OpenLocate Beacon Specification

Revision



# Contents

[Revision History 2](#_Toc149133880)

[Introduction 3](#_Toc149133881)

[Purpose of this specification 3](#_Toc149133882)

[Terminology 3](#_Toc149133883)

[Advertisement format 5](#_Toc149133884)

[Location Elements 6](#_Toc149133885)

[Properties element 6](#_Toc149133886)

[Transmit Power field 7](#_Toc149133887)

[Flags field 7](#_Toc149133888)

[Geo location element 7](#_Toc149133889)

[Floorplan relative location element 8](#_Toc149133890)

[Identity element 9](#_Toc149133891)

[URL element 9](#_Toc149133892)

[Signature element 10](#_Toc149133893)

[Extended elements 10](#_Toc149133894)

[Transmitter Requirements 10](#_Toc149133895)

[Receiver Requirements 11](#_Toc149133896)

[Appendix A – MAC generation and validation procedures 11](#_Toc149133897)

[Key Generation 11](#_Toc149133898)

[Composing the Message 11](#_Toc149133899)

[Generation of the Message Authentication Code 12](#_Toc149133900)

[Validation 12](#_Toc149133901)

[Appendix B – Sample payloads 12](#_Toc149133902)

[Minimal beacon 12](#_Toc149133903)

[Full beacon 13](#_Toc149133904)

[References 15](#_Toc149133905)

# Revision History

**Document status:** Draft

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Change request# (Optional) | Document version | Date | Prepared / Modified by | Reviewed by | Approved by | Section and text revised |
|  | 0.1 | 06/26/2023 | Ben Dunsbergen |  |  | Initial draft |
|  | 0.2 | 07/03/2023 | Ben Dunsbergen |  |  | Added sections for signature and calibrated RSSI |
|  | 0.3 | 10/22/2023 | Ben Dunsbergen |  |  | * Add extension options for signature * Add seq/frag field * Renamed Transmit Power element to Properties element and added Flags field |
|  | 0.4 | wip | Ben Dunsbergen |  |  | Typos |

# Introduction

## Purpose of this specification

This document describes the formatting and use of the . The is a BLE advertisement used to broadcast the physical location of the transmitter.

## Terminology

BLE devices that want to broadcast their physical location can choose to implement the . The goal is for receivers to understand the location of the transmitting device without any consultation of a remote API.

For example, the transmitter is a location anchor. It’s mounted at a fixed location and transmits s. The receivers are mobile devices that listen for s. The receivers can now calculate their own location relative to the location anchors without previous knowledge about the anchors and without consulting any other APIs.

The support two types of location:

* Absolute location using latitude/longitude/altitude.
* Location relative to a floorplan

The amount of location information made available in the is flexible, it can support either absolute or relative location or both. There may or may not be uncertainty information available. Also, the latitude in absolute coordinates is optional.

# Advertisement format

The will be available in either legacy format (BT 4) or current format (BT 5) or both. The main difference is that in the legacy format the payload length is only 31 bytes and multiple advertisement frames will be needed to convey the total content of the .

The data is embedded inside the Service Data Element as defined in [2], Section 1.11 using a 2-byte UUID.

The actual data in the consists of multiple “Location Elements”.

* Each advertisement frame can contain only one Service Data Element
* Subsequent advertisements can have differing contents (different location elements)
* Multiple location elements could occur within a single service data.
* location elements can occur in any order.

The usage of the Service Data Element is shown in Figure 1

A data elements with a dotted line

Description automatically generated with medium confidence

Figure : Service Data Element

The seq/frag field uses 4 bits for sequence number, 3 bits for fragment number and the final bit to indicate last-fragment. See Figure 2

A rectangular object with numbers

Description automatically generated with medium confidence

Figure : Seq/Frag field

The sequence# field is increased for every advertisement interval. After 16 intervals the value rolls over and restarts at 0.

The fragment number indicates the fragment. This is useful when multiple frames are required to transmit an entire .

The last fragment field is 0 if more fragments are expected within the advertisement interval. It is set to 1 if this is the final fragment.

## Location Elements

The tag/len field uses the 3 most significant bits to indicate the tag (location element type) and the remaining 5 bits to indicate the length of the value field. Note that the byte used for tag/len is not included in the length count.

A picture containing line

Description automatically generated

Figure : Tag/Len field

Table 1 shows the currently defined location elements and their tags:

| Tag | Name | Purpose | Notes |
| --- | --- | --- | --- |
| 0 | Properties | Indicates the Transmit Power and flags |  |
| 1 | Geo location | Location data in WGS84 coordinates |  |
| 2 | Floor location | Location relative a to a floorplan |  |
| 3 | Identity | Identity of the Transmitter | Helps with correlating the BLE MAC address to a more friendly identifier of the transmitter |
| 4 | URL | Points to a location with additional info regarding the transmitter |  |
| 5 | Signature | Ensure integrity and authenticity of the data |  |
| 6 | Reserved |  | Do not use |
| 7 | Extension | For future use | Once more than 7 tags are needed, the extension tag can be used to add more. |

Table : Tag definitions for the Location Element

## Properties element

The purpose of this element is to convey certain device properties that are relevant to location. This element consists of 1 byte for Transmit power followed by one byte for flags and in the future potentially additional fields. See Figure 4

A table with text and images

Description automatically generated

Figure : Properties Element

### Transmit Power field

This field provides a means for the transmitter to indicate its transmit power to the receiver. In turn the receiver can use this to better estimate distance from the received signal strength. The measured power is a single byte containing a signed integer that specifies the measured signal strength at 1 meter from the transmitter.

The usage for this field follows the Proximity Beacon Spec from Apple (iBeacons). See [4], Section 2.2.

### Flags field

This field contains a bitmap with flags as defined in Table 2

|  |  |  |
| --- | --- | --- |
| Name | Value | Usage |
| isMobile | 0x01 | 1 for mobile devices  0 for stationary devices |
| <reserved> | 0x02 … 0x80 | For future use |

Table : Device Flags

## Geo location element

The geolocation element follows the definitions in RFC 6225 [3], section 2.2.2. The layout is shown in Figure 5. It is 16 bytes long.

When this element is present, it is expected that the latitude and longitude fields have valid values, The altitude fields, and the uncertainty fields may be unspecified.

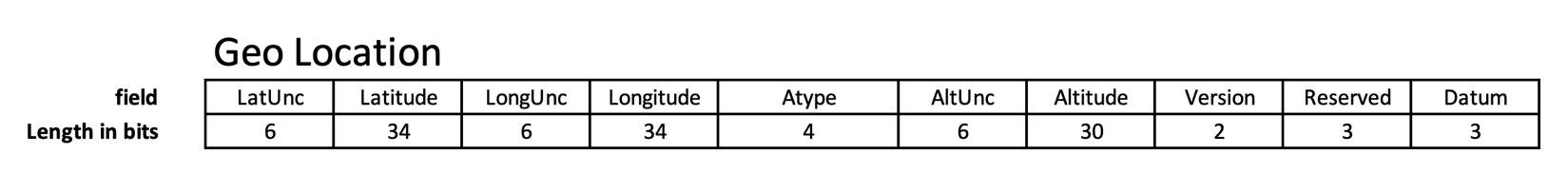


Figure : Geo location element

**Notes**:

* The Version field must be set to 1. Version 1 is the only version supported as of this version of the spec.
* The Datum field must be set to 1. WGS84 is the only supported Datum as of this version of the spec.
* The reserved field must be set to 0 by the transmitter and ignored by the receiver.

## Floorplan relative location element

This element provides location relative to a floorplan. This is useful only if the receiver has a means of obtaining floorplans ahead of time. The dimensions of the floorplan also need to be obtained out of band. The method of obtaining these floorplans and dimensions is outside of the scope of this specification.

The location is indicated using an x and y coordinate in 2D space relative to the floorplan. The floorplans are assumed to be rectangular. The origin (x=0; y=0) of this coordinate system is the top-left corner of the floorplan.

The uncertainty fields are intended to provide a region in which the actual location resides. If the uncertainty fields are specified, the uncertainty region is a rectangle, centered around the provided x, y coordinates. The sides of the region are 2\*X-Unc x 2\*Y-Unc. So, the actual horizontal offset is somewhere in the range between [X – X-Unc … X + X-Unc].

If the uncertainty fields are not specified, it means that the given X, Y coordinates are the best estimate and there is no indication of a confidence.

A picture containing text, font, screenshot, line

Description automatically generated

Figure : Floorplan relative location element

The fields are defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| field | Purpose | Range | Notes |
| X | Horizontal offset | 0-16,777,215 cm |  |
| Y | Vertical offset | 0-16,777,215 cm |  |
| X-Unc | Uncertainty in the X direction | 1-65,535 cm | 0 indicates uncertainty is unspecified.  All other values are encoded as the square root of the Uncertainty:  .  E.g. For an uncertainty of 100 cm, the value of X-Unc = 10 |
| Y-Unc | Uncertainty in the Y direction | 1-65,535 cm | 0 indicates uncertainty is unspecified.  All other values are encoded as the square root of the Uncertainty:  .  E.g. For an uncertainty of 100 cm, the value of Y-Unc = 10 |
| FloorId | Identifier that can be used to retrieve a floorplan |  |  |

Table : Floor Location fields

## Identity element

This element provides a means for the transmitter to publish additional identities for itself.

Depending on implementation, the receiver may see the BLE mac address of the receiver or in some cases (IOS) the receiver may only see a locally generated identifier. The additional identities can help the receiver to correlate the sender to a known entity.

The element supports three different types of identifiers: a mac address, a string, or an iBeacon identity. It is possible to advertise more than one identity in this element.

The format of the Identity element is shown in Figure 7.

A close-up of a number

Description automatically generated with low confidence

Figure : Identity element

The ID Types field is a bitmap indicating which identities are included. Usage of the ID Types field is specified in Table 4.

|  |  |  |
| --- | --- | --- |
| value | type | format |
| 0x01 | MAC address | 6 bytes |
| 0x02 | Text String | 1 byte length followed by 1...22 bytes UTF8 string. Note that this length limit is specifically for BT4 advertisements. For BT5 extended advertisements, the length may be longer. |
| 0x04 | iBeacon id | 16 bytes UUID followed by 2 bytes major# followed by 2 bytes minor# |

Table : Identity Types

If more than one identity is included, they must appear in the same order in which they are listed in Table 4. So, mac address first, string next, and iBeacon last.

## URL element

This element can be used to broadcast a http/https URL where more information can be found about the transmitter. The use of the URL is outside of the scope of this spec.

The format of this element is show in Figure 8.

A picture containing text, font, screenshot, line

Description automatically generated

Figure : URL Element

The URL scheme field and the Encoded URL field follow the definitions of the Eddystone-URL specification [5].

## Signature element

This element provides a data integrity check and an authenticity check for the . The first byte specifies the type of signature. As of this version of the spec only one signature type (AES-128-CMAC) is supported. The format for the signature element is shown in Figure 9.

A close-up of a document

Description automatically generated

Figure : Signature element

The size of the AES-128-CMAC Signature is 20 bytes. The content is timestamp followed by a Message Authentication Code (MAC). The algorithm used to generate the MAC is AES-128-CMAC.

Details for the generation and validation of the MAC are spelled out in Appendix A.

The timestamp field contains the number of seconds since Jan 1st, 1970. This is also known as Unix epoch time. The timestamp is encoded as an unsigned integer (no rollover until 2106). The timestamp represents UTC time. Not local time.

## Extended elements

Tag 7 is reserved for future elements that require a tag of 8 or higher, and/or elements with a length of 32 or greater.

An extended element contains 0xE0 in the tag/len field. This element itself starts with a 1-byte tag field, followed by a 1-byte length field, followed by the contents of the extended element.

Unlike all the other elements, the extended element holds the length field inside the payload and not inside the tag/len field. It is important for receivers to handle this element correctly so that they can properly parse these elements when they start appearing in the future.

The format is shown in Figure 10.

A picture containing text, font, screenshot, line

Description automatically generated

Figure : Extended elements

# Transmitter Requirements

**Subset of elements.** The transmitter can choose which location elements to include in the . The minimum required subset of elements is the Measured Power element and at least one of geolocation or floor location. Everything else is optional.

**Advertisement interval**. Every advertisement interval, the entire must be broadcasted. If a beacon requires multiple frames, then all fragments must be transmitted every advertisement interval. It is up to the transmitter to decide how the multiple frames are transmitted over time. Table 5 shows a few possible distributions for how 3 fragments (a, b, c) could be distributed over a 6 second advertisement interval.

A picture containing text, font, number, screenshot

Description automatically generated

Table : time distribution of multiple advertisements

**Ordering of elements**. Location elements can appear in any order. The only exception to that is that the signature element must be in the final fragment. It is recommended to combine elements in such a way that it minimizes the number of fragments.

# Receiver Requirements

For the most part, the receiver can simply parse each of the frames as it arrives, parse the contents, and extract the data out of it.

Only for signature checking, it is important to make sure that all fragments from a particular interval are reassembled into a complete beacon. The Seq/frag field can be used to reassemble all the fragments of an .

# Appendix A – MAC generation and validation procedures

## Key Generation

Usage of the signature field assumes that the transmitter and receiver share a secret passphrase. Secure exchange of this secret is out of the scope of this specification.

The secret is used to generate a key that is one of the required inputs for the AES-128-CMAC algorithm. To generate the key, use the PBKDF2 algorithm. In addition to the passphrase, PBKDF2 requires the following input parameters:

* Salt: : OpenLocate
* Hashing Algorithm : SHA256
* Iterations : 10,000

So, for example, the passphrase “HPE Aruba Networking” would generate the key:

3d93402b633288545eb07aa716eeccca

## Composing the Message

The message content is composed by concatenating the following:

* A 4-byte timestamp containing the Unix epoch time encoded as an unsigned integer.
* All the location elements concatenated together in the same order as they are defined in Table 1.
  + Include the tag/len field each element.
  + Do not include the signature element!
* If the identity element is present, then add nothing. If it is not present, add the 6-byte BLE transmitter mac address.

There’s no need for padding. The message can be of any length.

Here’s an example message that has a timestamp and 5 location elements: properties, geolocation, floor location, identity, and URL. (Line-breaks are added for readability. They’re not part of the actual message)

64A1D98F  
02CE00  
304c4ad6a705470c0ad9ae200000040041  
530F000C4E0D000A0A466966746820466C6F6F72  
72030011223344550A73657269616C23313233  
850168706507

## Generation of the Message Authentication Code

The Message Authentication Code is calculated using the AES-128-CMAC algorithm. The algorithm requires a key and a message which are explained above.

Following the same example key and message above, the generated MAC would be:

b6 5a e7 4d 51 95 33 04 20 a1 da 80 b8 82 f9 eb

Include the MAC along with the timestamp that was used in the message in the Signature element.

## Validation

The receiver is expected to generate the key and the message using the same steps as the transmitter, and then calculate a MAC locally.

Note that the ordering of the location elements during verification can be different than the order in which fragments were received.

The receiver should verify that:

* The timestamp is current.
* The received MAC matches the locally calculated MAC.

# Appendix B – Sample payloads

The implementer has flexibility as to which location elements are included in the . Below are example payloads of a minimal beacon and a fully featured beacon. Both examples use Bluetooth 4 regular advertisements.

## Minimal beacon

This example contains two elements: measured power and geo location. There’s just one advertisement that holds 2 elements. The measured power at 1m is measured as -50 dBm. The location in this example is the Aruba HQ building in San Jose, CA located at latitude 37.419243 and longitude -121.978808. In this example the altitude and uncertainties are not specified. The advertisement payload is:

191694FD090102CE0030004AD6A705030C0AD9AE000000000041

19 – length: 24 bytes  
 16 – Type: Service Data Element  
 94FD – UUID: FD94 in little-endian order  
 09 – Subtype: OpenLocate Beacon  
 01 – Seq/Frag: sequence# 0; fragment#0; last-frag=1  
 02 – Tag/Len: tag = 0 (measured power); Len = 1 byte  
 CE00 – Measured Power: -50 dBm; stationary  
 30 - Tag/Len: tag = 1(geolocation); Len = 16 bytes  
 004AD6A705030C0AD9AE000000000041 – LCI: see Table 6.

The interpretation of the LCI value is shown in Table 6.

| Field | Value | Binary Value | Interpretation |
| --- | --- | --- | --- |
| LatUnc | 0 | 000000 | Unspecified |
| Latitude | 0x04AD6A705 | 00 0100 1010 1101 0110 1010 0111 0000 0101 | 37.419243 |
| LongUnc | 0 | 00000 | Unspecified |
| Longitude | 0x30C0AD9AE | 11 0000 1100 0000 1010 1101 1001 1010 1110 | -121.978808 |
| AType | 0 | 0000 | No altitude specified |
| AltUnc | 0 | 000000 | N/A |
| Altitude | 0 | 00 0000 0000 0000 0000 0000 0000 0000 | N/A |
| Version | 1 | 01 | Version 1 |
| Reserved | 0 | 000 | N/A |
| Datum | 1 | 001 | WGS84 |

Table : LCI values

## Full beacon

This example contains all elements. The total beacon is too large for a single Bluetooth 4 advertisement frame. Therefore, it is split into 4 frames. This beacon advertises the following information:

* Frame 1
  + Properties: calibrated power = -50 dBm; device is stationary
  + Geolocation: Fifth floor of the Aruba HQ building in San Jose, CA, with uncertainty spanning the entire building
* Frame 2
  + Floor location: coordinates [31.5 ± 2.25, 25.7 ± 1.69] on “Fifth Floor”
* Frame 3:
  + Identities: mac address 00:11:22: 33:44:55 and string “serial#123”
  + URL: <https://www.hpe.com>
* Frame 4
  + Signature: generated with passphrase “HPE Aruba Networking” and timestamp 1688328591 (UTC: Sunday, July 2, 2023, 8:09:51 PM)

The payloads of these advertisements are:

Frame1: 191694FD094002CE00304C4AD6A705470C0AD9AE200000040041  
191694FD09 – Prefix  
 40 – Seq/Frag: seq=4; frag=0; last-frag=0  
 02CE00 – measured power element: -50 dBm  
 304C4AD6A705470C0AD9AE200000040041 – Location info. See Table 7.

Frame2: 191694FD0942530F000C4E0D000A0A466966746820466C6F6F72  
191694FD09 – Prefix  
 42 – Seq/Frag: seq=4; frag=1; last-frag=0  
 53 – Tag(2: floor location) / Len (19)  
 0F – X-Unc: 15 -> 2.25m  
 000C4E – X: 3150 -> 31.5m  
 0D – Y-Unc: 13 -> 1.69m  
 000A0A – Y: 2570 -> 25.7m  
 466966746820466C6F6F72 – “Fifth Floor”

Frame3: 1E1694FD094472030011223344550A73657269616C23313233850168706507  
1E1694FD09 – Prefix  
 44 – Seq/Frag: seq=4; frag=2; last-frag=0  
 72 – Tag(3: identity) / Len (18)  
 03 – Identity Types: 0x01 + 0x02 -> mac-address + string  
 001122334455 – Mac address: 00:11:22:33:44:55  
 0A – id string length  
 73657269616C23313233 – “serial#123”  
 85 – Tag (4: URL) / Length (5)  
 01 – URL scheme: “https://www.”  
 68706507 – encoded URL: “hpe.com”

Frame4: 1B1694FD0947B50064A1D98FB65AE74D5195330420A1DA80B882F9EB  
1B1694FD09 – Prefix  
 47 – Seq/Frag: seq=4; frag=3; last-frag=1  
 B5 – Tag (5: signature) / Len (21)  
 00 – SigType: AES-128-CMAC  
 64A1D98F - timestamp  
 B65AE74D5195330420A1DA80B882F9EB – Message Authentication Code

Alternatively, if Bluetooth 5 and extended advertisement is supported, the transmitter can opt to send the beacon in an extended advertisement frame. In this case there is a single advertisement and inside that, there is a single Service Data Element containing the entire beacon. The payload would be:

5C1694FD0941  
 02CE00  
 304C4AD6A705470C0AD9AE200000040041  
 530F000C4E0D000A0A466966746820466C6F6F72  
 72030011223344550A73657269616C23313233  
 850168706507   
 B50064A1D98F B65AE74D5195330420A1DA80B882F9EB

The table below shows the interpretation of the geolocation element:

| Field | Value | Binary Value | Interpretation |
| --- | --- | --- | --- |
| LatUnc | 19 | 010011 | 0.000488 degrees |
| Latitude | 0x04AD6A705 | 00 0100 1010 1101 0110 1010 0111 0000 0101 | 37.419243 |
| LongUnc | 17 | 010001 | 0.000195 degrees |
| Longitude | 0x30C0AD9AE | 11 0000 1100 0000 1010 1101 1001 1010 1110 | -121.978808 |
| AType | 2 | 0000 | Altitude in Floors |
| AltUnc | 0 | 000000 | N/A |
| Altitude | 0x00000400 | 00 0000 0000 0000 0000 0100 0000 0000 | 4 floors above ground floor. I.e., fifth floor |
| Version | 1 | 01 | Version 1 |
| Reserved | 0 | 000 | N/A |
| Datum | 1 | 001 | WGS84 |

Table : Geolocation data for the full beacon

# References

1. “Bluetooth Core Specification", Version 5.4, [https://www.bluetooth.com/specifications/specs/core-specification-5-4](https://www.bluetooth.com/specifications/specs/core-specification-5-4/)
2. “Supplement to the Bluetooth Core Specification”, Revision v11, <https://www.bluetooth.com/specifications/specs/core-specification-supplement-11/>
3. RFC 6225, “Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information”, July 2011, <https://datatracker.ietf.org/doc/html/rfc6225>
4. “Proximity Beacon Specification”, Release R1, Apple Inc., 2015-09-04, <https://developer.apple.com/ibeacon/>
5. “Eddystone-URL”, Google, Jul 4, 2016, <https://github.com/google/eddystone/blob/d522df8b5c60858ddd4dbaa69a53a0870e304579/eddystone-url/README.md>