

# Thread SDK 2.7.1.0 GA

Silicon Laboratories, Inc.

July 14, 2018

## 1 Release Highlights

### 1.1 Version 2.7.1.0 GA:

- GRL Test Harness support bug fixes
- Minor bug fixes related to sleep, child update request and network data synchronization
- Various bug fixes

### 1.2 Version 2.7.0.0 GA:

- Thread with Micrium OS Beta support
- DotDot DTLS security support
- DotDot resource directory and discovery support
- DotDot conformance testing support
- IAR EWARM 7.80.4 and GCC 7.2.1 support
- GRL Test Harness support bug fixes
- Various bug fixes

## 2 Using This Release

This release contains the following

- Silicon Labs Thread stack
- Constrained Application Protocol (CoAP) API for Thread
- Application Framework for Thread
- Application Framework for ZCL/IP
- Thread Sample Applications

For more information about the Thread SDK see UG103.11: Application Development Fundamentals: Thread.

If you are a first-time user, see QSG113: Getting Started with Silicon Labs Thread, for instructions on installing and configuring your development environment, building and flashing a sample application, and documentation references pointing to next steps.



For information on the Silicon Labs Thread Border Router Kit, how to set it up with an existing EM35x Development Kit, and how to demonstrate an example application, see QSG102: Thread Border Router Add-On Kit Quick Start Guide.

For information on how to build your own border router applications, see UG116: Developing Custom Border Router Applications.

## 2.1 Installation:

Installation of the Thread SDK is covered in the document QSG113.

## 2.2 Compatible Software:

The Thread SDK should be used in conjunction with the Silicon Labs Simplicity Studio V4 development platform. Simplicity Studio ensures that most software and tool compatibilities are managed correctly. Install software and board firmware updates promptly when you are notified.

This version of the Thread SDK is compatible with the following tool chains.

- IAR Embedded Workbench for ARM (IAR-EWARM) version 7.80.4. Follow instructions in QSG113: Getting Started with Silicon Labs Thread, to download from the support portal, and obtain a 30 day trial license.
- GCC (The GNU Compiler Collection) version 7.2.1 is provided with Simplicity Studio.

## 2.3 Support:

Development Kit customers are eligible for training and technical support. You can use the Silicon Laboratories web site <http://www.silabs.com/thread> to obtain information about all Silicon Labs Thread products and services, and to sign up for product support.

You can contact Silicon Laboratories support at <http://www.silabs.com/support>

# 3 Added Items

## 3.1 Version 2.7.1.0 GA:

### 3.1.1 New Plugins:

There are no new plugins in this release.

### 3.1.2 New APIs:

The following is a list of new APIs included with this release. A brief description is given for each. For additional documentation please refer the Thread API Reference Guide.

- `halCommonGetInt64uMillisecondTick`  
Returns a 64-bit integer that contains the number of millisecond ticks since system startup.

## 3.2 Version 2.7.0.0 GA:

### 3.2.1 New Plugins:

The following is a list of new plugins included with this release. A brief description is given for each. For additional information refer to the plugin description within Simplicity Studio and the "Silicon Labs Thread Framework README".

- **DHCP vNCP Library**  
This library includes DHCP functionality for the vNCP.
- **DHCP Stub Library for vNCP**  
Stub library for DHCP.
- **HAL vNCP Library**  
Common functionality for virtual network coprocessors (vNCPs).
- **Manufacturing Library for vNCP**  
This library provides support for various manufacturing test APIs.
- **Micrium RTOS**  
This plugin sets up Micrium OS and runs the application within a task.
- **Thread vNCP Stack Library with Silicon Labs TLS support**  
This provides the Thread networking layer, with support for mbed TLS crypto, and SSL/DTLS done via Silicon Labs TLS code.
- **vNCP Library**  
This plugin provides the ncp-side functionality for a Thread virtual NCP application. It runs the Thread stack within a Micrium OS Task separate from the App Framework task. The two tasks communicate just like an NCP and a host, but via Micrium OS queues instead of a serial port.
- **ZCL Core DTLS Manager CLI**  
This plugin provides CLI for DTLS Session Manager.

### 3.2.2 New APIs:

The following is a list of new APIs included with this release. A brief description is given for each. For additional documentation please refer the Thread API Reference Guide.

- **emberZclOtaBootloadIsWildcard**  
Checks if all bytes are 0xFF
- **emberGetPtaPwmOptions**  
Fetches packet traffic arbitration PWM options
- **emberGetPtaPwmOptionsReturn**  
Provides the result of a call to emberGetPtaPwmOptions
- **emberSetPtaPwmOptions**  
Configures packet traffic arbitration PWM options
- **emberSetPtaPwmOptionsReturn**  
Provides the result of a call to emberSetPtaPwmOptions

## 4 Changed Items

### 4.1 Version 2.7.1.0 GA:

There are no changed public APIs in this release.

### 4.2 Version 2.7.0.0 GA:

#### 4.2.1 Changed APIs:

- The "longId" field in EmberRipEntry has been renamed to macExtendedId.

#### 4.2.2 Other Changes:

- SysTick\_Handler has been changed to WEAK in hal-library to allow customer-supplied code (or the RTOS code) to supply their own SysTick ISR routine. While we have previously recommended that customer-supplied code should hook the SysTick interrupt by defining a "halSysTickIsr" function, after this change, we now recommend that customer-supplied code should hook the SysTick interrupt by defining a "void SysTick\_Handler(void)" ISR function.
- EFR32 IAR, EFR32 GCC, and EM3xx IAR applications now use the C startup routines found in platform/Device rather than those found in platform/base.
- Added priority escalation and the ability to use a PWM signal for request to the coexistence plugin.

## 5 Deprecated Items

### 5.1 Version 2.7.1.0 GA:

There are no deprecated public APIs in this release.

### 5.2 Version 2.7.0.0 GA:

There are no deprecated public APIs in this release.

## 6 Removed Items

### 6.1 Version 2.7.1.0 GA:

No plugins or public APIs have been removed in this release.

### 6.2 Version 2.7.0.0 GA:

#### 6.2.1 Removed Plugins:

The following plugin has been removed in this release.

- DTLS CLI  
Please use ZCL Core DTLS Manager CLI instead, along with ZCL application layer.

#### 6.2.2 Removed APIs:

No public APIs have been removed in this release.

## 7 Fixed Issues

### 7.1 Version 2.7.1.0 GA:

- 330229 Thread Test Harness: Network Data Synchronization issue after border router rejoin
- 313726 Thread Test Harness: Unexpected Server Data Notification (a/sd) messages from FTD
- 330488 Changes to UART behavior caused there to be a delay in receiving CLI commands in applications which include the idle-sleep plugin. An option has been added to the serial module in halconfig called "Enable wake from idle", which is set to True by default. As long as this option is True this should restore responsiveness.

## 7.2 Version 2.7.0.0 GA:

- 267572 The "longId" field in EmberRipEntry is too generic. It has been renamed to macExtendedId.
- 258485 Host code that writes to log files now conforms to linux-style line endings.
- 297804 ip-driver-app now reports an error rather than asserting when the application is sending a packet greater than EMBER\_IPV6\_MTU.
- 285563 platform/base/phy/phy.h is no longer included in the stack install.
- 304228 Support added for Max Child Timeout Diagnostic TLV
- 298111 An error is now printed when a bad length value is given to the "network-management set-vendor-info" CLI command rather than asserting.
- 297262 Address cache is now updated when 6LoWPAN contexts change
- 297255 The "on-mesh" flag in Border Router TLVs is now fully supported.

## 8 Open Issues

### 8.1 Version 2.7.1.0 GA:

- 331945 In the creation of new SL-Thread apps which use the zcl-core plugin there will be an incorrect upgrade rule displayed telling the user that the dtls-auth-params plugin has been enabled. This does not actually change anything, as this plugin is already enabled in all apps which use the zcl-core plugin.

### 8.2 Version 2.7.0.0 GA:

- 267704 NCP SPI Link Plugin is empty...
- 283490 Issues with vUART over SWO
- 260589 ip-driver-app seems to log messages regardless of "-l" options
- 281231 Enabling Serial 3 or USB functionality on EM358x and EM359x may cause memory management faults and other errors. As EM358x and EM359x USB support has been deprecated, please ensure that Serial 3 and USB functionality are disabled.
- 277510 nWAKE handshake does not complete between EM35x SPI NCP and host, causing the host to assert.
- 251287 To achieve the lowest current during sleep on EFR32xG12, EFR32xG13, and EFR32xG14 parts, you must turn on voltage scaling. However, the radio will not operate with voltage scaling turned on, so to turn it on you must also make sure to disable it after each wake-up. Furthermore, some resets will not turn off voltage scaling, so please ensure that it is disabled before attempting to turn on the radio. Note that there is a ramp when turning voltage scaling on or off, so enabling this feature may increase the time it takes to go to sleep or wake up.
- 313709 Add support for specifying/retrieving hop limit values for multicast CoAP UDP Messages
- 313577 The sensor-actuator sample application is not displaying info on WSTK display
- 311990 Remove prototype and documentation for emberSetRadioChannel. It is defined as a public API but is not supported.
- 277875 Network Analyzer does not decrypt messages sent within a DTLS session.
- 294835 OTA bootload is intermittently unable to detect the presence of the OTA server

## 9 Intended Behavior

### 9.1 Version 2.7.0.0 GA and 2.7.1.0 GA:

- Pre-built Thread NCP UART binaries with Software (XON/XOFF) flow control have been removed as the WSTK does not support software flow control through the USB serial interface. However, software flow control is still supported for Thread NCP UART and may be enabled with a plugin option in the NCP UART Link plugin of the Thread framework.
- Customers who have previously used Coexistence functionality in their EFR32 application should review the Coexistence HW Configurator peripheral to ensure that the settings match their application functionality.
- The ip-driver-app can't detect and recover if the length field of an IPv6 packet written to the TUN driver is corrupt. The sender of the packet should check to make sure the IPv6 length matches the length of the packet being sent. Added an assert to verify that the length does not exceed EMBER\_IPV6\_MTU (1280), which will catch length field corruption some of the time.
- Prior implementations of the light sample application included a temporary implementation of emberZclGetPublicKeyCallback. That callback is now implemented by the dtls-auth-params plugin which is automatically included for all ZCL based applications. What this means is that if you are importing an older version of the light sample application or more specifically the light-implementation.c file you will need to remove the implementation of emberZclGetPublicKeyCallback from that file. This also means that in the unlikely case that you have created your own application and have an implementation of emberZclGetPublicKeyCallback in your application you will need to compare your implementation to the one in the dtls-auth-params plugin and decide which version to keep. If you choose to use your implementation then you should disable the dtls-auth-params plugin. If you choose to use the dtls-auth-params plugin's implementation then you should remove your implementation from your application code.
- 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. The workaround is to modify these functions to use clock\_gettime with the CLOCK\_MONOTONIC source instead.
- Changes to UART behavior caused there to be a delay in receiving CLI commands in applications which include the idle-sleep plugin. To restore responsiveness, enable the "wake from activity on RX GPIO of application port" serial option.

## 10 Documentation Changes

### 10.1 Version 2.7.1.0 GA:

No documents have been added, modified, or removed in this release.

### 10.2 Version 2.7.0.0 GA:

#### 10.2.1 New Documents:

- AN1141: Thread Mesh Network Performance
- AN1143: Using Micrium OS with Silicon Labs Thread
- AN1142: Mesh Network Performance Comparison
- AN1154: Using Tokens for Non-Volatile Data Storage
- UG103.07: Non-Volatile Data Storage Fundamentals

### 10.2.2 Modified Documents:

- AN1115: Configuring Peripherals for 32-Bit Devices in Simplicity Studio
- AN1136: Using the Thread Test Harness in Auto Device Under Test Mode
- AN703: Using the Simulated EEPROM Version 1 and Version 2 for the EM35x and EFR32 SoC Platforms
- QSG113: Getting Started with Silicon Labs Thread
- UG162: Simplicity Commander Reference Guide
- UG266: Silicon Labs Gecko Bootloader User's Guide
- UG278: Zigbee Cluster Library over IP Users Guide

### 10.2.3 Removed Documents:

- AN1031: Thread Large Network Scalability

## 11 History

### 11.1 Version 2.6.2.0 GA:

- 298193 Fix issue where network diagnostics messages sent from a host application were being sent from an incorrect source port.
- 298229 Address issue with certification tests 5.3.8 5.3.9 and the new thread-test-harness-cli plugin.
- 298358 Address issue with certification tests 5.7.1 and the new thread-test-harness-cli plugin.
- 301031 Address issue with certification tests 8.3.1, 9.2.2, 9.2.4, 9.2.19 and the new thread-test-harness-cli plugin.
- 301014 Address issue with certification test 5.7.3 and the new thread-test-harness-cli plugin.
- 302078 Fix misleading MGMT\_GET response log message.
- 302081 Fix incorrect source address selection when responding to a network diagnostic multicast request.
- 302082 Ensure that SoCs loopback multicast network diagnostic requests.
- RAIL\_LIB-1928 Fix issue with truncated captured packets in network analyzer.

### 11.2 Version 2.6.1.0 GA:

- 299748 Add support for EFR32MG12P433F1024GM68, EFR32MG12P432F1024GM68, and EFR32MG12P232F1024GM68

### 11.3 Version 2.6.0.0 GA:

- 280224 Thread ZCL OTA bootload does not complete when the target device is a SED
- 276838 ping6 from host gives "wrong data byte" sometimes
- 274796 Simplicity Commander is not aware of TOKEN\_MFG\_THREAD\_JOIN\_KEY
- 269799 Add CoAP Diagnostic Queries from the host
- 263365 Mapping: getting a list of all nodes in network

- 280040 OTA Bootload plugins will not compile without the ZCL cluster enabled. If you receive the error, "'EMBER\_ZCL\_CLUSTER\_OTA\_BOOTLOAD" is undefined', you will need to edit your device's configuration. Please visit the 'ZCL over IP' tab of your configuration file. Under the general cluster section, enable Over the definitionAir Bootloading.
- 197474 ip-driver doesn't correctly setup IPv6 multicast
- 278882 Occassionally sleepy end node becomes orphaned after joining network
- 296396 The legacy Ember one-hop OTA bootloader has been deprecated. For new products which require OTA updating, please use either an Ember application or Gecko storage bootloader in conjunction with the storage/update mechanism of the chosen stack.
- 246164 Can't create ZCL binding when creating a reporting config
- 227955 The RNG algorithm has been upgraded on EM35x parts as the previous method could result in slight correlations. Please see platform/base/phy/em3xx/phy.h for further information.
- 290851 Several compiler warnings have been addressed.
- 292916 On the EFR32, updates to the HFXO CTUNE value will only be applied to HFXOSTEADYS-TATECTRL, with HFXOSTARTUPCTRL left unchanged.
- 275821 FEM Control plugin's GPIO usage should be managed by Hardware Configurator.
- 286972 Combine FEM control/LNA plugins

#### 11.4 Fixed issues in 2.5.0:

- 281491 PSKC is not properly propagated when using the external commissioning application with multiple border routers.
- 282097 Sleepy reference designs applications (occupancy, contact and dimmer) fail to receive a GUA after reset.
- MCUDT-13581 Absolute path in generated post-build batch file prevents it from being easily moved.
- 271100 Update default power values for reference designs applications.
- 275781 "Sensor Actuator Node is up" does not need to be printed periodically.
- 266389 ZCLIP discovery payload should be link-format+CBOR.
- 281257 Remove OTA server functionality from the light host sample application.
- DEV-1656 Add support for handling Child Update Requests from Parent.
- 280809 Use MAC extended id in Address Solicit / Release instead of EUI64.
- 281096 ZCLIP Profile IDs and Device IDs need to be updated.
- 279589 Update how nodes reattach to a network partition after losing connectivity to its parent.
- 278403 ZCLIP notification payload should include the Binding ID of the binding on the server for which the notification is sent to the client.
- 273399 Remove assert on OTA "addConfiguration" when reporting config table is full and instead send a response error code 500 with payload: status = INSUFFICIENT\_SPACE.
- 265767 ZCLIP OTA server code should automatically detect when new images are available.
- 223261 FEM plug-in description updates, improvements.



- 288709 Added USART3 defines to plugin/serial/com\_vcom.h.
- 279860 Coexistence: GRANT denial shows incorrect REQUEST deassert/re-assert behavior.
- 284767 HAL\_HAS\_INT64 is now defined for GCC Cortex-M builds, properly indicating the platform's support for 64-bit integers. A side effect of this is that GCC-built EmberZNet applications can now define an end device child table larger than 32.

### 11.5 Fixed issues in 2.4.0:

- 279374 PSKc TLV not included in active operational dataset
- 279368 Thread ip-driver-app assert: IP\_MODEM\_READ\_FORMAT\_ERROR
- 279743 OTA bootload client hits an assert if an OTA bootload server is not running
- 280040 Cannot compile OTA bootload plugins without ZCL cluster
- 279361 Thread GCC - Usage Fault hit joining network: CFSR.UNALIGNED
- 279112 Thread CoAP - Address cache not updated as first CoAP fails
- 277675 DNS CLI Plugin needs to be enabled on all reference designs
- 277865 When configuring gateway with net-management cli the border-router incorrectly moves through the state machine
- 277677 Add connection manager callback status to all reference designs
- 275566 Register all multicast addresses in the border router app needed for certification
- 279893 Address issue with GRL Thread test-harness test 5.3.2
- 263033 As a SED we are occasionally setting Frame Pending in the 802.15.4 ACK to TRUE
- 194945 Get all responses for CoAP GET to multicast address
- 198951 Callback or Handler for J\_PAKE decryption during join by discovery
- 221454 CBOR Encoder/Decoder Needs EMBER\_ZCLIP\_TYPE\_BINARY Support for Scenes
- 221460 CBOR Decoder Doesn't Handle Strings Correctly
- 222576 APIs for Sending to Bindings
- 224061 Group name support
- 227145 Add callback for each 802.15.4 packet so application can obtain last hop RSSI
- 233431 Add loopback for CoAP API to make consistent with UDP API
- 239914 Many header files don't have protection against multiple inclusion
- 241552 Add explanation to documentation on Return functions
- 242132 .well-known responses should include our UID
- 245075 Bounded signed attributes are broken
- 246165 Report configurations are not persisted
- 249392 Add server discovery functionality to ota-bootload-client plugin
- 249838 mfglib.h API documentation for needs to describe throttling logic for mfglibRxCallback

- 252875 Border router encounters memory fault
- 253119 Add adaptivity test mode for mfglib
- 255021 Support for Proxy Transmit Receive UDP Notification
- 255049 Do not include the idle/sleep plugin by default for SOC device type
- 255636 Update CBOR implementation to only support definite objects
- 255719 UIDs should now be encoded in base64url
- 255720 Validate Accept and Content-Format CoAP options when processing ZCLIP frames
- 255863 Generate maximum size of reportable attribute data
- 257858 Reset Reporting Configurations by endpointId
- 259957 CLI command load-ota-image should provide some indication that it is processing information rather than appearing hung
- 260251 tun0 interface fails to get a global address after an assert in the border-router-mgmt host app
- 260532 GCC Compiler support for Thread
- 261040 emberConfigureGateway is unclear with a change in parameter list - newBorderRouterFlag
- 261954 ip-driver-app is unable to reset the NCP
- 262991 Update reporting to be in line with the r30 specification
- 263366 Thread sample apps should not stub out DHCP
- 263398 local ip descriptions too minimal in "info" CLI command
- 264519 Broadcast Group ID changed from 0 to 0xffff
- 265117 Add PDF version of doxygen to stack release
- 265264 Continual OTA bootloading of a file the device has already successfully received
- 265270 Child Table TLV returns incorrect results for CoAP Diagnostic
- 265278 Support Mac Counters TLV in CoAP Diagnostic
- 265627 Stack asserts when sent empty CoAP URI.
- 265806 Restrict ip-driver-app to listen only on local interfaces
- 266031 ipDriverTick causes high CPU load when output buffer is full
- 266391 Preserve CBOR decoding error status to report proper ZCLIP error response
- 266603 Implement Set source address at application layer on SoC Gateway
- 267648 Correct handling of Report Configuration mandatory 'n' and 'x' parameters
- 267649 Undivided write is not implemented
- 267651 Wrong ZCLIP response returned for invalid attribute data type
- 267652 Ignore unrecognized command parameters in a ZCLIP request
- 267577 post-build.bat fails in xNCP Thread project

- 269124 Dynamically pull cluster data from xml
- 269506 Reporting Configuration takes too long to delete configs for multiple endpoints
- 269587 Support variable length ZCLIP text and binary strings
- 269665 Commissioning NCP/border-router to a network with Android app hangs during join
- 271077 Non OTA server responds to discovery
- 272515 Get OTA to bootstrap various images
- 273602 Get version of running image
- 273788 Remove commission-tester.c from distributed files in SDK
- 275566 Registering multicast addresses for the border router app.

### 11.6 Fixed issues in 2.3.1:

- 259956 Generated bootstrap image is too large for OTA bootloading targets to receive
- 259958 OTA bootloading support should exist in the built in sample apps for thread
- 260021 OTA Bootstrap Storage EEPROM plugin does not correctly limit compatible architectures
- 260023 OTA Bootstrap Server Policy plugin has "Allows Downgrades" plugin option listed twice
- 260024 After receiving a bad block transfer, OTA Bootstrap Client doesn't reschedule next block request
- 260871 dns-client was not put in to the Q2 SDK release
- 260932 Border router sends commission messages to 2 different UDP ports
- 264678 When OTA client misses packet, it incorrectly calculates the next block number
- 264733 Once an bootstrap fails to complete, the partial file must be erased from flash for bootloading to later succeed
- 265203 erasing bootloaded images is not recognized until after a reset
- 265439 OTA bootloading with an existing image in flash causes assertion: [ASSERT:ota-bootstrap-client.c:338]

### 11.7 Fixed issues in 2.3.0:

- Store and check MLE frame counters.
- The fallbackChannelMask argument of the emberCommissionNetwork API is now fully implemented.
- DTLS sessions can now be created and used at application layer.
- The ip-driver-app no longer terminates under high stress and instead now exerts flow control back to the IP stack.
- emberLeaderDataHandler is now called when a node becomes a leader. Previously this was only called when the node learned another node had become a leader.
- Fixed an issue with UDP port compression that was preventing some UDP ports from being used.
- Fixed an issue where emberStartScan was not getting a return when called after emberJoinNetwork or emberJoinCommissioned.
- Fixed an issue where a sleepy end device when reattaching to a network could lose its sleepy status.
- Fixed an issue that was causing a packet to show up with a Malformed CoAP option in Wireshark.

## 11.8 Fixed issues in 2.2.1:

- Fixed an issue where the first byte of the mfglib test packet was being dropped when the packet was passed from the NCP to the host.
- The border router now clears the global IP addresses on the host after a network reset on NCP.

## 11.9 Fixed issues in 2.2.0:

- The CoAP API has been significantly improved. Please refer to the notes within the "Detailed Description" section of the Coap module in the doxygen-generated documentation which is rooted at **THREAD\_HOME**/documentation/Thread-Doxygen/index.html. These notes provide a brief overview of the key parts of the API as well as tips on migrating from the old API to the new API. The API is also fully documented in the C header file, **THREAD\_HOME**/stack/include/coap.h. Here is a list of some of the improvements.
  - Internal CoAP functions have been removed from stack/include/coap.h.
  - CoAP responses can now be sent outside of the incoming message handler.
  - Applications will now receive all responses to requests sent to a multicast address.
  - Separate responses, as described in section 5.2.2 of RFC-7252, can now be sent using the improved CoAP API.
- The client and client-sleepy sample applications have been modified such that joining will only occur after a button press or after invoking the new join CLI command. In addition, these applications now generate a random join key and store it in the new **THREAD\_JOIN\_KEY** manufacturing token.
- The host ip-driver-app has been modified to work on Linux Kernel 4.4.16.
- A new application framework plugin, thread-test-harness-cli, has been added. This plugin provides all of the CLI required for testing against the GRL Thread Test Harness. This plugin is currently in beta, and is not guaranteed to pass all tests when run against the GRL Thread Test Harness.
- Compression of the link local multicast address has been corrected.
- In addition the the CoAP API changes described previously the following APIs have also changed.
  - New counters introduced.
    - \* **EMBER\_COUNTER\_PTA\_LO\_PRI\_REQUESTED** - The number of times a low priority packet traffic arbitration request has been made.
    - \* **EMBER\_COUNTER\_PTA\_HI\_PRI\_REQUESTED** - The number of times a high priority packet traffic arbitration request has been made.
    - \* **EMBER\_COUNTER\_PTA\_LO\_PRI\_DENIED** - The number of times a low priority packet traffic arbitration request has been denied.
    - \* **EMBER\_COUNTER\_PTA\_HI\_PRI\_DENIED** - The number of times a high priority packet traffic arbitration request has been denied.
    - \* **EMBER\_COUNTER\_PTA\_LO\_PRI\_TX\_ABORTED** - The number of times a low priority packet traffic arbitration transmission has been aborted.
    - \* **EMBER\_COUNTER\_PTA\_HI\_PRI\_TX\_ABORTED** - The number of times a high priority packet traffic arbitration transmission has been aborted.
  - New APIs for handling network data. These are described in more detail in the C header file located at **THREAD\_HOME**/stack/include/network-management.h.
    - \* emberGetNetworkData
    - \* emberGetNetworkDataReturn
    - \* emberNetworkDataChangeHandler

### 11.10 Fixed issues in 2.1.1:

- The default child timeout for sleepy end devices was changed from 300 seconds to 240 seconds. This aligns with the Thread 1.1 specification.
- Added new API `emberIsIpv6UnspecifiedAddress()` which checks if a given `EmberIpv6Address` is set to all zeroes which represents an unspecified address.
- When `emberGetRipEntry()` is called with an `0xFF` index to request all valid RIP table entries the stack will return valid entries through calls to `emberGetRipEntryReturn()` as it always has but now once all valid entries have been returned an extra zeroed-out entry is returned to indicate completion.
- When `emberGetGlobalPrefixes()` is called to request the list of global prefixes that we know about the stack will return valid entries through calls to `emberGetGlobalPrefixReturn()` as it always has but now once all valid entries have been returned an extra zeroed-out entry is returned to indicate completion.
- When `emberGetDhcpClients()` is called to request the list of DHCP clients that we know about the stack will return valid entries through calls to `emberGetDhcpClientReturn()` as it always has but now once all valid entries have been returned an extra zeroed-out entry is returned to indicate completion.
- When `emberGetGlobalAddresses()` is called to request list of global addresses configured on this device the stack will return valid entries through calls to `emberGetGlobalAddressReturn()` as it always has but now once all valid entries have been returned an extra zeroed-out entry (an IPv6 unspecified address) is returned to indicate completion.
- Fixed an issue introduced in Silicon Labs Thread 2.1.0 where the stack was dropping data requests with security enabled and thus preventing some sleepy end devices from sleeping.

### 11.11 Fixed issues in 2.1.0:

- Added new API `emberGetPtaOptions()` and corresponding callback `emberGetPtaOptionsReturn()` to get packet traffic arbitration configuration options.
- Added new API `emberSetPtaOptions()` and corresponding callback `emberSetPtaOptionsReturn()` to configure packet traffic arbitration options.
- The following API functions have been renamed:
  - `emberSendJoinerEntrust` to `emberSendEntrust`
  - `emberSetBeaconSteeringData` to `emberSetSteeringData`

### 11.12 Fixed issues in 2.0.0:

- DHCP is now an optional component within the Silicon Labs Thread stack. To include support for DHCP applications must link against `dhcp-library.a`. To exclude support for DHCP and thus save flash space applications should link against `dhcp-stub-library.a`.
- SW flow control is now supported between a host and NCP when connected through a UART interface. To enable, define `EMBER_APPLICATION_USES_SOFTWARE_FLOW_CONTROL` and `EMBER_SERIAL1_XONXOFF`.
- Added new API `emberSetRadioHoldOff()` and corresponding callback `emberSetRadioHoldOffReturn()` which gives both host and SoC applications the ability to enable or disable radio holdoff. When radio holdoff is enabled it configures `RHO_GPIO` in `BOARD_HEADER` as an input which, when asserted, will prevent the radio from transmitting. When radio holdoff is disabled it configures `RHO_GPIO` for its original default purpose.
- Added new API `emberGetPtaEnable()` and corresponding callback `emberGetPtaEnableReturn()` to get whether packet traffic arbitration is enabled or disabled.

- Added new API `emberSetPtaEnable()` and corresponding callback `emberSetPtaEnableReturn()` to enable or disable packet traffic arbitration.
- Added new API `emberGetAntennaMode()` and corresponding callback `emberGetAntennaModeReturn()` to get the current antenna mode.
- Added new API `emberSetAntennaMode()` and corresponding callback `emberSetAntennaModeReturn()` to set the current antenna mode.
- Added new callback `emberCounterValueHandler()` which is invoked to query the application for the counter value of an event defined by the given `EmberCounterType`.
- An additional parameter, `isStable`, has been added to the `emberGetGlobalPrefixReturn()` callback.
- Added support for `EMBER_MINIMAL_END_DEVICE`, an always-on end device like `EMBER_END_DEVICE`, but IP address discovery is performed by the parent on its behalf to help it conserve resources.
- Added the following new APIs to convert IPv6 addresses and prefixes to and from strings.
  - `emberIpv6AddressToString()`
  - `emberIpv6PrefixToString()`
  - `emberIpv6StringToAddress()`
  - `emberIpv6StringToPrefix()`
- Added new API `emberRadioGetRandomNumbers()` and corresponding callback `emberRadioGetRandomNumbersReturn()` to obtain true random numbers.
- The following callback functions have been renamed:
  - `emberDhcpServerChange` to `emberDhcpServerChangeHandler`
  - `emberAddressConfigurationChange` to `emberAddressConfigurationChangeHandler`
  - `emberExternalRouteChange` to `emberExternalRouteChangeHandler`
  - `emberMarkApplicationBuffers` to `emberMarkApplicationBuffersHandler`
  - `emberSlaacServerChange` to `emberSlaacServerChangeHandler`
  - `emberDeepSleepCallback` to `emberDeepSleepCompleteHandler`
- Address prefix length is now in bits instead of bytes. The following APIs and callbacks are affected by this change:
  - `emberConfigureGateway()`
  - `emberConfigureExternalRoute()`
  - `emberGetGlobalPrefixReturn()`
  - `emberDhcpServerChangeHandler()`
  - `emberRequestDhcpAddress()`
  - `emberRequestDhcpAddressReturn()`
  - `emberSlaacServerChangeHandler()`
  - `emberRequestSlaacAddress()`
  - `emberRequestSlaacAddressReturn()`
  - `emberGetGlobalAddresses()`
  - `emberExternalRouteChangeHandler()`
- The syntax for specifying prefixes in CLI commands has changed. The format for a prefix is now the first address in the block, a slash, and a decimal value equal to the size of the prefix in bits (e.g., `fe80::/64`, `2001:db8:1234::/48`). The following CLI commands are affected by this change:

- network-management commission ...
- network-management form ...
- network-management gateway ...
- network-management global-addresses ...

### 11.13 Fixed issues in 1.0.7:

- Added a new API `emberEnableHostJoinClient()` so applications can run the commissioning state machine on the host instead of the NCP. The behavior of this API is documented in `stack/include/network-management.h`.

### 11.14 Fixed issues in 1.0.6:

- Added a new API `emberSetCtune()` and corresponding callback `emberSetCtuneReturn()` to change the CTUNE value. Involves switching to HFRCO and turning off the HFXO temporarily. (Only valid on EFR32.)
- Added a new API `emberGetCtune()` and corresponding callback `emberGetCtuneReturn()` to get the CTUNE value. (Only valid on EFR32.)
- Added support to build NCP applications with optional extensions to initialization, main loop processing, event definition and handling, and host/NCP commands.
- Added NCP SPI and NCP UART sample applications. It is recommended these be used as the starting point for building customized NCP applications.

### 11.15 Fixed issues in 1.0.5:

- Join failure argument added to `emberNetworkStatusHandler()` callback.
- Host callback function `emberStateReturn` will now always be called after a call to `emberState`. Host applications must now implement `emberStateReturn()`.
- Messages sent to the “All Mesh Nodes” multicast address (`ff03::1`) are no longer delivered to rx-off-when-idle Children. The “All Thread Nodes” address (`ff33:40:jula prefixj:1`) can be used to multicast to these devices.

### 11.16 Fixed issues in 1.0.4:

- Added EFR32 SoC and NCP support.
- Added border router sample application.
- Added support for NCPs over SPI.
- Added standalone bootloaders to the SPI and UART NCPs. The included NCP images now must be used with `ezsp-spi-bootloader` for SPI or `serial-uart-bootloader` for UART.
- Added a new API `emberGetStandaloneBootloaderInfo()` and corresponding callback `emberGetStandaloneBootloaderInfoReturn()` to obtain the version of the installed standalone bootloader as well as the platform, micro, and PHY of the NCP.
- Added a new API `emberLaunchStandaloneBootloader()` and corresponding callback `emberLaunchStandaloneBootloaderReturn()` to inform the NCP to launch the standalone bootloader.
- Added a new API `emberGetMfgToken()` and corresponding callback `emberGetMfgTokenReturn()` to obtain a manufacturer token.

- Added a new API `emberSetMfgToken()` and corresponding callback `emberSetMfgTokenReturn()` to set a manufacturer token.
- The way the network status is communicated to the application has been made more consistent. There is a new callback, `emberNetworkStatusHandler()`, that is called whenever the network status changes. The return callbacks for the `form`, `join`, `resume`, and `attach` commands now indicated only whether the process was initiated. There is a new network status type, `EMBER_JOINED_NETWORK_ATTACHING`, for use when the node is attaching. The name of `EMBER_JOINED_NETWORK` has also been changed to `EMBER_JOINED_NETWORK_ATTACHED`. The new callback behavior is documented in `stack/include/network-management.h`.
- The mbed TLS functionality required by the Silicon Labs Thread stack is now released as a separate library, `mbedtls-library.a`. This new library must be included when linking an application that includes the Silicon Labs Thread stack.
- Changed `emberCoapMessageHandler` to have a single parameter, a pointer to an `EmberCoapMessage` struct, which replaces the previous discrete parameters. `EmberCoapMessage` also replaced `EmberAfCoapDispatchRequest` in the CoAP Dispatch plugin.
- Added a `macExtendedId` parameter to `emberStateReturn`.

### 11.17 Fixed issues in 1.0.3:

- Changed some constant names in `stack/include/network-management.h` to reflect their actual purpose. `EmberJoiningMode`:
  - `EMBER_JOINING_NO_STEERING` is now `EMBER_JOINING_ALLOW_ALL_STEERING`
  - `EMBER_JOINING_CLEAR_STEERING` is now `EMBER_JOINING_ALLOW_EUI_STEERING`
  - `EMBER_JOINING_CLEAR_STEERING_SMALL_EUI64` is now `EMBER_JOINING_ALLOW_SMALL_EUI_STEERING`

Flag value for `emberCommissionerStatusHandler`:

- `EMBER_JOINING_WITH_STEERING` is now `EMBER_JOINING_WITH_EUI_STEERING`
- Added a new API `emberConfigureExternalRoute()` to define an external route set, which is a route for a Thread network IPv6 packet that must traverse a border router and be forwarded to an exterior network.
- Removed a `macExtendedId` field mistakenly added to the Silicon Labs Thread token struct in the 1.0.2 release. NOTE: this breaks token compatibility between 1.0.2 and 1.0.3, so existing networks cannot be upgraded in place and must be factory reset.
- Removed options and hop limit arguments to `emberSendUdp()` and its associated CLI commands.
- Various minor cleanup tasks.

### 11.18 Fixed issues in 1.0.2:

- Added a destination IP address as the second argument to `emberCoapMessageHandler()`.
- Fixed a bug in which a sleepy end device would change to a powered end device if it failed to join.
- Added version and steering fields to the `EmberMacBeaconData` struct.
- Fixed a problem in which GUA addresses were being lost on reset.



- When configuring a border router as a DHCP server:  
The preferred lifetime should lie between **EMBER\_MIN\_PREFERRED\_LIFETIME\_SEC** and **EMBER\_MAX\_LIFETIME\_DELAY\_SEC**.  
The valid lifetime should lie between **EMBER\_MIN\_VALID\_LIFETIME\_SEC** and **EMBER\_MAX\_LIFETIME\_DELAY\_SEC**.  
As always, preferred and valid lifetimes are ignored for SLAAC prefixes.
- Fixed multicast relay timing to conform to the Thread specification.
- Various other minor bug fixes found during testing.

#### 11.19 Fixed issues in 1.0.1.1:

- An include file problem, which prevented host applications from compiling, has been resolved.
- A bug has been fixed that caused multicast messages to silently fail when attempting to send from a host to the mesh.