

Electronic Shelf Label Service

Bluetooth® Service Specification

- **Version:** v1.0.1
- **Version Date:** 2025-06-30
- **Prepared By:** Electronic Shelf Label Working Group

Abstract

The Electronic Shelf Label Service allows electronic shelf labels (ESLs) to be controlled and updated using Bluetooth wireless technology.



Version History

Version Number	Date (yyyy-mm-dd)	Comments
v1.0	2023-03-28	Adopted by the Bluetooth SIG Board of Directors.
v1.0.1	2025-06-30	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Andreas Hechenblaickner	SES-imagotag
Christian Friessnegg	SES-imagotag
Christopher Church	Qualcomm Technologies, Inc.
Daniel Misoph	Silicon Laboratories
Ealwan Lee	SKAIChips Co., Ltd
Guo Hui	Hanshow Technology Co., Ltd.
Horst Kautschitz	VusionGroup
Jussi Mäkinen	Nokia Corporation
Kurt Haller-Walzl	SES-imagotag
Károly Kovács	Silicon Laboratories
Laurence Richardson	Qualcomm Technologies International, Ltd.
Lauri Hintsala	Silicon Laboratories
Lorenzo Amicucci	Nordic Semiconductor
Min Liang	Beijing Hanshi Technology Co., Ltd.
Philip Maurer	SES-imagotag
Pirun Lee	Nordic Semiconductor
Ravi Bamidi	Silvair, Inc.
Robin Heydon	Qualcomm Technologies International, Ltd.
Simon Slupik	Silvair, Inc.
Yaping Ji	Hanshow Technology Co., Ltd.

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. All content within the specification, including notes, appendices, figures, tables, message sequence charts, examples, sample data, and each option identified 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.

Copyright © 2020–2025. All copyrights in the Bluetooth Specifications themselves are owned by Apple Inc., Ericsson AB, Intel Corporation, Google LLC, Lenovo (Singapore) Pte. Ltd., Microsoft Corporation, Nokia Corporation, and Toshiba Corporation. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.



Contents

1	Introduction	6
1.1	Change History	6
1.1.1	Errata incorporated in v1.0.1	6
1.2	Language	6
1.2.1	Language conventions	6
1.2.1.1	Implementation alternatives	7
1.2.1.2	Discrepancies	7
1.2.2	Reserved for Future Use	7
1.2.3	Prohibited	7
1.3	Table requirements	8
1.4	Conformance	8
1.5	Byte transmission order	8
2	Service	9
2.1	Service dependencies	9
2.2	Bluetooth Core Specification release compatibility	9
2.3	Transport dependencies	9
2.4	Application error codes	9
2.5	GATT sub-procedure requirements	9
2.6	Declaration	9
2.7	Behavior	9
2.7.1	Roles	10
2.7.2	Elements	10
2.7.2.1	Display element	10
2.7.2.2	LED element	10
2.7.2.3	Sensor element	11
2.7.3	ESL state machine	11
2.7.3.1	Unassociated state	12
2.7.3.2	Configuring state	12
2.7.3.3	Synchronized state	12
2.7.3.4	Updating state	13
2.7.3.5	Unsynchronized state	13
3	Service characteristics	14
3.1	ESL Address characteristic	14
3.1.1	Format	14
3.1.1.1	Group_ID	14
3.1.1.2	ESL_ID	15
3.1.2	Behavior	15
3.2	AP Sync Key Material characteristic	15
3.2.1	Format	15
3.2.2	Behavior	15
3.3	ESL Response Key Material characteristic	15
3.3.1	Format	15
3.3.2	Behavior	15
3.4	ESL Current Absolute Time characteristic	16
3.4.1	Format	16
3.4.2	Behavior	16
3.5	ESL Display Information characteristic	16



3.5.1	Format	16
3.5.2	Behavior	17
3.6	ESL Image Information characteristic	17
3.6.1	Format	17
3.6.2	Behavior	17
3.6.2.1	48-bit Object IDs	18
3.7	ESL Sensor Information characteristic	18
3.7.1	Format	18
3.7.1.1	Sensor_Type	19
3.7.2	Behavior	20
3.8	ESL LED Information characteristic	20
3.8.1	Format	20
3.8.1.1	sRGB LED type	20
3.8.1.2	Monochrome LED type	21
3.8.2	Behavior	21
3.9	ESL Control Point characteristic	21
3.9.1	Format of commands and responses	22
3.9.1.1	Command opcodes	22
3.9.1.2	Response opcodes	22
3.9.2	Command behavior	23
3.9.2.1	Ping	24
3.9.2.2	Unassociate from AP	24
3.9.2.3	Service Reset	25
3.9.2.4	Factory Reset	26
3.9.2.5	Update Complete	26
3.9.2.6	Read Sensor Data	26
3.9.2.7	Refresh Display	27
3.9.2.8	Display Image	28
3.9.2.9	Display Timed Image	29
3.9.2.10	LED Control	30
3.9.2.11	LED Timed Control	35
3.9.2.12	Vendor-specific tag	37
3.9.3	Response behavior	38
3.9.3.1	Error	38
3.9.3.2	LED State	39
3.9.3.3	Basic State	39
3.9.3.4	Display State	41
3.9.3.5	Sensor Value	41
3.9.3.6	Vendor-specific response	41
3.9.4	ESL Control Point Procedure Timeout	41
4	Acronyms and abbreviations	43
5	References	44

1 Introduction

This specification specifies a Generic Attribute Profile (GATT) service intended for use in ESL devices.

1.1 Change History

This section summarizes changes at a moderate level of detail and should not be considered representative of every change made.

1.1.1 Errata incorporated in v1.0.1

Section	Errata
2.7.3: ESL state machine	24214
2.7.3.3: Synchronized state	25180
2.7.3.5: Unsynchronized state	24212, 25180
3.9.2.2: Unassociate from AP	24211
3.9.2.4: Factory Reset	24211
5: References	27151

Table 1.1: Errata incorporated in v1.0.1

1.2 Language

1.2.1 Language conventions

In the development of a specification, the Bluetooth SIG has established the following conventions for use of the terms “shall”, “shall not”, “should”, “should not”, “may”, “must”, and “can”. In this Bluetooth specification, the terms in [Table 1.2](#) have the specific meanings given in that table, irrespective of other meanings that exist.

shall	—used to express what is required by the specification and is to be implemented exactly as written without deviation
shall not	—used to express what is forbidden by the specification
should	—used to express what is recommended by the specification without forbidding anything
should not	—used to indicate that something is discouraged but not forbidden by the specification
may	—used to indicate something that is permissible within the limits of the specification

must	—used to indicate either: 1. an indisputable statement of fact that is always true regardless of the circumstances 2. an implication or natural consequence if a separately-stated requirement is followed
can	—used to express a statement of possibility or capability

Table 1.2: Language conventions terms and definitions

1.2.1.1 Implementation alternatives

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

1.2.1.2 Discrepancies

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

1.2.2 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.

1.2.3 Prohibited

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

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

“Prohibited” is never abbreviated.



1.3 Table requirements

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

1.4 Conformance

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

1.5 Byte transmission order

All characteristics used with the ESL Service shall be transmitted with the least significant octet (LSO) first (i.e., little endian). Where the format is described in tables in this document, the LSO is the first octet in the topmost field of the table; the most significant octet 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 [\[3\]](#) for more information on octet ordering.

2 Service

2.1 Service dependencies

The ESL Service does not depend on any other service.

2.2 Bluetooth Core Specification release compatibility

The ESL Service is compatible with the Bluetooth Core Specification [2], Version 5.4 or later.

2.3 Transport dependencies

The ESL Service shall be used over Bluetooth Low Energy (LE) transport only.

2.4 Application error codes

The ESL Service does not use any Attribute Protocol (ATT) Application Error codes that are not already defined in Part B, Section 1 of the Core Specification Supplement [7].

2.5 GATT sub-procedure requirements

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

Table 2.1 summarizes additional GATT sub-procedure requirements beyond those required by all GATT Servers.

GATT sub-procedure	Requirements
Read Long Characteristic Value	M
Characteristic Value Write	M
Write Without Response	M
Characteristic Value Notification	M
Write Characteristic Descriptors	M

Table 2.1: GATT sub-procedure requirements

2.6 Declaration

The ESL Service shall be instantiated as a primary service.

The service Universally Unique Identifier (UUID) shall be set to «Electronic Shelf Label Service» as specified in [1].

2.7 Behavior

The ESL Service enables a Client to discover the capabilities of the ESL and to send configuration settings to an ESL on which the service is exposed.

The Bluetooth wireless technology that can be used to update and control the ESL is described by this specification.

2.7.1 Roles

The ESL Service is used as part of a system in which a centralized access point (AP) may act as the Client of the service. The AP is enabled to initiate a connection to discover the capabilities of the ESL and to configure the ESL.

The ESL Service is concerned only with the Server behavior. The Server resides in the ESL.

Therefore, any references in this document to the "Server" are to the ESL that exposes the ESL Service.

The term access point, abbreviated to "AP", is used in some of the characteristic names in this specification where they refer to values associated with the Client.

2.7.2 Elements

An ESL is composed of certain elements. Three types of independent, optional elements (a display, a light-emitting diode (LED), and a sensor) are described in [Section 2.7.2.1](#) to [Section 2.7.2.3](#).

Because there may be multiple displays, LEDs, and sensors per ESL, a display index number, an LED index number, and a sensor index number are used to enable multiple displays, LEDs, and sensors to be identified individually within a single ESL device.

The index numbers are unsigned integers. For each type of element that is supported by the ESL, the index numbering shall start at 0, and additional index numbers, if required, shall be assigned using consecutive numbers until an index number has been assigned to every instance of that element type. The use of indexing is described in more detail in [Section 2.7.2.1](#) to [Section 2.7.2.3](#) and in [Section 3](#).

2.7.2.1 Display element

An ESL may have zero or more displays, up to a maximum of 256, and the displays are addressed using index numbers from 0 to 255. The index is called `Display_Index`.

The display used in an ESL typically is a low-power display (e.g., an e-ink, or similar, display that does not require power to maintain a constant image).

The purpose of a display is to display an ESL image.

2.7.2.1.1 ESL image

An ESL may store zero or more images, up to a maximum of 256, and the images are addressed using index numbers from 0 to 255. The index is called `Image_Index`.

The stored images can subsequently be displayed on an ESL display. An image can therefore be pre-loaded onto a device ready for a short command to instruct the ESL to display the image on the display.

2.7.2.2 LED element

An ESL may have zero or more LEDs, up to a maximum of 256, and the LEDs are addressed using index numbers from 0 to 255. The index is called `LED_Index`.

The light is assumed to be an LED, which may be a single-color LED or an LED that can display one of multiple colors. Although this element is assumed to be an LED, equivalent light-emitting technologies may also be used.

The LED can flash to attract the attention of a human. The LED can be off, on, or flashing in a particular pattern of flashes.

Flashing the LED generally consumes less power than having the LED continuously illuminated, and different flashing patterns may be used to indicate different meanings.

2.7.2.3 Sensor element

An ESL may have zero or more sensors, up to a maximum of 256, and the sensors are addressed using index numbers from 0 to 255. The index is called `Sensor_Index`.

An ESL sensor is an element that can be used to read a physical property of the environment (e.g., the temperature of a refrigerator).

2.7.3 ESL state machine

An ESL can be in one of five possible states:

- Unassociated state
- Configuring state
- Synchronized state
- Updating state
- Unsynchronized state

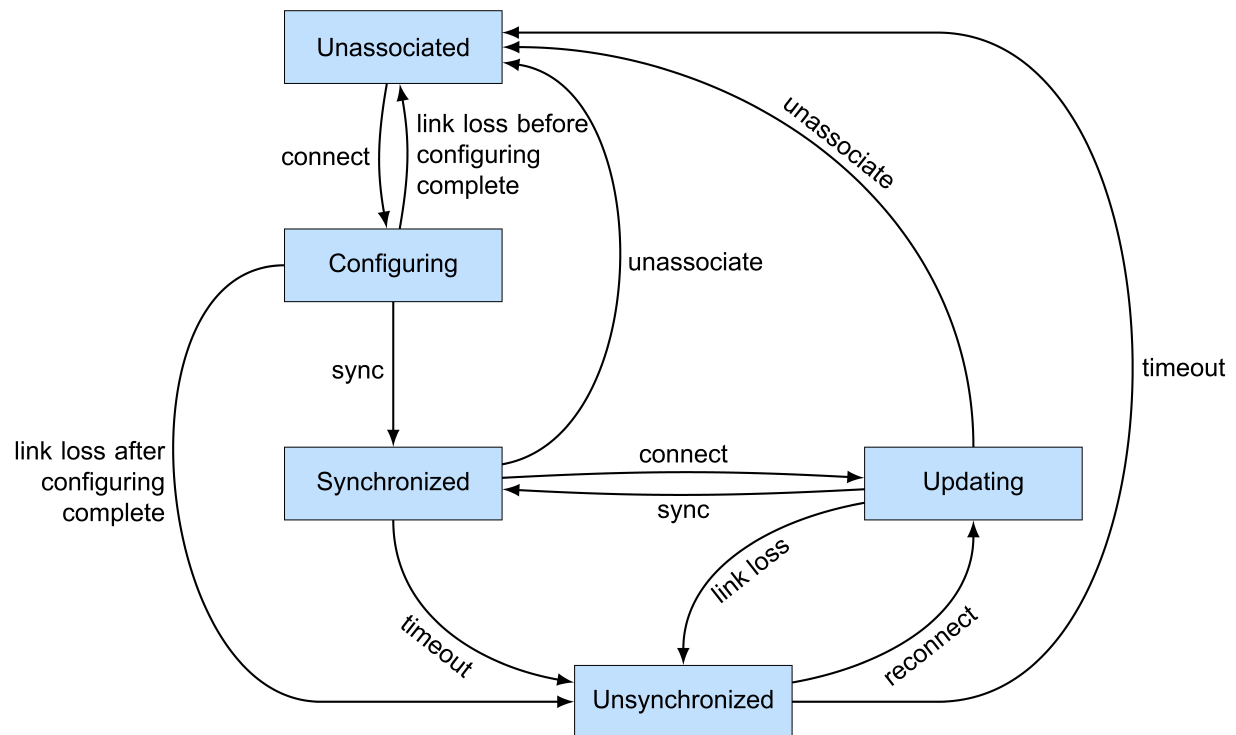


Figure 2.1: State machine

Figure 2.1 is illustrative; refer to the text for further details of the state transitions.

2.7.3.1 Unassociated state

The Unassociated state is used when an ESL has not yet been configured by an AP (i.e., the ESL has not yet been provisioned with the information required for the ESL to be able to synchronize with an AP as described in [Section 2.7.3.3](#)).

In the Unassociated state, the ESL shall enter the Generic Access Profile (GAP) Undirected Connectable mode and transmit advertising packets containing a Service UUID AD type that includes the UUID of the ESL Service.

When an ESL in the Unassociated state forms a connection with a Client, the ESL shall be in Bondable mode as described in the GAP (see [Vol 3], Part C, Section 9.4.3 of [2]). If the Bonding procedure is successfully completed with the Client, then the ESL shall transition to the Configuring state described in [Section 2.7.3.2](#).

2.7.3.2 Configuring state

The Configuring state shall be entered only from the Unassociated state.

The Configuring state allows an ESL that has not been configured to be configured by the bonded Client. State information is exposed by the ESL via the GATT characteristics described in [Section 3](#).

A Client may configure the ESL by writing new values to the writable characteristics. The configuration of the ESL has been completed when the Client has successfully written values to all of the following characteristics:

- ESL Address
- AP Sync Key Material
- ESL Response Key Material
- ESL Current Absolute Time

2.7.3.2.1 Link loss

If the connection is lost owing to link loss occurring in the Configuring state before the configuration of the ESL has been completed, then the ESL shall transition to the Unassociated state described in [Section 2.7.3.1](#). However, if the connection is lost owing to link loss after the configuration of the ESL has been successfully completed, the ESL shall transition to the Unsynchronized state described in [Section 2.7.3.5](#).

2.7.3.3 Synchronized state

The Synchronized state is a low-power and low-bandwidth communications scheme based on the Periodic Advertising with Responses feature described in the Bluetooth Core Specification [2]. This scheme enables thousands of ESLs to synchronize with a central AP, and for short messages to be sent by the AP and responded to by the ESLs. Requirements that apply in this state may be specified in a separate profile specification. The Synchronized state may be entered only from the Configuring state or the Updating state. The ESL transitions to the Synchronized state when the ESL completes the Periodic Advertising Sync Transfer procedure as the recipient and successfully synchronizes to a periodic advertising train transmitted by the Client, as described in Volume 6, Part B of the Bluetooth Core Specification [2].

If the ESL has not received a valid ESL message in a synchronization message from the AP within 60 minutes, or has been unable to respond to synchronization messages within 60 minutes, then the ESL shall transition to the Unsynchronized state.

A relevant profile specification may specify additional requirements that apply to transitions to and from the Synchronized state.

Larger data transfers may be accomplished by transitioning from the Synchronized state to the connection-oriented Updating state described in [Section 2.7.3.4](#).

2.7.3.4 Updating state

The Updating state may be entered only from the Synchronized state or the Unsynchronized state (see [Section 2.7.3.5](#)).

If the ESL is in the Unsynchronized state, then the ESL is already in a GAP connectable mode as described in [Section 2.7.3.5](#).

To transition to the Updating state from the Synchronized state, the AP can use the GAP Periodic Advertising Connection procedure in the specified subevent to initiate a connection to the ESL.

If the Client successfully initiates a connection, then the ESL shall transition to the Updating state only if the peer device is the Client with which the ESL is already associated. The Server shall then transition to the Updating state.

While in the Updating state, the ESL shall ignore any data present on the LE periodic advertising with responses logical transport.

When in the Updating state, the ESL Service may be used to reconfigure the ESL device by updating the values of one or more of the characteristics described in [Section 2.7.3.2](#). In addition, in the Updating state, the Object Transfer Profile [5] (if supported) may be used to write new image data to the device. Support for the Object Transfer Profile is provided by associating an Image_Index value with a specific image that is stored in the ESL device, as described in [Section 3.6](#). A 48-bit Object identifier (ID) can be derived from each Image_Index value for use in the Object Transfer Service [6] (see [Section 3.6.2.1](#)).

The Server shall reject any pairing requests that are received while the Server is in the Updating state.

2.7.3.4.1 Link loss

If the connection is lost owing to link loss occurring in the Updating state, then the ESL shall transition to the Unsynchronized state described in [Section 2.7.3.5](#).

2.7.3.5 Unsynchronized state

When in the Unsynchronized state, the ESL is associated with an AP (i.e., the ESL has been provisioned with the information required for the ESL to be able to synchronize with an AP as described in [Section 2.7.3.5](#)) but is not in a connection and is not synchronized.

In the Unsynchronized state, the ESL shall enter a GAP connectable mode. If a connection is formed with a Client, then the ESL transitions to the Updating state as described in [Section 2.7.3.4](#).

If the ESL is not moved to the Updating state within 60 minutes, then the ESL shall remove all bonding information with the AP; delete the values of the AP Sync Key Material, ESL Response Key Material, and ESL Address in internal storage; and delete all stored commands.

Following execution of removing the bonding information and deleting the values and commands, the ESL shall enter the Unassociated state.

3 Service characteristics

The ESL Service exposes the characteristics listed in [Table 3.1](#). These characteristics may be read and written (where the characteristic properties allow) to discover the capabilities of the ESL and to configure the ESL.

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
ESL Address	M	Write	None	Encryption required
AP Sync Key Material	M	Write	None	Encryption required
ESL Response Key Material	M	Write	None	Encryption required
ESL Current Absolute Time	M	Write	None	Encryption required
ESL Display Information	C.1	Read	None	Encryption required
ESL Image Information	C.1	Read	None	Encryption required
ESL Sensor Information	C.2	Read	None	Encryption required
ESL LED Information	C.3	Read	None	Encryption required
ESL Control Point	M	Write Without Response, Write, Notify	None	Encryption required

Table 3.1: ESL Service characteristics

C.1: Mandatory if the ESL supports one or more displays; otherwise Excluded.

C.2: Mandatory if the ESL supports one or more sensors; otherwise Excluded.

C.3: Mandatory if the ESL supports one or more LEDs; otherwise Excluded.

The characteristics listed in [Table 3.1](#) are described in [Section 3.1](#) to [Section 3.9](#).

3.1 ESL Address characteristic

3.1.1 Format

Each ESL Address consists of 15 bits and is split into a Group_ID and an ESL_ID. The Group_ID is a 7-bit value, and the ESL_ID is an 8-bit value, as shown in [Figure 3.1](#).



Figure 3.1: ESL Address format

3.1.1.1 Group_ID

Each ESL is a member of a group, identified by the Group_ID part of the ESL Address.

3.1.1.2 ESL_ID

The ESL_ID in the ESL Address characteristic shall be a value in the range 0x00 to 0xFE.

The value 0xFF is reserved for the Broadcast Address. Refer to a relevant profile specification where further requirements for the use of the Broadcast Address may be specified. For example, the Broadcast Address may be used in the profile to enable a message to be addressed simultaneously to all the ESLs that belong to the group identified by the Group_ID part of the ESL Address.

Because there is a maximum of 128 groups, and 255 ESL_IDs that are unique within each group, a total of 32,640 ESL devices can be allocated a locally unique address.

3.1.2 Behavior

The ESL Address is configured when the Client writes an ESL Address value to the ESL Address characteristic.

After an ESL Address has been configured, the ESL Address can be used in the Synchronized state. The Synchronized state is described in [Section 2.7.3.3](#).

3.2 AP Sync Key Material characteristic

3.2.1 Format

The AP Sync Key Material characteristic is a 24-octet value that has the same format as the Encrypted Data Key Material characteristic defined in [Vol 3], Part C, Section 12.6 of the Bluetooth Core Specification [2]. The AP Sync Key Material value enables an ESL to receive encrypted messages from the AP when the ESL is in the Synchronized state. Refer to a relevant profile specification where further details of the use of encryption keys in the Synchronized state may be specified.

3.2.2 Behavior

The AP Sync Key Material is configured when an AP, acting as a Client, writes its AP Sync Key Material value to the AP Sync Key Material characteristic. The AP Sync Key Material characteristic enables the transfer of the session key and initialization vector (IV) that are used in the Synchronized state.

3.3 ESL Response Key Material characteristic

The ESL Response Key Material characteristic value provides a key for use in the Encrypted Advertising Data feature described in the Bluetooth Core Specification [2]. The ESL Response Key Material value is intended to be different for each ESL in a system and enables ESL responses sent in the Synchronized state to be authenticated by the AP. Refer to a relevant profile specification where further details of the use of authentication in the Synchronized state may be specified.

3.3.1 Format

The ESL Response Key Material characteristic is a 24-octet value that has the same format as the Encrypted Data Key Material characteristic defined in [Vol 3], Part C, Section 12.6 of the Bluetooth Core Specification [2].

3.3.2 Behavior

The ESL Response Key Material is configured when an AP, acting as a Client, writes a value to the ESL Response Key Material characteristic.



The ESL Response Key Material characteristic enables the transfer of the session key and IV that are used by the ESL when the ESL sends responses to the AP.

3.4 ESL Current Absolute Time characteristic

3.4.1 Format

The ESL Current Absolute Time characteristic value is a 32-bit reference time integer value (uint32) with a resolution of 1 millisecond (ms).

3.4.2 Behavior

When a value is written to the ESL Current Absolute Time characteristic, the Server shall set its current system time to the value.

The Server shall increment the value by 1, every 1 ms. When the value has reached 0xFFFFFFFF, it shall roll over to 0x00000000.

The resulting value is referred to in this specification as the "absolute time".

The resolution of 1 ms results in a clock wrapping time of 49.7 days. Therefore, timed commands can be set up for slightly over 1 month in advance.

The ESL current absolute time concept does not specify an epoch; it is assumed that all ESLs are configured to have the same current absolute time value and, therefore, the timed commands will operate at approximately the same time. This allows the timed commands (e.g., the Timed Display Image command) to include an Absolute_Time parameter value so that all ESLs will execute the command at approximately the same absolute time.

3.5 ESL Display Information characteristic

3.5.1 Format

The ESL Display Information characteristic value consists of an array of one or more Display Data Structures. Each individual Display Data Structure shall consist of a set of fields that have the format shown in [Figure 3.2](#).

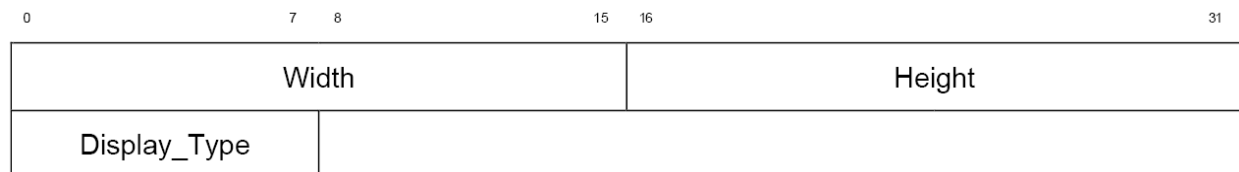


Figure 3.2: Display Data Structure format

The meaning of each field in [Figure 3.2](#) is described in [Table 3.2](#).

Field	Size (octets)	Description
Width (uint16)	2	The width of the display in pixels
Height (uint16)	2	The height of the display in pixels
Display_Type (uint8)	1	Enumeration as specified in [1]

Table 3.2: Display Data Structure fields

The value of the Display_Type field is from an enumeration maintained in the Bluetooth SIG Assigned Numbers [1], which identifies the colors that are supported by the display and may provide additional information about the display.

As each Display Data Structure has a fixed length of 5 octets, the total length of the Display Information characteristic value, measured in octets, is five times the number of displays supported by the ESL. The maximum number of displays that could be represented in the characteristic value is therefore 102 displays.

3.5.2 Behavior

The ESL Display Information characteristic shall return an array of one or more Display Data Structures when read by a Client.

When the ESL supports only one display, the display shall have the index value of 0.

When the ESL supports more than one display, the Display Data Structure associated with the index value of 0 shall be located to begin at the least significant bit position and the Display Data Structures shall be concatenated in order of increasing index value.

The Display Data Structure associated with the index value of N shall provide information about the display with the index value of N.

If the ESL does not support a display, then the ESL Display Information characteristic shall not be exposed. In this case, the characteristic would not be found by a Client during GATT Characteristic Discovery procedures.

The value of the ESL Display Information characteristic shall remain constant at least until the ESL is commanded to perform a factory reset.

3.6 ESL Image Information characteristic

3.6.1 Format

The ESL Image Information characteristic contains a single field, called Max_Image_Index, whose value is a 8-bit integer (uint8) in the range 0x00 to 0xFF.

3.6.2 Behavior

When read, the ESL Image Information characteristic shall return the Max_Image_Index value equal to the numerically highest Image_Index value supported by the ESL.

The Server shall allocate a unique index value, Image_Index, to each image that the Server can store, using consecutive numbering from 0x00 to Max_Image_Index. Because the index is zero-based, the maximum number of images that can be stored by the ESL is Max_Image_Index + 1. An Image_Index

value serves as a pointer to an individual image storage location. An image storage location may be occupied with an image or empty. The number of images that may be stored can be different from the number of displays supported by the ESL—a Client can select from among the stored images to choose an image to present on a display. If the ESL does not support the display of an image, then the ESL Image Information characteristic shall not be exposed. In this case, the characteristic would not be found by a Client during GATT Characteristic Discovery procedures.

The value of the ESL Image Information characteristic shall remain constant at least until the ESL is commanded to perform a factory reset.

3.6.2.1 48-bit Object IDs

Each image storage location has an associated 48-bit Object ID value that can be calculated from the Image_Index value for that location.

When a 48-bit Object ID is used (e.g., for use with the Object Transfer Service [6]), an Object ID can be derived for a given image storage location by adding the associated Image_Index value to the following base value: 0x000000000100 (i.e., 256, in decimal notation). A profile specification may apply additional requirements.

3.7 ESL Sensor Information characteristic

3.7.1 Format

The ESL Sensor Information characteristic value consists of an array of one or more Sensor Information Structures. Each individual Sensor Information Structure shall consist of a pair of fields (i.e., Size and Sensor_Type) having either the short or long format as shown in Figure 3.3 and Figure 3.4. The short format uses standard sensor IDs whereas the long format is for vendor-specific sensor IDs, as described in Section 3.7.1.1.

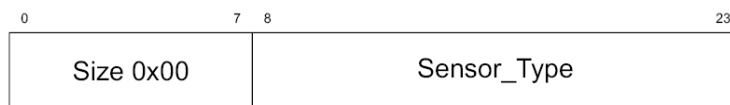


Figure 3.3: Sensor Information Structure short format, using a Property ID



Figure 3.4: Sensor Information Structure long format, using a vendor-specific ID

The two different Sensor Information Structure formats may be mixed within an array, in any order. The meaning of each field is described in [Table 3.3](#).

Field	Size (octets)	Description
Size	1	When Size is 0x00, the Sensor_Type field is a 16-bit value (2 octets), and is a Property ID assigned in the Mesh Device Properties specification [4]. When Size is 0x01, the Sensor_Type field is a 32-bit vendor-specific value (4 octets).
Sensor_Type	2 or 4	An ID that indicates a type of sensor.

Table 3.3: Sensor Information fields

3.7.1.1 Sensor_Type

Sensors typically provide data representing measurements. A Client can obtain the data from a sensor by using the Read Sensor Data command described in [Section 3.9.2.6](#). In order to interpret the data that has been read from a sensor, the format of the data produced by the sensor needs to be known. The Sensor_Type value identifies the data format as described in [Section 3.7.1.1.1](#) and [Section 3.7.1.1.2](#).

3.7.1.1.1 16-bit value

When the value of the Size field is 0x00, the Sensor_Type field shall contain a 16-bit value that represents a Property ID from the Mesh Device Properties. The meaning of the property, including the mapping between the Property ID and a characteristic UUID, is specified in the Mesh Device Properties specification [4]. The data provided by the sensor shall be in the format described by the property identified by the Property ID.

This indirect method of describing the format of the data enables support for a wide range of types of sensor to be supported. The property definition provides an additional layer of context on top of a characteristic definition. The property can provide additional information such as the type of the sensor and its intended application.

3.7.1.1.2 32-bit value

When the value of the Size field is 0x01, the Sensor_Type field shall contain a 32-bit value consisting of two 16-bit sub-fields as shown in [Figure 3.5](#).

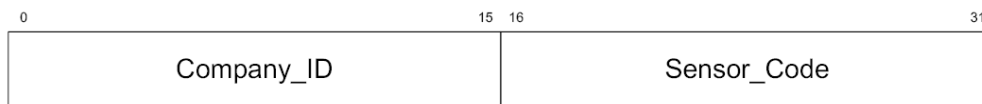


Figure 3.5: Vendor-specific sensor type sub-fields

The Company_ID sub-field shall contain a Company ID, a unique 16-bit number assigned by the Bluetooth SIG to a member company. Company ID values are defined in the Bluetooth SIG Assigned Numbers [1].

The Sensor_Code sub-field shall contain a number assigned to a vendor-specific sensor type by the member whose Company ID is in the Company_ID sub-field.

3.7.2 Behavior

The ESL Sensor Information characteristic shall return an array of one or more Sensor Information Structures when read by a Client.

When the ESL supports only one sensor, the sensor shall have the index value of 0.

When the ESL supports more than one sensor, the Sensor Information Structure associated with the index value of 0 shall be located to begin at the least significant bit position and the Sensor Information Structures shall be concatenated in order of increasing index value.

When the ESL supports more than one sensor, the Sensor Information Structure associated with the index value of N shall provide information about the sensor with the index value of N.

If the ESL does not support a sensor, then the ESL Sensor Information characteristic shall not be exposed. In this case, the characteristic would not be found by a Client during GATT Characteristic Discovery procedures.

The value of the ESL Sensor Information characteristic shall remain constant at least until the ESL is commanded to perform a factory reset.

3.8 ESL LED Information characteristic

3.8.1 Format

The ESL LED Information characteristic is an array of one or more octets in which each octet represents an LED that is supported by the ESL.

The two most significant bits of each octet identify the type of LED, as specified in [Table 3.4](#).

Bits 7 and 6	LED type
0b00	sRGB
0b01	Monochrome
0b10 to 0b11	RFU

Table 3.4: LED type enumeration

The remaining bits of the octet have a definition that depends on the LED type identified by bits 7 and 6. See [Section 3.8.1.1](#) for the sRGB LED type and [Section 3.8.1.2](#) for the monochrome LED type. The most important difference between these two LED types is that the sRGB LED type is able to change color so the color reported in this case is an instantaneous value, whereas the color reported for the monochrome LED type is a constant value.

3.8.1.1 sRGB LED type

An sRGB LED is a multi-color LED that can change color. The current color of the LED is identified by the value of bits 5 to 0, which are able to represent 63 possible colors as combinations of quantities of red, green, and blue, as shown in [Table 3.5](#). The RGB value exposed is a best approximation in the sRGB color space to the present color of the LED, in which 0b00 represents the minimum level and 0b11 represents the maximum level for a color component.

Bit numbers	Field
7 and 6	LED type = sRGB (see Table 3.4)
5 and 4	Blue
3 and 2	Green
1 and 0	Red

Table 3.5: Current color of a multi-color LED

3.8.1.2 Monochrome LED type

A monochrome LED is a single-color LED. The single color is identified by the value of bits 5 to 0, which are able to represent 63 possible colors as combinations of quantities of red, green, and blue, as shown in [Table 3.6](#). The RGB value exposed is a best approximation in the sRGB color space to the fixed color of the LED, in which 0b00 represents the minimum level and 0b11 represents the maximum level for a color component.

Bit numbers	Field
7 and 6	LED type = Monochrome (see Table 3.4)
5 and 4	Blue
3 and 2	Green
1 and 0	Red

Table 3.6: Fixed color of a monochrome LED

3.8.2 Behavior

The ESL LED Information characteristic shall return an array of one or more octets when read by a Client.

When the ESL supports only one LED, the LED shall have the index value of 0.

In the characteristic value, the octet associated with the index value of 0 shall be located at the LSO position and the following octets, if any, shall be in order of increasing index value.

The octet associated with the index value of N shall provide information about the LED with the index value of N.

If the ESL does not support an LED, then the ESL LED Information characteristic shall not be exposed. In this case, the characteristic would not be found by a Client during GATT Characteristic Discovery procedures.

3.9 ESL Control Point characteristic

The ESL Control Point characteristic allows a Client to write commands to an ESL while in a connection.

With respect to values written to ESL Control Point characteristic, the Server shall support both the GATT Write Without Response sub-procedure and the GATT Write Characteristic Value sub-procedure, allowing a Client implementation to use either GATT sub-procedure; see [Table 2.1](#) and [Table 3.1](#).

The Server shall send a response to each command written to the ESL Control Point characteristic; the response takes the form of a notification of the ESL Control Point characteristic as described in [Section 3.9.3](#).

3.9.1 Format of commands and responses

When written or notified, the format of the characteristic value shall be a single Tag Length Value (TLV) formatted element.

The TLV formatted element represents one ESL command or ESL response. Commands are written and responses are notified.

The length of the characteristic value is variable because TLVs are of variable length. The format of each TLV is shown in [Figure 3.6](#).



Figure 3.6: ESL TLV format

3.9.1.1 Command opcodes

The Tag and Length fields are each 4 bits in size. The combination of the Tag and Length field values (i.e., the value of the entire LSO) comprises an opcode. Therefore, there are 256 possible opcodes in total.

The total length of the Parameters field that follows the opcode can be calculated from value of the Length field; the total length of the Parameters field shall be equal to (Length + 1) octets. Therefore, there are 16 possible opcodes in which Length = 0b0000 that have 1 parameter octet containing additional data, 16 possible opcodes in which Length = 0b0001 that have 2 parameter octets containing additional data, 16 possible opcodes in which Length = 0b1111 that have 16 parameter octets containing additional data, etc.

The overall length of each TLV is therefore equal to (Length + 2) octets. For example, the Service Reset command described in [Section 3.9.2.3](#) has a Length nibble of 0b0000, meaning that it has a Parameters field consisting of 1 octet and an overall length, including the opcode itself, of 2 octets. The shortest possible TLV is 2 octets in size, and the longest is 17 octets.

Vendor-specific commands are supported by use of a vendor-specific tag (see [Section 3.9.2.12](#)).

Refer to [Section 3.9.2](#) for details on the ESL commands.

3.9.1.1.1 Additional requirements for commands

In each command, the first parameter, consisting of the octet immediately following the opcode, is the ESL_ID parameter. The ESL_ID parameter value shall be set to the value of the ESL_ID as described in [Section 3.1](#).

3.9.1.2 Response opcodes

As with the command opcode format, the response opcode is composed of the Length nibble (forming the most significant nibble (MSN) of the opcode) and the Tag nibble (forming the least significant nibble of the opcode). The total length of the Parameters field that follows the opcode can be calculated from value of the Length field; the total length of the Parameters field shall be equal to (Length + 1) octets.

Opcode values used in ESL responses have no connection with opcodes of the same numerical value used in ESL commands (i.e., the opcode numbers assigned in [Section 3.9.2](#) can be re-assigned for use in an ESL response with an unrelated meaning).

3.9.1.2.1 Additional requirements for responses

When notified, the characteristic value shall be an ESL response value as described in [Section 3.9.3](#).

3.9.2 Command behavior

The ESL Control Point characteristic may be written multiple times, representing a sequence of commands. The sequence of commands shall be processed in the order in which they are received. Sending an error response counts as having "processed" the command for the purposes of this requirement.

Some commands can be scheduled for execution in the future (e.g., the LED Timed Control and Display Timed Image commands). If such a timed command has been received but the time at which it will be executed has not yet been reached, then the command is described as "pending". As described in [Section 3.4](#), the absolute time is represented by an integer value that rolls over periodically. If a timed command is received, and the time at which the command will be executed is represented by an Absolute_Time parameter value that is lower than the current absolute time, then the command shall be executed only after the absolute time has wrapped around and then reached the value specified in the Absolute_Time parameter. However, when a timed command is received and the time specified by the Absolute_Time parameter value is more than 48 days (4,147,200,000 ms) in the future, the ESL shall reject the command by responding with the Error response: Implausible Absolute Time.

If an opcode is written to the ESL Control Point characteristic and the ESL_ID value specified within the opcode (as described in [Section 3.9.1.1.1](#)) does not match the ESL_ID of the ESL or matches the Broadcast Address, then the ESL shall reject the command by responding with the Error response: Invalid Parameter(s). If an opcode is received by an ESL and the opcode is not recognized by the ESL, then the ESL shall respond by sending the Error response: Invalid Opcode.

Opcode	Procedure	Description	Section
0x00	Ping	Does nothing but solicits a response	Section 3.9.2.1
0x01	Unassociate from AP	Requests the ESL to disassociate from the AP with which it was associated	Section 3.9.2.2
0x02	Service Reset	Sets the Service Needed flag to False	Section 3.9.2.3
0x03	Factory Reset	Requests the ESL to disassociate from the AP and revert to its original state	Section 3.9.2.4
0x04	Update Complete	Requests that the ESL return to the Synchronized state once synchronized	Section 3.9.2.5
0x10	Read Sensor Data	Requests a response with sensor data, or an indication that data is not yet available	Section 3.9.2.6
0x11	Refresh Display	Refreshes the current displayed image to keep the displayed image fresh on the display	Section 3.9.2.7

Opcode	Procedure	Description	Section
0x20	Display Image	Displays a pre-stored image on an ESL display	Section 3.9.2.8
0x60	Display Timed Image	Displays a pre-stored image on an ESL display at a specified time	Section 3.9.2.9
0xB0	LED Control	Turns on/off an LED with a color/flashing pattern	Section 3.9.2.10
0xF0	LED Timed Control	Turns on/off an LED with a color/flashing pattern at a specified time	Section 3.9.2.11
0x_F	Vendor-specific Tag	Allows vendors to specify their own commands	Section 3.9.2.12

Table 3.7: ESL commands

Any opcode that is not yet assigned in [Table 3.7](#) is RFU. In [Table 3.7](#), the underscore character (_) means that any value in the range 0x0 to 0xF is permitted. This allows commands of different lengths to be supported.

The ESL shall support the complete set of commands as detailed in [Section 3.9.2.1](#) to [Section 3.9.2.12](#).

3.9.2.1 Ping

The Ping command solicits an ESL response to be sent. The Ping command has no other effect.

3.9.2.1.1 Format

The Ping command consists of the opcode with no parameter other than the ESL_ID parameter:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1

Table 3.8: Ping command parameters

3.9.2.1.2 Response

If the Ping command is received, then the ESL shall respond by sending a Basic State response as described in [Section 3.9.3.3](#).

An Error response will not be sent in response to a Ping command because the Ping command shall be processed successfully.

3.9.2.2 Unassociate from AP

The Unassociate from AP command allows an ESL to be disassociated from the AP.

When the Unassociate from AP command is received, and the response has been sent, the ESL shall remove all bonding information with the AP; delete the values of the AP Sync Key Material, ESL Response Key Material, and ESL Address in internal storage; and delete all stored commands.

Following execution of the command, the ESL shall enter the Unassociated state.

3.9.2.2.1 Format

The Unassociate from AP command consists of the opcode with no parameter other than the ESL_ID parameter:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1

Table 3.9: Unassociate from AP command parameters

3.9.2.2.2 Response

If the Unassociate from AP command is received and no error condition occurs, then the ESL shall respond by sending a Basic State response as described in [Section 3.9.3.3](#).

However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#).

3.9.2.3 Service Reset

The Service Needed state may be True or False. The Service Needed state is reported via the Service Needed bit of the Basic State response described in [Section 3.9.3.3](#).

If a condition occurs that causes the ESL to set the Service Needed state to True, then the ESL shall set the value of the Service Needed bit to True. Once set to True, the value of the Service Needed bit shall remain True until it is successfully reset by the Client as follows.

When the Service Reset command is received by an ESL:

- If the Service Needed state is no longer True, then the ESL shall set the value that is reported in the Service Needed bit of the Basic State response to False.
- If a condition persists that requires the ESL to keep the Service Needed state set to True, then the ESL shall set the value that is reported in the Service Needed bit of the Basic State response to True (i.e., the ESL shall not change the value of the Service Needed bit to False).

3.9.2.3.1 Format

The Service Reset command consists of the opcode with no parameter other than the ESL_ID parameter:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1

Table 3.10: Service Reset command parameters

3.9.2.3.2 Response

If the Service Reset command is received and no error condition occurs, then the ESL shall respond by sending a Basic State response as described in [Section 3.9.3.3](#).

However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#).

The Service Reset command is idempotent; if the Service Reset command is received when the value of the Service Needed bit is already False, then this shall not be considered an error condition, and the command shall be processed normally.

3.9.2.4 Factory Reset

When the Factory Reset command is received by an ESL, the ESL shall initiate disconnection of the link with the AP.

Upon disconnection, the ESL shall become unassociated from any AP and shall revert to its original state before it was associated with an AP. The ESL shall remove all bonding information with the AP; delete the values of the AP Sync Key Material, ESL Response Key Material, and ESL Address in internal storage; and delete all stored commands and any stored image data that was written to the ESL.

If, after receipt of the Factory Reset command and prior to disconnection from the AP, the ESL receives any other command from the AP written to the ESL Control Point characteristic, then the other command shall be rejected with the error code: "Unspecified Error".

3.9.2.4.1 Format

The Factory Reset command consists of the opcode with no parameter other than the ESL_ID parameter:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1

Table 3.11: Factory Reset command parameters

3.9.2.4.2 Response

The Factory Reset command has no response.

3.9.2.5 Update Complete

When the Update Complete command is received by an ESL, the ESL shall synchronize with the AP and then disconnect the link with the AP.

3.9.2.5.1 Format

The Update Complete command consists of the opcode with no parameter other than the ESL_ID parameter:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1

Table 3.12: Update Complete command parameters

3.9.2.5.2 Response

The Update Complete command has no response.

3.9.2.6 Read Sensor Data

When the Read Sensor Data command is received and the Sensor_Index value is valid, the ESL shall obtain data from the sensor identified by the Sensor_Index value. However, if the sensor identified in the parameter value does not exist, the ESL shall send an Error response.

In addition to sensing environmental variables such as ambient temperature, certain sensor types may report device conditions such as battery level or hardware errors.



3.9.2.6.1 Format

The Read Sensor Data command has the following parameters:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1
Sensor_Index	8	Index to the sensor on the ESL

Table 3.13: Read Sensor Data command parameters

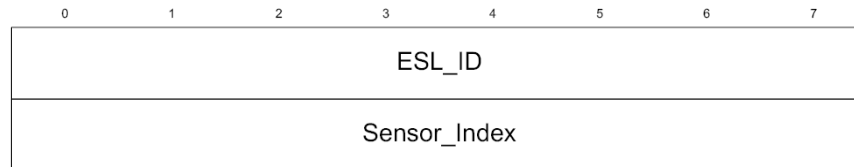


Figure 3.7: Read Sensor Data command parameters

3.9.2.6.2 Response

If the Read Sensor Data command is received and no error condition occurs, then the ESL shall respond by sending a Sensor Value response as described in [Section 3.9.3](#).

However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#).

3.9.2.7 Refresh Display

When the Refresh Display command is received, the Display_Index value is valid, and an image is being displayed on the specified display, then the ESL shall refresh the displayed image on the display specified in the command. However, if the display identified in the parameter value does not exist, or if there is no image currently being displayed on the specified display, the ESL shall send an Error response.

The Refresh Display command enables a display such as an e-ink display to be refreshed at the request of the Client. The process of refreshing a display is implementation-specific. For example, certain types of display technology need the display to be refreshed from time to time to prevent fading or ghosting of the image. The display might flash visibly while being refreshed and this process is typically performed as a night-time routine.

If the Refresh Display command is received, the Display_Index value is valid, and an image is being displayed on the specified display, but the display specified in the command does not need to be refreshed (e.g., because the display technology employed by the implementation uses a different solution), then this shall not be considered an error condition and the command shall be processed as if the display had been refreshed.

3.9.2.7.1 Format

The Refresh Display command has the following parameters:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1
Display_Index	8	Index to the physical display on the ESL

Table 3.14: Refresh Display command parameters



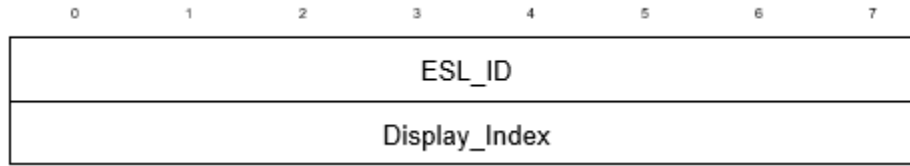


Figure 3.8: Refresh Display command parameters

3.9.2.7.2 Response

If the Refresh Display command is received and no error condition occurs, then the ESL shall respond by sending a Display State response as described in [Section 3.9.3](#). In the Display State response, the Display_Index parameter shall be set to the same value as received in the Refresh Display command, and the Image_Index parameter value shall be set to the image index of the image currently being displayed on the display identified by the Display_Index parameter.

However, if an error condition occurs, then the ESL shall send an Error response as described in [Section 3.9.3](#).

3.9.2.8 Display Image

The Display Image command changes the state of the display selected by a Display_Index value to display a stored image selected by an Image_Index value.

If no error condition occurs, then the ESL shall display the image on the display as specified by the values received in the command.

If the display or the image identified in the parameter values does not exist, then the ESL shall send an Error response.

3.9.2.8.1 Format

The Display Image command has the following parameters:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1
Display_Index	8	Index to the physical display on the ESL
Image_Index	8	Index to a stored image (see Section 3.6)

Table 3.15: Display Image command parameters

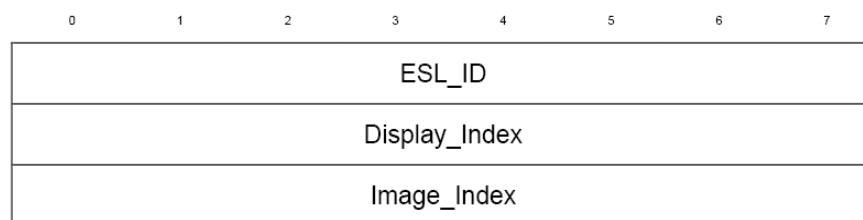


Figure 3.9: Display Image command parameters

3.9.2.8.2 Response

If the Display Image command is received and no error condition occurs, then the ESL shall respond by sending a Display State response as described in [Section 3.9.3](#).

When the ESL sends the Display State response, this response shall confirm that the command was successfully received by containing the same values for the Display_Index and Image_Index parameters as were received in the command.

However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#).

3.9.2.9 Display Timed Image

The Display Timed Image command changes the state of the display selected by a Display_Index value to display a stored image selected by an Image_Index value, at a given absolute time. The Display Timed Image command is acknowledged using a Display State response (see [Section 3.9.3.4](#)), but the command has no effect on the displayed image until the time specified by the Absolute_Time parameter.

If no error condition occurs, then the ESL shall display the image on the display as specified by the values received in the command, at the specified time.

If the display or the image identified in the parameter values does not exist, then the ESL shall send an Error response.

Although the Basic State response is not sent in response to the Display Timed Image command, see [Section 3.9.2.9.3](#) and [Section 3.9.3.3.3](#) for details of the effect of accepting a Display Timed Image command on the value of the Pending Display Update bit, which is exposed when the Basic State response is sent in response to another command.

After the specified time has been reached and the display has been updated as described above, the ESL shall set the Pending Display Update bit to False.

3.9.2.9.1 Handling multiple Display Timed Image commands

If the ESL supports the display of one or more images, then the ESL shall support having one, and only one, Display Timed Image command pending per display. If a Display Timed Image command is received while a Display Timed Image command is already pending, then the ESL shall handle this event as follows:

- If the value of the Absolute_Time parameter in the new command is equal to the value of the Absolute_Time parameter in the pending Display Timed Image command, then the newly received command shall replace the old pending command.
- Otherwise, the ESL shall send the Error response: Queue Full, as described in [Section 3.9.3.1](#), because it is not permitted to have more than one Display Timed Image command pending. In this case, the Display Timed Image command that was already pending remains unchanged.
- If the value of the Absolute Time parameter is zero (0x00000000), then the pending Display Timed Image command shall be deleted.

3.9.2.9.2 Format

The Display Timed Image command has the following parameters:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1
Display_Index	8	Index to the physical display on the ESL
Image_Index	8	Index to a stored image (see Section 3.6)
Absolute_Time	32	Time when the display changes state

Table 3.16: Display Timed Image command parameters

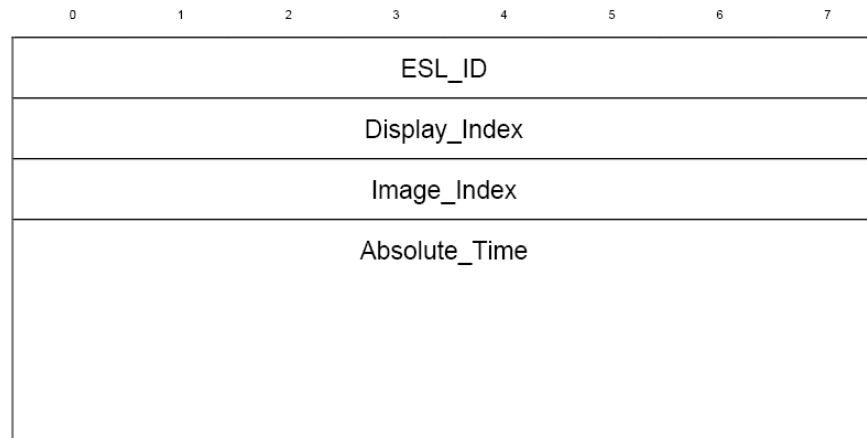


Figure 3.10: Display Timed Image command parameters

3.9.2.9.3 Response

If the Display Timed Image command is received and no error condition occurs, then the ESL shall respond by sending a Display State response as described in [Section 3.9.3](#).

When the ESL sends the Display State response, this response shall confirm that the command was successfully received by containing the same values for the Display_Index and Image_Index parameters as were received in the command. The ESL shall send the updated values in the Display State response despite the time when the display is set to change (as specified in the Absolute_Time parameter of the command) being in the future. However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#). The response shall be sent upon receipt of the command (i.e., the response is not delayed awaiting the timed execution of the command).

3.9.2.10 LED Control

The LED Control command changes the state of an LED. If no error condition occurs, then the ESL shall change the state of the LED in accordance with the values received in the command.

If the LED identified in the parameter values does not exist, then the ESL shall send an Error response.

The LED Control command is addressed to one LED at a time. The LED Control command may be received multiple times to control multiple LEDs.

If an LED Control command is received and the LED specified in the command is already active (where "active" has the meaning described in [Section 3.9.3.3.1](#)), then the new command shall

immediately supersede the previous command, and the effect of the previous command related to controlling the same LED shall then be terminated.

3.9.2.10.1 Format

The LED Control command has the following parameters:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1
LED_Index	8	Index of the LED
Color_Red	2	Red component (see Section 3.8)
Color_Green	2	Green component (see Section 3.8)
Color_Blue	2	Blue component (see Section 3.8)
Brightness	2	0 = 25% 1 = 50% 2 = 75% 3 = 100%
Flashing_Pattern	56	See Section 3.9.2.10.2.2
Repeat_Type	1	0: Number of times 1: Time duration
Repeats_Duration	15	Number of times to repeat, or time duration in increments of 1 second

Table 3.17: LED Control command parameters

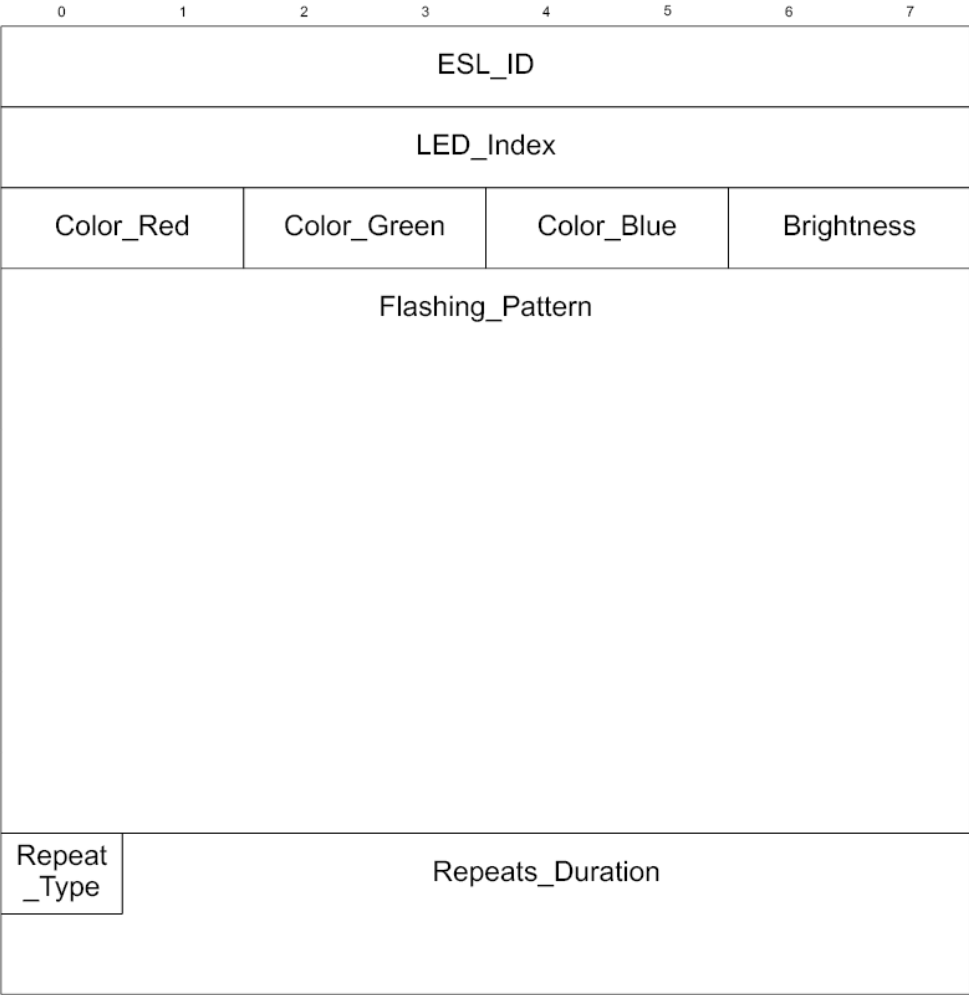


Figure 3.11: LED Control command parameters

3.9.2.10.2 Parameter information

3.9.2.10.2.1 Color

The values of the Color_Red, Color_Green, and Color_Blue fields shall have the same interpretation as specified for the ESL LED Information characteristic in [Section 3.8](#).

However, if the LED selected by the Index value is a monochrome LED, the value of the Color fields shall be ignored upon receipt.

3.9.2.10.2.2 Flashing_Pattern

The Flashing_Pattern fields contain values that indicate the flashing pattern setting of the LED, as shown in [Figure 3.12](#).



Figure 3.12: Flashing_Pattern fields

The Flashing_Pattern bits shall be interpreted by starting at the most significant bit, and progressing one bit at a time through the bits toward the least significant bit, in order of descending bit number. The dwell time for each bit (i.e., the elapsed time before progressing to the next bit) shall be equal to either the Bit_Off_Period field value or the Bit_On_Period field value, depending on whether the instantaneous bit value is 0 or 1, respectively, as described below.

The `Bit_On_Period` field represents a time period, consisting of an integer (uint8) value multiplied by 2 ms. The resulting value of `Bit_On_Period` is the length of time per bit for which the LED is kept on. The valid range is 2 ms to 510 ms; a `Bit_On_Period` value of 0 ms is invalid. This period applies only to the on state. The on duration can also be extended by setting consecutive bits to 1, as described below.

- If Bit_Off_Period and Bit_On_Period are set to the same value, the value of 0b10 (0xAAAAAAAA) means an LED flashing pattern that would be 50 percent on and 50 percent off, with the LED flashing continuously. Each on period would be Bit_On_Period in duration, and each off period would be Bit_Off_Period in duration. The duty cycle can be changed by using different values for Bit_Off_Period and Bit_On_Period.
- The value of 0b1010000010100000101000001010000010100000 (0xA0A0A0A0A0) would be on for 1 Bit_On_Period, then off for 1 Bit_Off_Period, on for 1 Bit_On_Period, off for 5 times the duration of Bit_Off_Period; on for 1 Bit_On_Period, then off for 1 Bit_Off_Period, on for 1 Bit_On_Period, off for 5 times the duration of Bit_Off_Period; etc.
- The value of 0b1010000000001010000000001010000000001111 (0xA00A00A00F) would be on for 1 Bit_On_Period, then off for 1 Bit_Off_Period, on for 1 Bit_On_Period, off for 9 times the duration of Bit_Off_Period; on for 1 Bit_On_Period, then off for 1 Bit_Off_Period, on for 1 Bit_On_Period, off for 9 times the duration of Bit_Off_Period; on for 1 Bit_On_Period, then off for 1 Bit_Off_Period, on for 1 Bit_On_Period, off for 9 times the duration of Bit_Off_Period; and then on for 4 times the duration of Bit_On_Period.

The Pattern field is 40 bits in size. However, the ESL shall ignore any leading zeroes in a pattern and treat only the remaining bits as meaningful (i.e., the meaningful part of the pattern begins only when the first bit set to 1 is encountered, reading from left to right). This rule enables flashing patterns whose length is not a submultiple of 40 bits to be used.

1. The value of 0b00001011100 (0x0B00000000) means an LED flashing pattern that starts with the LED being on for 1 Bit_On_Period, then off for 1 Bit_Off_Period, on for 2 times the duration of Bit_On_Period, and off for 32 times the duration of Bit_Off_Period. Only the 36 least significant bits are meaningful. If the duration specified by the Repeat_Type and Repeats_Duration fields has not expired, then the pattern repeats, starting from the first 1 again, with the leading zeroes being ignored.

- Under certain conditions, described in [Section 3.9.2.10.2.4](#), the value of the `Flashing_Pattern` field is ignored and in this case it may be set to any value.

The average brightness of the LED during the on period shall be determined by the value of the Brightness field. Notwithstanding the above requirement for the LED to be illuminated for the duration of an on period, the LED may be pulsed at an appropriate rate during the period when it is on to reduce the perceived brightness. For example, if the Brightness value is set to 25 percent, this may be achieved by illuminating the LED with a 25 percent duty cycle during the period when it is on. If the LED is pulsed on and off to adjust the perceived brightness, then the pulse rate should take into account the typical human flicker fusion threshold and any potential interactions with ambient lighting. However, the method of controlling the brightness is selected by the implementer.

The Repeat_Type bit enables the time for which the flashing pattern shall be displayed to be specified as either:

- If the value of the Repeats_Duration field is non-zero, then the flashing pattern shall be repeated continuously until the duration specified in the command has elapsed. After the duration specified in the command has elapsed, the LED shall be turned off.

If the value of the Repeats_Duration field is non-zero, then the LED shall be turned off after the duration specified by the Repeat_Type and Repeats_Duration fields expires.

- If Repeat_Type = 0, then the LED shall be turned off continuously.
- If Repeat_Type = 1, then the LED shall be turned on continuously.

If an LED has been turned on by an LED Control command containing a non-zero `Repeats_Duration` parameter value, then the LED will turn off automatically after expiration of the specified duration, as described in [Section 3.9.2.10.2.4](#).

However, if an LED has been turned on by an LED Control command containing a Repeats_Duration value of zero (i.e., the LED is switched on continuously), the LED may be turned off by sending a further LED Control command. The new command may turn the LED off with immediate effect, if the Repeats_Duration and Repeat_Type parameter values are both set to zero as described in [Section 3.9.2.10.2.4](#), or after expiration of a specified duration by including a non-zero Repeats_Duration parameter value as described above.

3.9.2.10.4 Response

If the LED Control command is received and no error condition occurs, then the ESL shall respond by sending an LED State response as described in [Section 3.9.3](#). The ESL sends the LED State response as confirmation that the command was successfully received.

However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#).

3.9.2.11 LED Timed Control

The LED Timed Control command changes the state of an LED at a given absolute time. The LED Timed Control command is acknowledged using the LED State response (see [Section 3.9.3.2](#)), but the command has no effect on the state of the LED until the time specified by the Absolute_Time parameter.

If no error condition occurs, then the ESL shall, at the specified time, change the state of the LED in accordance with the values received in the command.

If an LED Timed Control command is scheduled to start when the LED is already active (where "active" has the meaning described in [Section 3.9.3.3.1](#)), then the LED Timed Control command shall supersede the older command at the time when the LED Timed Control command is scheduled to start, and the effect of the older command related to controlling the same LED shall then be terminated.

If the LED identified in the parameter values does not exist, then the ESL shall send an Error response.

Although the Basic State response is not sent in response to the LED Timed Control command, see [Section 3.9.3.3.2](#) for details of the effect of accepting an LED Timed Control command on the value of the Pending LED Update bit, which is exposed when the Basic State response is sent in response to another command.

After the specified time has been reached and the LED state has been updated as described above, the ESL shall set the Pending LED Update bit to False.

3.9.2.11.1 Handling multiple LED Timed Control commands

If the ESL supports one or more LEDs, then the ESL shall support having one, and only one, LED Timed Control command pending per LED. If an LED Timed Control command is received while an LED Timed Control command is already pending, then the ESL shall handle this event as follows:

- If the value of the Absolute_Time parameter in the new command is equal to the value of the Absolute_Time parameter in the pending LED Timed Control command, then the newly received command shall replace the old pending command.
- Otherwise, the ESL shall send the Error response: Queue Full, as described in [Section 3.9.3.1](#), as it is not permitted to have more than one LED Timed Control command pending. In this case, the LED Timed Control command that was already pending remains unchanged.
- If the value of the Absolute Time parameter is zero (0x00000000), then the pending LED Timed Control command shall be deleted.

3.9.2.11.2 Format

The LED Timed Control command has the following parameters:

Field	Size (bits)	Description
ESL_ID	8	See Section 3.9.1.1.1
LED_Index	8	Index of the LED
Color_Red	2	Red component (see Section 3.8)
Color_Green	2	Green component (see Section 3.8)
Color_Blue	2	Blue component (see Section 3.8)
Brightness	2	0 = 25% 1 = 50% 2 = 75% 3 = 100%
Flashing_Pattern	56	See Section 3.9.2.10.2.2
Repeat_Type	1	0: Number of times 1: Time duration
Repeats_Duration	15	Number of times to repeat or time duration in increments of 1 second
Absolute_Time	32	Time when the LED changes state

Table 3.18: LED Timed Control command parameters

The Absolute_Time is the time when the LED Timed Control command shall be executed. Repeat_Type and Repeats_Duration shall start at the specified absolute time. The interpretation of the fields shall otherwise be the same as specified in [Section 3.9.2.10](#).

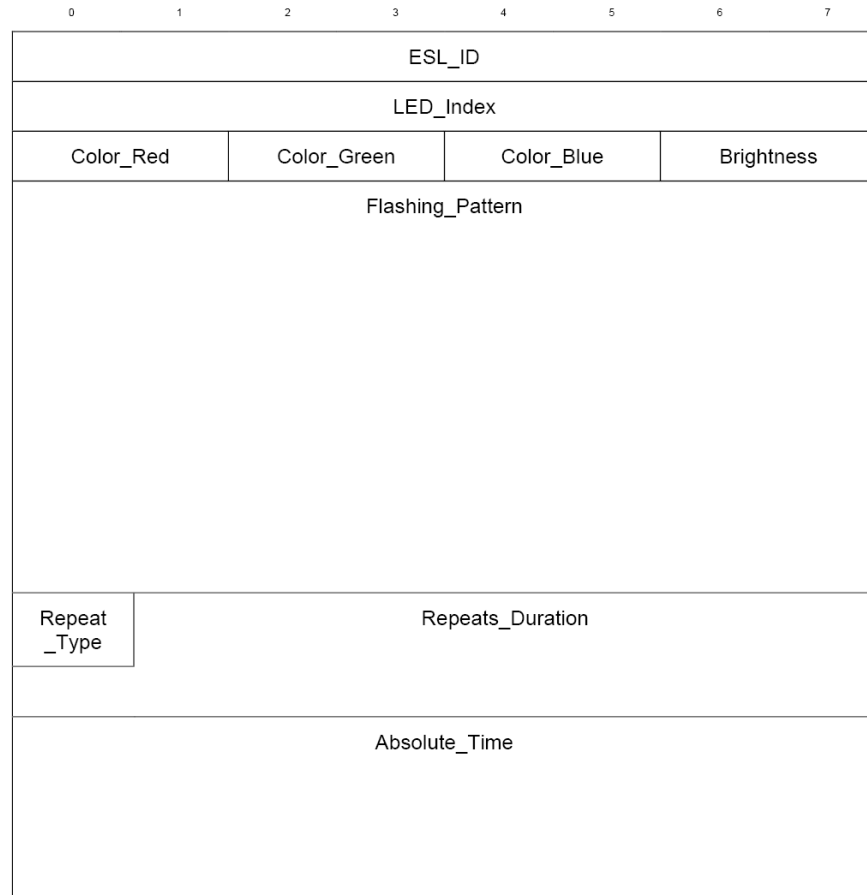


Figure 3.13: LED Timed Control command parameters

3.9.2.11.3 Response

If the LED Timed Control command is received and no error condition occurs, then the ESL shall respond by sending an LED State response as described in [Section 3.9.3](#). The ESL sends the LED State response as confirmation that the command was successfully received.

However, if an error condition occurs, the ESL shall send an Error response as described in [Section 3.9.3](#).

The response shall be sent upon receipt of the command (i.e., the response is not delayed awaiting the timed execution of the command).

3.9.2.12 Vendor-specific tag

All opcodes containing the Tag value "Vendor-specific" are vendor-specific.

If an opcode containing the vendor-specific tag is received by an ESL and the opcode is not recognized by the ESL, then the ESL shall respond by sending the Error response: Invalid Opcode.

Refer also to a relevant profile specification where requirements for identification of the vendor of the device may be specified.

3.9.2.12.1 Format

The format of the vendor-specific tag is specified by the vendor. However, the requirements described in [Section 3.9.1.1.1](#) continue to apply.

3.9.2.12.2 Response

Any type of response specified in [Section 3.9.3](#) may be sent, as specified by the vendor.

If the vendor-specific response code is used, then the contents of the response are also specified by the vendor.

3.9.3 Response behavior

When sent in a notification of the ESL Control Point characteristic, the characteristic value shall consist of one TLV formatted element using the TLV format specified in [Section 3.9.1](#). This value is referred to as the ESL response value.

The ESL response data types that may be sent in response to a particular command shall be constrained to the responses specified per command in [Section 3.9.2](#).

The ESL response data types and the parameter(s) that shall follow each ESL response opcode are described in [Table 3.19](#).

Opcode	Name	Description	Section
0x00	Error	The command could not be processed successfully	Section 3.9.3.1
0x01	LED State	Acknowledgment of a request to control an LED	Section 3.9.3.2
0x10	Basic State	General acknowledgment containing ESL status data	Section 3.9.3.3
0x11	Display State	Acknowledgment of a request to display an image	Section 3.9.3.4
0x_E	Sensor Value	Sensor report	Section 3.9.3.5
0x_F	Vendor-specific Response	Response data as specified by the vendor of the ESL	Section 3.9.3.6

Table 3.19: ESL response data types

Any opcode that is not yet assigned in [Table 3.19](#) is RFU.

In [Table 3.19](#), the underscore character (_) means that any value in the range 0x0 to 0xF is permitted. This allows variable length responses to be supported.

3.9.3.1 Error

The Error response has a single parameter that contains an error code as enumerated in [Table 3.20](#).

Error code (uint8)	Name	Description
0x00	RFU	Reserved for Future Use
0x01	Unspecified Error	Any error condition that is not covered by a specific error code below
0x02	Invalid Opcode	The opcode was not recognized
0x03	Invalid State	The request was not valid for the present ESL state

Error code (uint8)	Name	Description
0x04	Invalid Image_Index	The Image_Index value was out of range
0x05	Image Not Available	The requested image contained no image data
0x06	Invalid Parameter(s)	The parameter value(s) or length did not match the op-code
0x07	Capacity Limit	The required response could not be sent as it would exceed the payload size limit
0x08	Insufficient Battery	The request could not be processed because of a lack of battery charge
0x09	Insufficient Resources	The request could not be processed because of a lack of resources. This may be a temporary condition.
0x0A	Retry	The ESL is temporarily unable to give a full response (e.g., because the required sensor hardware was asleep) and the AP is asked to try the same command again
0x0B	Queue Full	The ESL is temporarily unable to add a further timed command to the queue of pending commands—the queue has reached its limit
0x0C	Implausible Absolute Time	The Absolute Time parameter value in the command is implausible
0x0D to 0xEF	RFU	Reserved for Future Use
0xF0 to 0xFF	Vendor-specific Error	Error codes defined by a vendor

Table 3.20: ESL response error codes

3.9.3.2 LED State

The LED State response is of fixed length and has one parameter as shown in [Table 3.21](#).

Field	Size (bits)	Description
LED_Index	8	Equal to the LED_Index parameter value received in the command

Table 3.21: LED State response parameters

3.9.3.3 Basic State

The Basic State response has a single parameter that consists of a 16-bit bitmap as shown in [Table 3.22](#).

Bit number	Name	Description	References
0	Service Needed	The ESL has detected a condition that needs service 0: False 1: True	Section 3.9.2.3
1	Synchronized	The ESL is synchronized to the AP 0: False 1: True	Section 2.7.3.3
2	Active LED	The ESL has one or more active LEDs 0: False 1: True	Section 3.9.3.3.1
3	Pending LED Update	The ESL has a timed LED update pending 0: False 1: True	Section 3.9.3.3.2
4	Pending Display Update	The ESL has a timed display update pending 0: False 1: True	Section 3.9.3.3.3
5 to 15	RFU	Reserved for Future Use	N/A

Table 3.22: Basic State response bitmap

3.9.3.3.1 Active LED

An LED shall be considered active if:

- the LED is the subject of an LED Control command (see [Section 3.9.2.10](#)) for which
 - the time specified by the Repeat_Type and Repeats_Duration parameters has not yet expired, and
 - the command requires the LED to be switched on for a least some of the time

or

- the LED is the subject of an LED Timed Control command (see [Section 3.9.2.11](#)) for which
 - the time specified by the Absolute_Time parameter has been reached or passed, and
 - the time specified by the Repeat_Type and Repeats_Duration parameters has not yet expired, and
 - the command requires the LED to be switched on for a least some of the time

3.9.3.3.2 Pending LED Update

If an LED Timed Control command has previously been accepted and not yet executed because the specified absolute time has not yet been reached, then the ESL shall set the value of the Pending LED Update bit in the Basic State response to True; otherwise the bit shall be set to False.

3.9.3.3.3 Pending Display Update

If a Display Timed Image command has previously been accepted and not yet executed because the specified absolute time has not yet been reached, then the ESL shall set the value of the Pending Display Update bit in the Basic State response to True; otherwise the bit shall be set to False.



3.9.3.4 Display State

The Display State response is of fixed length and has two parameters as shown in [Table 3.23](#).

Field	Size (bits)	Description
Display_Index	8	Equal to the Display_Index parameter value received in the command
Image_Index	8	The image index number of the image associated with the display identified by the Display_Index parameter

Table 3.23: Display State response parameters

3.9.3.5 Sensor Value

The Sensor Value response is of variable length and has two parameters as shown in [Table 3.24](#).

Field	Size (bits)	Description
Sensor_Index	8	Equal to the Sensor_Index parameter value received in the command
Sensor_Data	variable, up to 120 bits (15 octets)	Sensor data reported by the sensor identified by the Sensor_Index value

Table 3.24: Sensor Value response parameters

The Tag nibble contained in the opcode used in the response is a fixed value, but the length is variable and is indicated by the value of the Length nibble.

The total parameter size is equal to (Length + 1) octets as described in [Section 3.9.1.2](#). Because one octet in the response parameters is used for the Sensor_Index value, the ESL must therefore set the value of the Length nibble equal to the size of the associated Sensor_Data field, in units of octets.

The format of the sensor data for a given Sensor_Index value is known from the Sensor Information characteristic described in [Section 3.7](#).

3.9.3.6 Vendor-specific response

The Vendor-specific response has one or more parameters, specified by the vendor.

The length is variable and can be calculated from value of the Length nibble contained in the opcode used in the response. The total parameter size shall be equal to (Length + 1) octets as described in [Section 3.9.1.2](#).

3.9.4 ESL Control Point Procedure Timeout

In the context of the ESL Control Point characteristic, a procedure is started when a write to the ESL Control Point characteristic is successfully completed. The Server may take some time to process the request. The Server then sends a notification of the ESL Control Point characteristic containing a response, as described in [Section 3.9.3](#).



The ESL Control Point Procedure Timeout period, which is the maximum duration of elapsed time between the start of an ESL Control Point procedure and the transmission of the notification containing the response, shall be 30 seconds.

If the Server cannot complete the requested procedure within the ESL Control Point Procedure Timeout period, then the Server shall instead send an Error response within the ESL Control Point Procedure Timeout period. Error responses are enumerated in [Section 3.9.3.1](#). For example, to prompt the Client to repeat the same request later, the ESL may send the Error response: Retry.

If the Server cannot complete the requested procedure within the ESL Control Point Procedure Timeout period, and no other error response is relevant, then the Server shall send the Error response: Unspecified Error.

4 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
AP	access point
ATT	Attribute Protocol
ESL	electronic shelf label
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GSS	GATT Specification Supplement
ID	identifier
IV	initialization vector
LED	light-emitting diode
LSO	least significant octet
TLV	Tag Length Value
ms	millisecond
MSN	most significant nibble
RFU	Reserved for Future Use
UUID	Universally Unique Identifier

Table 4.1: Acronyms and abbreviations

5 References

- [1] Bluetooth Assigned Numbers, <https://www.bluetooth.com/specifications/assigned-numbers>
- [2] Bluetooth Core Specification, Version 5.4 or later
- [3] GATT Specification Supplement, <https://www.bluetooth.com/specifications/gss/>
- [4] Device Properties, <https://www.bluetooth.com/specifications/specs/device-properties>
- [5] Object Transfer Profile Specification, Version 1.0 or later
- [6] Object Transfer Service Specification, Version 1.0 or later
- [7] Supplement to the Bluetooth Core Specification, Version 13 or later