The Proprietary Flex SDK is a complete software development suite for proprietary wire-
less applications. Per its namesake, Flex offers two implementation options.

The first uses Silicon Labs RAIL (Radio Abstraction Interface Layer), an intuitive and eas-
ily-customizable radio interface layer designed to support both proprietary and standards-
based wireless protocols.

The second uses Silicon Labs Connect, an IEEE 802.15.4-based networking stack de-
signed for customizable broad-based proprietary wireless networking solutions that re-
quire low power consumption and operates in either the sub-GHz or 2.4 GHz frequency
bands. The solution is targeted towards simple network topologies.

The Flex SDK is supplied with extensive documentation and sample applications. All ex-
amples are provided in source code within the Flex SDK sample applications.

These release notes cover SDK version(s):

3.6.0.0 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/flex-sdk-connect-networking-stack](https://www.silabs.com/developers/flex-sdk-connect-networking-stack). Silicon Labs also strongly
recommends that you subscribe to Security Advisories for up-to-date information. For instructions, or if you are new to the Silicon Labs
Flex 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.
## Contents

1. **Connect Applications** ................................................................. 1
   1.1 New Items ................................................................................. 1
   1.2 Improvements ......................................................................... 1
   1.3 Fixed Issues ........................................................................... 1
   1.4 Known Issues in the Current Release ...................................... 1
   1.5 Deprecated Items ................................................................. 1
   1.6 Removed Items ....................................................................... 1

2. **Connect Stack** .......................................................................... 2
   2.1 New Items ............................................................................. 2
   2.2 Improvements ......................................................................... 2
   2.3 Fixed Issues ........................................................................... 2
   2.4 Known Issues in the Current Release ...................................... 2
   2.5 Deprecated Items ................................................................. 2
   2.6 Removed Items ....................................................................... 2

3. **RAIL Applications** ................................................................. 3
   3.1 New Items ............................................................................. 3
   3.2 Improvements ......................................................................... 3
   3.3 Fixed Issues ........................................................................... 3
   3.4 Known Issues in the Current Release ...................................... 3
   3.5 Deprecated Items ................................................................. 3
   3.6 Removed Items ....................................................................... 3

4. **RAIL Library** ........................................................................... 4
   4.1 New Items ............................................................................. 4
   4.2 Improvements ......................................................................... 4
   4.3 Fixed Issues ........................................................................... 5
   4.4 Known Issues in the Current Release ...................................... 6
   4.5 Deprecated Items ................................................................. 6
   4.6 Removed Items ....................................................................... 6

5. **Using This Release** ................................................................. 7
   5.1 Installation and Use ............................................................... 7
   5.2 Security Information ............................................................. 7
   5.3 Support .................................................................................. 8
1 Connect Applications

1.1 New Items

**Added in release 3.6.0.0**

- XG27 Support
- NCP – Host solution
  - Host: Connect Host Sink CLI Application
  - NCP: Connect – NCP Application
  - Security and OTA Bootloader support

1.2 Improvements

None

1.3 Fixed Issues

None

1.4 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 the TECH DOCS tab on [https://www.silabs.com/developers/flex-sdk-connect-networking-stack](https://www.silabs.com/developers/flex-sdk-connect-networking-stack).

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>652925</td>
<td>EFR32XG21 is not supported for &quot;Flex (Connect) - SoC Light Example DMP&quot; and &quot;Flex (Connect) - SoC Switch Example&quot;</td>
<td></td>
</tr>
<tr>
<td>1076409</td>
<td>OTA Bootloader is not working on Series2</td>
<td></td>
</tr>
<tr>
<td>1139850</td>
<td>DMP instabilities with XG27</td>
<td></td>
</tr>
</tbody>
</table>

1.5 Deprecated Items

None

1.6 Removed Items

None
2 Connect Stack

2.1 New Items
None

2.2 Improvements
None

2.3 Fixed Issues
None

2.4 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 the TECH DOCS tab on 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></td>
<td>When running the RAIL Multiprotocol Library (used for example when running DMP Connect+BLE), IR Calibration is not performed because of a known issue in the RAIL Multiprotocol Library. As result, there is an RX sensitivity loss in the order of 3 or 4 dBm.</td>
<td></td>
</tr>
<tr>
<td>501561</td>
<td>In the Legacy HAL component, the PA configuration is hard-coded regardless of the user or board settings.</td>
<td>Until this is changed to properly pull from the configuration header, the file ember-phy.c in the user's project will need to be modified by hand to reflect the desired PA mode, voltage, and ramp time.</td>
</tr>
<tr>
<td>711804</td>
<td>Connecting multiple devices simultaneously may fail with a timeout error.</td>
<td></td>
</tr>
</tbody>
</table>

2.5 Deprecated Items
None

2.6 Removed Items
None
3 RAIL Applications

3.1 New Items

**Added in release 3.6.0.0**
- XG27 Support
- Host – NCP support
  - Host: RAIL Host Simple CPC to Serial
  - NCP
    - RAIL - NCP Simple TRX with CPC Support (VCOM)
    - RAIL - NCP Simple TRX with CPC Support (SPI)

3.2 Improvements

**Changed in release 3.6.0.0**
- Amazon Sidewalk PHYs are added to the RAIL - SoC Range Test applications

3.3 Fixed Issues

None

3.4 Known Issues in the Current Release

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
</table>
| 1151826| On XG23, XG25 and XG28 Sidewalk profile is greyed out in radio configurator and cannot be selected. | Manually add a `profile_sidewalk.restriction` file like the other restriction files and update the “enabled” selection string to “Sidewalk”.
Then add to app/flex/component/rail/sl_flex_rail_package_assistant/sl_flex_rail_package_assistant.c line 243 end with this code:

```c
|| defined(RAIL0_CHANNEL_GROUP_1_PROFILE_SIDEWALK) || defined(RAIL0_CHANNELS_FOR_915_PROFILE_SIDEWALK)
```

3.5 Deprecated Items

None

3.6 Removed Items

**Removed in release 3.6.0.0**
-
4 RAIL Library

4.1 New Items

**Added in release 3.6.0.0**

- Added support for a new RX_DIRECT_SYNCHRONOUS_MODE_DATA RAIL RX data source to capture direct mode data in sync with the configured bit rate for the PHY. This requires a PHY that supports this mode of capture from the Radio Calculator and is only currently supported on the EFR32xG23.
- Added new RAIL_EnableCacheSynthCal function to enable the radio sequencer to cache calibration values instead of recalculating them on every RX and TX event. This allows you to lower the minimum transition time for most RAIL_StateTiming_t transitions in typical cases.
- Added a new RAIL_RX_OPTION_FAST_RX2RX which will force the radio sequencer to immediately transition to RXSEARCH to get ready to receive the next packet while still processing the previous one. This will minimize the RX to RX state transition time. This is only supported on chips that have RAIL_SUPPORTS_FAST_RX2RX set to true.
- Added RAIL_PacketTimeStamp_t::packetDurationUs field which is currently set only on EFR32xG25 for received OFDM packets.
- Added RAIL support for the MGM240L lighting modules.
- Added the new RAIL_WMBUS_Config API to allow configuring WMBUS and simultaneous M2O RX of T and C mode packets.
- Added a new API RAIL_SetTxFifoAlt() which provides a new start offset parameter to specify where the data begins in the TX FIFO.
- Added support for RAIL_IEEE802154_SupportsRxChannelSwitching on the EFR32xG21. This is also still supported at an alpha quality level on the EFR32xG24.
- Added RAIL_IEEE802154_SetRxToEnhAckTx() to allow IEEE 802.15.4 stacks to specify a different rxToTx state transition turnaround time for Enhanced ACKs, which generally need more time to construct and secure. Immediate ACKs will continue to use the rxToTx time specified in RAIL_IEEE802154_Config_t::timings.
- Added Tx packet duration information for EFR32xG22 and newer chips.

4.2 Improvements

**Changed in release 3.6.0.0**

- Added support for PHY-specific RSSI offsets on the EFR32xG27 and EFR32xG28 platforms.
- Updated Packet Trace on the EFR32xG25 and EFR32xG28 when using the Wi-SUN protocol to have a more informative PHY identifier and to support the whole channel number range.
- Fixed the RAIL_PA_CURVES_2P4_LP power curves on the EFR32xG24 to better match characterization data.
- Corrected the sign of the frequency error reported by RAIL_GetRxFreqOffset() when using OFDM on the EFR32xG25 to match how this was handled for other modulations (e.g., Freq_error=current_freq-expected_freq).
- Added new RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE to trigger RAIL_EVENT_ZWAVE_BEAM on all beam frames.
- Added RAIL_ZWAVE_GetBeamHomeldHash() to retrieve the beam frame's HomeldHash when handling that event and made sure that the HomeldHash byte is now present on PTI for Z-Wave beam frames even when NodeId does not match.
- Fixed an issue on EFR32ZG23 where multiple beam frames were lumped together on PTI as one large beam chain.
- Adjusted channel power restrictions for the 802.15.4 PHYs on new xGM210 modules.
- Added separate curves when the 20 dBm PA is used at 3.3 V and 1.8 V for EFR32xG24.
- Increased EFR32ZG13 and EFR32ZG14 Z-Wave long-range beam detect time to improve FLiRS performance.
### 4.3 Fixed Issues

**Fixed in release 3.6.0.0**

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>824355</td>
<td>Fixed an issue in IEEE802.15.4 MAC address filtering when receiving small OFDM packets.</td>
</tr>
<tr>
<td>832743</td>
<td>Clarified use of RAIL_SetNextTxRepeat() must be prior to initiating a transmit operation via API call and fixed an issue where it did not properly return an error when called while a transmit operation was in progress.</td>
</tr>
<tr>
<td>1055824</td>
<td>Fixed an issue with low-side synth injection (negative IF) on proprietary 2.4GHz PHYs when using EFR32xG22 and newer chips. This fix requires regenerating the PHY with the latest version of the Radio Configurator to work.</td>
</tr>
<tr>
<td>1058480</td>
<td>Fixed an RX FIFO corruption on EFR32xG25 that occurred when receiving/sending certain OFDM packets using FIFO mode.</td>
</tr>
<tr>
<td>1082274</td>
<td>Fixed an issue on the EFR32xG22, EFR32xG23, EFR32xG24, and EFR32xG25 chips that could cause the chip to lock up if the application attempts to re-enter EM2 within ~10 μs after wake-up and hits a &lt;0.5 μs timing window. If hit, this lockup requires a power-on reset to restore normal operation to the chip.</td>
</tr>
<tr>
<td>1083615</td>
<td>Fixed an issue for certain ramp time and power level combinations on the EFR32xG21 where the PA ramp would stop one power level short of the desired output level.</td>
</tr>
<tr>
<td>1090336</td>
<td>Fixed an issue in the &quot;RAIL Utility, Protocol&quot; component where BLE would be required to select a Zigbee PHY.</td>
</tr>
<tr>
<td>1090512</td>
<td>Fixed an issue in the &quot;RAIL Utility, PA&quot; component where certain functions would attempt to use the RAIL_TX_POWER_MODE_2P4GIG_HIGHEST macro even though they didn’t support it. This would result in undefined behavior previously, but will now correctly error.</td>
</tr>
<tr>
<td>1090728</td>
<td>Fixed a possible RAIL_ASSERT_FAILED_UNEXPECTED_STATE_RX_FIFO issue on EFR32xG12 with RAIL_IEEE802154_G_OPTION_GB868 enabled for a FEC-capable PHY which can happen when aborting a packet at frame detection, for instance by idling the radio.</td>
</tr>
<tr>
<td>1092769</td>
<td>Fixed an issue when using Dynamic Multiprotocol and BLE Coded PHYs where a transmit could underflow depending on what protocol was active when the PHY and syncword were loaded.</td>
</tr>
<tr>
<td>1096663</td>
<td>Fixed a compilation error in &quot;RAIL Utility, Coexistence&quot; component when the Coexistence WiFi TX GPIO is enabled.</td>
</tr>
<tr>
<td>1096665</td>
<td>Fixed a compilation issue in &quot;RAIL Utility, Coexistence&quot; component when the SL_RAIL_UTIL_COEX_WIFI_TX_PORT is defined.</td>
</tr>
<tr>
<td>1103966</td>
<td>Fixed an unexpected Rx packet abort on the EFR32xG25 when using the Wi-SUN OFDM option4 MCS0 PHY.</td>
</tr>
<tr>
<td>1104033</td>
<td>Fixed an issue in the RAIL_ZWAVE_ReceiveBeam function so that it idles the radio regardless of whether a beam is detected on the EFR32ZG23.</td>
</tr>
<tr>
<td>1104441</td>
<td>Fixed an issue with the &quot;RAIL Utility, Coexistence&quot; component counters for Zigbee that could prevent them from ticking as expected depending on how things are linked.</td>
</tr>
<tr>
<td>1105134</td>
<td>Fixed an issue when switching between certain PHYs that could cause the first received packet to be reported as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS. This issue could potentially impact EFR32xG22 and newer chips.</td>
</tr>
<tr>
<td>1105529</td>
<td>Fixed an issue on EFR32xG22 and later platforms when using a FrameType decoding PHY where a bad frame type packet was mis-reported as RAIL_RX_PACKET_ABORT_ABORTED instead of the proper RAIL_RX_PACKET_ABORT_FORMAT.</td>
</tr>
<tr>
<td>1109574</td>
<td>Fixed an issue on EFR32xG22 and newer chips where a radio sequencer assert could cause the application to hang in an ISR rather than report the assert via RAILCb_AssertFailed().</td>
</tr>
<tr>
<td>1118063</td>
<td>Fixed issue with recent RAIL_ZWAVE_OPTION_PROMISCUOUS_beam_MODE on EFR32xG13 and EFR32xG14 where the NodeId of the promiscuous beam was not properly recorded for RAIL_ZWAVE_GetBeamNodeId(), causing it to report 0xFF.</td>
</tr>
<tr>
<td>1126343</td>
<td>Fixed an issue on EFR32xG24 when using the IEEE 802.15.4 PHY where the radio could become stuck when doing an LBT transmit if a frame is received during the CCA check window.</td>
</tr>
<tr>
<td>1134223</td>
<td>Fixed an issue when using &quot;RAIL Utility, Coexistence&quot; component where the request line is left asserted after TX is aborted following a sync detect.</td>
</tr>
<tr>
<td>1135418</td>
<td>Fixed incorrect RAIL_RxPacketInfo_t::filterMask on received 802.15.4 Beacon frames, which now reflects which PanId and address the Beacon's Source PanId and Source Address match, if any. Note that RAIL generally accepts all Beacons so the filterMask can be 0x00.</td>
</tr>
<tr>
<td>1138522</td>
<td>Fixed an issue on the EFR32xG25 for SUN FSK PHYs where receiving a packet after calling RAIL_IEEE802154_Init() but before configuring RAIL_IEEE802154_ConfigGOptions could break reception.</td>
</tr>
</tbody>
</table>
### 4.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>1140569</td>
<td>Fixed a rare timing issue on EFR32xG24 where an ACK timeout might cause the next packet to be received as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS.</td>
<td></td>
</tr>
<tr>
<td>1150779</td>
<td>Fixed the 15.4 channel configurations on the MGM240PA32 and MGM240PB32 modules to use the correct frequency for channel 26.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>641705</td>
<td>Using direct mode (or IQ) functionality on EFR32xG23 requires a specifically set radio configuration that is not yet supported by the radio configurator. For these requirements, reach out to technical support who could provide that configuration based on your specification</td>
<td></td>
</tr>
<tr>
<td>732659</td>
<td>Infinite receive operations where the frame's fixed length is set to 0 are not working correctly on the EFR32xG23 series chips.</td>
<td></td>
</tr>
<tr>
<td>732659</td>
<td>On EFR32xG23:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Wi-SUN FSK mode 1a exhibits a PER floor with frequency offsets around ± 8 to 10 KHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Wi-SUN FSK mode 1b exhibits a PER floor with frequency offsets around ± 18 to 20 KHz</td>
<td></td>
</tr>
</tbody>
</table>

### 4.5 Deprecated Items

None

### 4.6 Removed Items

None
5 Using This Release

This release contains the following

- Radio Abstraction Interface Layer (RAIL) stack library
- Connect Stack Library
- RAIL and Connect Sample Applications
- RAIL and Connect Components and Application Framework

This SDK depends on Gecko Platform. The Gecko Platform code provides functionality that supports protocol plugins and APIs in the form of drivers and other lower layer features that interact directly with Silicon Labs chips and modules. Gecko Platform components include EMLIB, EMDRV, RAIL Library, NVM3, and mbedtls. Gecko Platform release notes are available through Simplicity Studio’s Documentation tab.

For more information about the Flex SDK v3.x see UG103.13: RAIL Fundamentals and UG103.12: Silicon Labs Connect Fundamentals. If you are a first time user, see QSG168: Proprietary Flex SDK v3.x Quick Start Guide.

5.1 Installation and Use

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

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

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/.

5.2 Security Information

Secure Vault Integration

When deployed to Secure Vault High devices, sensitive keys are protected using the Secure Vault Key Management functionality. The following table 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>Thread Master Key</td>
<td>Exportable</td>
<td>Must be exportable to form the TLVs</td>
</tr>
<tr>
<td>PSKc</td>
<td>Exportable</td>
<td>Must be exportable to form the TLVs</td>
</tr>
<tr>
<td>Key Encryption Key</td>
<td>Exportable</td>
<td></td>
</tr>
<tr>
<td>MLE Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>Temporary MLE Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>MAC Previous Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>MAC Current Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>MAC Next Key</td>
<td>Non-Exportable</td>
<td></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.
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.

5.3 Support

Development Kit customers are eligible for training and technical support. Use the Silicon Labs Flex web page 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.
Simplicity Studio

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