

Mesh Configuration Database Profile

Bluetooth® Profile Specification

- **Revision:** v1.0.1
- **Revision Date:** 2022-06-21
- **Group Prepared By:** Mesh Working Group

Abstract:

This Bluetooth specification defines fundamental requirements to enable an interoperable Mesh Configuration Database for use with the Bluetooth Mesh Profile specification v1.0 [\[1\]](#).



Revision History

Revision Number	Date	Comments
v1.0	2021-09-14	Adopted by the Bluetooth SIG Board of Directors.
v1.0.1	2022-06-21	Adopted by the Bluetooth SIG Board of Directors.

Version History

Versions	Changes
v1.0 to v1.0.1	Incorporated errata E17655, E18059, E18313, E18346, E18480, E18533, E18868, E18953, E18977. Incorporated issues ID18907, ID18908, ID18986.

Acknowledgments

Name	Company
Robin Heydon	Qualcomm Technologies International, Ltd.
Jori Rintahaka	Silicon Laboratories Inc.
Inga Stotland	Intel Corporation
Piotr Winiarczyk	Silvair, Inc.
Szymon Slupik	Silvair, Inc.
Robert D. Hughes	Intel Corporation
Victor Zhodzishsky	Cypress Semiconductor Corporation
Bogdan Alexandru	NXP Semiconductors
Max Palumbo	Katerra
Omkar Kulkarni	Nordic Semiconductor ASA
Aleksander Nowakowski	Nordic Semiconductor ASA
Prashant Pandey	ST Microelectronics



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 © 2017–2022. All copyrights in the Bluetooth Specifications themselves are owned by Apple Inc., Ericsson AB, Intel Corporation, 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	5
1.1	Language.....	5
1.1.1	Language conventions	5
1.1.2	Reserved for Future Use	6
1.1.3	Prohibited.....	6
1.2	Table requirements.....	6
1.3	Conformance	6
1.4	Bluetooth specification release compatibility	6
1.5	Terminology	6
2	Mesh Configuration Database format.....	7
2.1	Mesh Object.....	7
2.1.1	Provisioner object	8
2.1.2	Network key object	10
2.1.3	Application key object.....	11
2.1.4	Node object.....	12
2.1.5	Group object	21
2.1.6	Scene object.....	22
2.1.7	exclusionList object	22
3	Security considerations	23
4	Mesh Configuration Database JSON schema.....	24
5	Example Mesh Configuration Database	41
6	Acronyms and abbreviations	60
7	References.....	61
	Appendix A: Mesh Configuration Database management.....	62
A.1	Database creation.....	62
A.2	Provisioning complete	62
A.3	Processing a Config Composition Data Status message.....	63
A.4	Processing miscellaneous configuration messages.....	63
A.5	Key refresh	65
A.6	Node removal and IV update procedure	65
A.7	Scene maintenance	66



1 Introduction

This Bluetooth specification defines fundamental requirements to enable an interoperable Mesh Configuration Database for use with the Bluetooth Mesh Profile specification v1.0 [1]. The Mesh Profile specification defines procedures for provisioning devices as nodes and configuring nodes using a standard set of messages.

For the purpose of this specification, the term Mesh Manager denotes a device that supports the Provisioner role and the Configuration Client model as described in [1]. To enable the sharing of provisioning and configuration data between multiple Mesh Managers that may be from different manufacturers, this specification defines a standardized Mesh Configuration Database. This database can be exchanged between Mesh Managers. Mechanisms used to exchange Mesh Configuration Databases are not defined in this specification.

1.1 Language

1.1.1 Language conventions

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

shall	<u>is required to</u> – used to define requirements.
must	is used to express: a natural consequence of a previously stated mandatory requirement. OR an indisputable statement of fact (one that is always true regardless of the circumstances).
will	<u>it is true that</u> – only used in statements of fact.
should	<u>is recommended that</u> – used to indicate that among several possibilities one is recommended as particularly suitable, but not required.
may	<u>is permitted to</u> – used to allow options.
can	<u>is able to</u> – used to relate statements in a causal manner.
is	<u>is defined as</u> – used to further explain elements that are previously required or allowed.
note	Used to indicate text that is included for informational purposes only and is not required in order to implement the specification. Each note is clearly designated as a “Note” and set off in a separate paragraph.

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



1.1.2 Reserved for Future Use

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

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

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

The acronym RFU is equivalent to Reserved for Future Use.

1.1.3 Prohibited

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

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

"Prohibited" is never abbreviated.

1.2 Table requirements

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

1.3 Conformance

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

1.4 Bluetooth specification release compatibility

This specification is compatible with the Bluetooth Mesh Profile specification v1.0 [1].

1.5 Terminology

The terminology used throughout this specification is defined in the Mesh Profile specification [1], unless otherwise defined in this specification.



2 Mesh Configuration Database format

This specification defines a representation of the Mesh Configuration Database in JavaScript Object Notation (JSON), as defined by [2], and follows the format defined by the JSON Schema Draft 4 [3]. Guidelines for creating the Mesh Configuration Database are provided in Appendix A.

2.1 Mesh Object

A Mesh Configuration Database shall contain a single JSON object that describes the mesh network, called the Mesh Object. The Mesh Object shall only contain information about a single mesh network and associated subnets. The Mesh Object documents the current state of the mesh network known by the device that generates and maintains this database.

The Mesh Object contains the properties defined in Table 2.1.

Property Name	Type	Requirements
\$schema	URL	M
id	URL	M
version	string	M
meshUUID	string	M
meshName	string	M
timestamp	string	M
partial	boolean	M
provisioners	array of objects	M
netKeys	array of objects	M
appKeys	array of objects	M
nodes	array of objects	M
groups	array of objects	M
scenes	array of objects	M
networkExclusions	array of objects	O

Table 2.1: Mesh Object properties

The \$schema property contains a URL (see Section 4) that points to the version of the JSON Schema specification that is used to generate the Mesh Configuration Database schema, which corresponds to JSON Schema Draft 4 [3].

The id property contains a URL (see Section 4) that points to the version of the Mesh Configuration Database schema that defines the format for this Mesh Object.



The version property contains a string that represents a version number of the Mesh Configuration Database that this Mesh Object supports. The version property for the Mesh Configuration Database that is defined by this specification is “1.0.1”.

The meshUUID property is defined as a string that represents the 128-bit Universally Unique Identifier (UUID), which allows differentiation among multiple mesh networks. This string shall follow the UUID string representation format as defined by [7].

The meshName property contains a UTF-8 string, which should be a human-readable name for the mesh network. This is useful when a Mesh Manager tracks multiple mesh networks to enable a user to identify different mesh networks.

The timestamp property contains a string that represents the last time the Mesh Object has been modified. The timestamp is based on Coordinated Universal Time (UTC) and follows the “date-time” format as defined by JSON Schema Draft 4 [3], which is based on the Internet date/time format described in Section 5.6 of RFC 3339 [6]:

YYYY-MM-DDThh:mm:ssZ or YYYY-MM-DDThh:mm:ss+/- timeoffset

where “YYYY” denotes a year; “MM” denotes a two-digit month (01 to 12); “DD” denotes a two-digit day of the month (01 to 31); “hh” denotes a two-digit hour (00 to 23); “mm” denotes a two-digit minute (00 to 59); “ss” denotes a two-digit second (00 to 59); “Z” denotes the UTC time zone; and “timeoffset” denotes the offset between local time and UTC in the format of +hh:mm or -hh:mm.

The partial property indicates if this Mesh Configuration Database is part of a larger database.

The provisioners property contains an array of provisioner objects (see Section 2.1.1) that includes information about known Provisioners and ranges of addresses that have been allocated to these Provisioners.

The netKeys property contains an array of network key objects (see Section 2.1.2) that includes information about network keys used in the mesh network.

The appKeys property contains an array of application key objects (see Section 2.1.3) that includes information about application keys used in the mesh network.

The nodes property contains an array of node objects (see Section 2.1.4) that includes information about mesh nodes in the mesh network.

The groups property contains an array of group objects (see Section 2.1.5) that includes information about groups configured in the mesh network.

The scenes property contains an array of scene objects (see Section 2.1.6) that includes information about scenes configured in the mesh network.

The networkExclusions property contains the array of exclusionList objects (see section 2.1.7).

2.1.1 Provisioner object

A Provisioner is a mesh node that is capable of provisioning a device to the mesh network as described in [1], and is represented by a provisioner object in the Mesh Configuration Database.



The provisioner object contains the properties defined in [Table 2.2](#).

Property Name	Type	Requirements
provisionerName	string	M
UUID	string	M
allocatedUnicastRange	array of objects	M
allocatedGroupRange	array of objects	M
allocatedSceneRange	array of objects	M

Table 2.2: Provisioner object properties

The provisionerName property contains a UTF-8 string, which should be a human-readable name of the Provisioner associated with the mesh network.

The UUID property contains a string that represents the 128-bit Device UUID [1]. This string shall follow the UUID string representation format as defined by [7]. If the Provisioner can perform configuration procedures, then the nodes property of the Mesh Object (see Section 2.1) contains a node object (described in Section 2.1.4) that matches the Provisioner's UUID.

The allocatedUnicastRange property contains an array of unicast range objects (see Section 2.1.1.1).

The allocatedGroupRange property contains an array of group range objects (see Section 2.1.1.2).

The allocatedSceneRange property contains an array of scene range objects (see Section 2.1.1.3).

2.1.1.1 Unicast range object

The unicast range object represents the range of unicast addresses that the Provisioner can allocate to new devices when they are provisioned onto the mesh network, without needing to coordinate the node additions with other Provisioners. The unicast range object contains the properties defined in [Table 2.3](#).

Property Name	Type	Requirements
lowAddress	string	M
highAddress	string	M

Table 2.3: Unicast range object properties

The lowAddress and highAddress properties contain 4-character hexadecimal strings representing values from 0x0001 to 0x7FFF. The value of the lowAddress property shall be less than or equal to the value of the highAddress property.

Note: The process for allocating the range of unicast addresses is implementation specific.

2.1.1.2 Group range object

The group range object represents the range of group addresses that the Provisioner can allocate to newly created groups, without needing to coordinate the group additions with other Provisioners.



The group range object contains the properties defined in [Table 2.4](#).

Property Name	Type	Requirements
lowAddress	string	M
highAddress	string	M

Table 2.4: Group range object properties

The lowAddress and highAddress properties contain 4-character hexadecimal strings, representing values from 0xC000 to 0xFEFF. The value of the lowAddress property shall be less than or equal to the value of the highAddress property.

Note: The process for allocating the range of group addresses is implementation specific.

2.1.1.3 Scene range object

The scene range object represents the range of scene numbers that the Provisioner can use to register new scenes in the mesh network, without needing to coordinate the allocated scene numbers with other Provisioners.

The scene range is an object that contains the properties defined in [Table 2.5](#).

Property Name	Type	Requirements
firstScene	string	M
lastScene	string	M

Table 2.5: Scene range object properties

The firstScene and lastScene properties contain 4-character hexadecimal strings, representing values from 0x0001 to 0xFFFF. The value of the firstScene property shall be less than or equal to the value of the lastScene property.

Note: The process for allocating the range of scene numbers is implementation specific.

2.1.2 Network key object

The network key object represents the state of the mesh network key that is used for securing communication at the network layer [\[1\]](#).

The network key object contains the properties shown in [Table 2.6](#).

Property Name	Type	Requirements
name	string	M
index	integer	M
phase	integer	M

Property Name	Type	Requirements
key	string	M
minSecurity	string	M
oldKey	string	C.1
timestamp	string	M

Table 2.6: Network key object properties

C.1: Mandatory if the phase property is non-zero; otherwise excluded.

The name property contains a UTF-8 string, which should be a human-readable name for the mesh subnet associated with this network key.

The index property contains an integer from 0 to 4095 that represents the NetKey index for this network key.

The phase property contains an integer with a value of 0, 1, or 2 that represents the Key Refresh phase [1] for the subnet associated with this network key.

The key property contains a 32-character hexadecimal string that represents the 128-bit network key.

The minSecurity property contains a string with a value of either “insecure” or “secure”, which describes a minimum security level for a subnet associated with this network key. If all the nodes on the subnet associated with this network key have been provisioned using the Secure Provisioning procedure [1], then the value of minSecurity property for the subnet is set to “secure”; otherwise, the value of the minSecurity is set to “insecure”.

The oldKey property contains a 32-character hexadecimal string that represents the 128-bit network key, and shall be present when the phase property has a non-zero value, such as when the Key Refresh procedure is in progress. The value of the oldKey property contains the previous network key.

The timestamp property contains a string that represents the last time the value of the phase property has been updated. The timestamp is based on UTC and follows the “date-time” format as defined by JSON Schema Draft 4 [3], which is based on Internet date/time format described in Section 5.6 of RFC 3339 [6]:

YYYY-MM-DDThh:mm:ssZ or YYYY-MM-DDThh:mm:ss+/- timeoffset

where “YYYY” denotes a year; “MM” denotes a two-digit month (01 to 12); “DD” denotes a two-digit day of the month (01 to 31); “hh” denotes a two-digit hour (00 to 23); “mm” denotes a two-digit minute (00 to 59); “ss” denotes a two-digit second (00 to 59); “Z” denotes the UTC time zone; and “timeoffset” denotes the offset between local time and UTC in the format of +hh:mm or -hh:mm.

2.1.3 Application key object

The application key object represents the state of a mesh application key that is used for securing communication at the access layer [1].



The application key object contains the properties defined in [Table 2.7](#).

Property Name	Type	Requirements
name	string	M
index	integer	M
boundNetKey	integer	M
key	string	M
oldKey	string	C.1

Table 2.7: Application key object properties

- C.1: Mandatory if the key refresh phase of the network key to which this application key is bound is non-zero, and if this application key is being updated by the Mesh Manager; otherwise excluded.

The name property contains a UTF-8 string, which should be a human-readable name for the application functionality associated with this application key (e.g., “Home Automation”).

The index property contains an integer from 0 to 4095 that represents the AppKey index for this application key.

The boundNetKey property contains a corresponding NetKey index from the netKeys property of the Mesh Object. All application keys are bound to network keys that are documented in the Mesh Object. That is, the bound network key index corresponds to the index value of one of the network key entries in the Mesh Object’s netKeys array (see Sections 2.1 and 2.1.2).

The key property contains a 32-character hexadecimal string that represents the 128-bit application key.

The oldKey property contains a 32-character hexadecimal string that represents the 128-bit application key, and is present if the application key is updated during the Key Refresh procedure. The value of the oldKey property contains the previous application key.

2.1.4 Node object

The node object represents a configured state of a mesh node.

The node object contains the properties defined in [Table 2.8](#).

Property Name	Type	Requirements
UUID	string	M
unicastAddress	string	M
deviceKey	string	C.4
security	string	M
netKeys	array of objects	M



Property Name	Type	Requirements
configComplete	boolean	M
name	string	O
cid	string	C.1
pid	string	C.1
vid	string	C.1
crpl	string	C.1
features	object	C.1
secureNetworkBeacon	boolean	C.5
defaultTTL	integer	C.5
networkTransmit	object	C.5
relayRetransmit	object	C.5
appKeys	array of objects	M
elements	array of objects	M
excluded	boolean	M
heartbeatPub	object	C.2
heartbeatSub	object	C.3

Table 2.8: Node object properties

C.1: Present if node composition data page 0 is available; otherwise omitted.

C.2: Present if heartbeat publication is configured; otherwise omitted.

C.3: Present if heartbeat subscription is configured; otherwise omitted.

C.4: Present if the partial property is set to “false”; otherwise optional.

C.5: If the value is unknown, this property is excluded.

The UUID property contains a string that represents the 128-bit Device UUID [1]. This string shall follow the UUID string representation format as defined by [7].

The unicastAddress property contains a 4-character hexadecimal string representing a value from 0x0001 to 0x7FFF that is the primary unicast address of this node.

The deviceKey property contains a 32-character hexadecimal string that represents the 128-bit device key for this node.

The security property contains a string with two allowed values, “insecure” and “secure”, representing the level of security for the subnet on which the node has been originally provisioned.



The `netKeys` property contains an array of node network key objects (see Section 2.1.4.1) that includes information about the network keys known to this node.

The `configComplete` property contains a Boolean value that represents whether the Mesh Manager finished configuring this node. The property's value is set to "true" once the Mesh Manager determines that this node's configuration process is completed; otherwise, the property's value is set to "false".

The `name` property contains a UTF-8 string, which should be a human-readable name that can identify this node within the mesh network.

The `cid` property contains a 4-character hexadecimal string that represents a 16-bit Company Identifier (CID) assigned by the Bluetooth SIG [4]. The value of this property is obtained from node composition data [1].

The `pid` property contains a 4-character hexadecimal string that represents a 16-bit, vendor-assigned Product Identifier (PID). The value of this property is obtained from node composition data [1].

The `vid` property contains a 4-character hexadecimal string that represents a 16-bit, vendor-assigned product Version Identifier (VID). The value of this property is obtained from node composition data [1].

The `crpl` property contains a 4-character hexadecimal string that represents a 16-bit value indicating the minimum number of Replay Protection List (RPL) entries for this node. The value of this property is obtained from node composition data [1].

The `features` property contains a features object (see Section 2.1.4.2).

The `secureNetworkBeacon` property contains a Boolean value that represents whether the node is configured to send Secure Network beacons.

The `defaultTTL` property contains an integer from 0 to 127 that represents the default Time to Live (TTL) value used when sending messages [1].

The `networkTransmit` property contains a network transmit object (see Section 2.1.4.3).

The `relayRetransmit` property contains a relay retransmit object (see Section 2.1.4.4).

The `appKeys` property contains an array of node application key objects (see Section 2.1.4.5) that includes information about the application keys known to this node.

The `elements` property contains an array of element objects (see Section 2.1.4.6).

The `excluded` property contains a Boolean value that is set to "true" when the node is in the process of being deleted and is excluded from the new network key distribution during the Key Refresh procedure; otherwise, it is set to "false".

The `heartbeatPub` property contains a heartbeat publish object (see Section 2.1.4.7).

The `heartbeatSub` property contains a heartbeat subscribe object (see Section 2.1.4.8).

2.1.4.1 Node network key object

The node network key object represents the state of a network key distributed to a mesh node by a Mesh Manager.

The node network key object contains the properties defined in [Table 2.9](#).

Property Name	Type	Requirements
index	integer	M
updated	boolean	M

Table 2.9: Node network key object properties

The index property contains an integer from 0 to 4095 that represents the NetKey index for this network key. The network key index corresponds to the index value of one of the network key entries in the Mesh Object's netKeys array (see Sections 2.1 and 2.1.2).

The updated property contains a Boolean value that is set to “false”, unless the Key Refresh procedure is in progress and the network key has been successfully updated.

2.1.4.2 Features object

The features object represents the functionality of a mesh node that is determined by the set features that the node supports [1].

The features object contains at least one of the properties defined in [Table 2.10](#).

Property Name	Type	Requirements
relay	integer	C.1
proxy	integer	C.2
lowPower	integer	C.3
friend	integer	C.4

Table 2.10 Features object properties

- C.1: The value of the relay property is set to 2 if the node does not support the Relay feature. The value is set to 0 if the Relay feature is not enabled or to 1 if the Relay feature is enabled. If the state of the Relay feature is unknown, this property is excluded.
- C.2: The value of the proxy property is set to 2 if the node does not support the Proxy feature. The value is set to 0 if the Proxy feature is not enabled or to 1 if the Proxy feature is enabled. If the state of the Proxy feature is unknown, this property is excluded.
- C.3: The value of the lowPower property is set to 2 if the node does not support the Low Power feature or to 1 if the node supports the Low Power feature. If the support of the Low Power feature is unknown, this property is excluded.

- C.4: The value of the friend property is set to 2 if the node does not support the Friend feature. The value is set to 0 if the Friend feature is not enabled or to 1 if the Friend feature is enabled. If the state of the Friend feature is unknown, this property is excluded.

2.1.4.3 Network transmit object

The network transmit object represents the parameters of the transmissions of network layer messages originating from a mesh node [1].

The network transmit object contains the properties defined in [Table 2.11](#).

Property Name	Type	Requirements
count	integer	M
interval	integer	M

Table 2.11: Network transmit object properties

The count property contains an integer from 1 to 8 that represents the number of transmissions for network messages.

The interval property contains an integer from 10 to 320 that represents the interval in milliseconds between the transmissions.

2.1.4.4 Relay retransmit object

The relay retransmit object represents the parameters of the retransmissions of network layer messages relayed by a mesh node [1].

The relay retransmit object contains the properties defined in [Table 2.12](#).

Property Name	Type	Requirements
count	integer	M
interval	integer	M

Table 2.12 Relay retransmit object properties

The count property contains an integer from 1 to 8 that represents the number of transmissions for relay messages.

The interval property contains an integer from 10 to 320 that represents the interval in milliseconds between the transmissions.

2.1.4.5 Node application key object

The node application key object represents the state of an application key distributed to a mesh node by a Mesh Manager.

The node application key object contains the properties defined in [Table 2.13](#).



Property Name	Type	Requirements
index	integer	M
updated	boolean	M

Table 2.13: Node application key object properties

The index property contains an integer from 0 to 4095 that represents the AppKey index for this application key. The application key index corresponds to the index value of one of the application key entries in the Mesh Object's appKeys array (see Sections 2.1 and 2.1.3).

The updated property contains a Boolean value that is set to "false", unless the Key Refresh procedure is in progress and the application key has been successfully updated.

2.1.4.6 Element object

The element object represents a mesh element that is defined as an addressable entity within a mesh node [1].

The element object contains the properties defined in Table 2.14.

Property Name	Type	Requirements
name	string	O
index	integer	M
location	string	M
models	array of objects	M

Table 2.14: Element object properties

The name property contains a UTF-8 string, which should be a human-readable name that can identify an element within the node.

The index property contains an integer from 0 to 255 that represents the numeric order of the element within this node. Because the node has at least one element, the primary element, as required by the Mesh Profile specification [1], Section 2.3.4, the elements array of the node object (see Section 2.1.4) must contain at least one element object with the value of the index property set to zero, which corresponds to the primary element within the node.

The location property contains a 4-character hexadecimal string that represents a description of the element's location (defined in the GATT Bluetooth Namespace Descriptors section of the Bluetooth SIG Assigned Numbers webpage [4]).

The models property contains an array of model objects (see Section 2.1.4.6.1).

2.1.4.6.1 Model object

The model object represents a configured state of a mesh model.

The model object contains the properties defined in Table 2.15.



Property Name	Type	Requirements
modelId	string	M
subscribe	array of strings	M
publish	object	O
bind	array of integers	M

Table 2.15: Model object properties

The modelId property contains either a 4-character hexadecimal string that represents a Bluetooth SIG-defined model identifier [1] and [5], or an 8-character hexadecimal string that represents a vendor-defined model identifier. In the latter case, the first 4 characters correspond to a Bluetooth-assigned Company Identifier [4], and the last 4 characters contain a vendor-assigned model identifier.

The subscribe property contains an array of 4-character hexadecimal strings, with values from 0xC000 to 0xFFFE, that represents group addresses and/or an array of 32-character hexadecimal strings that represents virtual label UUIDs.

The publish property contains a publish object (see Section 2.1.4.6.1.1) that describes the configuration of this model's publication.

The bind property contains an array of integers that represents indexes of the application keys (see Section 2.1.3) to which this model is bound. Each application key index corresponds to the index values of one of the application key entries in the node's appKeys array (see Sections 2.1.4 and 2.1.4.5).

2.1.4.6.1.1 Publish object

The publish object represents parameters that define how the messages are published by a mesh model [1].

The publish object contains the properties defined in Table 2.16.

Property Name	Type	Requirements
address	string	M
index	integer	M
ttl	integer	M
period	object	M
credentials	integer	M
retransmit	object	M

Table 2.16: Publish object properties

The address property contains the publication address for the model. It can be either a 4-character hexadecimal string that represents a group or unicast address, or a 32-character hexadecimal string that represents virtual label UUIDs.



The index property contains an integer that represents an application key index (see Section 2.1.3), indicating which application key to use for the publication. The application key index corresponds to the index value of one of the application key entries in the node's appKeys array (see Sections 2.1.4 and 2.1.4.5).

The ttl property contains an integer from 0 to 127 that represents the Time to Live (TTL) value for published messages or an integer with a value of 255 that indicates that the node's default TTL is used for publication.

The period property contains a publish period object (see Section 2.1.4.6.1.2) that describes the interval between subsequent publications. If the value of this property is 0, the periodic publication is disabled (as defined in [1]).

The credentials property contains an integer of 0 or 1 that represents whether managed flooding security material [8] (0) or friendship security material (1) is used.

The retransmit property contains a retransmit object (see Section 2.1.4.6.1.3) that describes the number of times a message is published and the interval between retransmissions of the published messages.

2.1.4.6.1.2 Publish period object

The publish period object determines the interval at which messages are published by a model and is defined by two values: the number of steps and step resolution.

The publish period object contains the properties defined in Table 2.17.

Property Name	Type	Requirements
numberOfSteps	integer	M
resolution	integer	M

Table 2.17: Publish period object

The numberOfSteps property contains an integer from 0 to 63 that represents the number of steps used to calculate the publish period (as described in [1]).

The resolution property contains an integer that represents the publish step resolution in milliseconds. The allowed values are: 100, 1000, 10000, and 600000.

The publish period is calculated as a product of the number of steps and step resolution. For example, if the value of the numberOfSteps property is 3 and the value of the resolution property is 1000, the publish period is 3000 ms.

2.1.4.6.1.3 Retransmit object

The retransmit object is used to describe the number of times a message is published and the interval between retransmissions of the published messages.

The retransmit object contains the properties defined in Table 2.18.



Property Name	Type	Requirements
count	integer	M
interval	integer	M

Table 2.18: Retransmit object properties

The count property contains an integer from 0 to 7 that represents the number of retransmissions for published messages. A value of 0 represents no retransmissions (as described in [1]).

The interval property contains an integer from 50 to 1600, with a resolution of 50, that represents the interval in milliseconds between the transmissions.

2.1.4.7 Heartbeat publication object

The heartbeat publication object represents parameters that define the sending of periodic Heartbeat transport control messages [1].

The heartbeat publication object contains the properties defined in Table 2.19.

Property Name	Type	Requirements
address	string	M
period	integer	M
ttl	integer	M
index	integer	M
features	array of strings	M

Table 2.19: Heartbeat publication object properties

The address property represents the destination address for the heartbeat messages. It contains a 4-character hexadecimal string that represents a group or unicast address.

The period property contains an integer from 0 to 65536 that represents the cadence of periodical heartbeat messages in seconds.

The ttl property contains an integer from 0 to 127 that represents the Time to Live (TTL) value for the heartbeat messages.

The index property contains an integer that represents a network key index (see Section 2.1.2), indicating which network key to use for the heartbeat publication. The network key index corresponds to the index value of one of the network key entries in the node's netKeys array (see Sections 2.1.4 and 2.1.4.1).

The features property contains an array of strings that describe the features that trigger the sending of heartbeat messages when changed. The allowed values are "relay", "proxy", "friend", and "lowPower".

2.1.4.8 Heartbeat subscription object

The heartbeat subscription object represents parameters that define the receiving of periodical Heartbeat transport control messages [1].

The heartbeat subscription object contains the properties defined in [Table 2.20](#).

Property Name	Type	Requirements
source	string	M
destination	string	M

Table 2.20: Heartbeat subscription object properties

The source property contains the source address for Heartbeat messages that a node processes. It is a 4-character hexadecimal string that represents a unicast address.

The destination property represents the destination address for the Heartbeat messages. It contains a 4-character hexadecimal string that represents a group or unicast address.

2.1.5 Group object

If a node is part of a Group, at least one model of the node is subscribed to the Group's group address.

A Group may have a Parent Group. In this case, all the models of a node that are subscribed to the Group's address are also subscribed to the Parent Group's address.

For example, the Second-Floor Group is a parent of the Bedroom Group and the Guest Bedroom Group. In this case, at least one model of all the nodes of the Bedroom Group is subscribed to a group address or virtual label of the Bedroom Group and Second-Floor Group; and at least one model of all the nodes of the Guest Bedroom Group is subscribed to the group address or virtual label of the Guest Bedroom Group and the Second-Floor Group.

The group object contains the properties defined in [Table 2.21](#).

Property Name	Type	Requirements
name	string	M
address	string	M
parentAddress	string	M

Table 2.21: Group object properties

The name property contains a UTF-8 string, which should be a human-readable name for this group within the mesh network.

The address property contains a 4-character hexadecimal string from 0xC000 to 0xFEFF or a 32-character hexadecimal string of virtual label UUID, and is the address of the group.



The parentAddress property contains a 4-character hexadecimal string or a 32-character hexadecimal string and represents an address of a parent group in which this group is included. The parentAddress property contains a hexadecimal string that represents 0x0000, a number from 0xC000 to 0xFEFF, or a 32-character hexadecimal string of virtual label UUID. The value of the parentAddress property shall not be equal to the value of the address property. The value of 0x0000 indicates that the group is not included in another group (i.e., the group has no parent).

2.1.6 Scene object

The scene object contains the properties defined in [Table 2.22](#).

Property Name	Type	Requirements
name	string	M
number	string	M
addresses	array of strings	M

Table 2.22: Scene object properties

The name property contains a string that represents the name of this scene.

The number property contains a 4-character hexadecimal string that represents a value from 0x0001 to 0xFFFF that represents the scene number.

The addresses property contains an array of 4-character hexadecimal strings that represent the addresses of elements whose Scene Register state contains this scene as described in [\[5\]](#).

2.1.7 exclusionList object

The exclusionList object represents the unicast addresses that are excluded by a Mesh Manager for a particular IV index. The exclusionList object contains the properties defined in [Table 2.23](#).

Property Name	Type	Requirements
ivIndex	integer	M
addresses	array of strings	M

Table 2.23: exclusionList object properties

The ivIndex property contains the integer value of the IV index of the mesh network that was in use while the unicast addresses were marked as excluded.

The addresses property contains an array of 4-character hexadecimal strings, representing the excluded unicast addresses.

3 Security considerations

Because the Mesh Configuration Database contains sensitive information, such as network, application, and device keys, its security is critical to the overall security of the associated mesh network. Therefore, the Mesh Configuration Database should be stored in a secure, protected location and should be transferred between Mesh Managers using security best practices. If the Mesh Configuration Database is transferred or stored in an insecure way, then the associated mesh network is not secure.



4 Mesh Configuration Database JSON schema

This JSON schema is used to describe the format of a Mesh Object.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "https://www.bluetooth.com/specifications/specs/mesh-cdb-1-0-1-schema.json#",
  "version": "1.0.1",
  "title": "Mesh Configuration Database",
  "description": "A representation of the Mesh network to allow provisioning and/or
  configuration roles to be distributed between devices. An excerpt from the Mesh Configuration
  Database Profile Bluetooth specification is hosted here for the convenience of members of
  Bluetooth SIG. Use of this excerpt by members of Bluetooth SIG is governed by the membership
  and other related agreements between Bluetooth SIG and its members, including those
  agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Use of this
  excerpt by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement
  of the intellectual property rights of Bluetooth SIG and its members. Please refer back to
  the Mesh Configuration Database Profile (https://www.bluetooth.com/specifications/specs/mesh-configuration-database-profile-1-0-1/) Bluetooth specification for additional information.",
  "type": "object",
  "definitions": {
    "unicastAddress": {
      "type": "string",
      "name": "hexadecimal unicast address",
      "pattern": "^(([1-7][0-9a-fA-F]{3})|([0-7][1-9a-fA-F][0-9a-fA-F]{2})|([0-7][0-9a-fA-F][1-9a-fA-F][0-9a-fA-F])|([0-7][0-9a-fA-F][0-9a-fA-F][1-9a-fA-F]))$"
    },
    "timestamp": {
      "type": "string",
      "format": "date-time"
    },
    "credentials": {
      "type": "integer",
      "minimum": 0, "maximum": 1
    },
    "ttl": {
      "type": "integer",
      "minimum": 0, "maximum": 127
    },
    "ttlDefault": {
      "type": "integer",
      "minimum": 255, "maximum": 255
    },
    "featureState": {
      "type": "integer",
      "enum": [ 0, 1, 2 ]
    },
    "publish": {
```




```

    "type": "object",
    "properties": {
      "address": {
        "$ref": "#/definitions/anyAddress"
      },
      "index": {
        "$ref": "#/definitions/keyIndex"
      },
      "ttl": {
        "oneOf": [
          {
            "$ref": "#/definitions/ttl"
          },
          {
            "$ref": "#/definitions/ttlDefault"
          }
        ]
      },
      "period": {
        "type": "object",
        "properties": {
          "numberOfSteps": {
            "type": "integer",
            "minimum": 0, "maximum": 63
          },
          "resolution": {
            "type": "integer",
            "enum": [ 100, 1000, 10000, 600000 ]
          }
        }
      },
      "retransmit": {
        "type": "object",
        "properties": {
          "count": {
            "type": "integer",
            "minimum": 0, "maximum": 7
          },
          "interval": {
            "type": "integer",
            "enum": [ 50, 100, 150, 200, 250, 300, 350, 400, 450, 500,
550, 600, 650, 700, 750, 800, 850, 900, 950, 1000, 1050, 1100, 1150, 1200, 1250, 1300, 1350,
1400, 1450, 1500, 1550, 1600 ]
          }
        }
      },
      "required": [
        "count",
        "interval"
      ]
    }
  }

```

```

        ],
        "additionalProperties": false
    },
    "credentials": {
        "$ref": "#/definitions/credentials"
    }
},
"required": [
    "address",
    "index",
    "ttl",
    "period",
    "retransmit",
    "credentials"
],
"additionalProperties": false
},
"subscribe": {
    "type": "array",
    "items": {
        "oneOf": [
            {
                "$ref": "#/definitions/groupAddress"
            },
            {
                "$ref": "#/definitions/specialGroupAddress"
            },
            {
                "$ref": "#/definitions/UUID"
            }
        ]
    },
    "minItems": 0
},
"bind": {
    "type": "array",
    "items": {
        "$ref": "#/definitions/keyIndex"
    },
    "minItems": 0
},
"modelId": {
    "oneOf": [
        {
            "type": "string",
            "pattern": "^[0-9a-fA-F]{4}$"
        },
        {

```

```

        "type": "string",
        "pattern": "^([0-9a-fA-F]{8})$"
    }
]
},
"identifier": {
    "type": "string",
    "pattern": "^([0-9a-fA-F]{4})$"
},
"group": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
        "address": {
            "$ref": "#/definitions/groupAddressOrLabelUUID"
        },
        "parentAddress": {
            "$ref": "#/definitions/parentAddress"
        }
    },
    "required": [
        "name",
        "address",
        "parentAddress"
    ],
    "additionalProperties": false
},
"UUID": {
    "type": "string",
    "name": "hexadecimal UUID",
    "pattern": "^([0-9a-fA-F]{32})$"
},
"standardUUID": {
    "type": "string",
    "name": "UUID",
    "pattern": "^([0-9a-fA-F]{8})-([0-9a-fA-F]{4})-([0-9a-fA-F]{4})-([0-9a-fA-F]{4})-([0-9a-fA-F]{12})$"
},
"anyAddress": {
    "oneOf": [
        {
            "$ref": "#/definitions/unicastAddress"
        },
        {
            "$ref": "#/definitions/groupAddress"
        }
    ],

```

```

        {
            "$ref": "#/definitions/specialGroupAddress"
        },
        {
            "$ref": "#/definitions/allNodesAddress"
        },
        {
            "$ref": "#/definitions/UUID"
        }
    ]
},
"keyIndex": {
    "type": "integer",
    "minimum": 0, "maximum": 4095
},
"elementIndex": {
    "type": "integer",
    "minimum": 0, "maximum": 255
},
"appKey": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
        "index": {
            "$ref": "#/definitions/keyIndex"
        },
        "boundNetKey": {
            "$ref": "#/definitions/keyIndex"
        },
        "key": {
            "$ref": "#/definitions/key"
        },
        "oldKey": {
            "$ref": "#/definitions/key"
        }
    }
},
"required": [
    "name",
    "index",
    "key",
    "boundNetKey"
],
"additionalProperties": false
},
"nodeKey": {
    "type": "object",

```

```
"properties": {
  "index": {
    "$ref": "#/definitions/keyIndex"
  },
  "updated": {"type": "boolean"}
},
"required": [
  "index",
  "updated"
],
"additionalProperties": false
},
"element": {
  "type": "object",
  "properties": {
    "index": {
      "$ref": "#/definitions/elementIndex"
    },
    "location": {
      "$ref": "#/definitions/identifier"
    },
    "name": {
      "type": "string"
    },
    "models": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/model"
      },
      "minItems": 1
    }
  },
  "required": [
    "index",
    "location",
    "models"
  ],
  "additionalProperties": false
},
"model": {
  "type": "object",
  "properties": {
    "modelId": {
      "$ref": "#/definitions/modelId"
    },
    "subscribe": {
      "$ref": "#/definitions/subscribe"
    }
  },
```

```

        "publish": {
            "$ref": "#/definitions/publish"
        },
        "bind": {
            "$ref": "#/definitions/bind"
        }
    },
    "required": [
        "modelId",
        "subscribe",
        "bind"
    ],
    "additionalProperties": false
},
"groupAddress": {
    "type": "string",
    "name": "hexadecimal group address",
    "pattern": "^[c-fC-F][0-9a-eA-E][0-9a-fA-F]{2}|[c-eC-E][fF][0-9a-fA-F]{2})$"
},
"specialGroupAddress": {
    "type": "string",
    "name": "hexadecimal fixed group address excluding all nodes address",
    "pattern": "^[fF]{2}[0-9a-eA-E][0-9a-fA-F]|[fF]{3}[0-9a-eA-E])$"
},
"allNodesAddress": {
    "type": "string",
    "name": "hexadecimal all nodes address",
    "pattern": "^[fF]{4}$"
},
"groupAddressOrLabelUUID": {
    "oneOf": [
        {
            "$ref": "#/definitions/groupAddress"
        },
        {
            "$ref": "#/definitions/UUID"
        }
    ]
},
"unassignedAddress": {
    "type": "string",
    "enum": ["0000"]
},
"parentAddress": {
    "anyOf": [
        {
            "$ref": "#/definitions/groupAddressOrLabelUUID"
        }
    ],

```

```

        {
            "$ref": "#/definitions/unassignedAddress"
        }
    ]
},
"key": {
    "type": "string",
    "name": "key",
    "pattern": "^[0-9a-fA-F]{32}$"
},
"keyRefreshPhase": {
    "type": "integer",
    "minimum": 0, "maximum": 2
},
"securityLevel": {
    "type": "string",
    "enum": ["insecure", "secure"]
},
"netKey": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
        "index": {
            "$ref": "#/definitions/keyIndex"
        },
        "key": {
            "$ref": "#/definitions/key"
        },
        "oldKey": {
            "$ref": "#/definitions/key"
        },
        "minSecurity": {
            "ref": "#definitions/securityLevel"
        },
        "phase": {
            "$ref": "#/definitions/keyRefreshPhase"
        },
        "timestamp": {
            "$ref": "#/definitions/timestamp"
        }
    }
},
"required": [
    "name",
    "index",
    "phase",
    "key",

```

```

        "minSecurity"
    ],
    "additionalProperties": false
},
"networkRetransmit": {
    "type": "object",
    "properties": {
        "count": {
            "type": "integer",
            "minimum": 0, "maximum": 7
        },
        "interval": {
            "type": "integer",

"enum": [
    10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190,
    200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310
    ]
        }
    },
    "required": [
        "count",
        "interval"
    ],
    "additionalProperties": false
},
"heartbeatPeriod": {
    "type": "integer",
    "enum": [
        0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384,
        32768, 65536
    ]
},
"heartbeatPublication": {
    "type": "object",
    "properties": {
        "address": {
            "oneOf": [
                {
                    "$ref": "#/definitions/unicastAddress"
                },
                {
                    "$ref": "#/definitions/groupAddress"
                },
                {
                    "$ref": "#/definitions/specialGroupAddress"
                },
                {

```



```

        "$ref": "#/definitions/allNodesAddress"
    }
]
},
"period": {
    "$ref": "#/definitions/heartbeatPeriod"
},
"ttl": {
    "$ref": "#/definitions/ttl"
},
"index": {
    "$ref": "#/definitions/keyIndex"
},
"features": {
    "type": "array",
    "items": {
        "type": "string",
        "enum": ["relay",
            "proxy",
            "friend",
            "lowPower"]
    },
    "minItems": 0
}
},
"required": [
    "address",
    "period",
    "ttl",
    "index",
    "features"
],
"additionalProperties": false
},
"heartbeatSubscription": {
    "type": "object",
    "properties": {
        "source" : {
            "$ref": "#/definitions/unicastAddress"
        },
        "destination" : {
            "oneOf": [
                {
                    "$ref": "#/definitions/unicastAddress"
                },
                {
                    "$ref": "#/definitions/groupAddress"
                }
            ]
        }
    }
}

```

```

        {
            "$ref": "#/definitions/specialGroupAddress"
        },
        {
            "$ref": "#/definitions/allNodesAddress"
        }
    ]
}
},
"required": [
    "source",
    "destination"
],
"additionalProperties": false
},
"node": {
    "type": "object",
    "properties": {
        "UUID": {
            "$ref": "#/definitions/standardUUID"
        },
        "name": {
            "type": "string"
        },
        "unicastAddress": {
            "$ref": "#/definitions/unicastAddress"
        },
        "security": {
            "$ref": "#/definitions/securityLevel"
        },
        "deviceKey": {
            "$ref": "#/definitions/key"
        },
        "cid": {
            "$ref": "#/definitions/identifier"
        },
        "vid": {
            "$ref": "#/definitions/identifier"
        },
        "pid": {
            "$ref": "#/definitions/identifier"
        },
        "crpl": {
            "$ref": "#/definitions/identifier"
        },
        "features": {
            "type": "object",
            "properties": {

```

```

    "relay": {
      "$ref": "#/definitions/featureState"
    },
    "proxy": {
      "$ref": "#/definitions/featureState"
    },
    "friend": {
      "$ref": "#/definitions/featureState"
    },
    "lowPower": {
      "$ref": "#/definitions/featureState"
    }
  },
  "anyOf": [
    {"required": [ "relay" ]},
    {"required": [ "proxy" ]},
    {"required": [ "friend" ]},
    {"required": [ "lowPower" ]}
  ],
  "additionalProperties": false
},
"elements": {
  "type": "array",
  "items": {
    "$ref": "#/definitions/element"
  },
  "minItems": 1
},
"configComplete": {"type": "boolean"},
"netKeys": {
  "type": "array",
  "items": {
    "$ref": "#/definitions/nodeKey"
  },
  "minItems": 1
},
"appKeys": {
  "type": "array",
  "items": {
    "$ref": "#/definitions/nodeKey"
  },
  "minItems": 0
},
"secureNetworkBeacon": {
  "type": "boolean"
},
"defaultTTL": {
  "$ref": "#/definitions/ttl"
}

```

```

    },
    "networkTransmit": {
        "$ref": "#/definitions/networkRetransmit"
    },
    "relayRetransmit": {
        "$ref": "#/definitions/networkRetransmit"
    },
    "excluded": {"type": "boolean"},
    "heartbeatPub": {
        "$ref": "#/definitions/heartbeatPublication"
    },
    "heartbeatSub": {
        "$ref": "#/definitions/heartbeatSubscription"
    }
},
"required": [
    "UUID",
    "unicastAddress",
    "security",
    "netKeys",
    "appKeys",
    "elements",
    "configComplete",
    "excluded"
],
"additionalProperties": false
},
"provisioner": {
    "type": "object",
    "properties": {
        "provisionerName": {
            "type": "string"
        },
        "UUID": {
            "$ref": "#/definitions/standardUUID"
        }
    },
    "allocatedGroupRange": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "highAddress": {
                    "$ref": "#/definitions/groupAddress"
                },
                "lowAddress": {
                    "$ref": "#/definitions/groupAddress"
                }
            }
        }
    }
}

```

```
    },
    "required": [
        "lowAddress",
        "highAddress"
    ],
    "additionalProperties": false
},
"minItems": 1
},
"allocatedUnicastRange": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "highAddress": {
                "$ref": "#/definitions/unicastAddress"
            },
            "lowAddress": {
                "$ref": "#/definitions/unicastAddress"
            }
        }
    },
    "required": [
        "lowAddress",
        "highAddress"
    ],
    "additionalProperties": false
},
"minItems": 1
},
"allocatedSceneRange": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "firstScene": {
                "$ref": "#/definitions/identifier"
            },
            "lastScene": {
                "$ref": "#/definitions/identifier"
            }
        }
    },
    "required": [
        "firstScene",
        "lastScene"
    ],
    "additionalProperties": false
},
"minItems": 0
```

```

    }
  },
  "required": [
    "provisionerName",
    "UUID",
    "allocatedUnicastRange",
    "allocatedGroupRange",
    "allocatedSceneRange"
  ],
  "additionalProperties": false
},
"scene": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "addresses": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/unicastAddress"
      },
      "minItems": 0
    },
    "number": {
      "$ref": "#/definitions/identifier"
    }
  },
  "required": [
    "name",
    "addresses",
    "number"
  ],
  "additionalProperties": false
},
"exclusionList": {
  "type": "object",
  "properties": {
    "ivIndex": {
      "type": "integer"
    },
    "addresses": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/unicastAddress"
      },
      "minItems": 1
    }
  }
}

```

```
    },
    "required": [
        "ivIndex",
        "addresses"
    ],
    "additionalProperties": false
  }
},
"properties": {
  "$schema": {
    "type": "string"
  },
  "version": {
    "type": "string"
  },
  "meshName": {
    "type": "string"
  },
  "meshUUID": {
    "$ref": "#/definitions/standardUUID"
  },
  "timestamp": {
    "$ref": "#/definitions/timestamp"
  },
  "netKeys": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/netKey"
    },
    "minItems": 1
  },
  "appKeys": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/appKey"
    },
    "minItems": 0
  },
  "provisioners": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/provisioner"
    },
    "minItems": 1
  },
  "nodes": {
    "type": "array",
    "items": {
```

```

        "$ref": "#/definitions/node"
    },
    "minItems": 0
},
"groups": {
    "type": "array",
    "items": {
        "$ref": "#/definitions/group"
    },
    "minItems": 0
},
"scenes": {
    "type": "array",
    "items": {
        "$ref": "#/definitions/scene"
    },
    "minItems": 0
},
"networkExclusions": {
    "type": "array",
    "items": {
        "$ref": "#/definitions/exclusionList"
    },
    "minItems": 0
},
"partial": {
    "type": "boolean"
}
},
"required": [
    "$schema",
    "id",
    "version",
    "meshName",
    "meshUUID",
    "timestamp",
    "provisioners",
    "netKeys",
    "appKeys",
    "nodes",
    "groups",
    "scenes",
    "partial"
],
"additionalProperties": false
}

```


5 Example Mesh Configuration Database

The following is an example of a Mesh Configuration Database.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "https://www.bluetooth.com/specifications/specs/mesh-cdb-1-0-1-schema.json#",
  "version": "1.0.1",
  "meshUUID": "72C6BE40-444D-2081-BEAA-DDAD4E3CC21C",
  "meshName": "Brian and Mary's House",
  "timestamp": "2018-12-23T11:45:22-08:00",
  "netKeys": [
    {
      "name": "Home Network",
      "index": 0,
      "key": "5543950C57EDCC38E442964065720A8B",
      "phase": 0,
      "minSecurity": "secure",
      "timestamp": "2018-11-20T10:05:20-08:00"
    },
    {
      "name": "Molly's Network",
      "index": 1,
      "key": "6D4B5D861F6C7304C039FD846231E84D",
      "oldKey": "610BB4B4BEC1A5630D6712DE06B95B20",
      "phase": 1,
      "minSecurity": "insecure",
      "timestamp": "2018-10-10T06:22:34-08:00"
    }
  ],
  "appKeys": [
    {
      "name": "Home Automation",
      "index": 0,
      "boundNetKey": 0,
      "key": "3FA985DA6D4CA22DA05C7E7A1F11C783"
    },
    {
      "name": "Door Bell",
      "index": 2,
      "boundNetKey": 0,
      "key": "20283753615B66E63D34AF4A4A4E7336"
    },
    {
      "name": "Cat Flap",
      "index": 3,
      "boundNetKey": 1,
      "key": "958CED9C76F1F23A60746F9384CF1E5B",

```

```

    "oldKey": "C337ACF959314D9D06C8419158F04617"
  },
  {
    "name": "Home Automation Setup",
    "index": 1200,
    "boundNetKey": 0,
    "key": "A327BB410CF137455A4F234DF8134C78"
  }
],
"provisioners": [
  {
    "provisionerName": "Brian's Phone",
    "UUID": "70CF7C97-32A3-45B6-9149-4810D2E9CBF4",
    "allocatedGroupRange": [
      {"highAddress": "C1FF", "lowAddress": "C100"}
    ],
    "allocatedUnicastRange": [
      {"lowAddress": "0001", "highAddress": "00FF"},
      {"lowAddress": "0301", "highAddress": "03FF"}
    ],
    "allocatedSceneRange": []
  },
  {
    "provisionerName": "Mary's Phone",
    "UUID": "577C2832-B345-A694-4A48-10D2C9DFE456",
    "allocatedGroupRange": [
      {"highAddress": "C0FF", "lowAddress": "C000"},
      {"highAddress": "C2FF", "lowAddress": "C200"}
    ],
    "allocatedUnicastRange": [
      {"highAddress": "01FF", "lowAddress": "0100"}
    ],
    "allocatedSceneRange": [
      {"firstScene": "0001", "lastScene": "0040"}
    ]
  }
],
"nodes": [
  {
    "UUID": "559B8C97-32A3-45B6-9149-4865D288CBF3",
    "name": "Bedroom Light Switch",
    "deviceKey": "0FE48D03E41D26E5C5EA327A55E8A218",
    "unicastAddress": "0002",
    "security": "secure",
    "cid": "0003",
    "pid": "0003",
    "vid": "0003",
    "crpl": "0003",

```



```

"features": {
  "relay": 2,
  "proxy": 2,
  "friend": 2,
  "lowPower": 1
},
"elements": [
  {
    "index": 0,

    "location": "0001",
    "models": [
      {
        "modelId": "0000",
        "bind": [],
        "subscribe": []
      },
      {
        "modelId": "0002",
        "bind": [],
        "subscribe": []
      },
      {
        "modelId": "0004",
        "publish": {
          "address": "C023",
          "index": 0,
          "ttl": 7,
          "period": {
            "numberOfSteps": 9,
            "resolution": 10000
          },
          "retransmit": {
            "count": 2,
            "interval": 250
          },
          "credentials" : 1
        },
        "bind": [ 0 ],
        "subscribe": []
      }
    ]
  },
  {
    "index": 1,
    "location": "0002",
    "models": [
      {

```

```

        "modelId": "0004",
        "publish": {
            "address": "C024",
            "index": 0,
            "ttl": 7,
            "period": {
                "numberOfSteps": 9,
                "resolution": 10000
            },
            "retransmit": {
                "count": 2,
                "interval": 100
            },
            "credentials" : 1
        },
        "bind": [ 0 ],
        "subscribe": []
    }
]
}
],
"configComplete": true,
"netKeys": [ { "index": 0, "updated": false } ],
"appKeys": [ { "index": 0, "updated": false } ],
"networkTransmit": {
    "count": 2,
    "interval": 20
},
"defaultTTL": 3,
"excluded": false
},
{
    "UUID": "347F8A97-32D3-45E7-9149-486AA2887983",
    "name": "Bedroom Light",
    "unicastAddress": "0004",
    "cid": "0011",
    "pid": "0004",
    "vid": "0001",
    "crpl": "0020",
    "features": {
        "relay": 1,
        "proxy": 2,
        "friend": 1,
        "lowPower": 2
    },
    "elements": [
        {
            "index": 0,

```

```
"location": "0106",
"models": [
  {
    "modelId": "0000",
    "bind": [],
    "subscribe": []
  },
  {
    "modelId": "0002",
    "bind": [],
    "subscribe": []
  },
  {
    "modelId": "1203",
    "subscribe": [ "C023" ],
    "bind": [ 0 ]
  },
  {
    "modelId": "1204",
    "subscribe": [ "C023" ],
    "bind": [ 0 ]
  },
  {
    "modelId": "1206",
    "subscribe": [ "C023" ],
    "bind": [ 0 ]
  },
  {
    "modelId": "1207",
    "subscribe": [ "C023" ],
    "bind": [ 0 ]
  },
  {
    "modelId": "1204",
    "subscribe": [ "C023" ],
    "bind": [ 0, 1200 ]
  },
  {
    "modelId": "1206",
    "subscribe": [ "C023" ],
    "bind": [ 0 ]
  },
  {
    "modelId": "1207",
    "subscribe": [ "C023" ],
    "bind": [ 0, 1200 ]
  },
  {
```



```

        "modelId": "1000",
        "subscribe": [ "C023" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1004",
        "subscribe": [ "C023" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1006",
        "subscribe": [ "C023" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1007",
        "subscribe": [ "C023" ],
        "bind": [ 0, 1200 ]
    },
    {
        "modelId": "1002",
        "subscribe": [ "C023" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1301",
        "subscribe": [ "C023" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1302",
        "subscribe": [ "C023" ],
        "bind": [ 0, 1200 ]
    },
    {
        "modelId": "1307",
        "subscribe": [ "C023" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1308",
        "subscribe": [ "C023" ],
        "bind": [ 0, 1200 ]
    }
]
},
{
    "index": 1,

```

```

        "location": "0106",
        "models": [
            { "modelId": "1002",
              "subscribe": [],
              "bind": []
            },
            { "modelId": "130a",
              "subscribe": [],
              "bind": []
            }
        ]
    },
    {
        "index": 2,
        "location": "0106",
        "models": [
            { "modelId": "1002",
              "subscribe": [],
              "bind": []
            },
            { "modelId": "130b",
              "subscribe": [],
              "bind": []
            }
        ]
    }
],
"deviceKey": "89DDEFD69B45AEC29DFD545E901CEE64",
"security": "secure",
"configComplete": true,
"netKeys": [ { "index": 0, "updated": false } ],
"appKeys": [ { "index": 0, "updated": false },
              { "index": 1200, "updated": false } ],
"relayRetransmit": {
    "count": 3,
    "interval": 60
},
"defaultTTL": 4,
"excluded": false
},
{
    "UUID": "345E8227-3DD3-45CC-4149-4865D2334545",
    "name": "Bedside Light (Brian)",
    "deviceKey": "AAA4B4CA3AD5C6D6BE1A8D455EC40BE0",
    "unicastAddress": "0007",
    "security": "secure",
    "cid": "000A",
    "pid": "1204",

```

```
"vid": "0041",
"crpl": "0020",
"features": {
  "relay": 1,
  "proxy": 2,
  "friend": 0,
  "lowPower": 2
},
"configComplete": true,
"netKeys": [ { "index": 0, "updated": false } ],
"appKeys": [ { "index": 0, "updated": false },
              { "index": 1200, "updated": false } ],
"elements": [
  {
    "index": 0,
    "location": "0001",
    "models": [
      {
        "modelId": "0000",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "0002",
        "bind": [],
        "subscribe": []
      },
      {
        "modelId": "1203",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
      },
      {
        "modelId": "1204",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0, 1200 ]
      },
      {
        "modelId": "1206",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
      },
      {
        "modelId": "1207",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0, 1200 ]
      },
      {
```



```

        "modelId": "1000",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1004",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1006",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1007",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0, 1200 ]
    },
    {
        "modelId": "1002",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1301",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0 ]
    },
    {
        "modelId": "1302",
        "subscribe": [ "C023", "C024" ],
        "bind": [ 0, 1200 ]
    }
    ]
}
],
"excluded": false
},
{
    "UUID": "268E8CA7-32A3-45B6-9149-AA44BAAD2346",
    "name": "Bedside Light (Mary)",
    "deviceKey": "B619F5BA671AF2548BA219760294063B",
    "unicastAddress": "0105",
    "security": "secure",
    "cid": "000A",
    "pid": "1204",
    "vid": "0041",

```

```
"crpl": "0020",
"features": {
  "relay": 0,
  "proxy": 2,
  "friend": 1,
  "lowPower": 2
},
"secureNetworkBeacon": true,
"configComplete": true,
"netKeys": [ { "index": 0, "updated": false } ],
"appKeys": [ { "index": 0, "updated": false },
              { "index": 1200, "updated": false } ],
"elements": [
  {
    "index": 0,
    "location": "0001",
    "models": [
      {
        "modelId": "0000",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "0002",
        "bind": [],
        "subscribe": []
      },
      {
        "modelId": "0005",
        "subscribe": [ "C023", "C025" ],
        "bind": [ 0 ]
      },
      {
        "modelId": "1204",
        "subscribe": [ "C023", "C025" ],
        "bind": [ 0, 1200 ]
      },
      {
        "modelId": "1206",
        "subscribe": [ "C023", "C025" ],
        "bind": [ 0 ]
      },
      {
        "modelId": "1207",
        "subscribe": [ "C023", "C025" ],
        "bind": [ 0, 1200 ]
      },
      {
```

```

        "modelId": "1000",
        "subscribe": ["C023", "C025"],
        "bind": [ 0 ]
    },
    {
        "modelId": "1004",
        "subscribe": ["C023", "C025"],
        "bind": [ 0 ]
    },
    {
        "modelId": "1006",
        "subscribe": ["C023", "C025"],
        "bind": [ 0 ]
    },
    {
        "modelId": "1007",
        "subscribe": ["C023", "C025"],
        "bind": [ 0, 1200 ]
    },
    {
        "modelId": "1002",
        "subscribe": ["C023", "C025"],
        "bind": [ 0 ]
    },
    {
        "modelId": "1301",
        "subscribe": ["C023", "C025"],
        "bind": [ 0 ]
    },
    {
        "modelId": "1302",
        "subscribe": ["C023", "C025"],
        "bind": [ 0, 1200 ]
    }
    ]
}
],
"excluded": false
},
{
    "UUID": "CBFA8C97-32E3-45B6-9149-4822D28856FF",
    "name": "Molly's Access Tag",
    "deviceKey": "C7F91F91AB79F445925BC9B568CA53A9",
    "unicastAddress": "0109",
    "security": "insecure",
    "cid": "003F",
    "pid": "0001",
    "vid": "0001",

```

```

"crpl": "0008",
"features": {
  "relay": 2,
  "proxy": 2,
  "friend": 2,
  "lowPower": 2
},
"configComplete": true,
"netKeys": [ { "index": 1, "updated": true } ],
"appKeys": [ { "index": 3, "updated": false } ],
"elements": [
  {
    "index": 0,
    "location": "0100",
    "models": [
      {
        "modelId": "0000",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "0002",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "003F002A",
        "publish": {
          "address": "EA57D9F1975D4EBDA78A37D189AC58F4",
          "index": 3,
          "ttl": 0,
          "period": {
            "numberOfSteps": 2,
            "resolution": 600000
          },
          "retransmit": {
            "count": 2,
            "interval": 150
          },
          "credentials": 0
        },
        "subscribe": [],
        "bind": [ 3 ]
      }
    ]
  }
],
"excluded": false

```

```

},
{
  "UUID": "45708C97-32A3-45B5-E149-486AB247C4EE",
  "name": "Cat Flap",
  "deviceKey": "CA1C7F26BB557BD609777575CEF8E599",
  "unicastAddress": "010A",
  "security": "insecure",
  "cid": "000C",
  "pid": "0002",
  "vid": "0000",
  "crpl": "0008",
  "features": {
    "relay": 2,
    "proxy": 2,
    "friend": 2,
    "lowPower": 2
  },
  "secureNetworkBeacon": false,
  "configComplete": true,
  "netKeys": [ { "index": 1, "updated": true } ],
  "appKeys": [ { "index": 3, "updated": true } ],
  "elements": [
    {
      "index": 0,
      "location": "0000",
      "models": [
        {
          "modelId": "0000",
          "subscribe": [],
          "bind": []
        },
        {
          "modelId": "0002",
          "subscribe": [],
          "bind": []
        },
        {
          "modelId": "0015",
          "subscribe": [ "EA57D9F1975D4EBDA78A37D189AC58F4" ],
          "bind": [ 3 ]
        }
      ]
    }
  ],
  "excluded": false
},
{
  "UUID": "27668C97-32A3-45B5-6149-4865E288CBC2",

```

```
"name": "Door Bell",
"deviceKey": "FC846BF0D85AC4D9CEF63AA04E5507CE",
"unicastAddress": "0103",
"security": "secure",
"cid": "0007",
"pid": "0019",
"vid": "0000",
"crpl": "0010",
"features": {
  "relay": 2,
  "proxy": 2,
  "friend": 2,
  "lowPower": 1
},
"configComplete": true,
"netKeys": [ { "index": 0, "updated": false } ],
"appKeys": [ { "index": 2, "updated": false } ],
"defaultTTL": 9,
"elements": [
  {
    "index": 0,
    "location": "0100",
    "models": [
      {
        "modelId": "0000",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "0002",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "1001",
        "publish": {
          "address": "C103",
          "index": 2,
          "ttl": 9,
          "period": {
            "numberOfSteps": 1,
            "resolution": 1000
          },
          "retransmit": {
            "count": 3,
            "interval": 150
          },
          "credentials" : 0
        }
      }
    ]
  }
]
```

```

        },
        "subscribe": [],
        "bind": [ 2 ]
    }
]
}
],
"excluded": false
},
{
    "UUID": "EAA38997-3B43-45B6-9149-4865D2885555",
    "name": "Bell",
    "deviceKey": "31743AFD08DBACECBBA1297BDD0AAEFE",
    "unicastAddress": "0104",
    "security": "secure",
    "cid": "0007",
    "pid": "001A",
    "vid": "0003",
    "crpl": "0100",
    "features": {
        "relay": 0,
        "proxy": 1,
        "friend": 0,
        "lowPower": 2
    },
    "configComplete": true,
    "netKeys": [ { "index": 0, "updated": false } ],
    "appKeys": [ { "index": 2, "updated": false } ],
    "defaultTTL": 9,
    "elements": [
        {
            "index": 0,
            "location": "010C",
            "models": [
                {
                    "modelId": "0000",
                    "subscribe": [],
                    "bind": []
                },
                {
                    "modelId": "0002",
                    "subscribe": [],
                    "bind": []
                },
                {
                    "modelId": "1000",
                    "subscribe": [ "C103" ],
                    "bind": [ 2 ]
                }
            ]
        }
    ]
}

```

```

        }
    ]
}
],
"excluded": false
},
{
    "UUID": "70CF7C97-32A3-45B6-9149-4810D2E9CBF4",
    "name": "Brian's phone",
    "cid": "0008",
    "pid": "032B",
    "vid": "0005",
    "crpl": "0100",
    "features": {
        "relay": 0,
        "proxy": 0,
        "friend": 0,
        "lowPower": 2
    },
    "deviceKey": "27653BFE0EEEA5ECBBA68975DD0A0244",
    "unicastAddress": "0001",
    "security": "secure",
    "configComplete": true,
    "netKeys": [
        { "index": 0, "updated": false },
        { "index": 1, "updated": true }
    ],
    "appKeys": [
        { "index": 0, "updated": false },
        { "index": 2, "updated": false },
        { "index": 3, "updated": true },
        { "index": 1200, "updated": false }
    ],
    "elements": [
        {
            "index": 0,
            "location": "0000",
            "models": [
                {
                    "modelId": "0000",
                    "subscribe": [],
                    "bind": []
                },
                {
                    "modelId": "0002",
                    "subscribe": [],
                    "bind": []
                }
            ],
        }
    ],
}

```



```

        {
            "modelId": "0001",
            "subscribe": [],
            "bind": []
        }
    ]
}
],
"excluded": false
},
{
    "UUID": "577C2832-B345-A694-4A48-10D2C9DFE456",
    "name": "Mary's phone",
    "cid": "0001",
    "pid": "011a",
    "vid": "0004",
    "crpl": "0100",
    "features": {
        "relay": 0,
        "proxy": 0,
        "friend": 0,
        "lowPower": 2
    },
    "deviceKey": "CCA28B400D49ABECBBA33975F7665628",
    "unicastAddress": "01FF",
    "security": "secure",
    "configComplete": true,
    "netKeys": [
        { "index": 0, "updated": false },
        { "index": 1, "updated": true }
    ],
    "appKeys": [
        { "index": 0, "updated": false },
        { "index": 2, "updated": false },
        { "index": 3, "updated": true },
        { "index": 1200, "updated": false }
    ],
    "elements": [
        {
            "index": 0,
            "location": "0000",
            "models": [
                {
                    "modelId": "0000",
                    "subscribe": [],
                    "bind": []
                },
                {

```

```

        "modelId": "0002",
        "subscribe": [],
        "bind": []
      },
      {
        "modelId": "0001",
        "subscribe": [],
        "bind": []
      }
    ]
  },
  "excluded": false
}
],
"groups": [
  {
    "name": "Bedroom",
    "address": "C023",
    "parentAddress": "0000"
  },
  {
    "name": "Bedside (Brian)",
    "address": "C024",
    "parentAddress": "C023"
  },
  {
    "name": "Bedside (Mary)",
    "address": "C025",
    "parentAddress": "C023"
  },
  {
    "name": "Doorbell",
    "address": "C103",
    "parentAddress": "0000"
  },
  {
    "name": "Cat Flap",
    "address": "EA57D9F1975D4EBDA78A37D189AC58F4",
    "parentAddress": "0000"
  }
],
"scenes": [
  {
    "name": "Go to sleep",
    "addresses": ["0004"],
    "number": "0007"
  },

```



```
{
  {
    "name": "Wakeup",
    "addresses": ["0004"],
    "number": "0002"
  }
},
"partial": false
}
```

6 Acronyms and abbreviations

Acronym or Abbreviation	Meaning
CID	Company Identifier
JSON	JavaScript Object Notation
PID	Product Identifier
RPL	Replay Protection List
TTL	Time To Live
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
VID	Version Identifier

Table 6.1: Acronyms and abbreviations



7 References

- [1] Mesh Profile Specification, Version 1.0 or later
- [2] Internet Engineering Task Force (IETF), RFC 8259, "The JavaScript Object Notation (JSON) Data Interchange Format", December 2017, <https://tools.ietf.org/html/rfc8259>
- [3] JSON Schema, Specification Links Draft 4, <http://json-schema.org/specification-links#draft-4>
- [4] Bluetooth SIG Assigned Numbers, <http://www.bluetooth.com/specifications/assigned-numbers>
- [5] Mesh Model Specification, Version 1.0 or later
- [6] IETF, RFC 3339, "Date and Time on the Internet: Timestamps", July 2002, <https://tools.ietf.org/html/rfc3339>
- [7] IETF, RFC 4122, "A Universally Unique IDentifier (UUID) URN Namespace", July 2005, <https://www.ietf.org/rfc/rfc4122.txt>
- [8] Appropriate Language Mapping Table, <https://www.bluetooth.com/language-mapping/Appropriate-Language-Mapping-Table>



Appendix A: Mesh Configuration Database management

This Appendix provides guidelines for creating and managing objects in the Mesh Configuration Database when using the Mesh Profile specification [1]. This database can be created and updated by any device that provisions and/or configures the devices in the mesh network. Examples of such devices are: Provisioner, Configuration Manager, and Mesh Manager.

A.1 Database creation

The Mesh Configuration Database is created by the Provisioner that initializes the mesh network. The Provisioner creates the database and updates the Mesh object (see Section 2.1) as follows:

- The provisioners array contains at least one provisioner object with the information about the Provisioner that created the mesh network.
- The netKeys array contains at least one netKey object. One netKey object exists per mesh network key that is in use in the mesh network.
- The appKeys array contains zero or more appKey objects. One appKey object exists per mesh application key that is in use in the mesh network.
- The nodes array contains zero or more node objects. One node object exists for each node provisioned on the mesh network.
- The deviceKey property of one or more node objects can be omitted if the value of the partial property is set to “true”.
- If the Provisioner is capable of configuring and/or controlling the network (i.e., the Provisioner can send mesh messages and act as a mesh node), then the nodes array contains a node object with the same UUID as the UUID in the provisioner object.
- The groups array contains zero or more group objects, one for each group address that has been assigned in the mesh network.
- The scenes array contains zero or more scene objects, one for each scene number that has been assigned in the mesh network.

A.2 Provisioning complete

When the Provisioner successfully completes provisioning of a new node, a new node object (see Section 2.1.4) is created in the nodes array of the Mesh Configuration Database. The Provisioner creates the new node object as follows:

- The UUID property is set to the Device UUID of the new node.
- The unicastAddress and deviceKey properties are set to the corresponding values generated during the provisioning process.



- The security property is set according to the level of security for the subnet on which the node has been provisioned.
- The netKeys array contains a single node network key object that identifies the network key used during provisioning.
- The configComplete property is set to “false”.
- The appKeys and elements arrays are empty.
- The excluded property is set to “false”.
- Element objects are created in the elements array, with one object for each element.

A.3 Processing a Config Composition Data Status message

When the Mesh Manager processes a Config Composition Data Status message from a node, it updates the corresponding node object (see Section 2.1.4) as follows:

- The cid, vid, pid, and crpl properties are created and initialized with the values received in the message.
- If the features property is not already present for this node object, the features object is created. If the composition data indicates that the Relay, Proxy, Friend, or Low Power feature is not supported, the corresponding property is created in the features object with the value set to 2 (Unsupported). If the composition data indicates that the Low Power feature is supported, the lowPower property is created with the value set to 1 (Supported). The fact that the features object does not contain a relay, proxy, or friend property means that the composition data indicates support for the corresponding feature, but the feature has not yet been configured, i.e., enabled or disabled, and its state is unknown.
- For each element object, the location property is initialized and the models array is created and populated with model objects that each contain an initialized modelId property. Empty arrays are created for the subscribe and bind properties of the models' objects.

A.4 Processing miscellaneous configuration messages

While configuring a node, the Mesh Manager sends various messages to update or read the node's networking parameters. When a response from a node is received, the source address of the message identifies the node object that needs to be updated in the Mesh Configuration Database. The values from the received message are used to modify the object. As a consequence, the Mesh Manager updates the node object (see Section 2.1.4) as follows:

- When a Config Relay Status message, Config Friend Status message, or Config GATT Proxy Status message is received, the features object of the node object is created if it does not exist and is updated with the configured value indicated by the status messages.
- When a Config Beacon Status message is received, the secureNetworkBeacon property is created in the node object and is set to the configured value indicated by the status message.



- When a Config Default TTL Status message is received, the defaultTTL property is created in the node object and is set to the configured value indicated by the status message.
- When a Config Network Transmit Status message is received, the networkTransmit property is created in the node object, and the count property and the interval property of the network transmit object are set to the configured values indicated by the status message.
- When a Config Relay Status message is received, the relayRetransmit property is created in the node object, and the count property and the interval property of the relay retransmit object are set to the configured values as indicated by the status message.
- When a Config NetKey Status message is received in response to a NetKey Add message with the Status Code set to Success, the node network key object is added to the node's netKeys array.
- When a Config NetKey Status message is received in response to a NetKey Delete message with the Status Code set to Success, the corresponding node network key object is removed from the netKeys array. When the network key object is removed, the node application key objects for the application keys that are bound to this network key are removed from the appKeys array. For all the model objects in the models object array, if the bind array contains entries with the values of the AppKey indexes of the removed application keys, these entries are removed, and if the publish object contains the index property with the value of the AppKey index corresponding to any of the removed application keys, the publish object is removed.
- When a Config AppKey Status message is received in response to an AppKey Add message with the Status Code set to Success, the node application key object is added to the appKeys array.
- When a Config AppKey Status message is received in response to an AppKey Delete message with the Status Code set to Success, the corresponding node application key object is removed from the appKeys array. For all the model objects in the models object array, if the bind array contains an entry with the value of the AppKey index of this application key, the entry is removed, and if the publish object contains the index property with the value of the AppKey index of this application key, the publish object is removed.
- When a Config Model Subscription Status message is received with the Status Code message set to Success in response to a Config Model Subscription Add message, Config Model Subscription Virtual Address Add message, Config Model Subscription Delete message, Config Model Subscription Virtual Address Delete message, Config Model Subscription Overwrite message, or Config Model Subscription Virtual Address Overwrite message, the entry is added, removed, or updated in the subscribe property of the model object corresponding to the model instance indicated by the status message.
- When a Config Model Publication Status message is received with the Status Code message set to Success in response to a Config Model Publication Set message or Config Model Publication Virtual Address Set message, the publish object is added or removed from the model object corresponding to the model instance indicated by the status message.

- When a Config Model App Status message is received with the Status Code message set to Success in response to a Config Model App Bind message, the entry is added to the bind property of the model object corresponding to the model instance indicated by the status message.
- When a Config Model App Status message is received with the Status Code message set to Success in response to a Config Model App Unbind message, the entry is removed from the corresponding bind object.
- After the Mesh Manager completes sending and receiving messages during the initial configuration process, the configComplete property is set to “true”.

A.5 Key refresh

When the Key Refresh procedure is in progress, the Mesh Manager updates the Mesh object (see Section 2.1) as follows:

- If the network key is being updated as part of the Key Refresh procedure, the key property of the network key object is set to the value of the new network key, and the oldKey property of the network key object is set to the value of the network key that was used before the key refresh. The phase property of the network key object is set to the current phase of the Key Refresh procedure.
- If the application key is being updated as part of the Key Refresh procedure, the key property of the application key object is set to the value of the new application key, and the oldKey property of the application key object is set to the value of the application key that was used before the key refresh.
- When a Config NetKey Status message is received in response to a Config NetKey Update message with the Status Code set to Success, the updated property of the node network key object is set to “true”. Upon the successful completion of the Key Refresh procedure, the updated property is set back to “false”.
- When a Config AppKey Status message is received in response to a Config AppKey Update message with the Status Code set to Success, the updated property of the node application key object is set to “true”. Upon the successful completion of the Key Refresh procedure, the updated property is set back to “false”.

A.6 Node removal and IV update procedure

When node needs to be removed from the network, the Mesh Manager updates the Mesh object (see Section 2.1) as follows:

- To remove a node from the mesh network, the Mesh Manager first marks the node as excluded, by setting the excluded property of the node object to “true” and by creating or updating the exclusionList object with the corresponding IV Index in the networkExclusions property and adding the node's unicast addresses to the array of addresses.
- After this, the Mesh Manager performs the Key Refresh procedure [1] to update all the keys known to the excluded node.



- When the Key Refresh procedure is successfully completed, the Mesh Manager removes the node object corresponding to the excluded node from the nodes property of the Mesh object.

When the IV Update procedure [1] is executing in the mesh network, the Mesh Manager updates the Mesh object (see Section 2.1) as follows:

- After the IV update procedure, when the network transitions to an IV Normal Operation state with a higher IV index, the Mesh Manager removes the exclusionList object containing the ivIndex property value that is lower than the current IV index of the network, by a count of two (or more), from the networkExclusions property array.

A.7 Scene maintenance

When the Mesh Manager creates, updates, or deletes scenes in the mesh network, the Mesh Manager maintains the information about the scenes stored on each particular node. As a consequence, the Mesh Manager adds the information about the scenes to the Mesh object (see Section 2.1) as follows:

- The Mesh Manager may create a scene object with an empty addresses array and later add addresses of the nodes on which the scenes are stored.
- When a Scene Register Status message is received with the Status Code indicating “Success” in response to a Scene Store message, Scene Delete message, or Scene Register Get message, the source address of the message is added or removed from the addresses array of the corresponding scenes.
- When a node is removed from the mesh network, all its unicast addresses are removed from the addresses array of all scenes the node was added to.