Bluetooth® mesh SDK 6.1.1.0 GA
Gecko SDK Suite 4.4
May 2, 2024

Bluetooth mesh is a new topology available for Bluetooth Low Energy (LE) devices that enables many-to-many (m:m) communication. It’s optimized for creating large-scale device networks, and is ideally suited for building automation, sensor networks, and asset tracking. Our software and SDK for Bluetooth development supports Bluetooth Mesh and Bluetooth 5.3 functionality. Developers can add mesh networking communication to LE devices such as connected lights, home automation, and asset tracking systems. The software also supports Bluetooth beaconing, beacon scanning, and GATT connections so Bluetooth mesh can connect to smart phones, tablets, and other Bluetooth LE devices.

This release includes features supported by the Bluetooth mesh specification version 1.1.

These release notes cover SDK versions:

6.1.1.0 released May 2, 2024
6.1.0.0 released April 10, 2024
6.0.1.0 released February 14, 2024
6.0.0.0 released December 13, 2023

Compatibility and Use Notices

For more information about security updates and notices, see the Security chapter of the Gecko Platform Release notes installed with this SDK or on the Silicon Labs Release Notes page. Silicon Labs also strongly recommends that you subscribe to Security Advisories for up-to-date information. For instructions, or if you are new to the Silicon Labs Bluetooth mesh SDK, see Using This Release.

Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 9.40.1
- Using wine to build with the iarBuild.exe command line utility or IAR Embedded Workbench GUI on macOS or Linux could result in incorrect files being used due to collisions in wine’s hashing algorithm for generating short file names.
- Customers on macOS or Linux are advised not to build with IAR outside of Simplicity Studio. Customers who do should carefully verify that the correct files are being used.

GCC (The GNU Compiler Collection) version 12.2.1, provided with Simplicity Studio.
- Link-time optimization feature of GCC has been disabled, resulting in a slight increase of image size.
## Contents

1. **New Items** .......................................................................................................................................................................................... 3  
   1.1 **New Features** ............................................................................................................................................................................. 3  
   1.2 **New APIs** .................................................................................................................................................................................... 4  
2. **Improvements** ..................................................................................................................................................................................... 7  
3. **Fixed Issues** ....................................................................................................................................................................................... 8  
4. **Known Issues in the Current Release** ................................................................................................................................................ 9  
5. **Deprecated Items** ............................................................................................................................................................................. 10  
6. **Removed Items** ................................................................................................................................................................................ 11  
7. **Using This Release** .......................................................................................................................................................................... 12  
   7.1 **Installation and Use** ................................................................................................................................................................. 12  
   7.2 **Security Information** ................................................................................................................................................................. 12  
   7.3 **Support** ..................................................................................................................................................................................... 13
1 New Items

This release of the Gecko SDK (GSDK) will be the last with combined support for all EFM and EFR devices, except for patches to this version as needed. Starting in mid-2024 we will introduce separate SDKs:

- The existing Gecko SDK will continue with support for Series 0 and 1 devices.
- A new SDK will cater specifically to Series 2 and 3 devices.

The Gecko SDK will continue to support all Series 0 and 1 devices with no change to the long-term support, maintenance, quality, and responsiveness provided under our software policy.

The new SDK will branch from Gecko SDK and begin to offer new features that help developers take advantage of the advanced capabilities of our Series 2 and 3 products.

This decision aligns with customer feedback, reflecting our commitment to elevate quality, ensure stability, and enhance performance for an exceptional user experience across our software SDKs.

1.1 New Features

Added in release 6.0.1.0

Changes in SLC components:

A third BT Mesh Role was added beside Provisioner and Provisionee role – a Custom BT Mesh Role, where the application gets the freedom to implement a custom role. For example, the Provisioner or Provisionee role could be selected runtime.

Added in release 6.0.0.0

New Networked Lighting Control (NLC) example apps:

- `btmesh_soc_nlc_basic_lightness_controller` for demonstration of the BT Mesh NLC Basic Lightness Controller Profile
- `btmesh_soc_nlc_basic_scene_selector` for demonstration of the BT Mesh NLC Basic Scene Selector Profile
- `btmesh_soc_nlc_dimming_control` for demonstration of the BT Mesh NLC Dimming Controller Profile
- `btmesh_soc_nlc_sensor_ambient_light` for demonstration of the BT Mesh NLC Ambient Light Sensor Profile
- `btmesh_soc_nlc_sensor_occupancy` for demonstration of the BT Mesh NLC Occupancy Sensor Profile (People count)

Changes in example apps:

- `btmesh_soc_sensor_server` was deleted and its functionality was split in 3 examples:
  - `btmesh_soc_sensor_thermometer` for demonstration of Sensor Server Model with a thermometer
  - `btmesh_soc_nlc_sensor_occupancy` for demonstration of the BT Mesh NLC Occupancy Sensor Profile (people count)
  - `btmesh_soc_nlc_sensor_ambient_light` for demonstration of the BT Mesh NLC Ambient Light Sensor Profile
- `btmesh_soc_switch` was renamed to `btmesh_soc_switch_ctl`, whose purpose is to demonstrate the usage of the Light CTL Client Model. The example no longer controls scenes (Scene Client)
- `btmesh_soc_light` was renamed to `btmesh_soc_light_ctl`
  The example no longer demonstrates the LC Server model and Scene Server, Scheduler Server and Time Server Models
- `btmesh_soc_hsl` was renamed to `btmesh_soc_light_hsl`
  The example no longer demonstrates the LC Server model and Scene Server, Scheduler Server and Time Server Models
Changes in all example apps:

DFU image updates are generated by a Python script instead of create_bl_files.bat/sh files

Support for Mesh Composition Data Pages 1, 2, 128, 129, 130 was added for all examples, these pages are automatically generated by the BT Mesh Configurator tool.

New SLC components:

- **btmesh_nlc_basic_lightness_controller** for demonstration of the BT Mesh NLC Basic Lightness Controller Profile
- **btmesh_nlc_basic_lightness_controller_profile_metadata** for Composition Data Page 2 NLC support for Basic Lightness Controller Profile
- **btmesh_nlc_basic_scene_selector** for demonstration of the BT Mesh NLC Basic Scene Selector Profile
- **btmesh_nlc_basic_scene_selector_profile_metadata** for Composition Data Page 2 NLC support for Basic Scene Selector Profile
- **btmesh_nlc_dimming_control** for demonstration of the BT Mesh NLC Dimming Controller Profile
- **btmesh_nlc_dimming_control_profile_metadata** for Composition Data Page 2 NLC support for Dimming Controller Profile
- **btmesh_nlc_ambient_light_sensor** for demonstration of the BT Mesh NLC Ambient Light Sensor Profile
- **btmesh_nlc_ambient_light_sensor_profile_metadata** for Composition Data Page 2 NLC support for Ambient Light Sensor Profile
- **btmesh_nlc_occupancy_sensor** for demonstration of the BT Mesh NLC Occupancy Sensor Profile (People count)
- **btmesh_nlc_occupancy_sensor_profile_metadata** for Composition Data Page 2 NLC support for Occupancy Sensor Profile
- **btmesh_generic_level_client_ext** for extending the Generic Base component with Generic Move Unacknowledged and Generic Delta Unacknowledged messages
- **ncp_btmesh_ae_server** for enabling the Silabs Configuration Server vendor model for the node to allow data transfer over Advertisement Extension
- **ncp_btmesh_ae_server** for enabling the Silabs Configuration Client vendor model for the node.
- **ncp_btmesh_user_cmd** for demonstrating the communication between an NCP host and NCP target using BGAPI user messages, responses and events.

1.2 New APIs

Added in release 6.1.0.0

BGAPI additions:

New commands have been added to the node class to associate scan response data with Mesh provisioning and Mesh proxy service advertisements. The scan response data associated with Mesh proxy service advertisements can be set individually for each network key, so it can contain data encrypted with that key, but it is up to the application to manage that. The new commands are:

- **sl_btmesh_node_set_proxy_service_scan_response**: Set scan response data for proxy service advertisement
- **sl_btmesh_node_clear_proxy_service_scan_response**: Clear scan response data for proxy service advertisement
- **sl_btmesh_node_set_provisioning_service_scan_response**: Set scan response data for provisioning service advertisement
- **sl_btmesh_node_clear_provisioning_service_scan_response**: Clear scan response data for provisioning service advertisement

A new command has been added to the vendor model class to set model behavior options. Currently there is one option that controls whether a work buffer is allocated from the heap for each vendor model for message reception reporting. The default value (1) allocates
a buffer, which increases event reporting resilience when a device is under high load at the expense of additional heap memory usage.

The new command is:

- `sl_btmesh_vendor_model_set_option`: Set a vendor model behavior option

New commands have been added to the diagnostic class to report friendship-related events. The new commands are:

- `sl_btmesh_diagnostic_enable_friend`: Enable generation of friendship-related diagnostic events
- `sl_btmesh_diagnostic_disable_friend`: Disable generation of friendship-related diagnostic events
- `sl_btmesh_diagnostic_get_friend`: Retrieve friendship-related diagnostic counters

New events added to the diagnostic class are as follows:

- `sl_btmesh_diagnostic_friend_queue`: Event for message being added to friendship message queue
- `sl_btmesh_diagnostic_friend_relay`: Event for message being relayed to a LPN
- `sl_btmesh_diagnostic_friend_remove`: Event for message being removed from friendship message queue

**Added in release 6.0.0.0**

Changes in SLC components:

**ncp_btmesh_dfu** component's `ncp_btmesh_dfu.h` has a new API

```c
void sl_btmesh_ncp_dfu_handle_cmd(void *data, bool *cmd_handled);
```

**btmesh_provisioning_decorator** component doesn't restart provisioning after provisioning is failed

**btmesh_lighting_server**'s `sl_btmesh_lighting_server.h` has a new API

```c
void sl_btmesh_update_lightness(uint16_t lightness, uint32_t remaining_ms);
```

**btmesh_event_log** has more granular configurability options

**btmesh_ctl_client**'s `sl_btmesh_ctl_client.h` has an API change

Instead of

```c
void sl_btmesh_set_temperature(uint8_t new_color_temperature_percentage);
```

the new API is

```c
void sl_btmesh_ctl_client_set_temperature(uint8_t temperature_percent);
```

```c
void sl_btmesh_ctl_client_set_lightness(uint8_t lightness_percent);
```

**BGAPI additions:**

A new BGAPI class for device diagnostics has been added. It provides the application with Mesh stack statistics counters and an event-based reporting of network PDU relaying and proxying, which can be activated and deactivated as needed.

The BGAPI commands in the diagnostic class are:

- `sl_btmesh_diagnostic_init`: Initialize the diagnostic component
- `sl_btmesh_diagnostic_deinit`: Deinitialize the diagnostic component
- `sl_btmesh_diagnostic_enable_relay`: Enable event-based reporting of network PDU relaying/proxying activity
- `sl_btmesh_diagnostic_disable_relay`: Disable event-based reporting of network PDU relaying/proxying activity
- `sl_btmesh_diagnostic_get_relay`: Get the number of relayed/proxied network PDUs so far
- `sl_btmesh_diagnostic_get_statistics`: Get mesh stack statistics counters
- `sl_btmesh_diagnostic_clear_statistics`: Zero mesh stack statistics counters
The BGAPI event in the diagnostic class is:

- **sl_btmesh_diagnostic_relay**: Event reporting that a network PDU has been relayed or proxied by the stack
2 Improvements

**Changed in release 6.1.0.0**

The diagnostic class BGAPI command for retrieving statistics has been changed to retrieve chunks of data instead of retrieving all data at once. The caller should supply the size of the chunk it requests along with the offset of the chunk in the statistics data, and the call will return with as much data as can be supplied, given the request constraints.

**Changed in release 6.0.0.0**

A provisioner or a node can now configure itself using the configuration client model and its own primary address as the destination for the messages. This can replace self-configuration by test BGAPI commands.

Code optimization can result in slightly smaller firmware images than before, depending on the feature set used.

Code optimization can result in slightly smaller RAM usage than before, depending on the feature set used.

The Mesh stack no longer requires or supports the deprecated BLE advertiser and scanner components. Instead, it uses the current versions of each (legacy advertiser and legacy scanner for non-extended advertisements, and extended advertiser and extended scanner for extended advertisements). Applications that use both the BLE and the Mesh BGAPIs should no longer use the deprecated BLE advertiser and scanner components either.
3 Fixed Issues

### Fixed in release 6.1.0.0

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1235337</td>
<td>Made GATT service discovery more robust on an overloaded device.</td>
</tr>
<tr>
<td>1247422</td>
<td>Made vendor model reception more robust on an overloaded device.</td>
</tr>
<tr>
<td>1251498</td>
<td>Fixed when Lighting message, including transition time, leads to an incorrect error message in logs.</td>
</tr>
<tr>
<td>1252252</td>
<td>Fixed when a Generic Move message leads to a dim up, which could overflow to a dim down.</td>
</tr>
<tr>
<td>1254356</td>
<td>Fixed a regression with friend subsystem deinitialization.</td>
</tr>
<tr>
<td>1276121</td>
<td>Fixed application key index truncation at BGAPI level when embedded provisioner invokes a key refresh procedure.</td>
</tr>
</tbody>
</table>

### Fixed in release 6.0.0.1

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1226000</td>
<td>Extended Provisioner BGAPI function for checking node identity to also check private node identity.</td>
</tr>
<tr>
<td>1206620</td>
<td>Fixed problems caused by missing BGAPI events during high load to correct firmware verification problems.</td>
</tr>
<tr>
<td>1230833</td>
<td>Fixed friend subsystem deinitialization so that reinitialization works without resetting the device.</td>
</tr>
<tr>
<td>1243565</td>
<td>Fixed crash that could occur if provisioner initialization failed, for example because of malformed DCD.</td>
</tr>
<tr>
<td>1244298</td>
<td>Fixed reporting of spurious extra octets in the Register Status event of the Scene Client model.</td>
</tr>
<tr>
<td>1243556</td>
<td>Automatic node initialization removed for BT Mesh application components. Now all components can be used also in Provisioner role.</td>
</tr>
</tbody>
</table>

### Fixed in release 6.0.0.0

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>360955</td>
<td>The interval between first and second attention timer event could be other than one second.</td>
</tr>
<tr>
<td>1198887</td>
<td>Private beacon random advertiser address is the same for all subnets while it should be different.</td>
</tr>
<tr>
<td>1202073</td>
<td>Btmesh_ncp_empty example does not have enough RAM on BRD4182 with GCC compiler.</td>
</tr>
<tr>
<td>1202088</td>
<td>Btmesh_soc_switch example does not have enough RAM on BRD4311 and BRD4312 with IAR compiler.</td>
</tr>
<tr>
<td>1206714</td>
<td>Proxy server should emit a beacon over proxy connection when a subnet is added to the proxy server.</td>
</tr>
<tr>
<td>1206715, 1211012, 1211022</td>
<td>Support for device composition data page 2, 129 and 130 should be present in configuration server model as well as large composition data server model when remote provisioning is supported</td>
</tr>
<tr>
<td>1211017</td>
<td>Periodic publishing of location information should alternate between global and local location when both are known</td>
</tr>
<tr>
<td>1212373</td>
<td>Resource leak in proxy connection handling after several hundred proxy connections have been opened and closed</td>
</tr>
<tr>
<td>1212854</td>
<td>Pull mode MBT transfer to an LPN does not complete successfully</td>
</tr>
<tr>
<td>1197398, 1194443</td>
<td>DFU distributor application is currently not able to handle more than 60 nodes successfully</td>
</tr>
<tr>
<td>1202088</td>
<td>Btmesh_soc_switch_ctl example compiles on all boards with IAR compiler.</td>
</tr>
</tbody>
</table>
4 Known Issues in the Current Release

Issues in bold were added since the previous release.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>401550</td>
<td>No BGAPI event for segmented message handling failure.</td>
<td>Application needs to deduce failure from timeout / lack of application layer response; for vendor models an API has been provided.</td>
</tr>
<tr>
<td>454059</td>
<td>A large number of key refresh state change events are generated at the end of KR process, and that may flood NCP queue.</td>
<td>Increase NCP queue length in the project.</td>
</tr>
<tr>
<td>454061</td>
<td>Slight performance degradation compared to 1.5 in round-trip latency tests was observed.</td>
<td></td>
</tr>
<tr>
<td>624514</td>
<td>Issue with re-establishing connectable advertising if all connections have been active and GATT proxy is in use.</td>
<td>Allocate one more connection than is needed.</td>
</tr>
<tr>
<td>841360</td>
<td>Poor performance of segmented message transmission over GATT bearer.</td>
<td>Ensure that the underlying BLE connection’s Connection interval is short; ensure that ATT MTU is large enough to fit a full Mesh PDU; tune the minimum connection event length to allow multiple LL packets to be transmitted per connection event.</td>
</tr>
<tr>
<td>1121605</td>
<td>Rounding errors may cause scheduled events to trigger at very slightly different times than expected.</td>
<td></td>
</tr>
<tr>
<td>1226127</td>
<td>Host provisioner example can be stuck when it starts to provision a second node.</td>
<td>Restart the host provisioner app before provisioning the second node.</td>
</tr>
<tr>
<td>1204017</td>
<td>Distributor is not able to handle parallel self FW Update and FW Upload.</td>
<td>Don’t run self FW update and FW upload in parallel.</td>
</tr>
</tbody>
</table>
5 Deprecated Items

Deprecated in release 6.0.0.0

The BGAPI command `sl_btmesh_node_get_networks()` has been deprecated. Use `sl_btmesh_node_key_key_count()` and `sl_btmesh_node_get_key()` instead.

The BGAPI commands `sl_btmesh_test_set_segment_send_delay()` and `sl_btmesh_test_set_sar_config()` have been deprecated. Use `sl_btmesh_sar_config_set_sar_transmitter()` and `sl_btmesh_sar_config_server_set_sar_receiver()` instead.
6 Removed Items

Removed in release 6.0.0.0

The BGAPI commands `sl_btmesh_test_set_local_config()` and `sl_btmesh_test_get_local_config()` have been removed.

The BGAPI commands `sl_btmesh_node_get_statistics()` and `sl_btmesh_node_clear_statistics()` have been removed.
7 Using This Release

This release contains the following

- Silicon Labs Bluetooth mesh stack library
- Bluetooth mesh sample applications

If you are a first time user, see QSG176: Silicon Labs Bluetooth Mesh SDK v2.x Quick-Start Guide.

7.1 Installation and Use

The Bluetooth mesh SDK is provided as part of the Gecko SDK (GSDK), the suite of Silicon Labs SDKs. To quickly get started with the GSDK, install Simplicity Studio 5, which will set up your development environment and walk you through GSDK installation. Simplicity Studio 5 includes everything needed for IoT product development with Silicon Labs devices, including a resource and project launcher, software configuration tools, full IDE with GNU toolchain, and analysis tools. Installation instructions are provided in the online Simplicity Studio 5 User’s Guide.

Alternatively, Gecko SDK may be installed manually by downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/gecko_sdk for more information.

The GSDK default install location has changed with Simplicity Studio 5.3 and higher.

- Windows: C:\Users\<NAME>\SimplicityStudio\SDKs\gecko_sdk
- MacOS: /Users/<NAME>/SimplicityStudio/SDKs/gecko_sdk

Documentation specific to the SDK version is installed with the SDK. Additional information can often be found in the knowledge base articles (KBAs). API references and other information about this and earlier releases is available on https://docs.silabs.com/.

7.2 Security Information

Secure Vault Integration

This version of the stack is integrated with Secure Vault Key Management. When deployed to Secure Vault High devices, mesh encryption keys are protected using the Secure Vault Key Management functionality. The table below shows the protected keys and their storage protection characteristics.

<table>
<thead>
<tr>
<th>Key</th>
<th>Exportability on a node</th>
<th>Exportability on Provisioner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network key</td>
<td>Exportable</td>
<td>Exportable</td>
<td>Derivations of the network key exist only in RAM while network keys are stored on flash</td>
</tr>
<tr>
<td>Application key</td>
<td>Non-exportable</td>
<td>Exportable</td>
<td>In Provisioner’s case, applied to Provisioner’s own device key as well as other devices’ keys</td>
</tr>
<tr>
<td>Device key</td>
<td>Non-exportable</td>
<td>Exportable</td>
<td></td>
</tr>
</tbody>
</table>

Keys that are marked as “Non-Exportable” can be used but cannot be viewed or shared at runtime.

Keys that are marked as “Exportable” can be used or shared at runtime but remain encrypted while stored in flash.

For more information on Secure Vault Key Management functionality, see AN1271: Secure Key Storage.
Security Advisories

To subscribe to Security Advisories, log in to the Silicon Labs customer portal, then select Account Home. Click HOME to go to the portal home page and then click the Manage Notifications tile. Make sure that ‘Software/Security Advisory Notices & Product Change Notices (PCNs)’ is checked, and that you are subscribed at minimum for your platform and protocol. Click Save to save any changes.

7.3 Support

Development Kit customers are eligible for training and technical support. Use the Silicon Labs Bluetooth mesh web page to obtain information about all Silicon Labs Bluetooth products and services, and to sign up for product support.

Contact Silicon Laboratories support at http://www.silabs.com/support.
Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!