Silicon Labs is a leading vendor in Bluetooth hardware and software technologies, used in products such as sports and fitness, consumer electronics, beacons, and smart home applications. The core SDK is an advanced Bluetooth 5.4-compliant stack that provides all of the core functionality along with multiple API to simplify development. The core functionality offers both standalone mode allowing a developer to create and run their application directly on the SoC, or in NCP mode allowing for the use of an external host MCU.

These release notes cover SDK version(s):

6.2.0.0 GA released October 9, 2023
6.1.0.0 GA released July 26, 2023
6.0.0.0 GA released June 7, 2023

Compatibility and Use Notices

For information about security updates and notices, see the Security chapter of the Gecko Platform Release notes installed with this SDK or on the TECH DOCS tab on https://www.silabs.com/developers/bluetooth-low-energy. Silicon Labs also strongly recommends that you subscribe to Security Advisories for up-to-date information. For instructions as well as notes on using Secure Vault features, or if you are new to the Silicon Labs Bluetooth SDK, see Using This Release.

Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 9.20.4.

- 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 10.3-2021.10, provided with Simplicity Studio.
1 New Items

1.1 New Features

Added in release 6.2.0.0

Bluetooth Connection Statistics

New feature component bluetooth_feature_connection_statistics provides the functionality to collect connection-specific statistics and report them to applications upon request. The statistics are also reported upon connection closing.

ESL AP

Multiconnection: Support is added to ESL AP script for opening and maintaining multiple parallel connections to Tags.

Encryption: Encrypted NCP communication is added and is made the default choice between ESL Core C library and ESL AP NCP.

Simple Communication Interface

Robust configuration option is added to the component to make the UART communication more resistant to garbage on the line at boot.

Added in release 6.1.0.0

User-Controlled TX Power of a Bluetooth Connection

New feature component bluetooth-feature-user-power-control is added to support changing the TX power of a Bluetooth connection. In RCP mode. Use the vendor command 0xfc30 to set the TX power of a connection. In NCP and SoC modes, use the sl_bt_connection_set_tx_power command. This feature is mutually exclusive with the standard Bluetooth LE Power Control feature (bluetooth-feature-power-control).

Testing Network Co-Processor (NCP) Communication

New host example is added for testing Network Co-Processor (NCP) communication between host and EFR32 target. It demonstrates the communication between an NCP host and NCP target using BGAPI user messages, responses and events. It can be used as a starting point for creating custom commands or for testing purposes.

Network Co-Processor (NCP) User Command

New component is added for handling BGAPI User Commands. It demonstrates the communication between an NCP host and NCP target using BGAPI user messages, responses and events. It can be used as a starting point for creating custom commands or for testing purposes.

ESL AP

• Core C library – Multiconnection: Support is added for opening and maintaining multiple parallel connections to Tags in the ESL Core C library. Note that multiconnection support is not yet available in ESL AP Python script!

• Application Python script – command line: ESL AP command line auto-complete and command history functionality is added for convenience.

ESL Tag

• e-paper driver: ESL Tag e-paper display driver is published (experimental quality).

ESL Tag and ESL AP NCP

ESL HW support is extended to xG13 chip family, tripled the number of supported radio boards. ESL Tag and ESL AP NCP pre-built binaries are provided in Simplicity Studio for convenience.

Added in release 6.0.0.0

Controller-Based Privacy

Controller-based privacy is now supported. The controller is able to perform address resolving for up to 8 devices. Include component ‘bluetooth_feature_resolving_list’ to use controller address resolving. The host stack continues to perform address resolving for resolvable private addresses that were not resolved by the controller, or to support the use case where the controller address resolving feature is not used.
Filter Accept List

The Filter Accept List is enhanced by the support of address resolving in the controller. Use component ‘bluetooth_feature_accept_list’ for the enhanced feature.

Connection Data Length Update

The stack supports data length update on Bluetooth connections. The component ‘bluetooth_feature_connection’ has a new configuration SL_BT_CONFIG_CONNECTION_DATA_LENGTH for setting default data length and run-time APIs are provided for using the feature.

Resource Usage Report

User applications can query the resource usage in the Bluetooth stack at run time with the ‘bluetooth_feature_resource_report’ component.

Transmission Queue Reports on Bluetooth Connections

The ‘bluetooth_feature_resource_report’ component supports tracking and reporting the number of packets and bytes queued for transmission on a specific connection.

Transmission Events for GATT Characteristic Notifications

The Bluetooth host stack now supports getting an event when the transmission of a GATT notification has completed on a connection. This functionality is provided by the ‘bluetooth_feature_resource_report’ component when the application specifically enables connection TX queue reporting using the commands in that class.

Accurate Bluetooth Address Types in BGAPI

It is possible now to select API semantics that use accurate Bluetooth address types with the ‘bluetooth_feature_use_accurate_api_address_types’ component. In the default Bluetooth stack configuration, most BGAPI commands and events use coarse address typing that only differentiates between public (value 0) and any random (value 1) address types. This component changes how the Bluetooth stack interprets and reports Bluetooth address types. When this component is included, Bluetooth commands and events that include an address type will systematically use the sl_bt_gap_address_type_t enumeration to indicate the accurate type.

Periodic Advertising with Responses (PAwR)

The Bluetooth stack now supports advertising with and synchronizing to Periodic Advertising with Responses (PAwR). The Bluetooth stack has new components ‘bluetooth_feature_pawrAdvertiser’ for advertising with PAwR, ‘bluetooth_feature_sync_scanner’ for scanning for synchronization, and ‘bluetooth_feature_periodic_sync’ and ‘bluetooth_feature_pawr_sync’ for managing the sync to periodic advertising trains of different types. The new BGAPI classes corresponding to these components supersede some functionality in the ‘bluetooth_feature_sync’ component. Deprecated functionality and the replacements are shown in the documentation of the affected BGAPI classes, commands, and events.

PAwR-Aware Connection Scheduling Algorithm

The component ‘bluetooth_feature_connection_pawr_scheduling’ provides a connection scheduling algorithm that has been designed for use especially with applications that use several concurrent connections while advertising using PAwR. The algorithm tries to distribute the connections such a way that they are distracted as little as possible by the PAwR train.

Support for Electronic Shelf Label (ESL) Service and Profile

The ‘ESL Tag Core’ software component implements the ESL tag role of the Electronic Shelf Label Profile and is fully compliant with the Bluetooth ESL specification.

The Access Point role is also implemented by the bt_host_esl_ap host sample app for NCP hosts, and is also fully compliant with the Bluetooth ESL specification.

Support for Object Transfer Service (OTS) and Profile

Object Transfer Service and Profile is implemented by the ‘Object Client’ and ‘Object Server’ software components in accordance with the Object Transfer Profile specification. This enables transferring binary large objects via a BLE connections.

Vendor HCI Command for Fetching Connection Parameters

Vendor HCI command 0xfc2a is added for fetching the current parameters of a Bluetooth connection.
1.2 New APIs

Added in release 6.2.0.0

**sl_bt_connection_read_statistics command**: Read the statistic values collected on a specified connection.

**sl_bt_evt_connection_statistics event**: Provide the connection statistic values when the application includes the component bluetooth_feature_connection_statistics.

Vendor HCI command 0xfc321 and event 0xef: Read the statistic values collected on a specified connection in the RCP mode.

**sl_bt_evt_system_stopped event**: Indicate that stopping the Bluetooth stack using the command sl_bt_system_stop_bluetooth has completed.

**sl_bt_system_forcibly_stop_bluetooth command**: Stop the Bluetooth stack immediately without performing the ACL. This is a termination procedure for connections that were open. This command is available only when the application includes the Bluetooth on-demand start component.

Added in release 6.1.0.0

**sl_bt_connection_set_tx_power command**: Set the transmit power of a connection.

**sl_bt_accept_list_add_device_by_bonding command**: Add a device to the Filter Accept List based on its bonding handle.

**sl_bt_accept_list_add_device_by_address command**: Add a device to the Filter Accept List based on its identity address.

**sl_bt_accept_list_remove_device_by_bonding command**: Remove a device from the Filter Accept List based on its bonding handle.

**sl_bt_accept_list_remove_device_by_address command**: Remove a device from the Filter Accept List based on its identity address.

**sl_bt_accept_list_remove_all_devices command**: Remove all devices from the Filter Accept List.

**sl_bt_resolving_list_add_device_by_bonding command**: Add a device to the Resolving List based on its bonding handle.

**sl_bt_resolving_list_add_device_by_address command**: Add a device to the Resolving List based on its identity address.

**sl_bt_resolving_list_remove_device_by_bonding command**: Remove a device from the Resolving List based on its bonding handle.

**sl_bt_resolving_list_remove_device_by_address command**: Remove a device from the Resolving List based on its identity address.

**sl_bt_resolving_list_remove_all_devices command**: Remove all devices from the Resolving List.

**sl_bt_connection_set_default_data_length command**: Set the default preferred maximum TX payload length for new connections.

**sl_bt_connection_set_data_length command**: Update the maximum TX payload length and maximum packet TX time of a Bluetooth connection.

**sl_bt_evt_connection_data_length event**: Report a change to the maximum payload length or TX time in either direction of a connection.

**sl_bt_connection_forcibly_close command**: Forcefully close a Bluetooth connection without performing the ACL Termination procedure.

**sl_bt_resource_get_status command**: Get the present memory buffer usage status.

**sl_bt_resource_set_report_threshold command**: Set low and high thresholds of memory buffer usage reports.

**sl_bt_resource_enable_connection_tx_report command**: Enable tracking and reporting data packet TX status of future new connections.

**sl_bt_resource_get_connection_tx_status command**: Get the data packet TX status of a connection.

**sl_bt_resource_disable_connection_tx_report command**: Disable tracking and reporting data packet TX status of future new connections.

**sl_bt_evt_resource_status event**: Indicates that the memory buffer usage has crossed a threshold.

**sl_bt_evt_gatt_server_notification_tx_completed event**: Indicates that one or more GATT notifications have been transmitted.
**sl_bt_scanner_set_parameters_and_filter command**: Set scan parameters and the scanning filter policy for subsequent scanning operations.

**sl_bt_pawrAdvertiser_start command**: Start PAwR advertising on an advertising set.

**sl_bt_pawrAdvertiser_set_subevent_data command**: Set data to be sent in the subevent of an active PAwR train.

**sl_bt_pawrAdvertiser_create_connection command**: Initiate a connection request to a device that is synchronized to the active PAwR train.

**sl_bt_pawrAdvertiser_stop command**: Stop PAwR advertising on an advertising set.

**sl_bt_evt_pawrAdvertiser_subevent_data_request event**: Triggered when the Bluetooth stack is ready to accept data for subevents of the PAwR train.

**sl_bt_evt_pawrAdvertiser_subevent_data_request event**: Triggered if subevent data was successfully set but the attempt to transmit the subevent data has failed.

**sl_bt_evt_pawrAdvertiser_response_report event**: Report the status and data of a used response slot of an active PAwR train.

**sl_bt_syncScanner_set_sync_parameters command**: Configure synchronization parameters for synchronizing to periodic advertising trains.

**sl_bt_syncScanner_open command**: Start establishing synchronization with the specified periodic advertiser in parallel with other advertisers given in previous invocations of this command.

**sl_bt_syncUpdateSync_parameters command**: Update synchronization parameters for a periodic sync that was already established.

**sl_bt_evt_periodicSync_opened event**: Triggered when synchronization is established to a periodic advertising train that does not have subevents or response slots.

**sl_bt_evt_periodicSync_transfer_received event**: Triggered when synchronization transfer is received for a periodic advertising train that does not have subevents or response slots.

**sl_bt_evt_periodicSync_report event**: Triggered when data for a periodic advertising train that does not have subevents or response slots is received and accepted by the reporting mode currently set to the train.

**sl_bt_evt_pawrSync_opened event**: Triggered after synchronization is established to a PAwR train.

**sl_bt_evt_pawrSync_transfer_received event**: Triggered after synchronization transfer is received for a PAwR train.

**sl_bt_pawrSync_set_sync_subevents command**: Specify the subevents that this device will synchronize to on the specified PAwR train.

**sl_bt_evt_pawrSync_subevent_report event**: Triggered when subevent data for PAwR train is received and accepted by the reporting mode currently set to the train.

**sl_bt_pawrSync_set_response_data command**: Set the data to be sent in the specified response slot of a subevent of an active PAwR train.

**sl_syncUpdateSync_parameters command**: Update synchronization parameters for a periodic sync that was already established.

**sl_gattDiscoverCharacteristicDescriptors command**: Discover all descriptors of a GATT characteristic in a remote GATT database.

New option in **sl_bt_sm_configure command**: Support the new option of rejecting a pairing when the remote device uses debug keys.
2 Improvements

2.1 Changed Items

**Changed in release 6.2.0.0**

**ESL AP**

Various stability and performance improvements are provided. The introduction of multiconnection feature allows parallelization of Tag configuration, and hence improves system performance significantly.

**Changed in release 6.1.0.0**

**Configurable IQ Sample Quality Analysis**

IQ sample quality analysis in the AoA host examples is now enabled only in case of debug log level.

**ESL AP**

Various stability improvements are provided.

**ESL AP Command Line Interface – Multiconnection**

Changes ESL AP CLI to enable future support for multiple simultaneous connections to Tags.

**OTS Image Transfer**

Stability improvements are provided.

**ESL Core Library - Multiconnection**

Adds support for opening and maintaining multiple parallel connections to Tags in the ESL Core C library.

**Changed in release 6.0.0.0**

**Amazon FreeRTOS BLE HAL Component**

The Amazon FreeRTOS BLE HAL component `iot_ble` has been updated to use new BGAPI classes for scanning and advertising. If a BLE HAL application uses scanning, the application must add `blueooth_feature_legacy_scanner` to the application project. If a BLE HAL application uses advertising, the application must add `blueooth_feature_legacy_advertiser` to the application project.

**Prevent BGAPI Command Calls From Interrupt Handler Mode**

The stack now checks if a BGAPI command is called in interrupt handler mode, and returns error SL_STATUS_IRQ if so.

2.2 Changed APIs

**Changed in release 6.1.0.0**

**sl_bt_evt_l2cap_command_rejected event**: New parameter `cid` is appended to the parameter list for specifying the local channel identifier.

**sl_simple_timer_start** renamed to **app_timer_start**.

**sl_simple_timer_stop** renamed to **app_timer_stop**.

**esl_image_chunk_received** callback: New parameter `offset` is added.

**Changed in release 6.0.0.0**

**sl_bt_evt_connection_parameters event**: This is no longer used for communicating the TX data length change on the connection. Data length update comes from new event `sl_bt_evt_connection_data_length`.
**sl_bt_evt_l2cap_le_channel_open_request event**: New parameter ‘remote_cid’ is appended to the event parameter list for specifying the corresponding channel identifiers of the peer device.

**sl_bt_evt_l2cap_le_channel_open_response event**: New parameter ‘remote_cid’ is appended to the event parameter list for specifying the corresponding channel identifiers of the peer device.

**sl_bt_sm_configure command**: Option "Allow connections only from bonded devices" is not supported when the application includes the 'bluetooth_feature_external_bonding_database' feature.

**Connection handle returned from sl_bt_connection_open command and sl_bt_evt_connection_opened event**: The stack now uses a handle value assignment schematic such that a newly assigned handle contains an additional sequence number. This change is to solve possible handle collisions in race conditions when connections are opened and closed very frequently.

### 2.3 Intended Behavior

When configuring the HFXO frequency in SL_DEVICE_INIT_HFXO_FREQ, make sure to set the SLDEVICE_INIT_DPLL_FREQ to twice the HFXO frequency in sl_device_init_dpll_config.h for proper function of Bluetooth.
3 Fixed Issues

Fixed in release 6.2.0.0

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1145112</td>
<td>The even-connection-scheduling feature now distributes connections evenly over common connection interval with any number of connections (up to 32 connections). In addition, the PAwR-aware connection scheduling is fixed at the same time.</td>
</tr>
<tr>
<td>1164422</td>
<td>Validation of the AUX_SYNC_SUBEVENT_RSP PDU has been improved to catch invalid data fields.</td>
</tr>
<tr>
<td>1170496</td>
<td>Now the RX data length of the connection remains unchanged when setting the TX data length with command sl_bt_connection_set_data_length(). The Bluetooth stack sets the default RX data length of a connection to maximum. The RX data length can be configured by the peer device during the feature exchange and connection procedures.</td>
</tr>
<tr>
<td>1174893</td>
<td>Fix an issue that was affecting the Bluetooth RCP application, where the response to HCI LE Periodic Advertising Parameters v2 command was missing the advertising handle.</td>
</tr>
<tr>
<td>1179646</td>
<td>Improved tolerance of ESL Tag example to air transmission interference.</td>
</tr>
<tr>
<td>1190534</td>
<td>Fix an issue that would cause connection creation over Periodic Advertising with Responses (PAwR) to fail if the synchronized device had a static random identity address but the application did not include Bluetooth scanner component.</td>
</tr>
<tr>
<td>1190989</td>
<td>Fix an issue where ESL Tag becomes inaccessible after timeout in Unsynchronized state.</td>
</tr>
<tr>
<td>1191503</td>
<td>Add Get-Coex-Parameters vendor-specific command to Bluetooth HCI Coex plugin.</td>
</tr>
</tbody>
</table>

Fixed in release 6.1.0.0

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1063952</td>
<td>Fix an issue in the Bluetooth link layer that may cause a multiprotocol application using the BLE advertisements to crash after running for a long time.</td>
</tr>
<tr>
<td>1114654</td>
<td>Update the NCP user commands of the bt_ncp example and add new host example bt_host_ncp_test that can be used together to test and verify the communication between the host and target devices.</td>
</tr>
<tr>
<td>1131599</td>
<td>The config file template now provides an example of how to disable angle filtering, required for the AoA positioning example, which must use unfiltered angle reports from the locators.</td>
</tr>
<tr>
<td>1131945</td>
<td>Fix an issue in the AppLoader that the TX power is too low on EFR32[M</td>
</tr>
<tr>
<td>1144105</td>
<td>Fix an issue that the first connection packet was not received by the peripheral if the peripheral was advertising using LE Coded PHY with S2 coding.</td>
</tr>
<tr>
<td>1145112</td>
<td>Even connection scheduling is fixed to distribute connections evenly over common connection interval with any number of connections (up to 32 connections).</td>
</tr>
<tr>
<td>1151844</td>
<td>Fix various issues in ESL Access Point CLI and logging.</td>
</tr>
<tr>
<td>1152274</td>
<td>Fix a dependency in the link-layer that causes unnecessary extended advertising functions compiled into the application even the application only uses the legacy advertising component. This issue results in increased code size.</td>
</tr>
<tr>
<td>1154896</td>
<td>Provide extended board support for Bluetooth NCP ESL AP and Bluetooth SoC ESL Tag examples.</td>
</tr>
<tr>
<td>1157633</td>
<td>Fix an issue in the GATT server that the client characteristic notification status may be out of sync if the remote GATT client device has lost the bonding with the local device. In this fix, when the remote GATT client device has lost bonding keys and the local device has enabled bonding confirmations to allow re-bonding, the GATT server will erase client-supported features and client characteristic notification settings of the GATT client if a re-bonding happens on an unencrypted connection between the GATT client and local devices. If re-bonding happens when the connection has already been encrypted, the settings are not erased.</td>
</tr>
<tr>
<td>1164860</td>
<td>Fix a bug that causes the Bluetooth link layer to transmit invalid Tx and Rx maximum times in the LLCP_LENGTH_REQ and LLCP_LENGTH_RSP packets to the remote device.</td>
</tr>
<tr>
<td>1159523</td>
<td>Fix an issue where the ESL access point running on Linux could not connect to ESL Tags.</td>
</tr>
</tbody>
</table>
### Fixed Issues

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>731981</td>
<td>Document UG434 is updated to only describe the priorities of interrupts used by the Link Layer and radio. Other interrupts, e.g., the USART interrupt, that are not used by the Bluetooth stack are no longer listed. Check the priority settings in the corresponding component to determine whether it uses the default priorities or has its own setting.</td>
</tr>
<tr>
<td>1074048</td>
<td>Fix an issue in Apploader that the address and address type are sent in wrong order in the advertising data packet. This issue results in the address type becoming part of the address and the most significant byte of the address becoming the address type.</td>
</tr>
<tr>
<td>1084247</td>
<td>Fix a memory leak if a connection update is received from the remote device at the same time as the connection is being closed by the local device.</td>
</tr>
<tr>
<td>1099142</td>
<td>Fix a pairing failure when the remote device address is a resolvable private address that has been resolved in link layer.</td>
</tr>
<tr>
<td>1104191</td>
<td>Fix a peripheral connection performance issue with simultaneous scanning.</td>
</tr>
<tr>
<td>1105786</td>
<td>Fix the extended connection backoff procedure behavior. Now, when the Link Layer fails multiple times to receive a AUX_CONNECT_RSP after a AUX_CONNECT_REQ is sent, an increasingly random number of connectable extended advertisements are skipped before attempting to connect again.</td>
</tr>
<tr>
<td>1118901</td>
<td>Update the documentation of the sl_bt_evt_connection_tx_power event to explicitly list the situations in which this event may get triggered.</td>
</tr>
<tr>
<td>1124748</td>
<td>Fix a possible MIC failure case when encrypting a connection with the bluetooth_feature_external_bonding_database component.</td>
</tr>
<tr>
<td>1124749</td>
<td>Fix an issue that could cause the Bluetooth stack to enter a live-lock situation where no progress is made if buffer resources are exhausted when the stack is generating critical events that require a retry in out-of-memory situations.</td>
</tr>
<tr>
<td>1124749</td>
<td>Update the documentation of the sl_bt_external_signal API that, if the Platform Core Interrupt API has been configured to use the CORE_ATOMIC_METHOD_BASEPRI as the implementation method of atomic sections, this function must not be called from an interrupt handler with a priority higher than CORE_ATOMIC_BASE_PRIORITY_LEVEL.</td>
</tr>
<tr>
<td>1132204</td>
<td>Improve that ATT PDU validation in the GATT server for the case that it receives more data than defined in the data length field of the L2CAP packet. The GATT server will discard such invalid packet.</td>
</tr>
<tr>
<td>1132430</td>
<td>Clarify that the transmission specified in the sl_bt_test_dtm_tx_v4 command is restricted by the global TX power setting that could be set using the sl_bt_system_set_tx_power command.</td>
</tr>
<tr>
<td>1135759</td>
<td>The sl_bt_system_linklayer_configure command now returns error codes properly. Previously it returned SL_STATUS_OK always even when the low level returns an error.</td>
</tr>
<tr>
<td>1135862</td>
<td>Improve the ATT response length validation to prevent the stack from being corrupted if the ATT response contains a too large invalid length value.</td>
</tr>
<tr>
<td>1135899</td>
<td>Fix an issue that the GATT client operation under command sl_bt_gatt_discover_primary_services goes into an infinite loop until the GATT operation timeout occurs if the end group handle in the response from GATT server is 0.</td>
</tr>
<tr>
<td>1137820</td>
<td>Fix an issue that the GATT client continues reading the value infinitely until the GATT operation timeout occurs if the GATT server always claims that it has more data to read for an attribute. This has been fixed such that the GATT client will stop the operation when the data offset is bigger than the attribute's maximum length.</td>
</tr>
<tr>
<td>1142135</td>
<td>Fix a memory leak when debug keys (set with command with sl_bt_sm_set_debug_mode) are used for pairing. The issue does not exist when pairing with normal keys.</td>
</tr>
<tr>
<td>1143357</td>
<td>Fix an issue of possible duplicate handle assignment in the stack in race conditions when a new connection is opened in the time period between the stack generating a sl_bt_evt_connection_opened event and the user application receiving the event for a disconnected connection.</td>
</tr>
<tr>
<td>1143531</td>
<td>Fix an issue that the EDIV and random values were always set to all zero, potentially causing encryption to fail when reconnected to a peripheral that has been paired with legacy pairing.</td>
</tr>
<tr>
<td>1143589</td>
<td>Improve the channel identifier validation in L2CAP COC disconnection requests and responses to comply with the specification.</td>
</tr>
<tr>
<td>1145778</td>
<td>Fix an issue in the stack that the LTK key is stored incorrectly in RAM after doing the bonding. This issue may cause encryption failure if the key is refreshed later on that same connection. This issue does not affect persistent bonding data.</td>
</tr>
</tbody>
</table>
# Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on [https://www.silabs.com/developers/bluetooth-low-energy](https://www.silabs.com/developers/bluetooth-low-energy) in the Tech Docs tab.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>361592</td>
<td>The sync_data event does not report TX power.</td>
<td>None</td>
</tr>
<tr>
<td>368403</td>
<td>If setting CTE interval to 1, a CTE request should be sent in every connection interval. But it is sent only in every second connection interval.</td>
<td>None</td>
</tr>
<tr>
<td>641122</td>
<td>The Bluetooth stack component does not provide a configuration for RF antenna path.</td>
<td>This is an issue specifically for BGM210P. One workaround is to manually update the configuration in <code>sl_bluetooth_config.h</code> in text edit mode. If the OTA with Apploader is used, include the <code>bluetooth_feature_ota_config</code> component in application project. Call command <code>sl_bt_ota_set_rf_path()</code> to set the RF path for OTA mode.</td>
</tr>
<tr>
<td>650079</td>
<td>LE 2M PHY on EFR32[B</td>
<td>M]G12 and EFR32[B</td>
</tr>
<tr>
<td>682198</td>
<td>The Bluetooth stack has an interoperability issue on the 2M PHY with a Windows PC.</td>
<td>No workaround exists. For application development and testing, the disconnection can be avoided by disabling 2M PHY with <code>sl_bt_connection_set_preferred_phy()</code> or <code>sl_bt_connection_set_default_preferred_phy()</code>.</td>
</tr>
<tr>
<td>730692</td>
<td>4-7% packet error rate is observed on EFR32M</td>
<td>BG13 devices when RSSI is between -25 and -10 dBm. The PER is nominal (as per the datasheet) both above and below this range.</td>
</tr>
<tr>
<td>756253</td>
<td>The RSSI value on a Bluetooth connection returned by the Bluetooth API is incorrect on EFR32M</td>
<td>B1, EFR32M</td>
</tr>
<tr>
<td>845506</td>
<td>When the Bluetooth_feature_afh component for AFH is included, the feature initialization always enables AFH.</td>
<td>To include the component but not to enable AFH at device boot, change the parameter value from 1 to 0 in the function call of <code>sl_btctrl_init_afh()</code> in <code>sl_bt_stack_init.c</code>.</td>
</tr>
<tr>
<td>1031031</td>
<td>Changing the configuration in the <code>bt_aoa_host_locator</code> application results in the application crashing.</td>
<td>None</td>
</tr>
</tbody>
</table>
5 Deprecated Items

Deprecated in release 6.0.0.0

Component bluetooth_feature_whitelisting

Command sl_bt_gap_enable_whitelisting

Command sl_bt_sync_set_parameters

Command sl_bt_sync_open

Command sl_bt_evt_sync_opened

Command sl_bt_evt_sync_transfer

Command sl_bt_evt_sync_data

Command sl_bt_sm_add_to_whitelist

The txsize filed in command sl_bt_evt_connection_parameters
6 Removed Items

Removed from release 6.0.0.0

Component bluetooth_feature_dfu

Component bluetooth_apploader_migration_util

Component bluetooth_feature_default
7 Multiprotocol Gateway and RCP

7.1 New Items

Added in release 6.0.0.0

Added a new application z3-light_ot-ftd_soc that demonstrates Zigbee and OpenThread Concurrent Multiprotocol functionality. It features a router on the Zigbee side and a Full Thread Device (FTD) on the OpenThread side. See the project description or app/framework/scenarios/z3/z3-light_ot-ftd_soc/readme.html for details.

First GA-quality release of CPC GPIO Expander module. The Co-Processor Communication (CPC) General Purpose Input/Output (GPIO) Expander is a software component designed to enable a Host device to utilize a Secondary device’s GPIOs as if they were its own. With the CPC GPIO Expander, the Host device can seamlessly integrate with the Secondary device and make use of its GPIO capabilities. See https://github.com/SiliconLabs/cpc-gpio-expander/README.md for documentation.

Added antenna diversity and coex EZSP command support to Zigbee.

Added better assert reporting to Zigbee.

Added bt_host_empty application (option: -B for the run.sh script) to the multiprotocol docker container.

Zigbee now includes an implementation of emberGetRestoredEui64() which loads the CREATOR_STACK_RESTORED_EUI64 token from the host_token.nvm file.

The multiprotocol container now sets the size of syslog to 100 MB by default. Users are able to change the size by modifying the "/etc/logrotate.d/rsyslog" and "/etc/rsyslog.d/50-default.conf" files and restarting the rsyslog service inside the container.

7.2 Improvements

Changed in release 6.1.0.0

Radio_url can now take a list of IIDs. This is to control IIDs a host can accept spinel frames from other than its own IID. RCP uses IID zero for the received broadcast packet to deliver a spinel frame to all the connected hosts/endpoints. An example of the new radio_url is 'spinel+cpc://cpcd_0?iid=1&iid-list=0'.

CLI commands used by z3-light_ot-ftd_soc sample application have been moved from the project into a new component called ot_up_cli.

Changed in release 6.0.0.0

Reduced CPC Tx and Rx queue sizes to fit the DMP NCP on the MG13 family.

Configured options on the multiprotocol RCP projects to provide ~3.3k in RAM savings, particularly for the MG1 part. This was accomplished by

- Reducing
  - The number of user CPC endpoints to 0
  - Tx CPC queue size to 15 from 20
  - Rx buffer count to 15
- Disabling OpenThread RTT logs

For further savings, customers can look into reducing the Tx and Rx queue sizes further. Note that the downside to this change would be a reduction in message throughput due to added retries. Also, customers can look into reducing the NVM cache size based on need. As a last resort, customers may also choose to disable CPC security on both the RCP and the host. We do not recommend the last option.

Changed zigbee_ble_event_handler to print scan responses from legacy advertisements in the DMPLight(Sed) app.

The rcp-xxx-802154 apps now by default support 192 µsec turnaround time for non-enhanced acks while still using 256 µsec turnaround time for enhanced acks required by CSL.
### 7.3 Fixed Issues

**Fixed in release 6.2.0.0**

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1090921</td>
<td>An issue has been fixed that caused Z3GatewayCpc not being able to form the zigbee network in a noisy environment.</td>
</tr>
<tr>
<td>1190859</td>
<td>An EZSP issue when sending mfglib packets CMP RCP setup has been fixed.</td>
</tr>
</tbody>
</table>

**Fixed in release 6.1.0.0**

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1113498</td>
<td>Resolved several issues that caused intermittent failures in zigbeed when attempting to join 100 or more Zigbee devices to the network.</td>
</tr>
<tr>
<td>1135805</td>
<td>An issue has been fixed that caused failure on the host if a communication error occurred when reading the NCP version.</td>
</tr>
<tr>
<td>1153055</td>
<td>An argument parsing issue has been fixed in the CLI command ping_ipaddr of the ot_up_cli component.</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>1078323</td>
<td>Resolved issue where Z3GatewayCPC asserts when there is a communication failure with the NCP during address table initialization. We will now try to reconnect to the NCP upon failure.</td>
</tr>
<tr>
<td>1080517</td>
<td>Z3GatewayCPC now automatically handles a reset of the NCP (CPC secondary).</td>
</tr>
<tr>
<td>1117789</td>
<td>Fixed an issue where modifying OPENTHREAD_CONFIG_PLATFORM_RADIO_SPINEL_RX_FRAME_BUFFER_SIZE caused a linker error when building Zigbeed.</td>
</tr>
<tr>
<td>1118077</td>
<td>In the CMP RCP, Spinel messages were being dropped under heavy traffic load due to CPC not keeping up with the incoming packets. Fixed this by bundling all Spinel messages ready to be sent over CPC into one payload on the RCP and unbundling them on the host. This dramatically improves the efficiency of CPC so that it can keep up with the incoming radio traffic.</td>
</tr>
<tr>
<td>1129821</td>
<td>Fixed null pointer dereference in Zigbeed in an out-of-buffer scenario while receiving packets.</td>
</tr>
<tr>
<td>1139990</td>
<td>Fixed an assert in the OpenThread Spinel code that could be triggered when joining many Zigbee devices simultaneously.</td>
</tr>
<tr>
<td>1144268</td>
<td>Fixed an issue where excessive radio traffic can cause the Zigbee-BLE NCP to get into a state where it continually executes the NCP and CPC initialization.</td>
</tr>
<tr>
<td>1147517</td>
<td>Fixed an issue with Z3GatewayCPC on startup that could cause the reset handling of the secondary to not work correctly.</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>118077</td>
<td>In the CMP RCP, Spinel messages were being dropped under heavy traffic load due to CPC not keeping up with the incoming packets. Fixed this by bundling all Spinel messages ready to be sent over CPC into one payload on the RCP and unbundling them on the host. This dramatically improves the efficiency of CPC so that it can keep up with the incoming radio traffic.</td>
</tr>
</tbody>
</table>

---

### 7.4 Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on [https://www.silabs.com/developers/gecko-software-development-kit](https://www.silabs.com/developers/gecko-software-development-kit).

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>811732</td>
<td>Custom token support is not available when using Zigbeed.</td>
<td>Support is planned in a future release.</td>
</tr>
<tr>
<td>937562</td>
<td>Bluetoothctl <code>advertise on</code> command fails with rcp-uart-802154-blehci app on Raspberry Pi OS 11.</td>
<td>Use btmgmt app instead of bluetoothctl.</td>
</tr>
<tr>
<td>1074205</td>
<td>The CMP RCP does not support two networks on the same PAN id.</td>
<td>Use different PAN ids for each network. Support is planned in a future release.</td>
</tr>
<tr>
<td>1122723</td>
<td>In a busy environment the CLI can become unresponsive in the z3-light_ot-ftd_soc app.</td>
<td>This app is released as experimental quality and the issue will be fixed in a future release.</td>
</tr>
<tr>
<td>1124140</td>
<td>z3-light_ot-ftd_soc sample app is not able to form the Zigbee network if the OT network is up already.</td>
<td>Start the Zigbee network first and the OT network after.</td>
</tr>
</tbody>
</table>
### 7.5 Deprecated Items

None

### 7.6 Removed Items

None
8 Using This Release

This release contains the following

- Silicon Labs Bluetooth stack library
- Bluetooth sample applications

For more information about the Bluetooth SDK see [https://docs.silabs.com/bluetooth/latest/](https://docs.silabs.com/bluetooth/latest/). If you are new to Bluetooth see UG103.14: Bluetooth LE Fundamentals.

8.1 Installation and Use

The Bluetooth 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](https://github.com/SiliconLabs/gecko_sdk) for more information.

Simplicity Studio installs the GSDK by default in:

- (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/](https://docs.silabs.com/).

8.2 Security Information

Secure Vault Integration

When deployed to Secure Vault High devices, sensitive keys such as the Long Term Key (LTK) 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>Wrapped Key</th>
<th>Exportable / Non-Exportable</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Long Term Key (LTK)</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>Local Long Term Key (legacy only)</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>Remote Identity Resolving Key (IRK)</td>
<td>Exportable</td>
<td>Must be Exportable for future compatibility reasons</td>
</tr>
<tr>
<td>Local Identity Resolving Key</td>
<td>Exportable</td>
<td>Must be Exportable because the key is shared with other devices.</td>
</tr>
</tbody>
</table>

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

Wrapped 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](https://silabs.com/).
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.

8.3 Support

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

You can 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!