Reserved for Future Use
RPA	Resolvable Private Address
RSSI	Received Signal Strength Indication
SIG	Special Interest Group
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
WENS	Wearable Exposure Notification Service

Table 7.1: Acronyms and abbreviations

8 References

[1] Bluetooth Core Specification, Version 4.0 and later

[2] Bluetooth SIG Assigned Values, https://www.bluetooth.com/specifications/assigned-numbers

[3] Core Specification Supplement

[4] GATT Specification Supplement, Version 2 or later

[5] Apple/Google, “Privacy-Preserving Contact Tracing,” https://www.apple.com/covid19/contacttracing, https://www.google.com/covid19/exposurenotifications

[6] ROBERT protocol, https://github.com/ROBERT-proximity-tracing/documents

[7] Battery Service, Version 1.0 or later

[8] Device Time Service

[9] Device Time Profile

[10] Device Information Service, Version 1.1 or later

[11] Wearable Extension to Exposure Notification System FRD

[12] Bond Management Service, Version 1.0 or later

Appendix A RACP procedure examples

This section contains informative examples for how to use the RACP procedures to perform common operations. Examples are shown in Table A.1.

How to:	Opcode	Operator	Operand
Request total number of ENS Records	0x04	0x01	0x05
Request number of ENS Records greater than specific Sequence Number value	0x04 <minimum>	0x03	0x01
Request number of ENS Records greater than specific Timestamp value	0x04 <minimum>	0x03	0x02
Request all ENS Records	0x07	0x01	0x00
Request all ENS Records greater than specific Sequence Number value	0x07 <minimum>	0x03	0x01
Request all ENS Records greater than specific Timestamp value	0x07 <minimum>	0x03	0x02
Request missing ENS Records based on missing Sequence Number values	0x07 <minimum> <maximum>	0x04	0x01
Request first (oldest) ENS Record	0x07	0x05	Not included
Request last (newest) ENS Record	0x07	0x06	Not included
Delete all ENS Records	0x02	0x01	0x00
Delete all ENS Records with Sequence Number less than a specific value	0x02 <maximum>	0x02	0x01
Delete all ENS Records older than (less than) specific Timestamp value	0x02 <maximum>	0x02	0x02

Table A.1: RACP procedure examples

Appendix B RACP reference material

This section contains an extract of the Record Access Control Point section of GSS [4] Version 2 for reference only and will be deleted before final adoption. Additional opcodes for use with the WENS specification are shown in red text within Table B.2.

B.1 Record Access Control Point

B.1.1 Description

This control point is used with a service to provide basic management functionality for a record database. This enables functions including counting records, transmitting records, and clearing records based on filter criterion. The filter criterion in the Operand field is defined by the service that references this characteristic, as is the format of a record (which may be comprised of one or more characteristics) and the sequence of transferred records.

B.1.2 Definition

The structure of this characteristic is defined in Table B.1.

Field	Data Type	Size (in octets)	Description
Opcode	uint8	1	See Table B.2
Operator	uint8	1	See Table B.3
Operand	struct	0–18	See Table B.4

Table B.1: Record Access Control Point characteristic

B.1.3 Opcode, Operator, and Operand/Filter fields

The Opcode values and associated Operator and Operand values are defined as shown in Table B.2.

Opcode Value	Definition	Operator	Operand	Description
0x00	Reserved for Future Use	N/A	N/A	N/A
0x01	Report stored records	Value from Operator table	Filter parameters (as appropriate to Operator and Service)	Following record transmission, the response to this control point is Opcode 0x06.
0x02	Delete stored records	Value from Operator table	Filter parameters (as appropriate to Operator and Service)	The response to this control point is Opcode 0x06.
0x03	Abort operation	Null	Not included	The response to this control point is Opcode 0x06.

Opcode Value	Definition	Operator	Operand	Description
0x04	Report number of stored records	Value from Operator table	Filter parameters (as appropriate to Operator and Service)	The normal response to this control point is Opcode 0x05. For error conditions, the response is Opcode 0x06.
0x05	Number of stored records response	Null	Number of Records (Field size defined by Service)	This is the normal response to Opcode 0x04.
0x06	Response Code	Null	Request Opcode, Response Code value	See Table B.5
0x07	Combined Report	Value from Operator table	Filter parameters (as appropriate to Operator and Service)	Following record transmission, the response to this control point is Opcode 0x08.
0x08	Combined Report Response	Null	Number of Records (Field size defined by Service)	This is the normal response to Opcode 0x07
0x09–0xFF	Reserved for Future Use	N/A	N/A	N/A

Table B.2: Record Access Control Point characteristic Opcode values

The Operator values are defined in Table B.3.

Operator Value	Definition	Operand Notes
0x00	Null	Varies by Opcode
0x01	All records	No Operand used
0x02	Less than or equal to	Operand contains at least a maximum value
0x03	Greater than or equal to	Operand contains at least a minimum value
0x04	Within range of (inclusive)	Operand contains at least a minimum value, maximum value pair
0x05	First record (i.e., oldest record)	No Operand used
0x06	Last record (i.e., most recent record)	No Operand used

Operator Value	Definition	Operand Notes
0x07–0xFF	Reserved for Future Use	N/A

Table B.3: Record Access Control Point characteristic Operator values

The operands and filter types (“Operand” column of Table B.2) correspond to the Opcode values (0x00–0xFF) defined in the Opcode field (also from Table B.2).

Key	Operand Value
0x00	N/A
0x01	Filter parameters (as appropriate to Operator and Service)
0x02	Filter parameters (as appropriate to Operator and Service)
0x03	Not included
0x04	Filter parameters (as appropriate to Operator and Service)
0x05	Number of Records (Field size defined per service)
0x06	Request Opcode, Response Code value
0x07	Filter parameters (as appropriate to Operator and Service)
0x08	Number of Records (Field size defined by Service)
0x09–0xFF	Reserved for Future Use

Table B.4: Opcode Operand/Filter Correspondence

The Response Code values associated with Opcode 0x06 are defined as follows:

Response Code Value	Definition	Description
0x00	Reserved for Future Use	N/A
0x01	Success	Normal response for successful operation.
0x02	Opcode not supported	Normal response if unsupported Opcode is received.
0x03	Invalid Operator	Normal response if Operator received does not meet the requirements of the service (e.g., Null was expected).
0x04	Operator not supported	Normal response if unsupported Operator is received.
0x05	Invalid Operand	Normal response if Operand received does not meet the requirements of the service.

Response Code Value	Definition	Description
0x06	No records found	Normal response if request for records resulted in no records meeting criteria.
0x07	Abort unsuccessful	Normal response if request for Abort cannot be completed.
0x08	Procedure not completed	Normal response if unable to complete a procedure for any reason.
0x09	Operand not supported	Normal response if unsupported Operand is received.
0x0A–0xFF	Reserved for Future Use	N/A

Table B.5: Record Access Control Point characteristic Response Code values

Appendix C Memory usage estimations

This section contains informative analysis of non-volatile memory usage in a WENS device. The analysis shows a few exemplary cases, and different devices may have differing requirements. Memory requirements for a device may vary based on intended usage, expected interval between uploads to the Client, and other factors.

C.1 Relevant factors for memory usage estimation

C.1.1 Record size

A record will be created and stored for each PI that is received. Each record will have data as specified in Table 4.2, including one or more LTVs from Table 4.4.

C.1.2 Scan interval and scan duration

As a wearable scans more frequently, and because it scans for a longer duration, it is likely that a wearable may receive more records, which would require more storage. For this analysis, it is assumed that a wearable typically will scan for roughly (N * advertising interval), so that it will typically receive N advertisements from each nearby transmitter during each scan window. N=1 may be the most common case, but other cases can be supported.

C.1.3 Average number of transmitters in range

The average number of nearby transmitters over a 24-hour period will impact the required storage. If more devices are nearby and considered to be in range, the storage requirements will increase. When considering the average number of transmitters in range, it may make sense to break the typical day into sections (home, commute, work, other) and how many people will be in range in each of these scenarios. It is also important to understand whether a minimum RSSI threshold will be applied to records to determine whether they are “in range,, especially for scenarios where a device may be able to receive advertisements that do not have a high enough RSSI to be within the exposure range (typically about 2 meters).

An example is shown below of a working adult who lives with a family of five people. From 7 p.m. to 7 a.m., this person is home and within range of at most four other devices, perhaps less while sleeping. The rest of the day involves activities with more people nearby, although even in crowded scenarios, there is a limit to how many people will typically be within a 2-meter radius. Even on crowded subways, a device held on the body will have a limit to how many other devices it can successfully receive.

	Transmitters in Range	Hours per Day
Home	4	12
Work	12	8
Commute	14	2
Social	10	2

Average: Eight receivable transmitters, on average, over a 24-hour period

C.1.4 Required days of storage

An ENS or a device may require a minimum number of days of record storage. Some devices may require 14 days of storage or more, while other wearables may be able to safely assume that they can periodically upload their data to a Client, and therefore may be able to support a shorter number of required days of storage.

The required days of storage for a wearable may be significantly shorter than for a phone or similar device in cases where the wearable reliably uploads data to the Client. Frequent uploads of data are typically encouraged to allow for quick notification if a potential exposure has occurred. A wearable worn by a child may be within range of a parent's phone multiple times per day, where the wearable can upload recent contact records, or an elderly user of a wearable in a care facility may be within range of an Internet-connected Client multiple times per day. In these cases, the ENS requirement for N days of storage can be met with storage on the parent's phone, or in the local cloud, allowing the wearable to operate with significantly less storage requirements, perhaps only 1 to 2 days of storage.

C.1.5 Compression

This specification defines the on-air transmission protocol for WENS operation, but it does not specify the format for record storage on a device. A device with constraints on memory size may choose to perform simple, lossless compression. As an example, consider a system with PIs that change every 15 minutes, a scan interval of 1 minute, and a scan duration similar to the advertising interval. In this case, a device might receive 15 advertisements with the same PI, which may require 15 nearly identical records. To save storage space, a device may choose to store one larger record for each PI, recording the RSSI and time of each, but only storing all other common information once. This can greatly reduce storage requirements without the loss of any information. When transferring this data wirelessly to the Client, the wearable can convert this compressed record into 15 individual records for transmission.

C.2 Scenario 1: Apple/Google ENS, uncompressed

C.2.1 Record format

In this scenario, a wearable can have sufficient storage to follow the Apple/Google Exposure Notification Bluetooth Specification [5] fully.

- For simplicity of the device, records are stored exactly as they will be transmitted on air.
- Recommended parameters are used for Scan Off Time and Advertising Interval.
- The analysis includes 7 days of storage, assuming uploads to the Client at least every 7 days.
- Eight devices are assumed to be transmitting in range, on average, across a typical 24-hour period.

Record Fields	Bytes
Length	2
Timestamp	4
Sequence Number	3
LTV1: RSSI	3

Record Fields	**Bytes**
LTV2: ENS-specific Data	22
Total	**34**

C.2.2 Results

Record Fields	**Value**	**Units**
Scan Off Time	60	seconds
Scan On Time	300	ms
Advertisement Interval (average)	250	ms
RPI Interval	10	minutes
Expected receptions per RPI Interval	12.00	records
Average # of Transmitters in Range	8	devices
Records/day	13824	records
Storage/day	470	kB
Days of storage	7	days
Minimum Memory Size	**3290**	**kB**

Note: If this device reliably uploads data to a phone every 1 to 2 days, storage can be reduced up to 7x.

C.3 Optional compression for reduced storage area

A WENS-compliant device may optionally choose to do lossless compression on records to reduce storage area. The Server must still transfer individual records using compliant formats, but there is no restriction on internal storage format. A simple software routine can un-compress the data into a compliant format prior to transmission.

C.3.1 Compressed record format

In this scenario, data is compressed as described above in Section C.1.5. Pathloss is used in place of RSSI and Tx power; the far-side transmitter will likely use the same power for all advertisements in a given period, so there should be minimal loss of information. If required, a single value can be added to the record to store the far-side Tx power for this period. If the Tx power is expected to fluctuate more frequently, another 15 bytes can be added without significantly impacting required memory size.

Record Fields	**Bytes**
Record Length	2
ENS-specific Data	20
Sequence Number	3

Record Fields	**Bytes**
RSSI	12
Timestamp	48
Total	**85**

C.3.2 Compressed storage requirements

When using optional compression, storage requirements can be significantly smaller, especially for cases where multiple advertisements are recorded from the same transmitter. For the same scenario shown in Section C.2, storage is reduced by 4.8x.

Description	**Value**	**Units**
Scan Interval	60	seconds
Scan Duration	300	ms
Advertisement Interval (average)	250	ms
RPI Interval	10	minutes
Expected receptions per RPI Interval	12.00	records
Average # of transmitters in range	8	devices
Records/day	1152	records
Storage/day	97.92	kB
Days of storage	**7**	days
Minimum memory size	685.44	kB

Note: If this device reliably uploads data to a phone every 1 to 2 days, storage can be reduced up to 7x.

C.4 Local storage of PIs or keys

Some ENSs may prefer the method of downloading PIs to the Server. In one example, a device may need to store PIs for 7 days, with a 10-minute PI interval and a maximum 33 bytes of storage per PI.

Description	**Bytes**	**Units**
Days of storage	7	days
PIs/day	144	PI/day
Storage per PI	33	bytes
Total Storage Required	33,264	bytes

A device may also store keys locally, either downloaded from the Client or self-generated; these will require a relatively small amount of memory, assuming that there is only one 20-byte key per day in most ENSs.