

Release Notes

Product: Application Framework for EmberZNet 5.7.0
Release Date: February 22, 2016

1 Overview

This release contains the GA release of Application Framework for use with EmberZNet 5.7 for the EFR32 and EM3xx platforms. This release must be used with Silicon Labs Simplicity Studio 3.3.

2 Getting Started

The source code for Application Framework is bundled with the EmberZNet 5.7 installer. Once the stack has been installed, follow these steps to add the stack to Ember Desktop.

1. Start Ember Desktop by selecting **Start** ⇒ **All Programs** ⇒ **Ember** ⇒ **Ember Desktop** ⇒ **Ember Desktop**
2. Open the preferences panel by selecting **File** ⇒ **Preferences** from the Ember Desktop menu.
3. Select **Network Applications** from the list on the left of the preferences panel. Installed stacks and applications will be listed, with an Add... button on the right panel.
4. Click the **Add** button and browse to the directory where EmberZNet 5.7 is installed
5. Click the **OK** button to save the changes and close the preferences panel

Once the stack is loaded, Application Framework projects can be created and configured using Ember Desktop.

1. Select **File** ⇒ **New** ⇒ **Application Framework Configuration** from the Ember Desktop menu.
2. Select the appropriate stack for your application (e.g., EmberZNet 5.7 EM35X).
3. Click the **OK** button.

2.1 Support

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

3 The 5.7 Release

This release includes a number of new features, improvements, and bug fixes. Customers upgrading from previous releases are strongly encouraged to carefully review this section. It is important to become familiar with the changes in this release before attempting to migrate applications.

Version 5.7.0

3.1 New Features

3.1.1 ZigBee 3.0 Support

This release contains support for ZigBee 3.0 functionality.

There were some changes made to application framework constants to align them with the naming found with in the ZigBee 3.0 document suite. The macro `ZCL_PROFILE_WIDE_COMMAND`, which indicated the ZCL frame control bits that are used for global ZCL commands, has been renamed to `ZCL_GLOBAL_COMMAND`. The former macro still exists in the master application framework header to support legacy code.

3.2 Configuration Changes

3.2.1 Transient Key Timeout Configuration

There is now a way to configure how long a trust center stores a transient key in RAM for a joining device to use before it updates its trust center link key per the Base Device Specification. The global variable can also be configured at runtime with AppBuilder using the Security Core Library plugin for the EM35X framework and the NCP Configuration plugin for the EM35X-EZSP framework.

3.2.2 Application Manufacturer Code

AppBuilder allows users to enter a manufacturer code for their application. In previous releases, if the manufacturer code was not set in the AppBuilder user interface, it would default to the Ember manufacturer code (0x1002). Now, the value will default to 0x0000 if it is not set. Users are recommended to use their own manufacturer ID for their application by setting it in the AppBuilder user interface under the ZCL global tab.

3.2.3 Default Response Policy

The default response policy is a value set in AppBuilder that dictates whether the disable default response bit in the ZCL header is set or cleared by default. The ZCL6 specification takes a stronger position on setting this bit for response-style frames (see section 2.4.1.1.4), so the default response policy value in AppBuilder will now default to `CONDITIONAL`, meaning the bit will be set on response-style frames. Users wishing to comply with the ZCL6 specification should consider the value of their default response policy in AppBuilder.

3.3 Removed APIs

No APIs were removed in this release.

3.4 Removed Callbacks

No callbacks were removed in this release.

3.5 Removed CLI Commands

No CLI commands were removed in this release.

3.6 Removed Plugins

There was one plugin remove in this release.

3.6.1 Network Commissioner

The Network Commissioner plugin sought to give users an example of a commissioning state machine per the Base Device Specification. This functionality has been moved to sample applications, in order to save code space. Users wishing to view an example of this functionality should study the ZigBee 3.0 sample applications, as they better exemplify a ZigBee 3.0 commissioning state machine.

3.7 Changed APIs

There was a couple of changes to API's in this release.

3.7.1 emberAfGetDifference

The API `emberAfGetDifference` now compares data sizes up to 8 bytes on platforms that support the `uint64_t` primitive type. Images that are built with the `HAL_HAS_INT64` precompiler symbol will support this type. Before, the API only supported data sizes of 4 bytes or fewer. There should be no action needed from users as compilers should promote values of size 4 bytes and fewer to 8 byte quantities with the same decimal value.

3.7.2 emberAfPluginNetworkSteeringStart

The API `emberAfPluginNetworkSteeringStart` will now try to join a network using an install code derived key, the centralized security default key, and the distributed security default key. Before, the plugin was unable to join a distributed or security without the caller having to pass in what kind of network they were looking for. Now, the plugin will try to join all of those network types automatically. Because of this change, the API now takes no arguments.

3.7.3 Finding and Binding

Due to the update to the Find and Bind plugins, see [3.10.2](#), the names of the finding and binding APIs have changed slightly. The API `emberAfPluginFindAndBindTarget` has been renamed to `emberAfPluginFindAndBindTargetStart`. The API `emberAfPluginFindAndBindInitiator` has been renamed to `emberAfPluginFindAndBindInitiatorStart`. The functionality of the APIs has stayed the same.

3.8 Changed Callbacks

Please see below for the callbacks that were changed in this release.

3.8.1 Network Steering Callbacks

The Network Steering plugin will now try to join networks with each of the three supported ZigBee 3.0 key types: an install code derived key, the default centralized security key, and the default distributed security key. Because of this, the callback `emberAfPluginNetworkSteeringCompleteCallback` has been simplified to pass the state at which the plugin completed a successful join. Please see [3.7.2](#) for more information on this change.

3.8.2 Update TC Link Key Callbacks

The Update TC Link Key plugin has changed the name of its callback `emberAfPluginUpdateTcLinkKeyCompleteCallback` to `emberAfPluginUpdateTcLinkKeyStatusCallback`. The new name of the callback more accurately reflects what notification the callback is representing. Previously, the callback would only be called once on completion. Now, the callback notification is more generic, and it may be called multiple times in a link key update process. Furthermore, this callback has changed slightly to help clients communicate better with the plugin through the callback.

The Update TC Link Key plugin uses the callback `emberAfPluginUpdateTcLinkKeyStatusCallback` to notify its client of an update in the trust center link key update process. The callback implementation now

lets the client return a boolean value to notify the plugin when the client is done receiving key exchange status updates. This allows the client to stop receiving updates from the plugin if it receives an `EmberKeyStatus` value that it deems an error in the process.

3.8.3 Find and Bind Callbacks

The Find and Bind plugin has been reworked to include support for creating multicast bindings. The application can use the callback `emberAfPluginFindAndBindInitiatorBindTargetCallback` to tell the plugin which type of binding they wish to create for that target. For this to happen, the callback now takes a binding entry and a pointer to a group name. If the application would like for the binding to be created as multicast, it should change the type member of the binding entry struct to `EMBER_MULTICAST_BINDING`, enter the group ID in identifier member of the binding entry struct, and optionally write the group name to the pointer. Previous implementations wishing to continue to use unicast bindings should not need to change other than possibly using the new binding entry struct for data on the target.

3.9 Changed CLI Commands

There were a couple of CLI commands changed in this release.

3.9.1 Network Steering CLI

The Network Steering plugin has been improved with runtime configurable options to determine specific parts of the network steering process. Because of this, the `plugin network-steering start` command now takes one 32-bit integer as a command. This 32-bit integer argument allows the user to control pieces of the network steering process when issuing the command from the command line interface. See `network-steering.h` for these specific options.

3.9.2 Color Control CLI

The Color Control ZCL CLI commands have been updated with a new format to match the rest of the ZCL CLI commands. Previously, the color control cli commands were of the format `zcl colorcontrol`. Now, the color control cli commands are of the format `zcl color-control`.

3.10 Changed Plugins

The following changes have been made to a plugin in this release.

3.10.1 Network Steering

The Network Steering plugin will now try to join networks with each of the three supported ZigBee 3.0 key types: an install code derived key, the default centralized security key, and the default distributed security key. Please see [3.7.2](#) for more information on this change.

3.10.2 Finding and Binding

The Find and Bind plugin has been split into two plugins for the purpose of code space efficiency. There are two new plugins, Find and Bind Initiator and Find and Bind Target, which contain the functionality for a finding and binding initiator and a finding and binding target, respectively. Upon users opening applications that used the Find and Bind plugin, AppBuilder will enable both of these new plugins, which together make up the old plugin. Users should decide whether or not they need both sides of the finding and binding process in their application and enable or disable either of the plugins if they wish to save code size. See [3.7.3](#) for slight changes made to the names of the Find and Bind APIs.

3.11 Deprecated APIs

No APIs were deprecated in this release.

3.12 Deprecated Callbacks

No callbacks were deprecated in this release.

3.13 Deprecated CLI Commands

No CLI commands were deprecated in this release.

3.14 Deprecated Plugins

No plugins were deprecated in this release.

3.15 New APIs

No APIs were added in this release.

3.16 New Callbacks

There was one new callback added in this release.

3.16.1 Network Steering Get Node Type

The network steering plugin now provides the ability for a user to set their node type dynamically at join time. For example, if a user wishes to join as a sleepy end device for one type of network, but an RX on when idle end device for another network, they can use this callback to do that. By default, the callback will perform the behavior that was previously hardcoded in the plugin. Therefore, previous implementations will not need to use this callback if they wish to continue functioning the same way.

3.17 New CLI Commands

There were no new CLI commands added in this release.

3.18 New Plugins

There are no new plugins in this release.

4 Known/Fixed Issues

4.1 Fixed Issues

4.1.1 Fixed Issues in EmberZNet 5.7.0

- N/A

4.2 Known Issues

- Issue 163938: IAR reports broken options when opening EFR32 target workspace created by AppBuilder
- Issue 159611: ZCL attribute tokens creator codes likely to change if you add or remove one or more attributes. For instance, if you add an attribute whose cluster ID or attribute ID is not the largest numerically, then this attribute gets inserted into a list of creator codes and makes all creator codes after it to be different (shifted).
- Issue 159014: AppBuilder for AFv2 applications limits radio TX power levels to +8 dBm

- Issue 136337: Price Client Plugin does not work properly in SE 1.1b. Workaround: Select an older SE 1.1 spec version or disable Price Client plugin.
- Issue 136274: Node State Cache incorrectly indicates no nodes in trace file.
- Issue 125595: Makefile generation does not reference the "How is path saved?" field for additional files.
- Issue 110166: Extended Ember Desktop Decoder's events window to be customizable
- Issue 92566: APS Alarm message always decodes to say missing packets
- Issue 66116: EZSP frames for EZSP_ZLL_NETWORK_OPS and EZSP_ZLL_SET_INITIAL_SECURITY_STATE are not decoding properly by Ember Desktop.
- Issue 66095: ISA3 Firmware Updates may cause Ember Desktop to hang, requiring a restart of both the ISA3 and Ember Desktop
- Issue 65929: Global channel change via Sniffer Settings dialog doesn't impact EM35x sniffers
- Issue 65898: ISD cannot find JRE on some 64-bit Windows 7 machines
- Issue 65867: Active Scan from Sniffer Settings dialog doesn't accurately report results.
- Issue 65687: ISD Console renders ASCII 0x81 character incorrectly
- Issue 65541: Ember Desktop Timeline view sometimes misses data in large captures
- Issue 173569: calculateTimer() should use MILLISECOND_TICKS_PER_SECOND for increased accuracy.
- Issue 160105: emberAfGroupsClusterRemoveAllGroupsCallback does not remove all scenes associated with groups removed. Only scenes associated with ZCL_SCENES_GLOBAL_SCENE_GROUP_ID are removed.
- Issue 158920: SE 1.2 Calendar Client Description, Interface and Dependencies are not accurate.
- Issue 146795: Added Alarm Cluster functionality to IAS Zone Plugin.
- Issue 137490: Update Reporting plugin to send one report for multiple attributes whenever possible.
- Issue 126087: Sleepy end device would return NO_LOCAL_RESOURCES when a coordinator initiates key establishment with it in Multi-networking.
- Issue 123399: Non-sleepy endpoint does not always keep correct network parameters on reset in multi-networking.
- Issue 121707: Reporting does not send to a group address.
- Issue 119828: ota-client.c does not use the server EUI64 in Partner Link Key Exchange.
- Issue 117894: Image Block Response does not properly handle the one time block request delay.
- Issue 113213: Request Block Delay value on client is misread by server
- Issue 108582: minBlockRequestPeriod is incorrectly compared to imageBlockRequestMinRequestPeriodSeconds
- Issue 108245: Old Custom Cluster XML is unable to be replaced by new XML
- Issue 105658: Network Find plugin timeout value does not always reset after state machine finishes
- Issue 101644: Add a callback to the check-in interval for poll control.
- Issue 92147: ZLL Scan Response Should Be Sent at Power 0 rather than last-used power level

- Issue 87104: Support for OTA Upgrade End Request status REQUIRE_MORE_IMAGE
- Issue 83798: Image Integrity Tag generation in image builder for an OTA file
- Issue 66944: Duplicate Key Confirm Response message can lock up KE plugin state machine
- Issue 66786: "zcl ota server reload" doesn't properly reload image info when using OTA Simple Storage plugin
- Issue 66785: Messaging Client plugin should differentiate between Cancel Msg command and timed out / replaced message
- Issue 66508: Framework should avoid sending unicast loopback messages with APS security since stack doesn't support this
- Issue 66221: DRLC Cancel command can cause new events to be scheduled incorrectly