Zigbee EmberZNet SDK 8.0.1.0 GA

Simplicity SDK Suite 2024.6.1
July 24, 2024

Silicon Labs is the vendor of choice for OEMs developing Zigbee networking into their products. The Silicon Labs Zigbee platform is the most integrated, complete, and feature-rich Zigbee solution available.

Silicon Labs EmberZNet SDK contains Silicon Labs' implementation of the Zigbee stack specification.

These release notes cover SDK version(s):

8.0.1.0 released July 24, 2024
8.0.0.0 released June 5, 2024

Zigbee

KEY FEATURES

Zigbee

- Zigbee re-architecture to support stack and framework to be fully RTOS based
- Memory management update
- New clock manager component integrated in Zigbee applications
- Removed support for Series 0/1
- Added support for series 3 – Alpha
- Added support for new platform xG26, xG22E
- Zigbee GP 1.1.2
- Support Sleepy NCP use case
- New application for Zigbee Unified Test Harness - Alpha

Multiprotocol

- Alpha support for OpenWRT on host processor of multiprotocol RCP solution
- Alpha support for Concurrent Zigbee and Matter over OpenThread, with DMP BLE

Compatibility and Use Notices

For information about security updates and notices, see the Security chapter of the Platform Release Notes installed with this SDK or on the TECH DOCS tab on https://www.silabs.com/developers/zigbee-emberznet. 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 Zigbee EmberZNet SDK, see Using This Release.

Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 9.40.1.

- Usingwine 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.

The EZSP protocol version for this release is 0x0E.

Contents

1 New Items

Simplicity SDK is an embedded software development platform for building IoT products based on our Series 2 and Series 3 wireless and MCU devices. It integrates wireless protocol stacks, middleware, peripheral drivers, a bootloader, and application examples – a solid framework for building power-optimized and secure IoT devices.

The Simplicity SDK offers powerful features such as ultra-low power consumption, strong network reliability, support for a large number of nodes, and abstraction of complex requirements like multiprotocol and pre-certification. Additionally, Silicon Labs provides over-the-air (OTA) software and security updates to remotely update devices, minimize maintenance costs, and enhance the end-user product experience.

Simplicity SDK is a follow-on from our popular Gecko SDK, which will continue to be available, providing long-term support for our Series 0 and Series 1 devices. For additional information on the Series 0 and Series 1 devices, please reference: Series 0 and Series 1 EFM32/EZR32/EFR32 device (silabs.com).

1.1 Important Changes

Important Changes in release 8.0.0.0

- Migration from EmberStatus to sl_status_t

  In this release, the status code for function calls, error codes, events, and APIs have been migrated from 1 byte EmberStatus to 4bytes sl_status_t type enumeration.

  New sl_status_t values are either added to the Zigbee space (0x0C00) or to the generic space (0x0000). In the latter case, it is added values starting at 0x0051, maintaining the enum.
- As part of the move to SL_ naming conventions in ZNet v8.0 alongside type names and API names, most enumerations in Zigbee code now follow an SL_ZIGBEE_ prefix instead of EMBER_.
- Types renaming

  This consists of renaming all types at the stack level and AppFramework level from the old “Ember” or “Em” prefix naming to the new sl_zigbee and sli_zigbee prefix naming.
- Public API renaming

  All public stack-level APIs will adopt the sl_zigbee prefix, while the AppFramework public APIs will adopt the sl_zigbee_af prefix.
- Rename all public variables

  To allow the RTOS thread safety while accessing public/global variables, all the global variables are renamed as internal variables. Reading or writing these variable directly is not recommended. The method to read or write these variables are solely through setters/getters. Based on application running RTOS or bare metal, the getter/setter is redirected appropriately.
- SoC/HOST API aligning

  Every stack API signature is aligned to its host version.
- Stack Buffer Management now uses dynamic heap allocation to acquire initial heap memory

  Reworked internal stack buffer initialization to acquire memory for buffer heap via library code callback.

  Removed previous 'Packet Buffer Count' configuration item in favor of directly specifying the amount of heap to allocate (in bytes). For a few releases we have been using the 'new' buffer system but maintained the old (and inaccurate) way of configuring and allocating the memory for it. With this the amount of buffer heap to be allocated for packet buffers can be specified by choosing one of the pre-configured options (tiny, small, medium, large, huge) or by specifying a custom heap size (4-byte aligned). This heap memory will be shared by both the Packet Buffer content and the associated metadata (8-bytes of overhead per packet buffer).
- Updated memory management

  Zigbee code has traditionally used packet buffers for dynamic memory allocation. Packet buffers are allocated from SL_HEAP and are compacted / garbage collected in the Zigbee main task tick. Packet buffers are therefore not thread safe.

  In prior implementations where there was only one RTOS task, it was okay to allocate packet buffers in both stack and application framework code. However, with the change in architecture to run these in different RTOS tasks, it is possible for the application framework task to be preempted by the stack task. Unfortunately, this also means that a buffer whose pointer was valid can now be compacted and moved during the garbage collection / compaction phase.

  Therefore, we scrubbed all of the public APIs and the application framework code to change all the signatures to no longer use any packet buffers. Any API that previously used buffers now uses a pointer / length pair in its place.
- Update to callback return type

  In this release all the callbacks are updated to void type return.
- Added mutex protection for the App Framework task to provide a thread-safe way for calling its APIs from other RTOS tasks.

1.2 New Components

New in release 8.0.0.0

- Inter Process Communication

  Added a new component named “zigbee_ipc”.

  This component adds support for interprocess communication on a Zigbee RTOS device.
- Byte utilities

  Added a new utility component named “byte_utilities” outside Zigbee protocol for data store and fetch routines.

  As part of that, moved the header file byte-utilities.h from protocol/zigbee/stack/include/ to util/plugin/byte_utilities.

  Since this is not specific to Zigbee, the API has been renamed, from sl_zigbee_reverse_mem_copy() to sl_util_reverse_mem_copy().
- Separated a new component "Zigbee System Common (App Framework Task)" from Zigbee System Common, to better allow for apps that do not bring in the full ZCL and app framework but still use the app framework's RTOS task.
- RTOS Configurations

1.3 New APIs

New in release 8.0.0.0

- A new event initialization function sl_zigbee_af_isr_event_init is added for the events that can be activated in ISR.

  Note that with this release the APIs have been renamed. For details, see the Zigbee v8.0.0 section of https://docs.silabs.com/.

1.4 New Platform Support

New in release 8.0.1.0

- New hardware xG27.

New in release 8.0.0.0

- New hardware xG26 and xG22E.
- A new platform component, Clock Manager, is integrated into all Zigbee sample applications for clock configuration.

1.5 New Documentation

New in release 8.0.0.0

None.

1.6 New Application

New in release 8.0.0.0

- Zigbee Unified Test Harness SoC application is introduced as a new application in this release.
- Added a new ZigbeeMinimalRtos app to Zigbee offerings. This is a test application with a bare minimum RTOS based Zigbee stack. This application is non-compliant since it does not have all the necessary component and configurations of a fully functional Zigbee application.
- A new configuration for the SPI based NCP as a sleepy NCP is introduced in this release. This serves the use case of a host-SPI NCP configuration to be used as a sleepy node in the network. The host controls the sleep and wake up of the SPI NCP. See *AN711: SPI Host Interfacing Guide for Zigbee* for more details.

1.7 Intended Behavior

Users are reminded that Zigbee unsynchronized CSL transmissions are subject to protocol preemption at the radio scheduler. In the SleepyToSleepy applications, BLE can and will preempt a Zigbee CSL transmission, which will terminate the transmission. Scheduler preemption is more common for unsynchronized CSL, given that a potentially lengthy wake up frame sequence may be used. Users wishing to adjust transmission priorities may use the DMP Tuning and Testing component to do so. Users may also consult *UG305: Dynamic Multiprotocol User's Guide* for more information.

2 Improvements

Changed in release 8.0.0.0

- The API sl_zigbee_af_acquire_lock() is used to acquire the mutex, and sl_zigbee_af_release_lock() is used to release it. These APIs should be called whenever another application RTOS task needs to access App Framework data or APIs, including when calling APIs related to event scheduling.
- sl_zigbee_send_reply, sl_zigbee_set_reply_binding, and sl_zigbee_note_senders_binding now take in the original sender's node ID as a parameter, and sl_zigbee_send_reply now uses a length and pointer to a flat array for its message parameter.
- Updated the signature of incoming route handlers to adopt sl_zigbee_rx_packet_info_t struct.
- Removed message buffers from code that runs in the app framework context, to make it thread-safe with stack code.
- Related to this change, zigbee_form_and_join's default maximum number of networks remembered is now 5 for both host and SoC.
- sl_zigbee_set_mac_filter_match_list has a new second argument for the input list's length.
- sl_zigbee_match_descriptors_request now takes in uint16_t arrays rather than message buffers for its cluster list parameters.

  Its new signature is:

  ```
  sl_zigbee_match_descriptors_request(sl_802154_short_addr_t target,
                                      uint16_t profile,
                                      uint8_t inCount,
                                      uint16_t* inClusters,
                                      uint8_t outCount,
                                      uint16_t* outClusters,
                                      sl_zigbee_aps_option_t options)
  ```
- Added mutex protection for App Framework event queue so App Framework event APIs can be called from multiple tasks.
- sl_zigbee_gp_tx_queue_entry_t no longer contains the message buffer field asdu.

  Added new API sl_zigbee_gp_get_tx_queue_entry_from_queue_index(uint8_t index, sl_zigbee_gp_tx_queue_entry_t* tx_queue, uint8_t* payload, uint16_t* payload_len) to retrieve an entry from the GP TX queue by queue position.
- The SE1.4a specification requires a Multi-MAC Joining End Device to rejoin on the same interface it had joined. The rejoin may be initiated as a result of parent loss or a reset of the node. In both cases, the stack API emberFindAndRejoinNetworkWithReason is called internally to start the rejoin process, and this needs the rejoin channel mask as an argument.

  This channel mask is provided by the call to API sl_zigbee_internal_update_multi_mac_rejoin_channel_mask_for_selection_or_joining_device_handler that is implemented as source and as part of the application project. A default implementation can be found in the ember-configuration.c as a WEAK function so that a user override is possible.
- The function signature of GSDK 4.4's emberUpdateMultiMacRejoinChannelMaskForSelectionOrJoiningDevice has been changed to become uint32_t sl_zigbee_internal_update_multi_mac_rejoin_channel_mask_for_selection_or_joining_device_handler(uint32_t rejoinMask). This change is for consistency with its intended behavior as a privileged callback that does not traverse the IPC layer.
- Improved parameter validation in function sli_zigbee_af_gp_make_addr.
- Move sl_mac_set_cca_threshold() sl_mac_get_ed_cca_threshold() sl_mac_set_csma_params() sl_mac_get_csma_params() to stack/include/stack-info.h so these public API can be used by customer
- Separated a new component "Zigbee System Common (App Framework Task)" from Zigbee System Common, to better allow for apps that do not bring in the full ZCL and app framework but still use the app framework's RTOS task.

3 Fixed Issues

Fixed in release 8.0.1.0

ID #	Description
1308602	sl_zigbee_mac_filter_match_list has been fixed to work properly in RTOS builds. The API now fully relies on list_length to parse the length of the passed-in filter list, and the length of a MAC filter match list has a limit of 32 (SL_802154MAC_FILTER_MATCH_LIST_MAX_LENGTH). Other ref. 1305214
1312106	Fixed an issue where the non-stub implementation of sl_zigbee_child_index() was incorrectly placed inside the Zigbee R22 Support library, causing it to return an invalid index for builds that do not use that library. Other ref. 1312065
1312369	Made SL_LEGACY_HAL_WDOG_IRQHandler() weak function.
1326010	Fixed an issue that created an excess default response from Z3Gateway for OTA cluster command handling. Other ref. 1300935

Fixed in release 8.0.0.0

ID #	Description
1036893	Fixed an issue that caused the OTA cluster component to install the legacy boot-loader interface component as a dependency. Other ref. 1252145
1156458	Fixed an issue that caused the gpd sensor to fail to commission when the environment is busy. Other ref. 1294349
1196863	Fixed an issue that the response destination of green power message is invalid when the customer does not use Silabs app framework. Other ref. 1224393
1206040	Calling emberRemoveChild() during a secure rejoin attempt by an end device can potentially lead to an extra decrement of the Child Count, potentially leading to a Child Count of -1 (255), inhibiting end devices from joining/rejoining due to an indicated lack of capacity in the Beacon. Other ref. 1227819, 1227826
1211249	Fixed an issue that prevented coex counters to increment as expected in Z3Gateway/zigbeed.
1213770	Fixed an issue where the trust-center-backup plugin was broken due to the EZSP frame not returning token contents properly.
1223985	Fixed an issue that caused the end device network move to work incorrectly in very busy environments. Other ref. 1223904
1225294	The stack was keeping track of recently seen unicast packets at the NWK layer in an attempt to provide rudimentary duplicate protection. However, unicast messages were being added to this table before checking for successful decryption, which could result in improper duplicate filtering if a device with the same node ID as an existing neighbor used an NWK sequence number that collided with that of the neighbor in question. In resolving this issue, the use of an NWK unicast table for duplicate tracking by the stack was removed as security frame counter checks already provide more robust duplicate filtering without incurring the same risk.
1227738	Fixed issue that caused the APS Verify Key Confirm message processing error. Other ref. 1255175
1228301	The USART0 can be configured to be used with SPI or IOStream components but not both. Misconfiguration of USART0 to be used by both can lead to unintended behavior such as SPI storage for OTA image or/and the IOstream for CLI or print not working as expected. The default configuration is fixed by using EUSART for SPI and recommended IOstream for vcom in the default applications project configurations. It is highly recommended to ensure this conflict is not present when using custom board or a different board that is not supported by default when using the SPI and IOstream. Other ref. 1252330

ID #	Description
1228403	Child Table search functions within the stack are inconsistent in use of 0x0000 versus 0xFFFF for node ID return value representing invalid/empty entries, leading to problems checking for unused entries in APIs like emberRemoveChild(). Other ref. 1207580
1229001	Fixed an issue where a retrieved PAN ID could contain values other than 0x0000 if the device was not joined to a network.
1232297	Fixed an issue where emberSetOutgoingNwkFrameCounter and emberSetOutgoingApsFrameCounter did not work on 64-bit host applications (returning EMBER_BAD_ARGUMENT).
1240366	ZDO Bind/Unbind Requests refused for access/permission reasons should return EMBER_ZDP_NOT_AUTHORIZED status rather than EMBER_ZDP_NOT_PERMITTED status as per Zigbee specifications.
1242066	Added validation script to warn user when EMBER_AF_PLUGIN_OTA_CLIENT_SIGNATURE_VERIFICATION_SUPPORT is set but zigbee_ota_client_signature_verify_support isn't enabled.
1248853	Fix a typo error in component catalog macro SL_CATALOG_ZIGBEE_OTA_STORAGE_COMMON_PRESENT.
1249347	Zigbee Direct: Improved handling of Leave Network Characteristic.
1249489	Fixed the display issue with macro definitions in gp-types.h documentation.
1249871	Corrected the RTOS macro at diagnostic.c.
1250702	Zigbee Direct: Improved stability of BLE connection to ZVD.
1250849	Fixed an issue that led to commissioning failure with GP Combo Host-NCP model.
1251512	Fixed an issue that incorrect initial value of status in emRadioTransmit caused assertion in lower mac.
1251566	Fixed an issue that caused build failure when enabling expanded reporting table on Z3Gateway.
1254054	Fixed the gppTunnelingDelay parameter calculation in green power client command processing.
1258585	Fixed an issue that led to wrong status in read sink/proxy table attribute response.
1258607	Fixed a compile issue on the host apps which use ezsp spi.
1258636	Optional Mask parameter for "net multi-phy-start" CLI command was being ignored and always treated as 0.
1259309	Fixed a compile issue on the host apps which use ezsp spi.
1259936	Fixed an issue that GP server did not drop a frame when the FC equaled the stored FC.
1262368	Removed code for deprecated MAC command handling.
1266351	Fixed an issue to improve multi hop joining of the devices. Upon device announcement, the address pair is added to the address cache.
1269700	The initialization value for the green power security frame counters in non-volatile memory is now set to 0 instead of 0xFFFFFFFF.
1269856	Stack events not marked interrupt-safe should not be scheduled in ISR context as this could create a race condition resulting in the event being dropped from the event queue. An instance of this prevented sub-GHz devices from properly scheduling MAC backoffs when this race condition was triggered.
1272229	Fixed an issue that prevented the application framework to send out ZCL default response for success case.
1273359	Fixed some proxy test failures, primarily with broadcast transmit busy and with the groupcast alias option. Other ref. 1268035
1273484	Fixed some GP ZUTH test failures at Verify GP Pairing Sent step.
1275092	Added checks for the ZCL data array to accommodate the longest string length to 253.
1275096	Fix Coverity warning in buffered ota block write. This is not a functional issue since the Zigbee OTA block without fragmentation would never be more than 128 bytes, which is the EEPROM page size.
1275104	Fixed missing initialization of local variables before use.
1275586	Fixed an issue that dropped a ZLL commissioning packet for the use case when one endpoint is disabled while other endpoints are still enabled. The fix is added to check all the enabled endpoints.

ID #	Description
1276049	If a dual-PHY NCP device were performing an Energy Scan on the sub-GHz band while the 2.4GHz radio was active, large bursts of incoming Beacon traffic in response to incoming Beacon Requests on the 2.4GHz radio channel could occupy lots of the stack's buffer memory (allocated from the Heap segment) while the many sub-GHz channels were being scanned, potentially leading to buffer shortages and EZSP Overflow conditions if the Heap size (which determines how many buffers are available to the system) was not sufficiently large. This Overflow condition could prevent the Scan Complete Handler from reaching the host application, causing the app's scan state machine to be stuck forever. The Energy Scanning code for dual PHY devices now discards any Beacon packets received on the 2.4GHz radio during the scanning of a sub-GHz channel, thus preventing a large influx of beacons from creating the aforementioned problem. Note this does not prevent normal operation of the network on 2.4GHz, as Beacons are only used for potential PAN ID conflict detection when the device isn't performing an Active Scan, and PAN ID conflicts can still be detected by other routers in the network during this time or by the coordinator once the Energy Scan has completed.
1277340	Triggering a child removal operation for a sleepy end device child when an Association Response or Rejoin Response is still pending delivery for that child can result in two erasures for the same child entry, leaving the emberChildCount() off by -1. If this is the last child in the table to be removed, this can lead to an underflow of the child count, which inhibits any further join/rejoin attempts by any end devices until a reset or LeaveNetwork is performed and which may result in an assert failure (citing child.c) in certain releases.
1277344	If child table is full and an existing child of the local device tries to rejoin, the rejoin is incorrectly rejected, causing the child entry to be removed and forcing the child to rejoin again to get reassociated with that parent.
1279109	Fixed an issue that caused a crash when a multi-network apps tries steering on the secondary network.
1280058	Fixed compile issues when adding test harness zigbee 3.0 component to a standard application.
1282579	Fixed an issue with IAR malloc smart selection which led to crash in some ncp apps.
1289413	The fragmentation plugin miscalculates available payload for a unicast fragment, which may trigger assert failures or other unexpected behavior in the NCP.
1293923	Fixed an issue where the network steering plugin could act as if the device was already on a network when it was not, leaving the plugin stuck in this invalid state until reset. This could be triggered by calling the network leave CLI command with precise enough timing while network steering was in the middle of joining a network and with a MAC scan still pending.
1294936	Corrected print format in new ZUTH app.
1296504	Increased the broadcast table size of new ZUTH app, helping to avoid busy broadcast issue in some sink commission tests

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/zigbee-emberznet in the Tech Docs tab.

ID #	Description	Workaround
N/A	The following apps/component is not supported in this release: EM4 support.	Feature will be enabled in subsequent releases.
193492	emberAfFillCommandGlobalServerToClientConfigureReporting macro is broken. The filling of buffer creates incorrect command packet.	Use the "zcl global send-me-a-report" CLI command instead of the API.
278063	Smart Energy Tunneling plugins have conflicting treatment/usage of address table index.	No known workaround
289569	Network-creator component power level picklist doesn't offer full range of supported values for EFR32	Edit the range <-8..20> specified in the CMSIS comment for EMBER_AF_PLUGIN_NETWORK_CREATOR_RADIO_POWER in the <sdk>/protocol/zigbee/app/framework/plugin/network-creator/config/network-creator-config.h file. For example, change to <-26..20>.
295498	UART reception sometimes drops bytes under heavy load in Zigbee+BLE dynamic multiprotocol use case.	Use hardware flow control or lower the baud rate.
312291	EMHAL: The halCommonGetIntxxMillisecondTick functions on Linux hosts currently use the gettimeofday function, which is not guaranteed to be monotonic. If the system time changes, it can cause issues with stack timing.	Modify these functions to use clock_gettime with the CLOCK_MONOTONIC source instead.
338151	Initializing NCP with a low packet buffer count value may cause corrupt packets.	Use the 0xFF reserved value for packet buffer count to avoid the too-low default value
387750	Issue with Route Table Request formats on end device.	Under Investigation
400418	A touchlink initiator cannot link to a non-factory-new end-device target.	No known workaround.
424355	A non-factory-new sleepy end device touchlink target-capable initiator is not able to receive a device information response in certain circumstances.	Under Investigation
465180	The Coexistence Radio Blocker Optimization item "Enable Runtime Control" may block proper Zigbee operation.	Optional 'Wi-Fi Select' Control of Blocker Optimization should be left "Disabled".
480550	The OTA cluster has its own built-in fragmentation method, hence it should not use APS fragmentation. Although, in case APS encryption is enabled it grows the payload of the ImageBlockResponses to a size where the APS fragmentation is activated. This could lead to the OTA process failing.	No known workaround
486369	If a DynamicMultiProtocolLightSoc forming a new network has child nodes remaining from a network it has left, emberAfGetChildTableSize returns a non-zero value in startIdentifyOnAllChildNodes, causing Tx 66 error messages when addressing the "ghost" children.	Mass-erase the part if possible before creating a new network or programmatically check the child table after leaving the network and delete all children using emberRemoveChild prior to forming a new network.
495563	Joining SPI NCP Sleepy End Device Sample App doesn't short poll, therefore the joining attempt fails at the state of Update TC Link Key.	The device that wishes to join should be in Short Poll mode before attempting to join. This mode can be forced by the End Device Support plugin.
497832	In Network Analyzer the Zigbee Application Support Command Breakdown for the Verify Key Request Frame mistakenly references the part of the payload that indicates the frame Source Address as the Destination Address.	No known workaround

ID #	Description	Workaround
519905 521782	Spi-NCP may very rarely fail to start up bootloader communication using the 'bootload' CLI command of the ota-client plugin.	Restart the bootload process
620596	NCP SPI Example for BRD4181A (EFR32xGMG21) nWake default pin defined cannot be used as a wake-up pin.	Change the default pin for nWake from PD03 to a EM2/3 wake-up-enabled pin in the NCP-SPI Plugin.
631713	A Zigbee End Device will report address conflicts repeatedly if the plugin "Zigbee PRO Stack Library" is used instead of "Zigbee PRO Leaf Library".	Use the "Zigbee PRO Leaf Library" instead of the "Zigbee PRO Stack Library" plugin.
670702	Inefficiencies within the Reporting plugin can lead to significant latency based on data write frequency and table size, which may interfere with customer application code, including event timing.	If doing frequent writes, consider checking reporting conditions and sending reports manually rather than using the plugin.
708258	Uninitialized value in groups-server.c via addEntryToGroupTable() can create a spurious binding and cause groupcast reporting messages to be sent.	Add "binding.clusterId = EMBER_AF_INVALID_CLUSTER_ID;" after "binding.type = EMBER_MULTICAST_BINDING;"
757775	All EFR32 parts have a unique RSSI offset. In addition, board design, antennas and enclosure can impact RSSI.	When creating a new project, install the RAIL Utility, RSSI component. This feature includes the default RSSI Offset Silabs has measured for each part. This offset can be modified if necessary after RF testing of your complete product.
758965	ZCL cluster components and ZCL command discovery table are not synchronized. Therefore, when enabling or disabling a ZCL cluster component, implemented commands will not be enabled/disabled in the corresponding ZCL Advanced Configurator command tab.	Manually enable/disable discovery for the desired ZCL commands in the ZCL Advanced Configurator.
765735	The OTA update fails on Sleepy End Device with enabled Page Request.	Use Block Request instead of Page Request.
845649	Removing CLI:Core component does not eliminate EEPROM cli calls to sl_cli.h.	Delete the eeprom-cli.c file that calls the sl_cli.h. Additionally, calls to sl_cli.h as well as sl_cli_command_arg_t in the ota-storage-simple-eeprom can be commented out.
857200	ias-zone-server.c allows for a binding to be created with a "0000000000000000" CIE address and posteriorly does not allow further bindings.	No known workaround
1019961	Generated Z3Gateway makefile hardcodes "gcc" as CC	No known workaround
1039767	Zigbee router network retry queue overflow issue in multi thread RTOS use case.	Zigbee Stack is not thread-safe. As a result, calling Zigbee stack APIs from another task is not supported in OS environment and may put the stack into "non-working" state. Refer to the following App note for more information and workaround using event handler. https://www.silabs.com/documents/public/application-notes/an1322-dynamic-multiprotocol-bluetooth-zigbee-sdk-7x.pdf .
1064370	The Z3Switch sample application only enabled one button (instance : btn1) by default that leads to mismatch in button description in the projectfile.	Workaround: Install the btn0 instance manually during Z3Switch project creation.
1161063	Z3Light and potentially other applications report incorrect cluster revision values.	Manually update the cluster revision attribute to their appropriate revision.
1164768, 1171478, 1171479	ERROR: ezspErrorHandler 0x34 reported repeatedly during mfglib receive mode	To reduce the error messages printed, configure EMBER_AF_PLUGIN_GATEWAY_MAX_WAIT_FOR_EVENT_TIMEOUT_MS on the host app to 100, so the callback queue is freed more quickly.
1254368	**Fragmentation of new Zigbee 2023 ZDO messages is not supported in this release**	**No known workaround.** **This will be supported in the next planned release.**

ID #	Description	Workaround
1297976	The Z3Gateway built on Raspberry Pi with kernel 6.6 fails to connect to NCP over SPI. The primary reason being the failure to export and set direction of the gpios during the initialization. This is tested to work on kernel 6.1.	Workaround is to redefine the GPIOs as described here. Note the following information on GPIOs that maps to the SPI NCP interface. On kernel 6.6, running sudo cat /sys/kernel/debug/gpio That will display the GPIOs, those are of interest are the following ones, since those are the IOs interfacing the SPI NCP. gpio-520 (GPIO8) gpio-534 (GPIO22) gpio-535 (GPIO23) gpio-536 (GPIO24) Then redefine the GPIO for SPI NCP interface from above sysfs in spi-protocol-linux.c as #define NCP_CHIP_SELECT_GPIO "520" #define NCP_HOST_INT_GPIO "534" #define NCP_RESET_GPIO "535" #define NCP_WAKE_GPIO "536"
1300935	There is an additional default response generated when the incoming OTA server command ZCL message has default response enabled.	As a workaround, set the 'disable default response bit' in the OTA Image Block Request command handler by adding: command->buffer[0] &= ~ZCL_DISABLE_DEFAULT_RESPONSE_MASK; to sl_zigbee_af_ota_server_incoming_message_raw_cb. When this is done, only one response to an error is sent, and for successful block image requests, no default response is set.
1305214	sl_zigbee_mac_filter_match_list does not properly assign the right filter list if called over the IPC.	Workaround: Call sli_zigbee_stack_mac_filter_match_list directly.
1306512	It is not possible to set the CTUNE value over EZSP command with manufacturing lib.	No known workaround.

5 Deprecated Items

Deprecated in release 8.0.0.0

- The Toolchain: Legacy HAL Compatible Memory Layout component has been removed.

  This component was only applicable to images using the long-removed Ember bootloaders, which have since been replaced by the Gecko Bootloader.

- The DMP applications are no longer available for MG22/MGM22 parts.

  Dynamic Multiprotocol applications use RTOS, which needs more flash (for an SoC OTA upgradable use case) and RAM (for more stack and heap configurations). These parts do not have enough flash and RAM to meet the default sample application configurations. Hence, the sample applications no longer support those variants of the chips and modules.

6 Removed Items

Removed in release 8.0.0.0

None.

7 Multiprotocol Gateway and RCP

7.1 New Items

Added in release 8.0.0.0

OpenWRT alpha support has been added for zigbeed, OTBR and Z3Gateway applications. Zigbeed and OTBR are now provided in IPK package format as well. See *AN1333: Running Zigbee, OpenThread, and Bluetooth Concurrently on a Linux Host with a Multiprotocol Co-Processor* for details.

7.2 Improvements

Changed in release 4.4.1.0

The zb_ble_dmp_print_ble_connections() API is defined in the zigbee_ble_event_handler component and referenced in the zigbee_ble_dmp_cli component. For applications that use the zigbee_ble_dmp_cli component, but NOT the zigbee_ble_event_handler, you will need to add an empty stub for this function in your app.c file as follows:

```
void zb_ble_dmp_print_ble_connections(void)

{

}
```

Changed in release 8.0.0.0

None.

7.3 Fixed Issues

Fixed in release 8.0.1.0

ID #	Description
1300848	An issue were fixed where Z3Gateway in OpenWRT environment couldn't start EZSP communication caused by mismatching termios control characters running on OpenWRT and other environments.

Fixed in release 8.0.0.0

ID #	Description
1231021	Avoid an OTBR assert by recovering the RCP rather than passing unhandled transmit errors to the sub mac.
1242948	Removed spurious test asserts from zigbeed.
1244459	Fixed issue where MAC-retried Indirect transmissions via RCP can result in a source match table entry for child being removed despite messages pending.
1245988	Fixed an issue where Zigbeed did not restart when performing a Trust Center Backup and Restore Reset Node action.
1282264	Fixed issue that could have interrupted radio transmit operations by clearing the transmit fifo prematurely, causing underflow.
1288653	Zigbee/OT/BLE SOC app will now print connection info upon receiving the CLI command "plugin ble gap print-connections".
1292537	Zigbee/BLE NCP application is now properly showing up in Simplicity Studio UI.
1252365	Added Coexistence plugin back into Zigbee NCP/OpenThread RCP sample application.
1293853	Reduced RAM footprint for zigbee_ncp-ot_rcp-spi and zigbee_ncp-ot_rcp_uart on MG21.
1124140	Fixed an issue where setting SL_OPENTHREAD_RADIO_RX_BUFFER_COUNT to a value other than 1 caused z3-light_ot-ftd_soc to not send Zigbee beacons after the OpenThread network is up.

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/simplicity-software-development-kit.

ID #	Description	Workaround
937562	Bluetoothctl ‘advertise on’ command fails with rcp-uart-802154-blehci app on Raspberry Pi OS 11.	Use btmgmt app instead of bluetoothctl.
1074205	The CMP RCP does not support two networks on the same PAN id.	Use different PAN ids for each network. Support is planned in a future release.
1122723	In a busy environment, the CLI may become unresponsive in the z3-light_ot-ftd_soc app.	No known workaround.
1209958	The ZB/OT/BLE RCP on MG24 and MG21 can stop working after a few minutes when running all three protocols.	Will be addressed in a future release.
1221299	Mfglib RSSI readings differ between RCP and NCP.	Will be addressed in a future release.

7.5 Deprecated Items

None.

7.6 Removed Items

Removed in release 8.0.0.0

None.

8 Using This Release

This release contains the following:

- Zigbee stack
- Zigbee Application Framework
- Zigbee Sample Applications

For more information about Zigbee and the EmberZNet SDK see *UG103.02: Zigbee Fundamentals*.

If you are a first-time user, see *QSG180: Zigbee EmberZNet Quick-Start Guide for SDK 7.0 and Higher*, for instructions on configuring your development environment, building and flashing a sample application, and documentation references pointing to next steps.

8.1 Installation and Use

The Zigbee EmberZNet SDK is provided as part of the Simplicity SDK, the suite of Silicon Labs SDKs. To quickly get started with the Simplicity SDK, install Simplicity Studio 5, which will set up your development environment and walk you through Simplicity SDK 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, Simplicity SDK may be installed manually by downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/simplicity_sdk for more information.

Simplicity Studio installs the Simplicity SDK by default in:

- (Windows): C:\Users\<NAME>\SimplicityStudio\SDKs\simplicity_sdk
- (MacOS): /Users/<NAME>/SimplicityStudio/SDKs/simplicity_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/.

8.2 Security Information

Secure Vault Integration

For applications that choose to store keys securely using the Secure Key Storage component on Secure Vault-High parts, the following table shows the protected keys and their storage protection characteristics that the Zigbee Security Manager component manages.

Wrapped Key	Exportable / Non-Exportable	Notes
Network Key	Exportable	
Trust Center Link Key	Exportable	
Transient Link Key	Exportable	Indexed key table, stored as volatile key
Application Link Key	Exportable	Indexed key table
Secure EZSP Key	Exportable	
ZLL Encryption Key	Exportable	
ZLL Preconfigured Key	Exportable	
GPD Proxy Key	Exportable	Indexed key table
GPD Sink Key	Exportable	Indexed key table
Internal/Placeholder Key	Exportable	Internal key for use by Zigbee Security Manager

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.

User applications never need to interact with the majority of these keys. Existing APIs to manage Link Key Table keys or Transient Keys are still available to the user application and now route through the Zigbee Security Manager component.

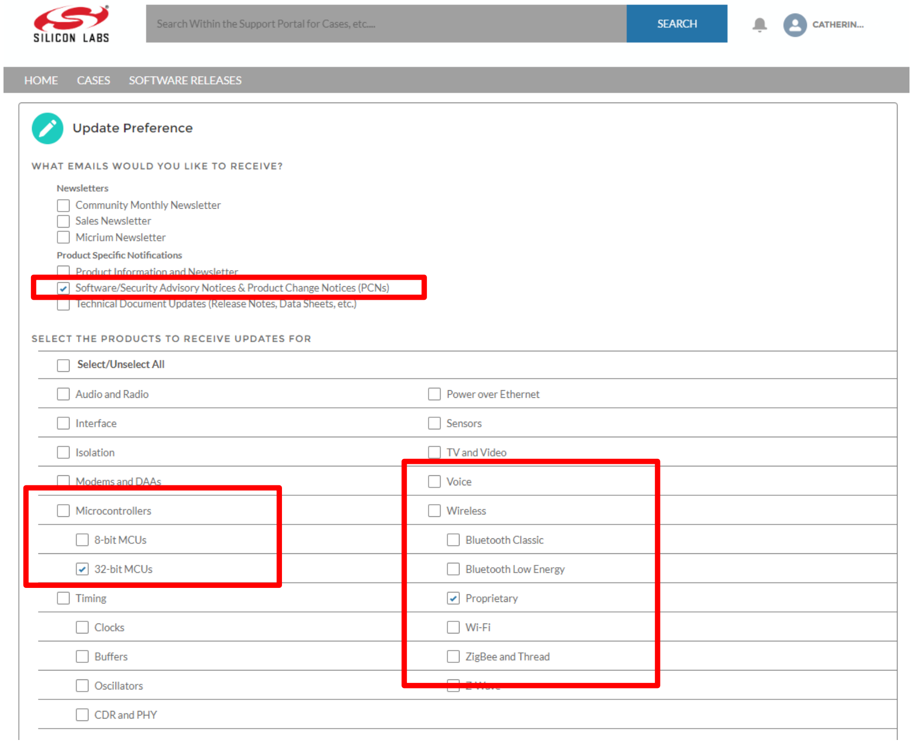
Some of these keys may become non-exportable to the user application in the future. User applications are encouraged to not rely on the exporting of keys unless absolutely necessary.

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.

The following figure is an example:



Update Preference

WHAT EMAILS WOULD YOU LIKE TO RECEIVE?

Newsletters

□ Community Monthly Newsletter
□ Sales Newsletter
□ Micrium Newsletter

Product Specific Notifications

□ Product Information and Newsletter
☑ Software/Security Advisory Notices & Product Change Notices (PCNs)
□ Technical Document Updates (Release Notes, Data Sheets, etc.)

SELECT THE PRODUCTS TO RECEIVE UPDATES FOR

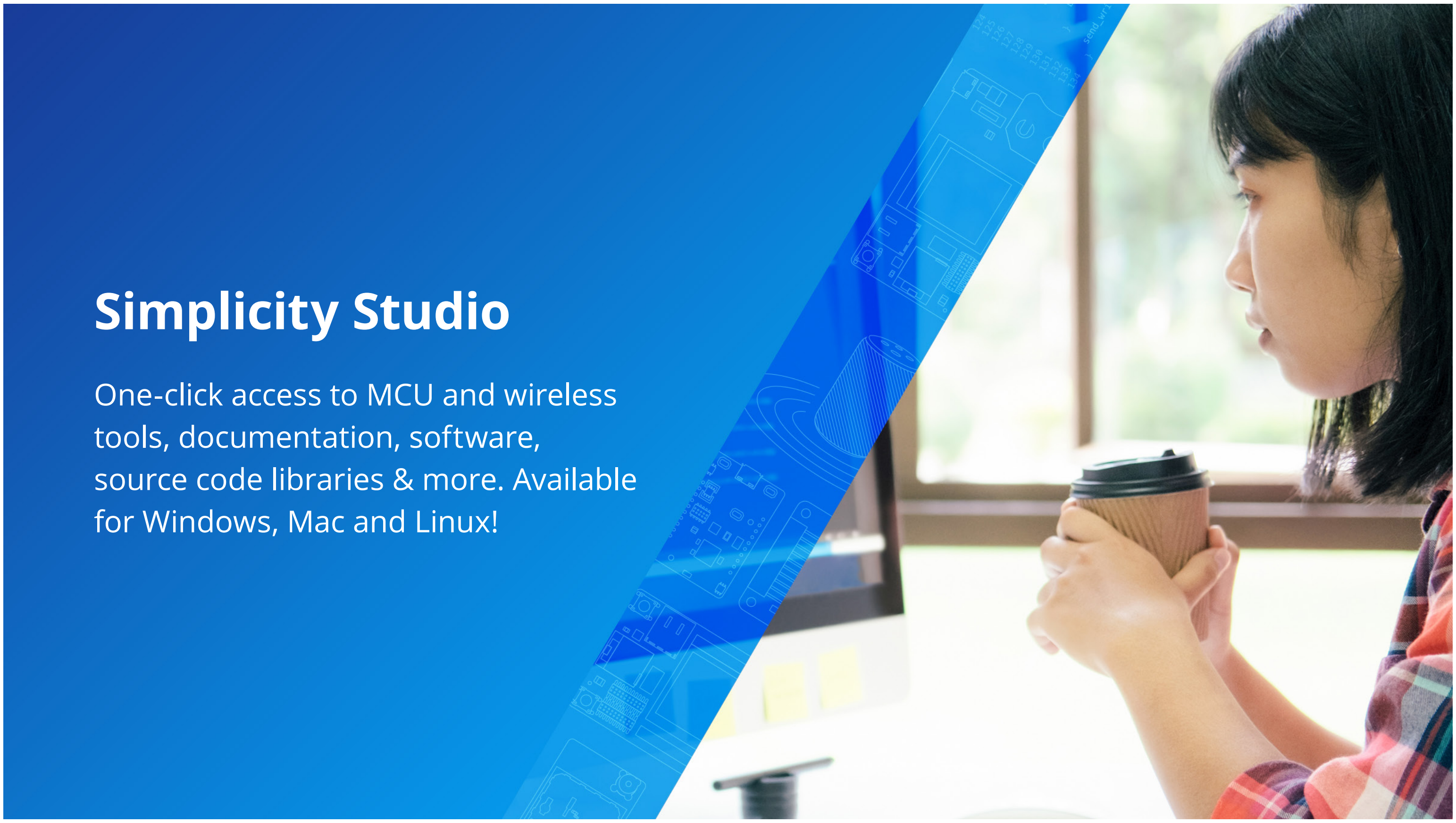
□ Select/Unselect All

□ Audio and Radio
□ Interface
□ Isolation
□ Modems and DAAs
□ Microcontrollers
  □ 8-bit MCUs
  ☑ 32-bit MCUs
□ Timing
  □ Clocks
  □ Buffers
  □ Oscillators
  □ CDR and PHY

□ Power over Ethernet
□ Sensors
□ TV and Video
□ Voice
□ Wireless
  □ Bluetooth Classic
  □ Bluetooth Low Energy
  ☑ Proprietary
  □ Wi-Fi
  □ ZigBee and Thread
  □ Z-Wave

8.3 Support

Development Kit customers are eligible for training and technical support. Use the Silicon Laboratories Zigbee web page to obtain information about all Silicon Labs Zigbee 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!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications. **Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project**

Trademark Information
Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals®, WiSeConnect , n-Link, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, Precision32®, Simplicity Studio®, Telegesis, the Telegesis Logo®, USBXpress® , Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com