QSG113: Getting Started with Silicon Labs Thread

This quick start guide provides basic information on configuring, building, and installing applications for the EM35x and Mighty Gecko (EFR32MG) family of SoCs using the Silicon Labs Thread stack version 2.2 and higher and Simplicity Studio version 4. While the focus is on development with the Wireless Gecko (EFR32) portfolio on the Wireless Starter Kit (WSTK), the instructions also apply to development for the EM35x/ISA3.

This guide is designed for developers who are new to Silicon Labs Thread and the Silicon Labs development hardware. It provides instructions to get started using the example applications provided with the Silicon Labs Thread stack.

KEY FEATURES

- Product overview
- Setting up your development environment
- Installing Simplicity Studio and Silicon Labs Thread
- Creating an example application network
- Using the Network Analyzer

1 Product Overview

Before following the procedures in this guide you must have

- Purchased your development hardware (see http://www.silabs.com/products/wireless/mesh-networking/thread/Pages/thread.aspx for more information):
  - Mighty Gecko (EFR32MG) Mesh Networking Kit

    or
  - EM35x Development Kit
- Downloaded the required software components, as described below. A card included in your development hardware kit contains a link to a Getting Started page, which will direct you to links for the Silicon Labs software products.

Note: If you are installing an EM3x development kit, do not install Ember Desktop as documented in the Quick Start Guide. Instead, install Simplicity Studio as noted below.

1.1 Software Components

See the stack release notes for version restrictions and compatibility constraints for the stack and these component. To develop Silicon Labs Thread applications, you will need the following. Installation instructions are provided in the section **Install Simplicity Studio and the Silicon Labs Thread Stack**:

- IAR Embedded Workbench for ARM (IAR-EWARM) 7.80, used as a compiler in the Simplicity Studio development environment. Download the supported version from the Silicon Labs Support Portal. Refer to the "QuickStart Installation Information" section of the IAR installer for additional information about the installation process and how to configure your license. Once IAR-EWARM is installed, the next time Simplicity Studio starts it will automatically detect and configure the IDE to use IAR-EWARM.
- The Simplicity Studio version 4 development environment, which incorporates AppBuilder. If you do not have version 4, please connect to http://www.silabs.com/products/mcu/Pages/simplicity-studio-v4.aspx to download it. AppBuilder is an interactive GUI tool that allows you to configure a body of Silicon Labs-supplied code to implement applications. Online help for AppBuilder and other Simplicity Studio modules is provided.
- The Silicon Labs Thread stack, an advanced implementation of a wireless protocol stack, installed through Simplicity Studio. The stack API is documented in online API reference as well as other documents available through Simplicity Studio. The stack is delivered as a collection of libraries that you can link to your applications. A description of each library is provided in the development environment. The release notes contain details on the folders installed along with their contents.
- Simplicity Commander, installed along with Simplicity Studio. A GUI with limited functionality can be accessed through Simplicity Studio's Tools menu. Most functions are accessible through a CLI invoked by opening a command prompt in the Simplicity Commander directory (\SiliconLabs\SimplicityStudio\v4\developer\adapter_packs\commander). See *UG162: Simplicity Commander Reference Guide* for more information.

While Simplicity Studio and Simplicity Commander can be run on a Mac OS or Linux machine, these instructions assume you are working with a Microsoft Windows-based PC. If you are using a non-Windows system, IAR-EWARM must be run via WINE or some other form of emulator or virtual machine.

Although you will not need them for the tasks in this Getting Started guide, you may wish to become familiar with the ISA3 manufacturing utilities available for the EM35x development environment. The ISA3 utilities are installed by Simplicity Studio automatically as an "adapter pack" when EMxxx part support is enabled in your installation. If these tools were installed as part of your Simplicity Studio setup, you can find them at `{SimplicityStudio_root}/v4/developer/adapter_packs/em3xx/utils`. These utilities may also be downloaded separately from the support portal. If you have installed them separately from a standalone, executable package, the installer modifies your PATH environment variable so that the command line utilities can be easily executed from a Windows Command Prompt. Note that, if you are using an EM35x Development Kit, you may have already installed these utilities during initial setup if you connected directly to your PC and needed to configure a static IP. See *UG107: ISA3 Utilities Guide*, for detailed information on the ISA3 utilities.

Finally, if you are working with an EM35x development kit and you want to use the USB interface of the breakout board for UART connectivity, download a driver for the FTDI USB <-> Serial converter from http://www.ftdichip.com/Drivers/VCP.htm.

1.2 Support

You can access the Silicon Labs support portal at https://www.silabs.com/support by clicking the "Email-Support" link and logging in with your self-registered credentials. Use the support portal to contact Customer Support for any questions you might have during the development process.

1.3 Documentation

Stack documentation is accessed through Simplicity Studio, as described in section **Accessing Documentation and Other Resources**. Simplicity Studio also provides links to hardware documentation and other application notes. See the release notes for further details about your Silicon Labs Thread software.

2 Setting Up Your Development Environment

2.1 Connect your Hardware

Connect your development hardware to the PC on which you will install Simplicity Studio. By having it connected when Simplicity Studio installs, Simplicity Studio will automatically obtain the relevant additional resources it needs.

Note: Throughout this document, the term 'debug adapter' refers to either the ISA3 (for the EM35x) or WSTK (for the EFR32).

2.1.1 EFR32 Wireless Starter Kit (WSTK)

Connect your WSTK, with radio board mounted, to your PC using a USB cable.

Note: For best performance in Simplicity Studio, be sure that the power switch on your WSTK is in the Advanced Energy Monitoring or "AEM" position, as shown in the following figure.

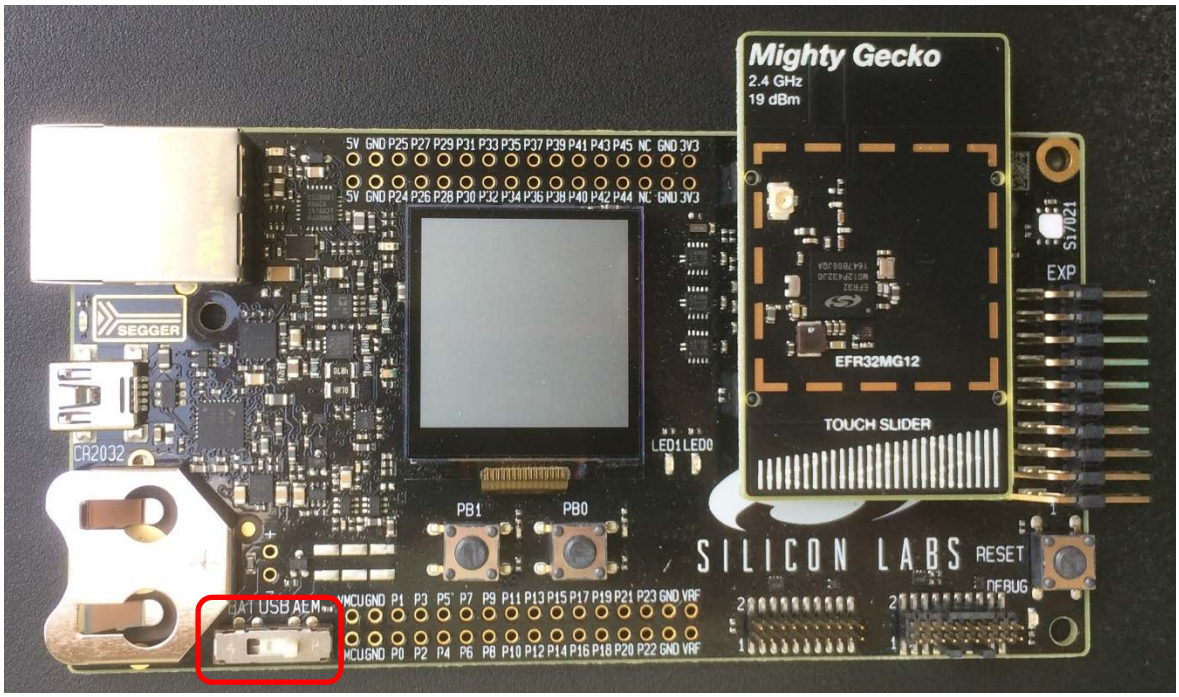


Figure 1. EFR32MG12 on a WSTK

2.1.2 EM35x Development Kit

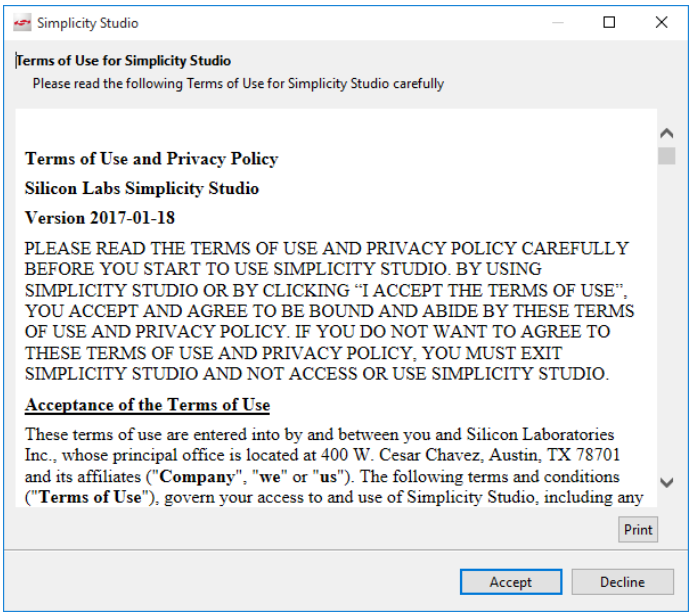
Follow the instructions in the Quick Start Guide included in the development kit to set up a development environment connected to your computer.

2.2 Register your Development Kit

In order to install the Silicon Labs Thread stack from Simplicity Studio, you must first create an account on the support portal and then register your kit on https://siliconlabs.force.com/KitRegistration, using your Mighty Gecko Kit or EM35x Development Kit serial number Be sure to record your account username and password as you will use it to log in to Simplicity Studio. You can register your kit through Simplicity Studio during installation if you prefer.

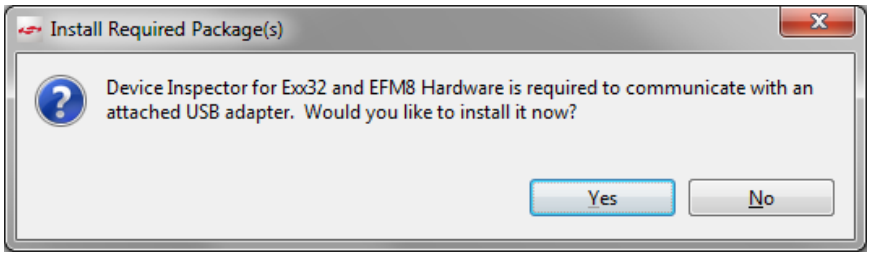
2.3 Install Simplicity Studio and the Silicon Labs Thread Stack

1. Run the Simplicity Studio installation application.
2. When Simplicity Studio first launches, it presents a Terms of Use dialog. Click **Accept** to continue.

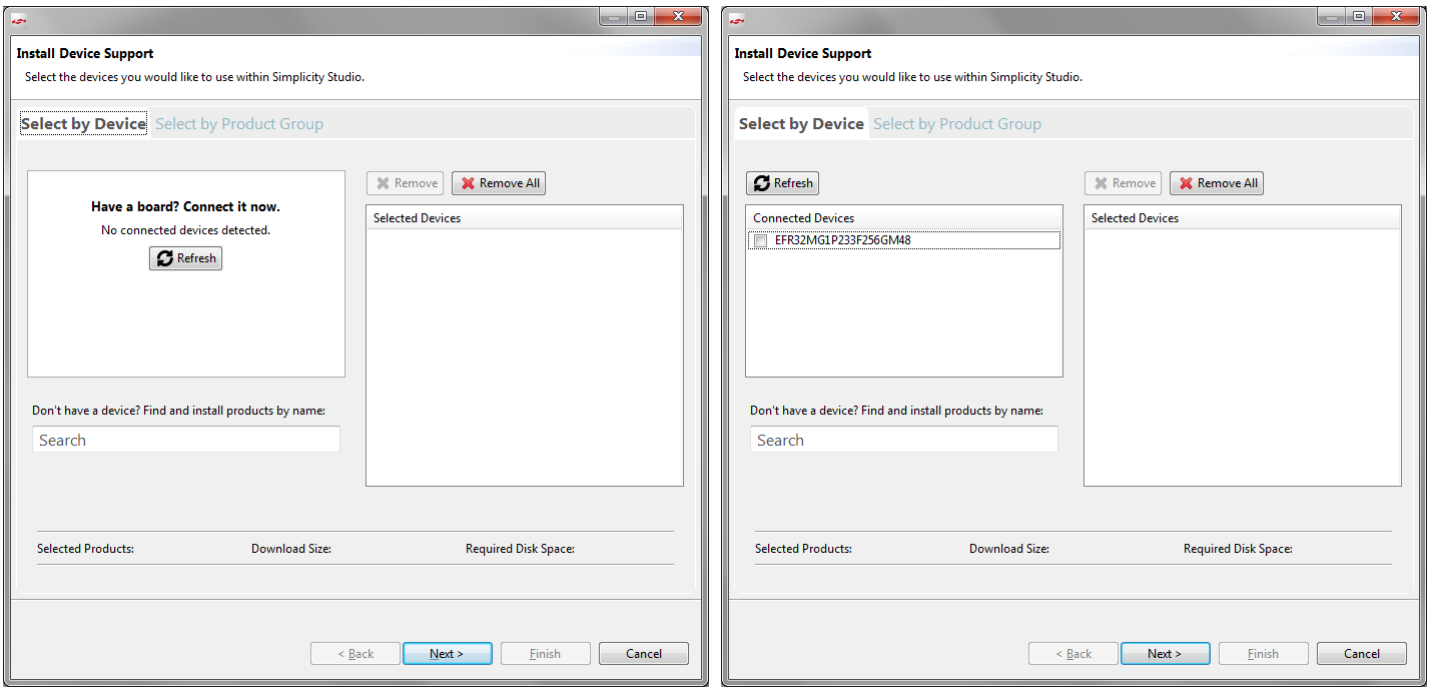


3. Next you are invited to log in. Log in using your support account username and password. Although you can skip log in here, you must be logged in and have registered your development kit to download a protected stack such as Silicon Labs Thread.

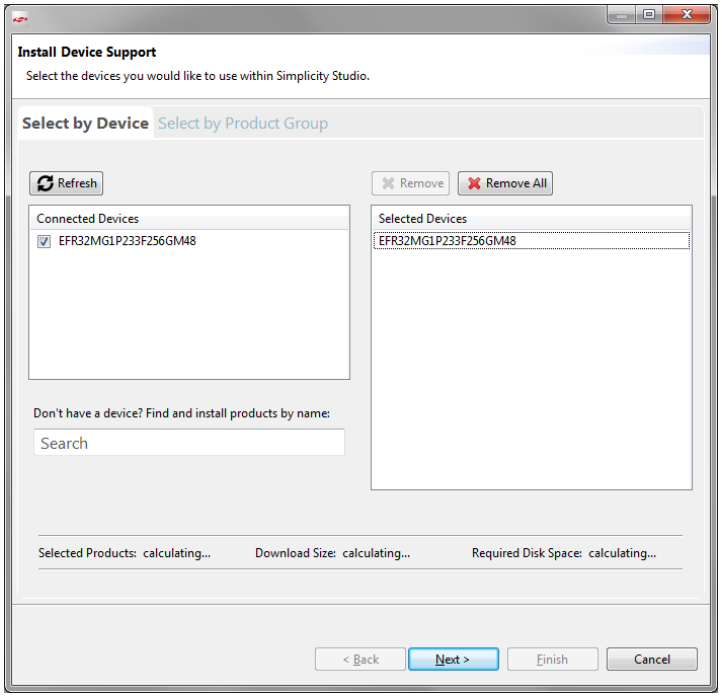
4. After login, Simplicity Studio adds software information. Once initial software installation is complete, Simplicity Studio checks for connected hardware. If you have the WSTK connected by USB cable, Simplicity Studio will detect the USB cable and prompt you to download a Device Inspector. Click **Yes**.



5. An Install Device Support dialog appears. After a short delay, it shows your connected device. If the connected device does not show, click **Refresh**. The following figure shows the Install Device Support dialog before and after the connected device is displayed.



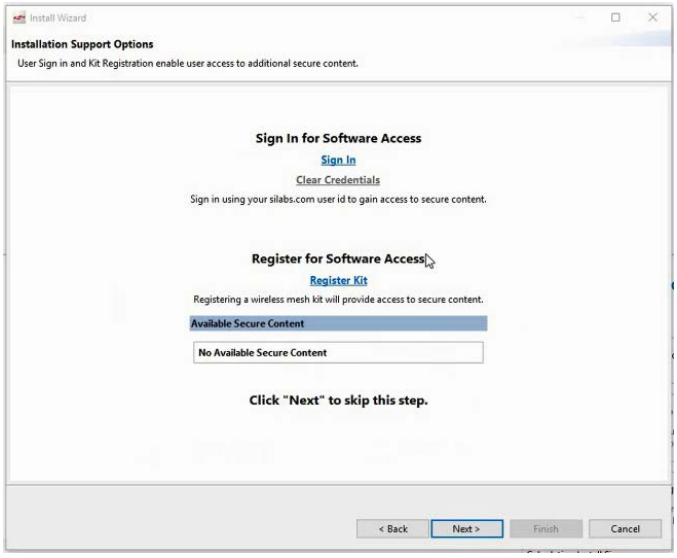
6. Click the checkbox next to the device to select it. Selecting the device allows Simplicity Studio to present the relevant software packages for you to install. Click **Next**.



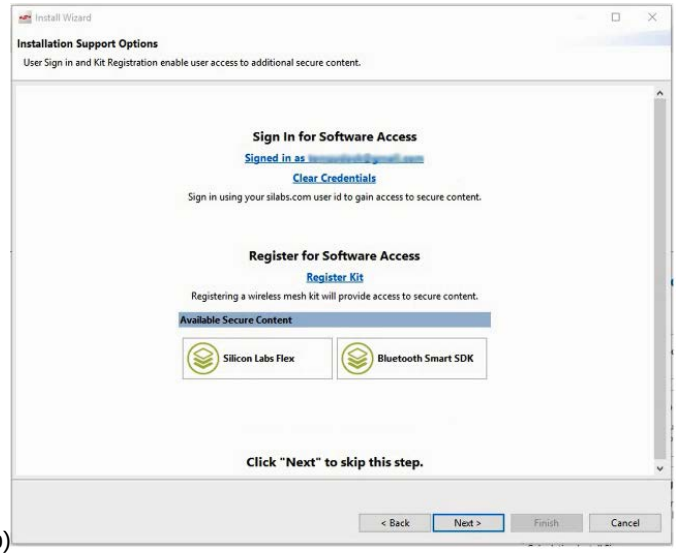
Note: You can also click the **Select by Product Group** tab to install device support for all devices in one or more product groups.

7. The next dialog varies depending on whether you have signed in and registered your kit. If you have not signed in, you have no access to restricted content and must sign in first (see the following figure a). If you If you have signed in but not registered the kit, you can see some restricted content but not Silicon Labs Thread (see the following figure b). Click **Register Kit**.

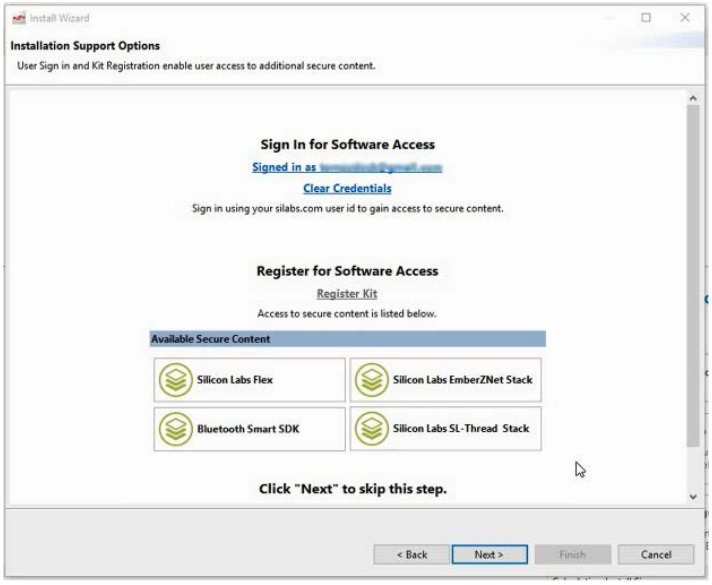
a)



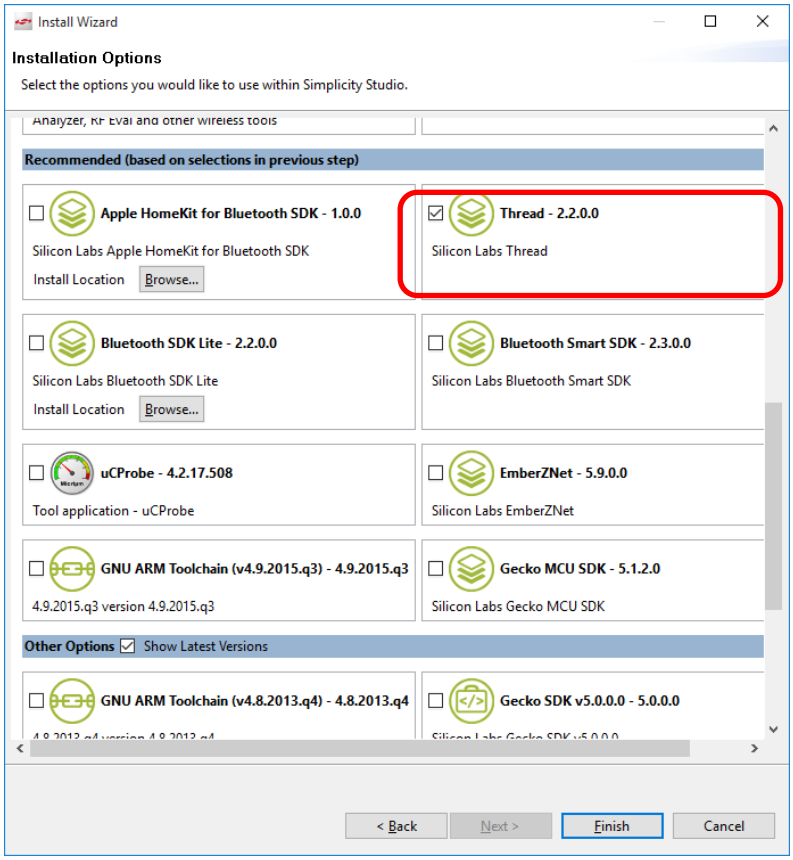
b)



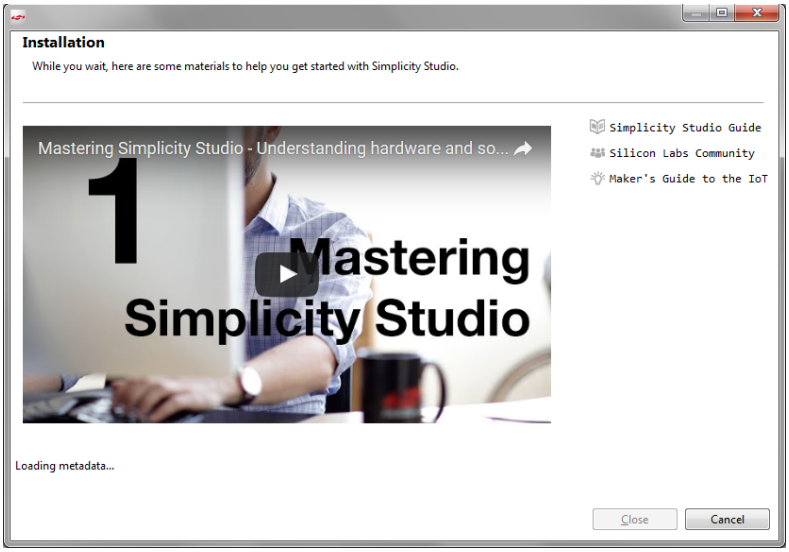
8. If you have already signed in and registered your kit, and see Silicon Labs Thread on the list of accessible components, click **Next**.



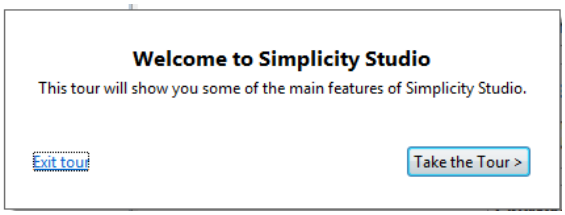
9. The **Installation Options** dialog shows the tools and software packages that can be installed. By default the list is filtered by the product connected to your computer, or that you selected in the Solutions area of the Launcher view. By default, the current versions of all Silicon Labs stacks that are compatible with this part are checked. Uncheck any you do not wish to install. Previous stack versions are shown under **Other Options**. Click **Finish**.



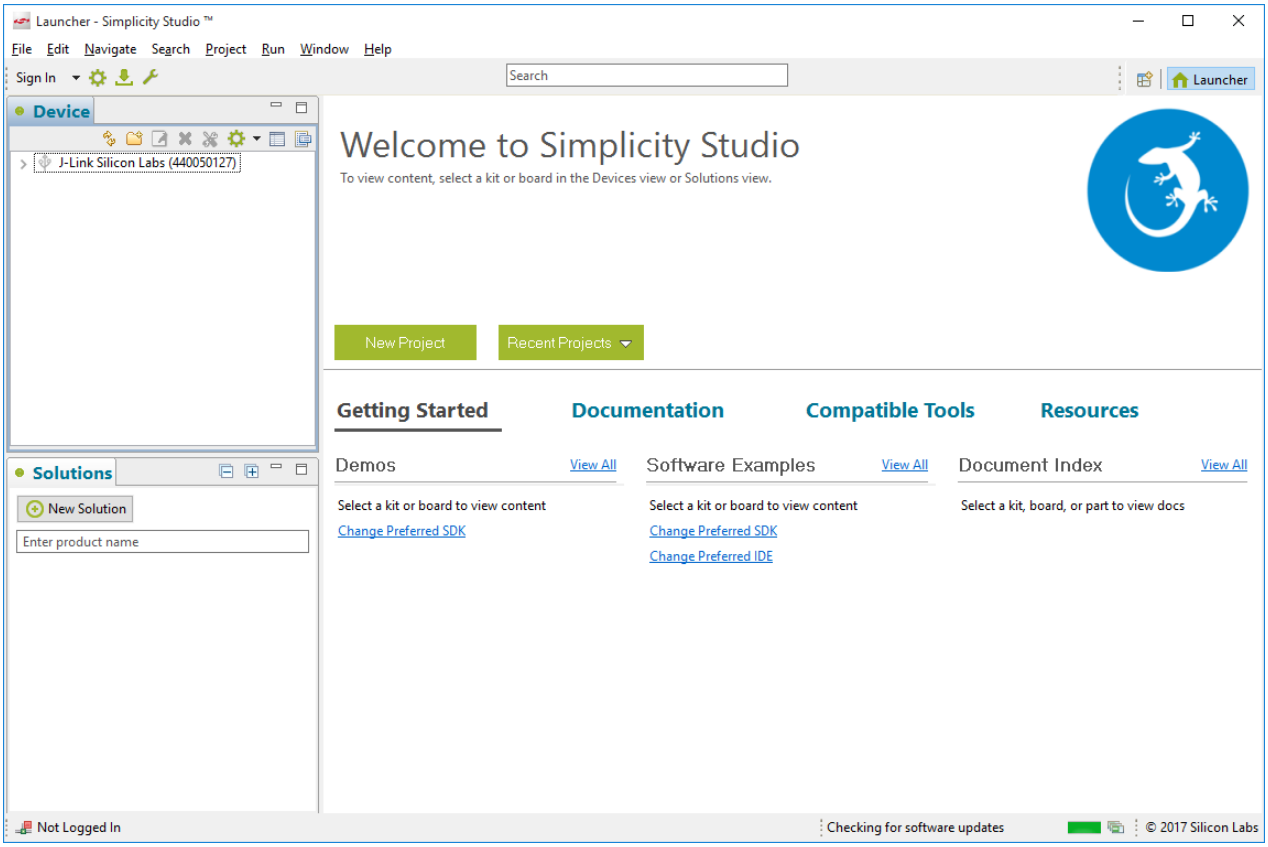
Installation will take several minutes. During installation, Simplicity Studio offers you viewing and reading options to learn more about the environment.



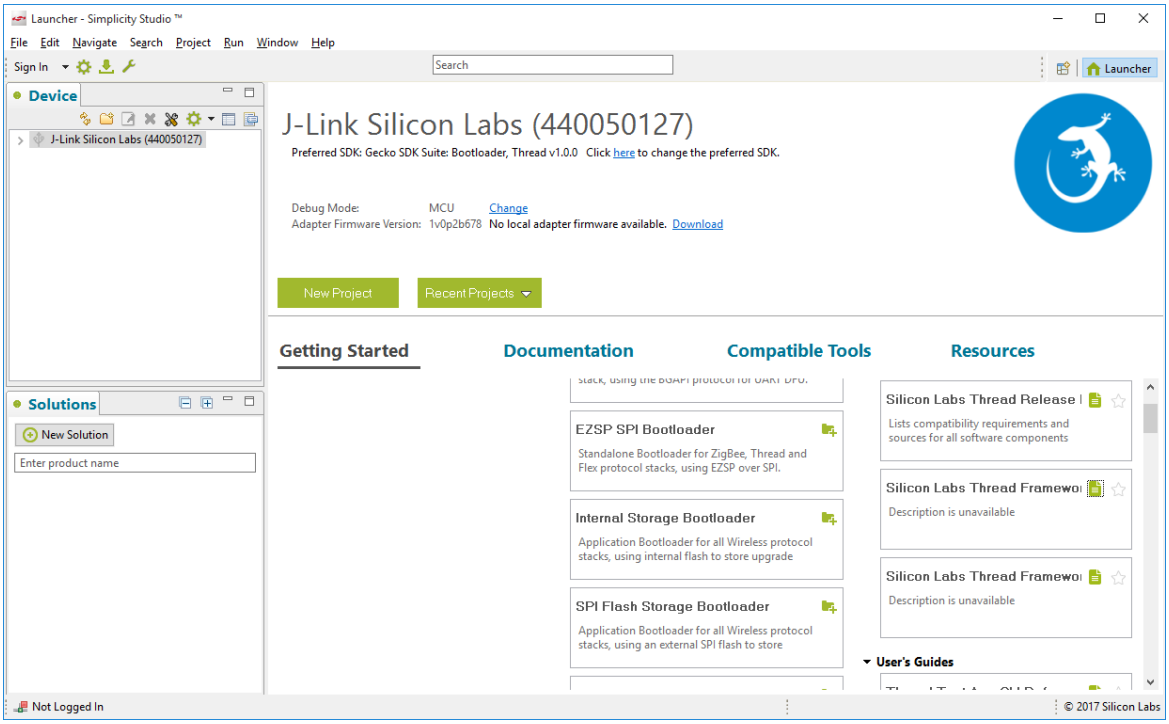
10. After installation is complete, restart Simplicity Studio.
11. When Simplicity Studio restarts, you are invited to take a tour. To clear this option now or at any time during or after the tour, click **Exit tour**.



12. The Launcher perspective opens, but it is not yet fully populated. Click the connection entry (J-link or EMISA3) in the devices tab.



13. The Launcher perspective then is populated with the software components and functionality associated with your hardware and stack, Update your device firmware as described in section **Updating Adapter Firmware**.



Finally, open the Release Notes listed under Document Index and check for software version requirements, in particular for IAR-EWARM. To install IAR-EWARM:

1. Go to the Silicon Labs support portal at https://www.silabs.com/support.
2. At the bottom of the page click **E-mail Support Request**.
3. If you are not already signed in, sign in.
4. Click the Software Releases tab, and search for the IAR version required for your Silicon Labs Thread version (in this case 7.80).
5. Download the IAR package (takes approximately 1 hour).
6. Install IAR.
7. In the IAR License Wizard, click **Register with IAR Systems to get an evaluation license**.
8. Complete the registration and IAR will provide a 30-day evaluation license.

3 Functionality in the Launcher Perspective

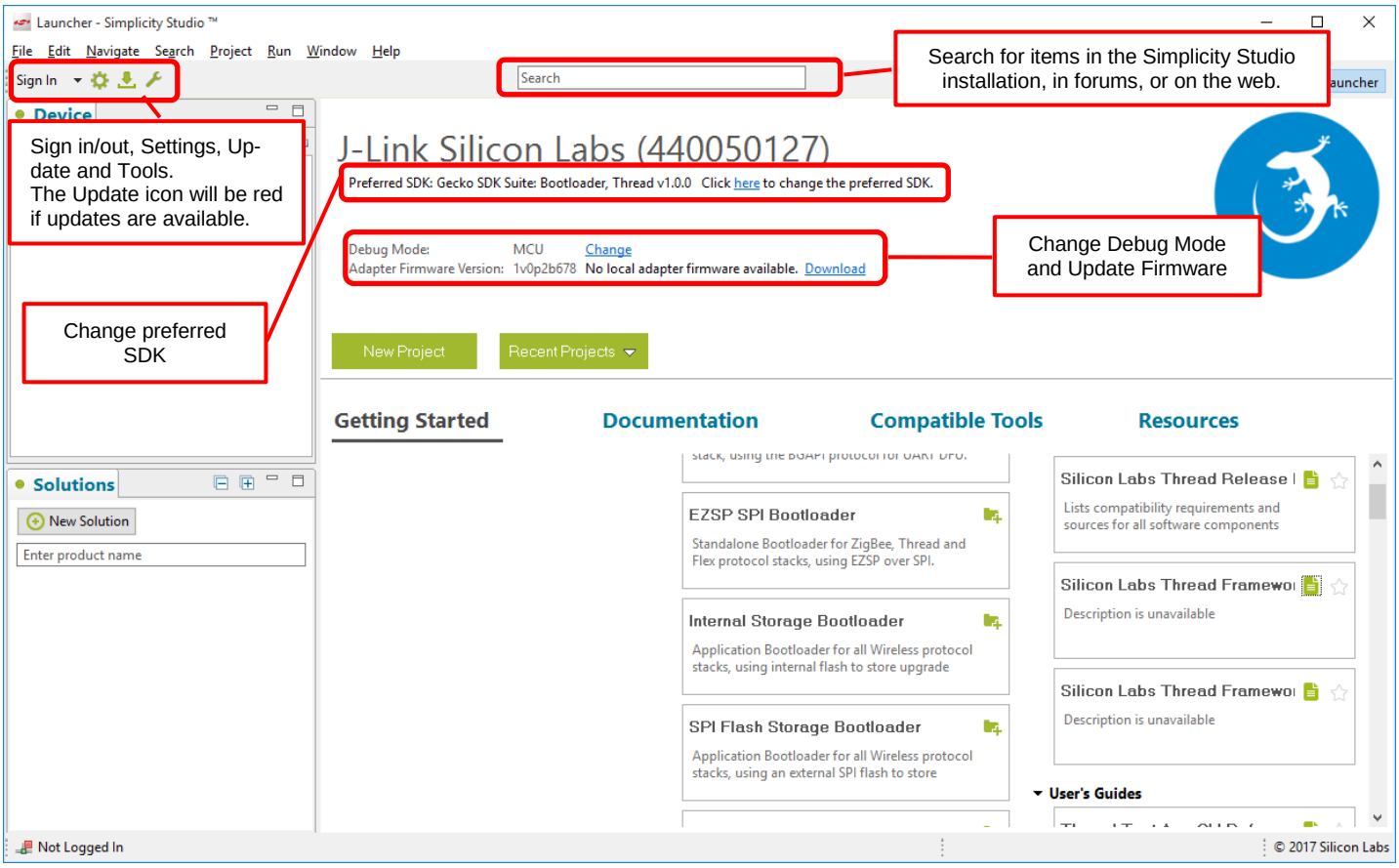
In the Launcher perspective you can perform a number of functions. Additional information on some of these is provided later in the section.

- Sign in (if you are not signed in already)
- Open application settings
- Update your software and firmware (see below for more information)
- Open tools or search

  As you open tools or perspectives, buttons for them are displayed in the upper right. You can use those buttons to easily navigate back to the Launcher perspective or to other perspectives.
- Change your preferred SDK (see below for more information)
- Change debug mode
- Update adapter firmware (see below for more information)
- Access demos, examples, documentation, and other resources (see below for more information)

Note: Perspectives are made up of a number of tiles or panes, called views, as well as the content in those views. You can change the layouts of various perspectives by expanding or relocating views, or adding or removing views. If you want to return to the default layout, right-click the perspective button in the upper right and select **Default.**

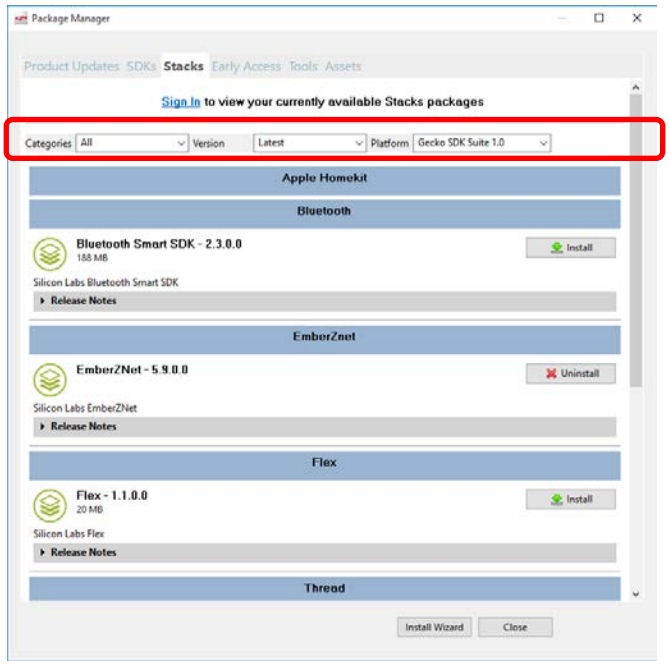
Under the Devices view on the left of the Launcher perspective is a Solutions view. If you developing for complex networks with a number of different parts involved, you can add them all to the solution and then select the one you are working on from the list. You do not need to have the hardware connected to your computer.



3.1 Downloading Updates

The Download Update icon will be red if updates are available. If Simplicity Studio detects an available update, and you are in another perspective, you will be notified that an update is available.

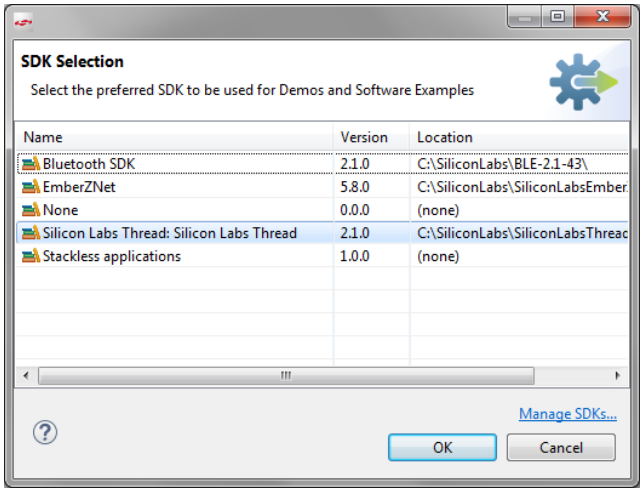
When you click the Download Update icon or accept updates in the notification dialog, Simplicity Studio shows you available updates in the Package Manager dialog. You can update all or select individual updates for installation. Click the tabs in the Package Manager dialog to see other components available for installation. Use the filters to reduce long lists.



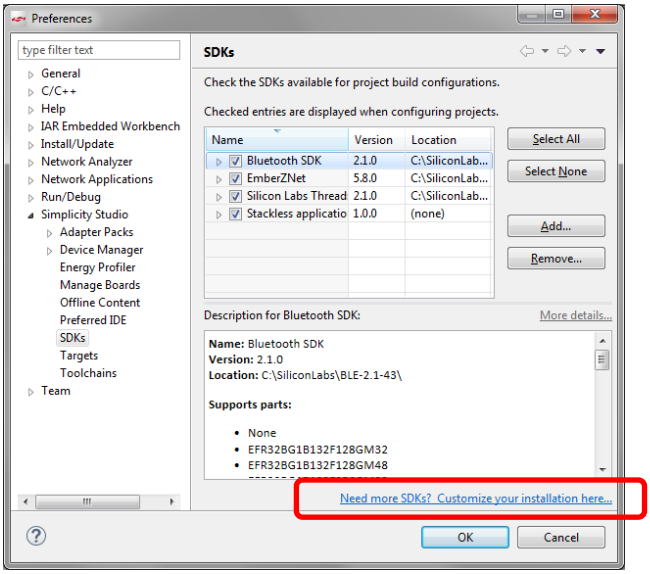
Note: If you are using a new device or product family, you can use the **Install Wizard** button to access the installation interface provided during initial installation. This makes installation of all components related to a selected device(s) or an entire family easier.

3.2 Changing the Preferred SDK

If you have more than one stack or SDK installed, you can change the preferred (or currently active) SDK by clicking one of the links. The SDK selection interface is displayed. Click an SDK, and then click **OK.**



If the SDK you want to use is not displayed, in the SDK Selection dialog click **Manage SDKs.** If you have already installed the SDK, click **Add**, and browse to its location. If you need to install a new SDK, in the Preferences dialog, click **Need more SDKs …**

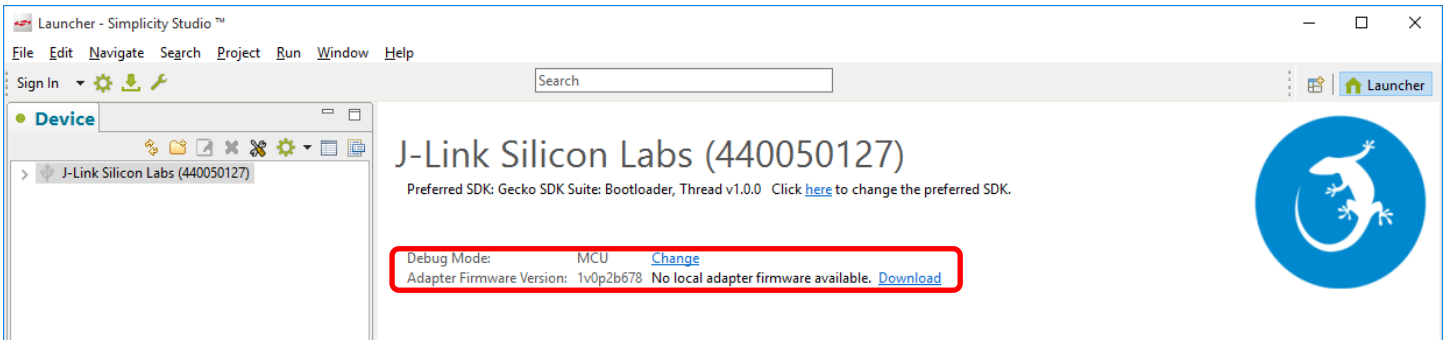


This updates your system and opens the Package Manager interface, the same as if you clicked the Download Updates icon. Click on either the SDK or the Stack tabs to see and install a new item. Then return to the SDK Selection dialog and select the recently installed SDK.

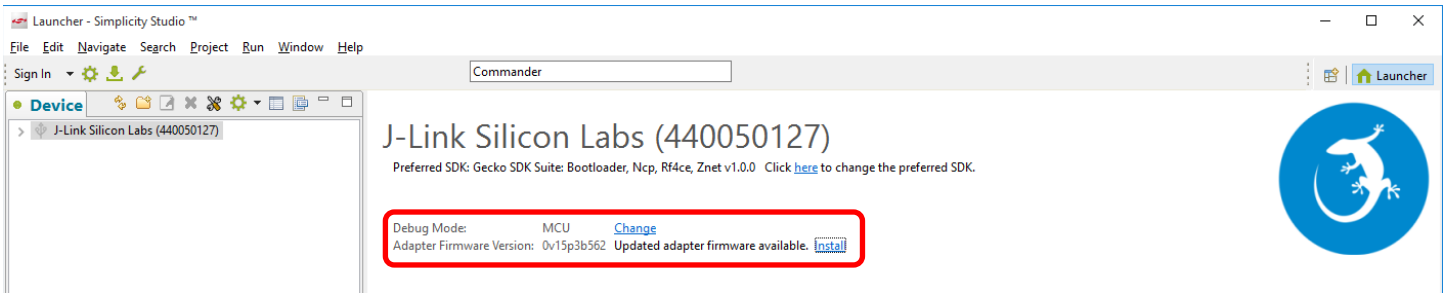
Note: If you are changing to an SDK you have installed outside of Simplicity Studio, first add it, which takes you back to the Launcher perspective. Then click one of the links to change the preferred SDK, select the SDK, and click **OK**.

3.3 Updating Adapter Firmware

Initially the Launcher perspective may display “No local adapter firmware available.” Click **Download** to download any updates.

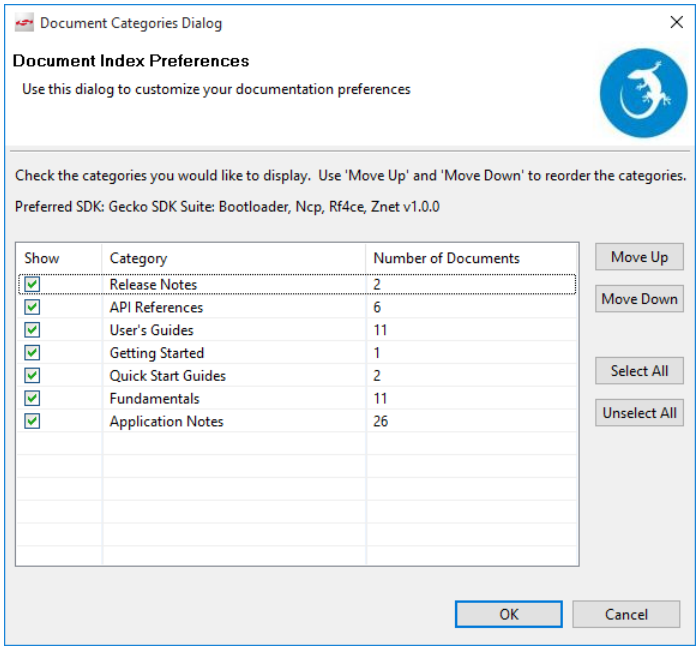


If an update is available, click **Install** to install the firmware.

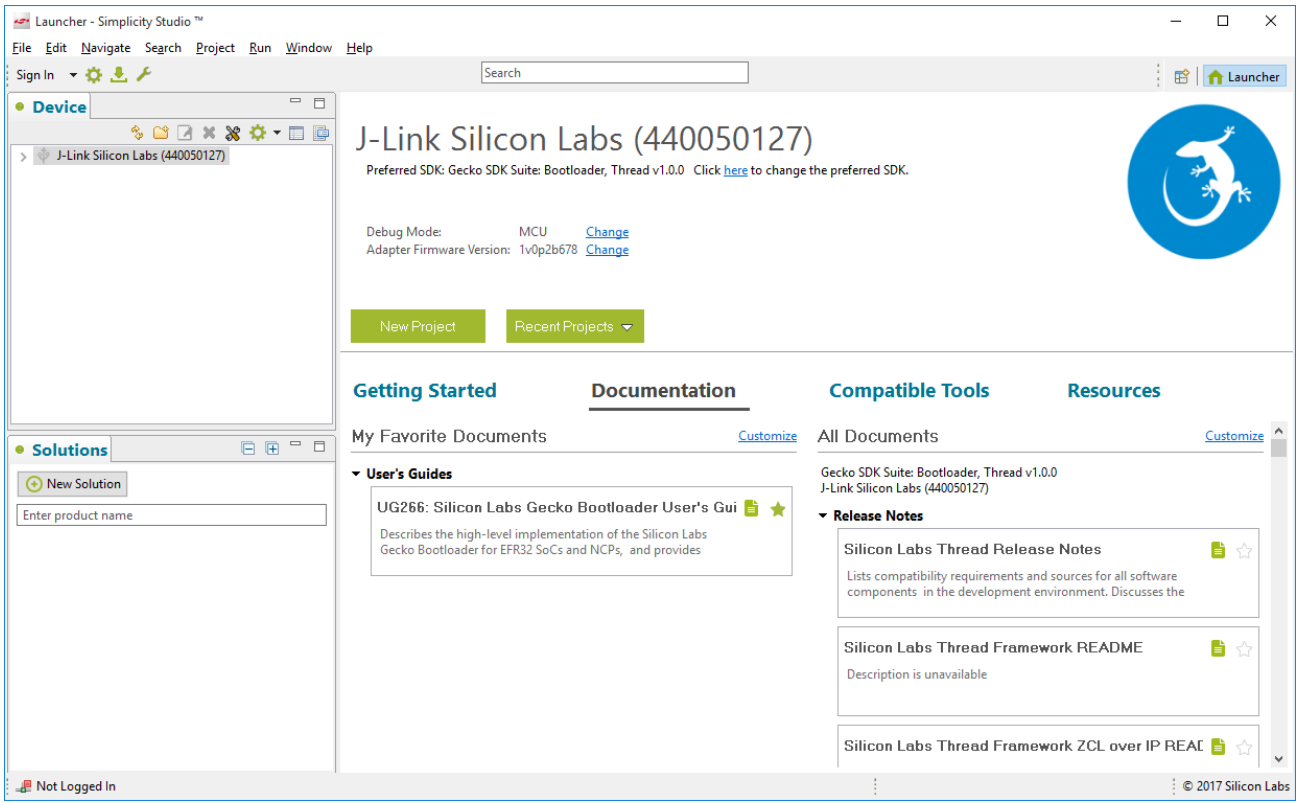


3.4 Accessing Documentation and Other Resources

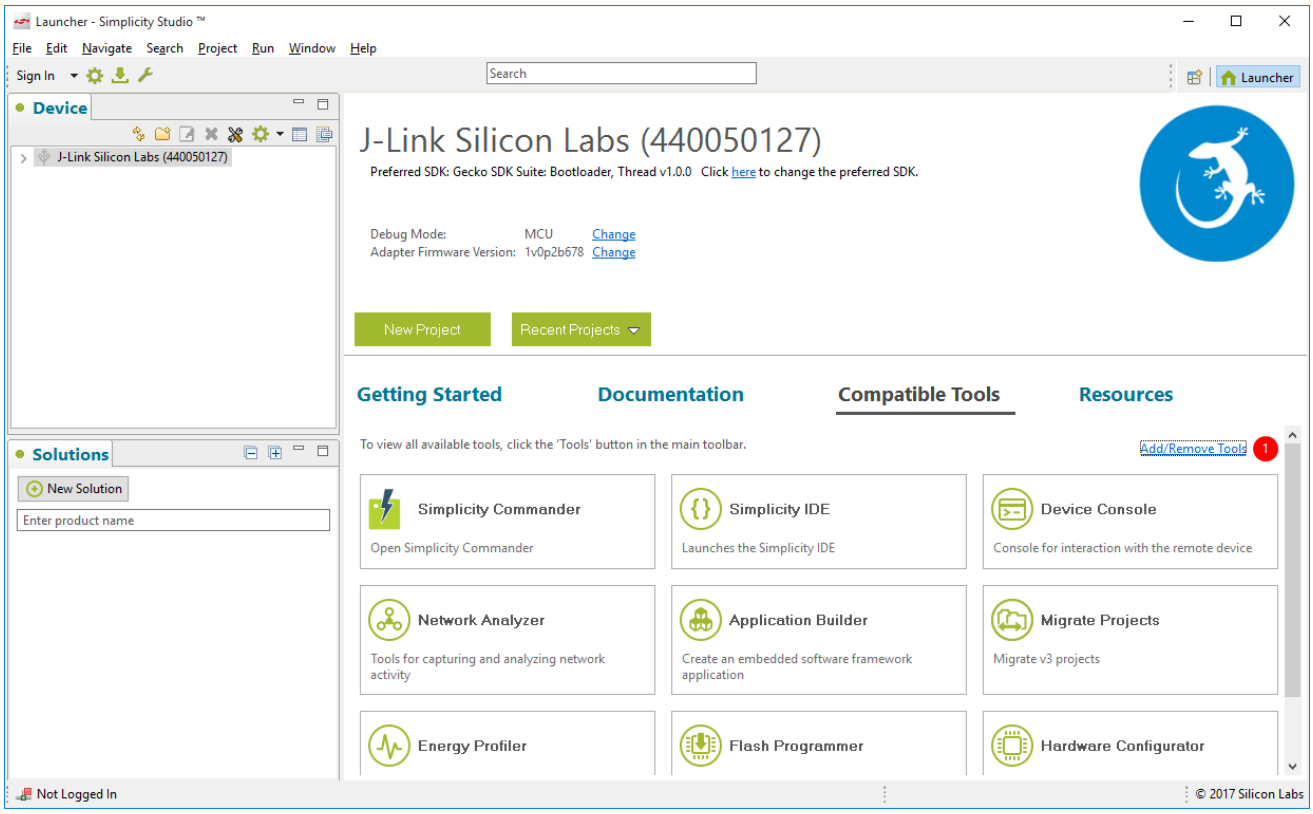
The **Getting Started** tab provides access to demos, example applications, and stack related documentation. To customize the documentation index, click **Customize**. You can select the categories to show and the order in which they will be listed. Click **OK**.



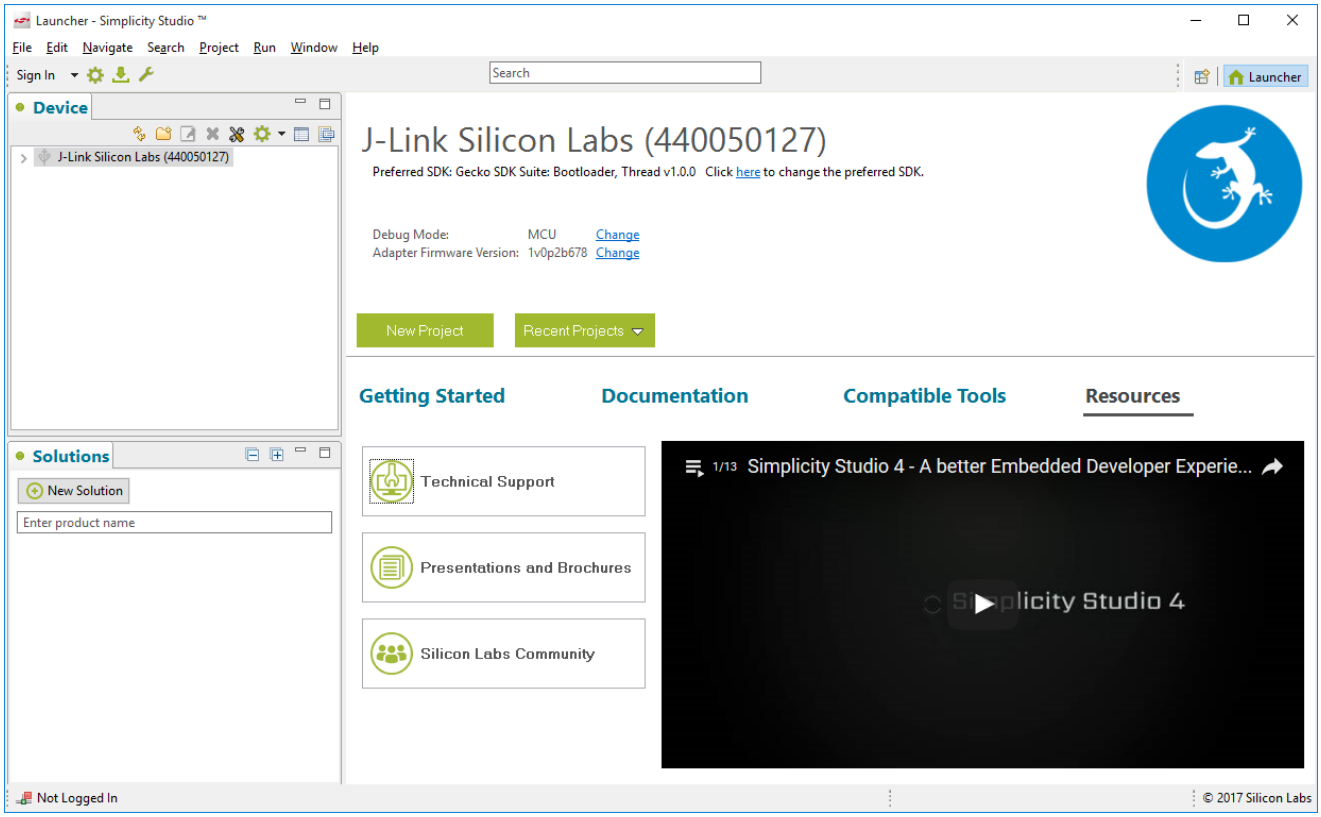
The **Documentation** tab lists documentation about the stack and about the hardware. By default, only Application Notes are displayed. Click Customize to change the categories and the order in which they are displayed. Click the star icon on any document to move it to the My Favorite Documents list. You can customize this list's categories and category order as well.



The **Compatible Tools** tab is an alternative way to access the tools available through the Tools dropdown.



The **Resources** tab provides access to support, marketing collateral, and the Silicon Labs community.



4 Working with Example Applications

In these instructions you will compile and load two example applications, Client and Server, to create a simple sensor-sink Thread network in which the Client broadcasts data to the Server. When the applications are loaded and the Client has joined the network, you can observe traffic across the network in the console, and also in Network Analyzer.

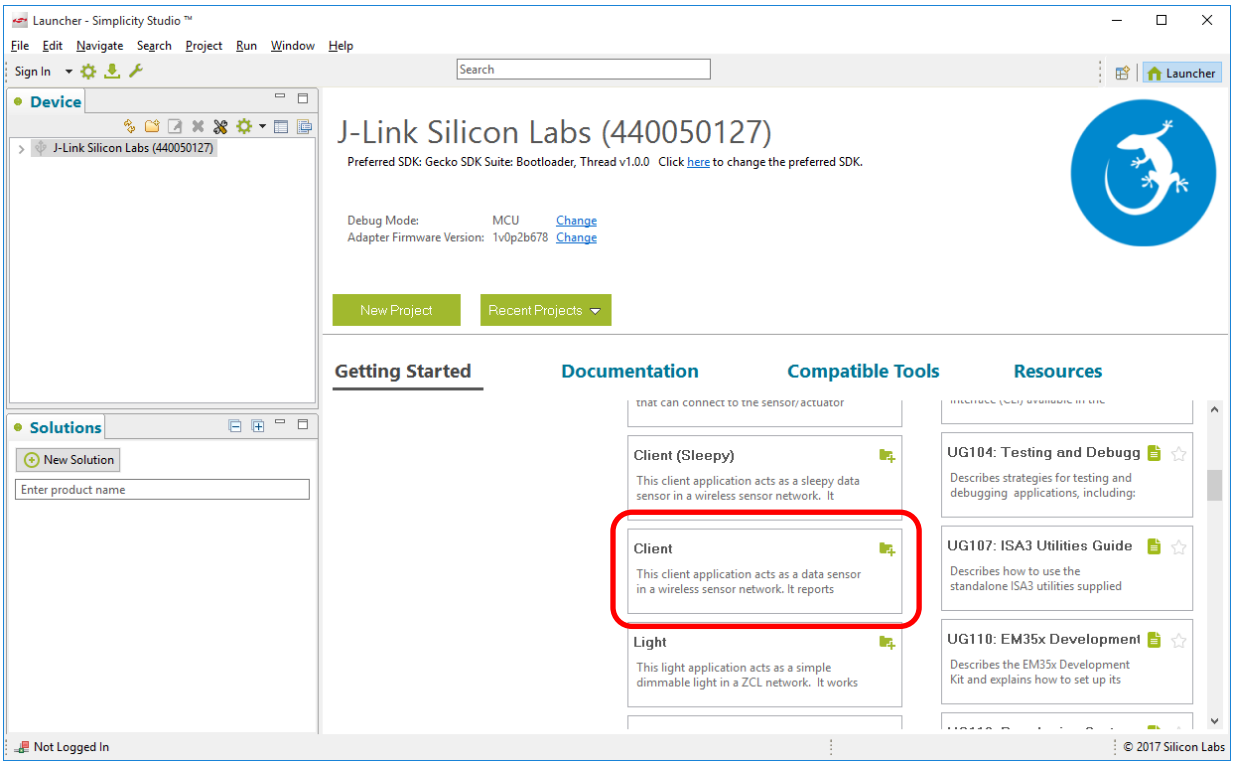
When working with example applications in Simplicity Studio, you will execute the following steps:

1. Select an example application.
2. Generate application files.
3. Compile and flash the application (and, the first time, a bootloader) to the radio board.
4. Interact with the application.

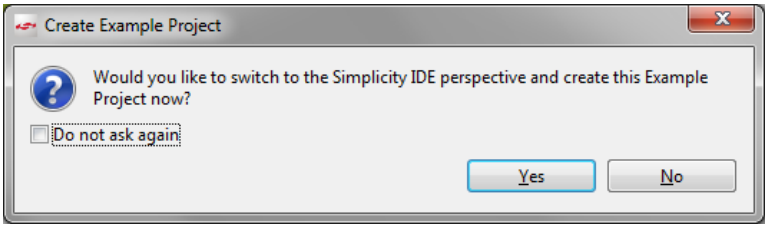
These steps are described in detail in the following sections. These procedures are illustrated for a WSTK with an EFR32MG. Except as noted, the procedure for an ISA3 connected to an EM35x is the same.

4.1 Selecting an Example Application

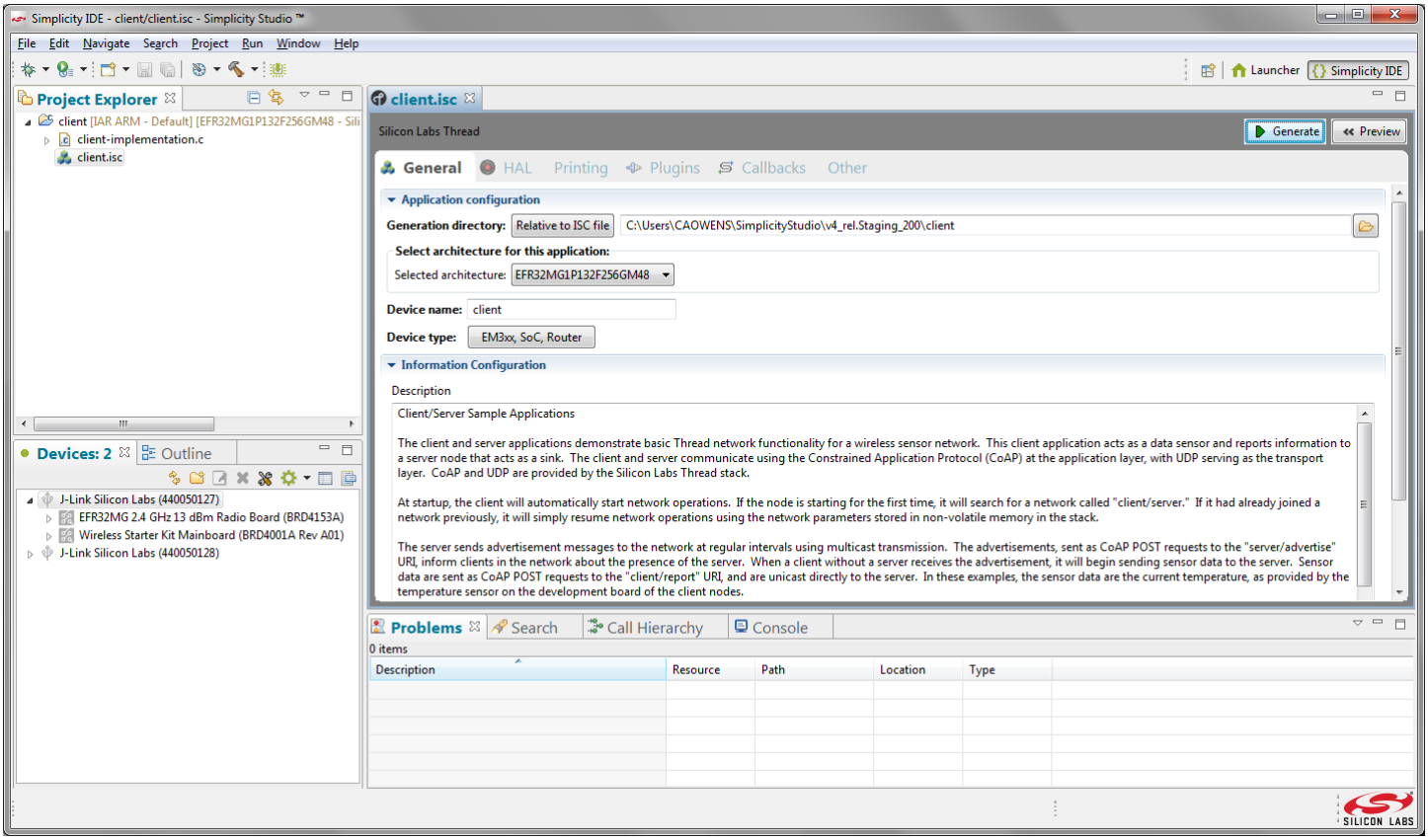
1. In the Launcher perspective, click an example application, in this case **Client**. Your project will be based on this example, and on the device you have selected in the Devices or Solutions tabs on the left.



2. You are asked if you want to switch to the Simplicity IDE. Click **Yes**.



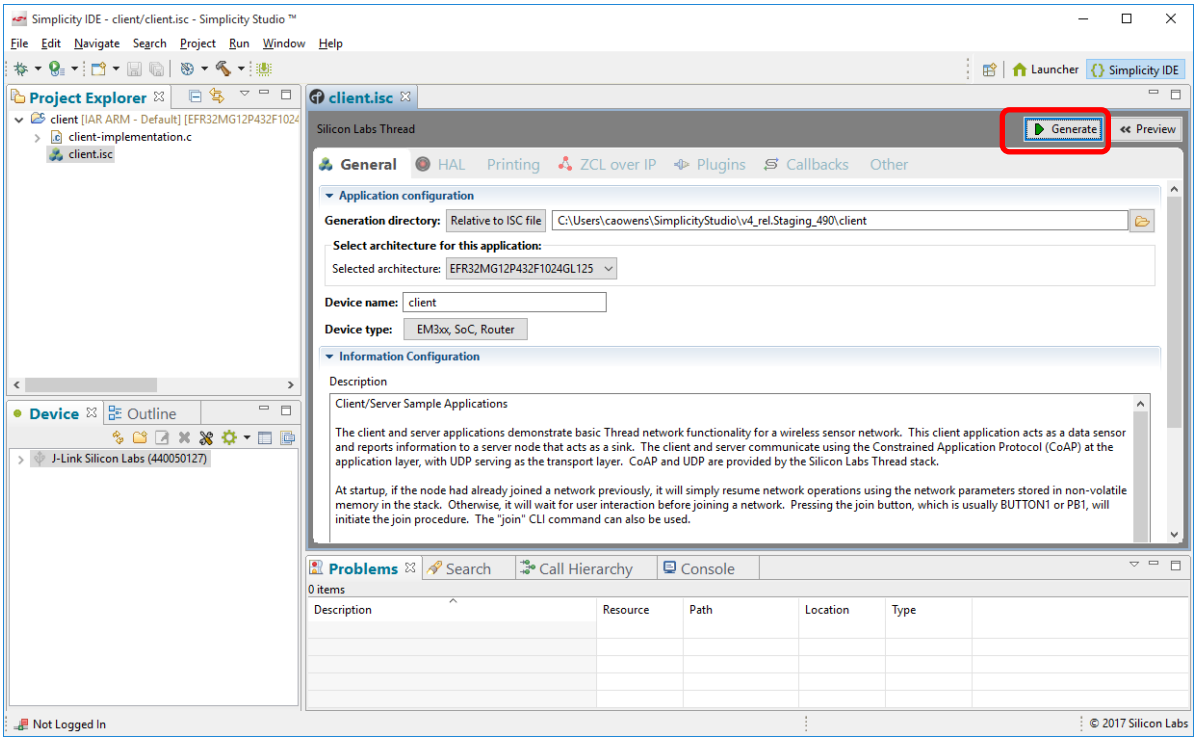
3. Simplicity IDE opens with the new project in AppBuilder view, and the focus on the General tab.



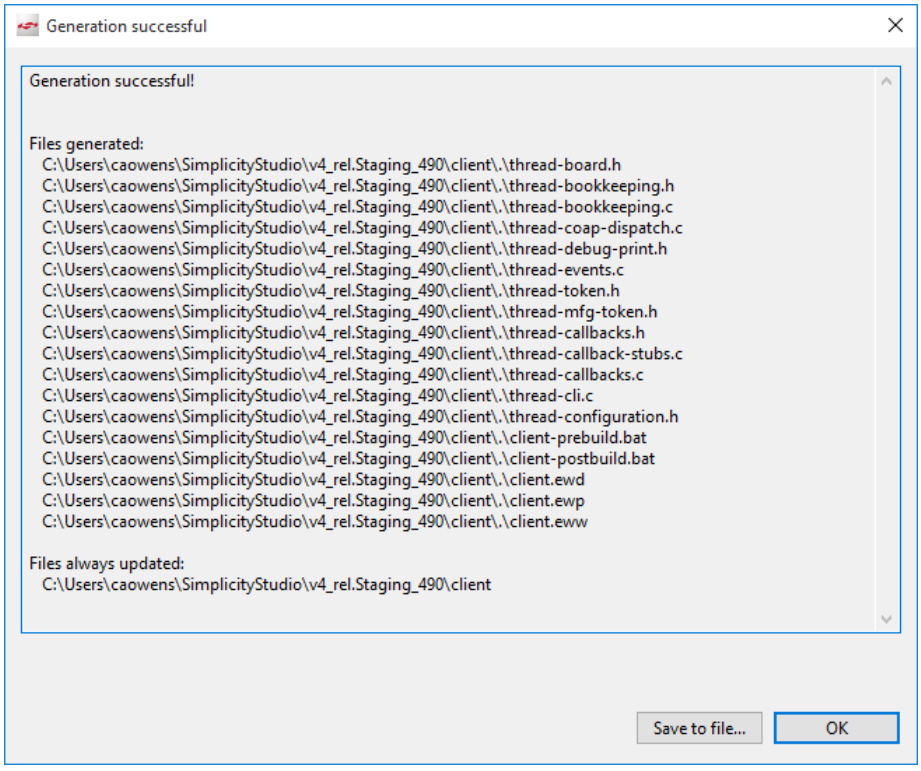
Note: You now have a Simplicity IDE button next to the Launcher button in the upper right.

4.2 Generating the Application Source Files

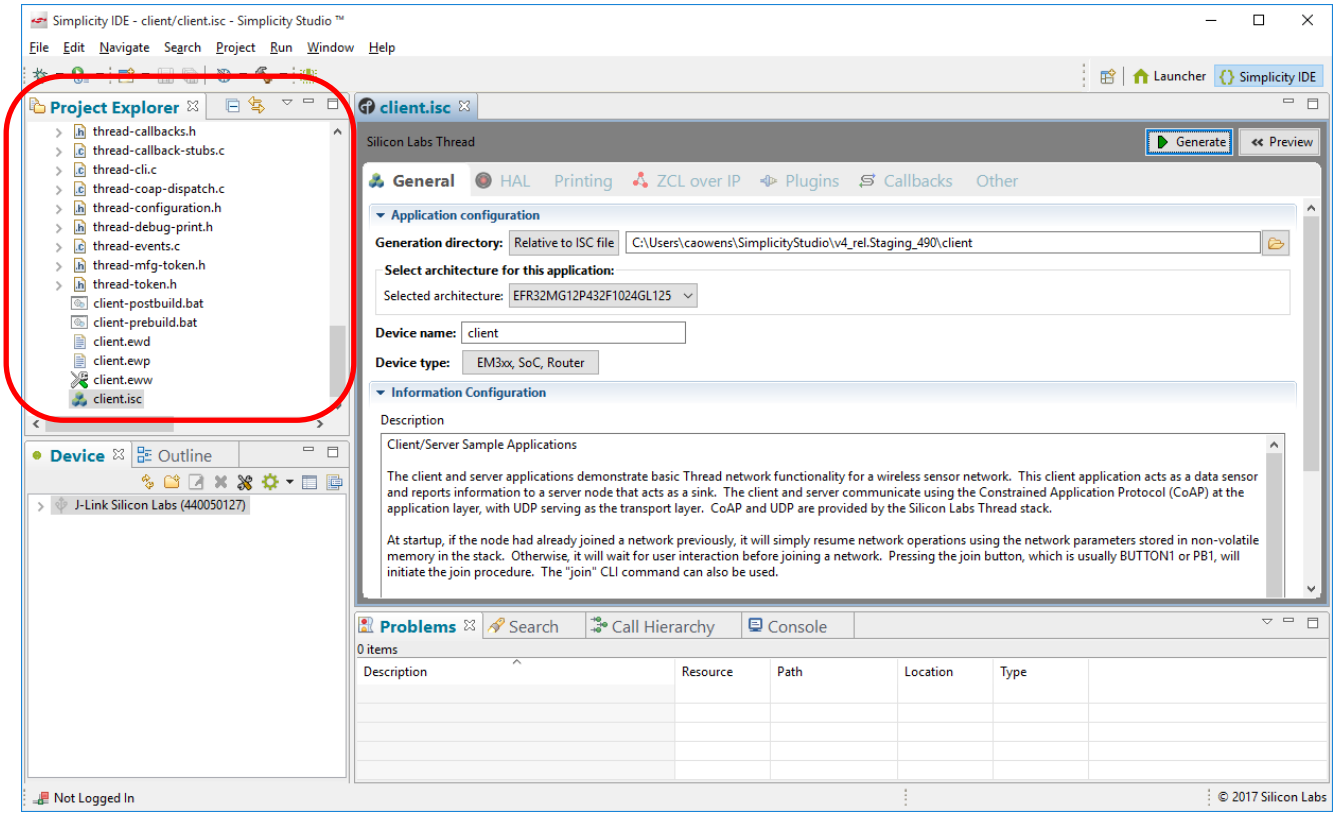
1. In the Simplicity IDE, click **Generate.**



2. Once generation is complete, a dialog reporting results is displayed. Click **OK**.



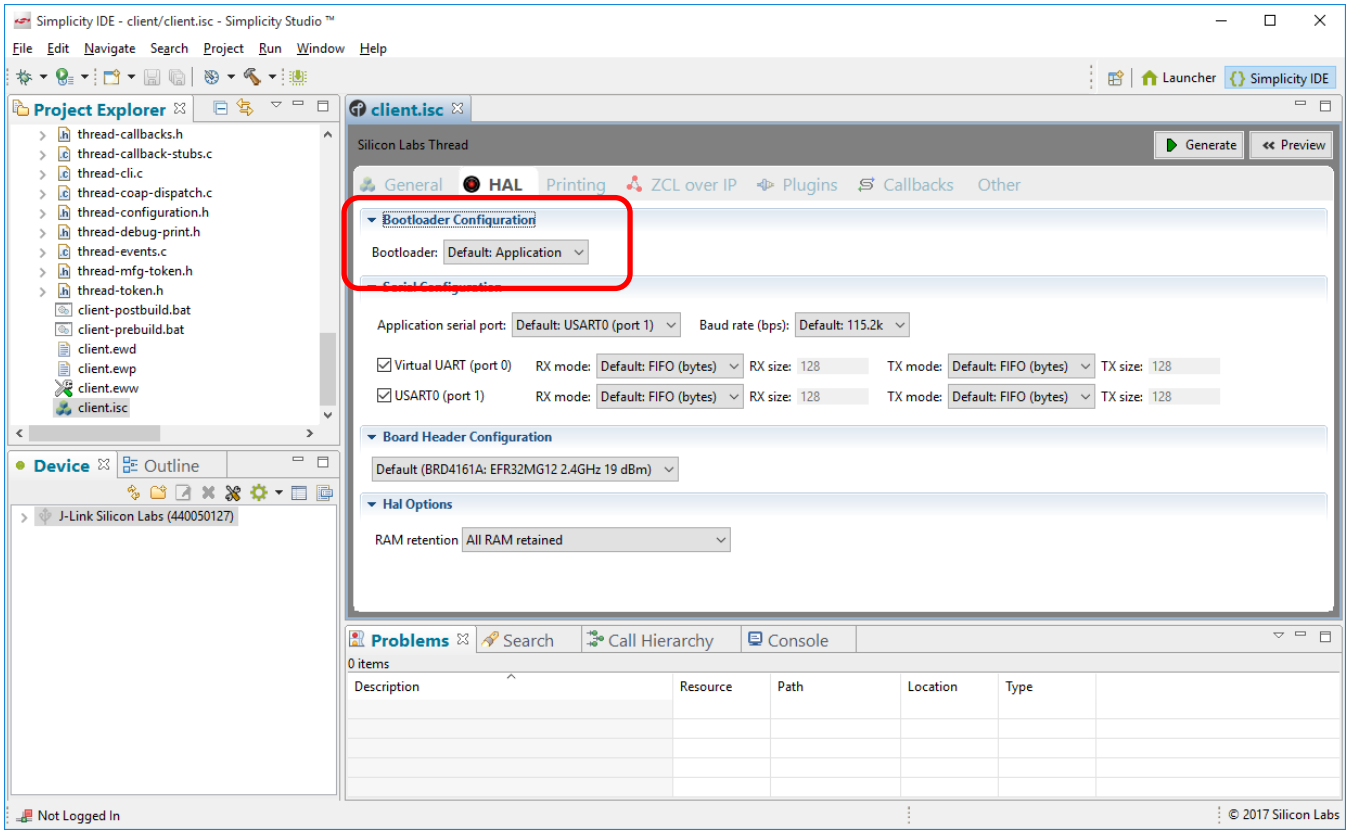
The generated files are shown in the Project Explorer view.



4.3 Compiling and Flashing the Application

4.3.1 About Bootloading

Because this sample application is built with a bootloader (configured under HAL configuration tab), you will need to load a bootloader before you run the application for the first time.



A bootloader is a program stored in reserved flash memory that can initialize a device, update firmware images, and possibly perform some integrity checks. Silicon Labs networking devices use bootloaders that perform firmware updates in two different modes: standalone (also called standalone bootloaders) and application (also called application bootloaders). An application bootloader performs a firmware image update by reprogramming the flash with an update image stored in internal or external memory. Silicon Labs recommends that you always flash a bootloader image along with your application, so that flash memory usage is appropriately allocated from the beginning. For more information about bootloaders see *UG103.6: Application Development Fundamentals: Bootloading*.

In March of 2017, Silicon Labs introduced the Gecko Bootloader, a code library configurable through Simplicity Studio's IDE to generate bootloaders that can be used with a variety of Silicon Labs protocol stacks. The Gecko Bootloader can be used with EFR32MG1/EFR32BG1 (EFR32xG1) however, beginning with the EFR32MG12/ EFR32BG12/ EFR32FG12 (EFR32xG12) platform, it and all future Mighty Gecko, Flex Gecko, and Blue Gecko releases will use the Gecko Bootloader only. Legacy Ember bootloader applications for use with specific protocols such as Silicon Labs Thread and platforms including the EM3x will continue to be provided for use with those platforms.

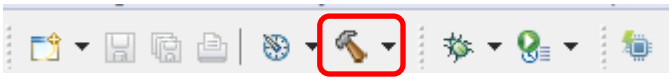
The bootloaders work with specialized firmware update image formats. The legacy Ember bootloader update images end in extension .ebl; Gecko Bootloader update images end in extension .gbl. When you build an application both .s37 and update image files are generated. The update image file format depends on the hardware you selected. EBL files are generated for EM3x and EFR32xG1. GBL files are generated for EFR32xG12 and later devices. If you want to use the Gecko Bootloader on EFR32xG1 devices, you must convert the .S37 file using Simplicity Commander, as described in *UG162: Simplicity Commander User Guide*.

Note: When working with the Gecko Bootloader, you must use Simplicity Commander to enable some configuration option such as security features. See *UG266: Simplicity Studio Gecko Bootloader User's Guide*.

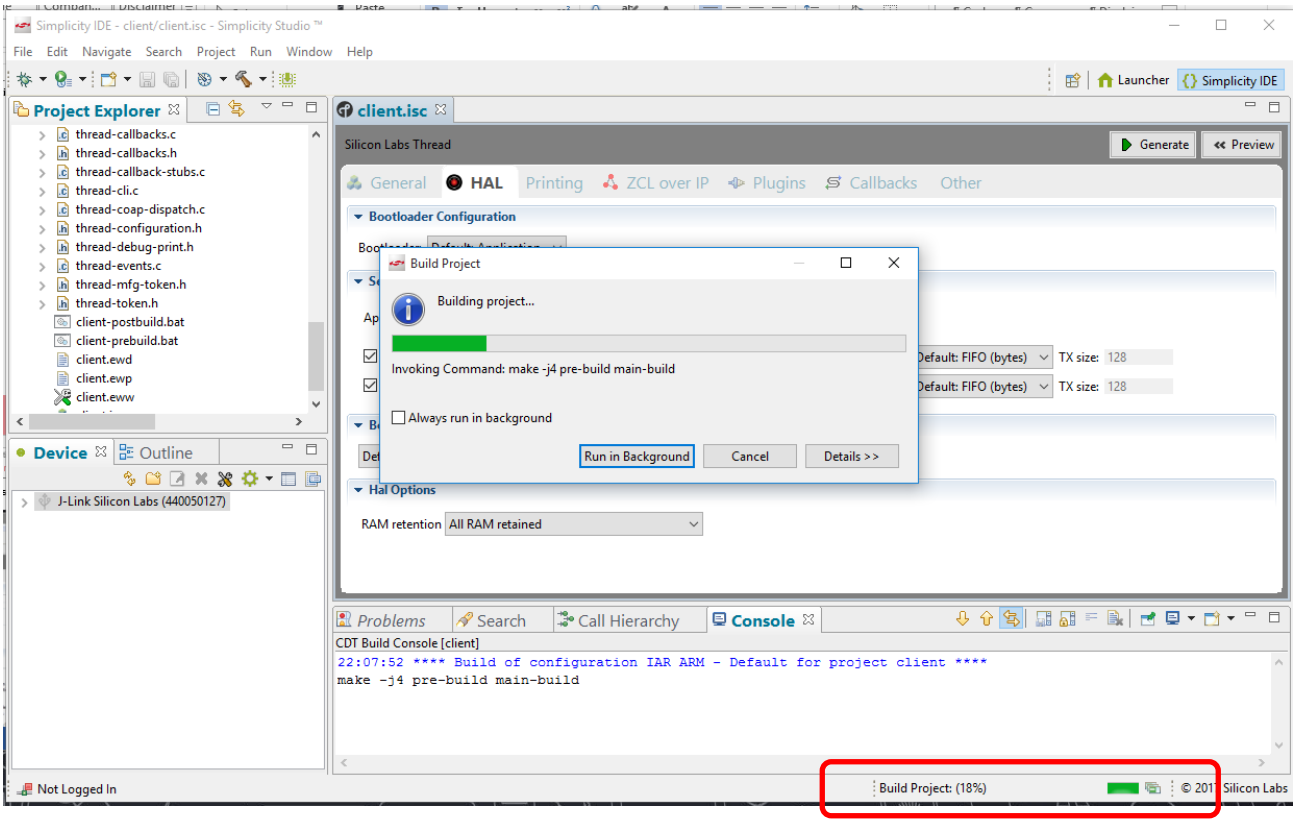
4.3.2 Building and Flashing Files

Note: The following procedure differs slightly depending on whether you have connected your debug adapter to your computer by USB (the default for the WSTK/EFR32 configuration) or by Ethernet (the default using an EM3x/ISA3 configuration).

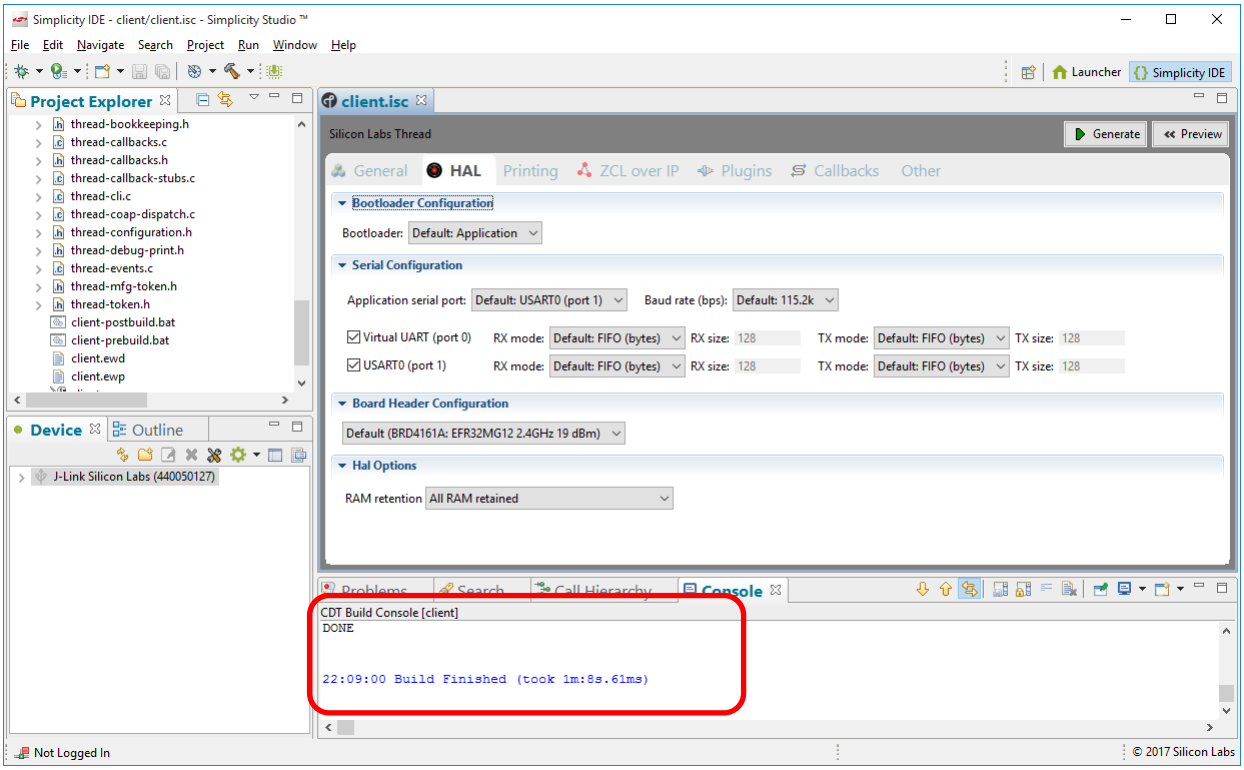
1. After you generate your project files, click the **Build** control in the top tool bar. If the Build control is not enabled, click the device. Your sample application will compile based on its build configuration. You can change the build configuration at any time in the Project Explorer View by right clicking on the project and going to **Build Configurations > Set Active**.



During the build, progress is reported both in a window, which can be run in the background, and also in the lower right.

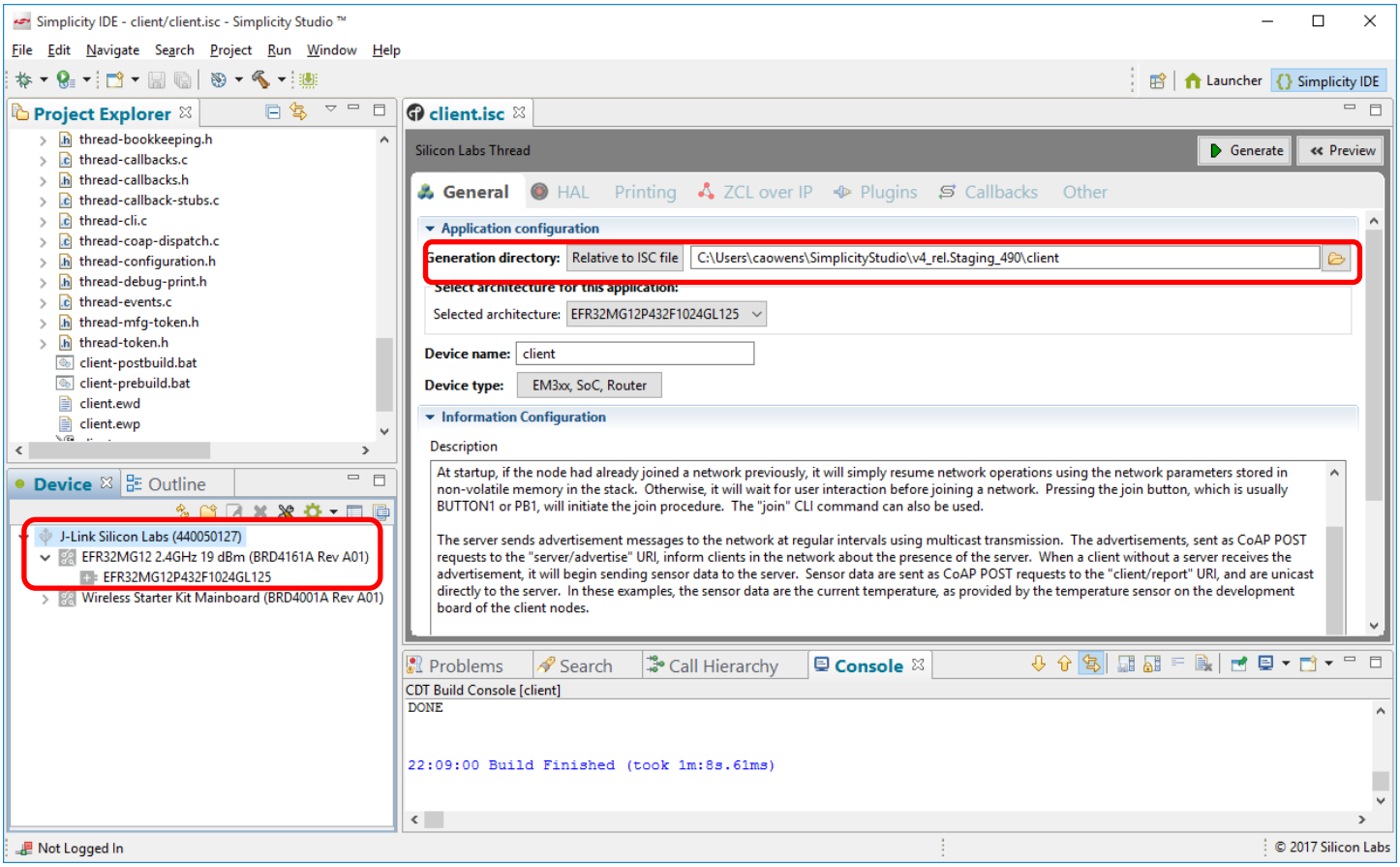


Build completion is reported in the Build Console.

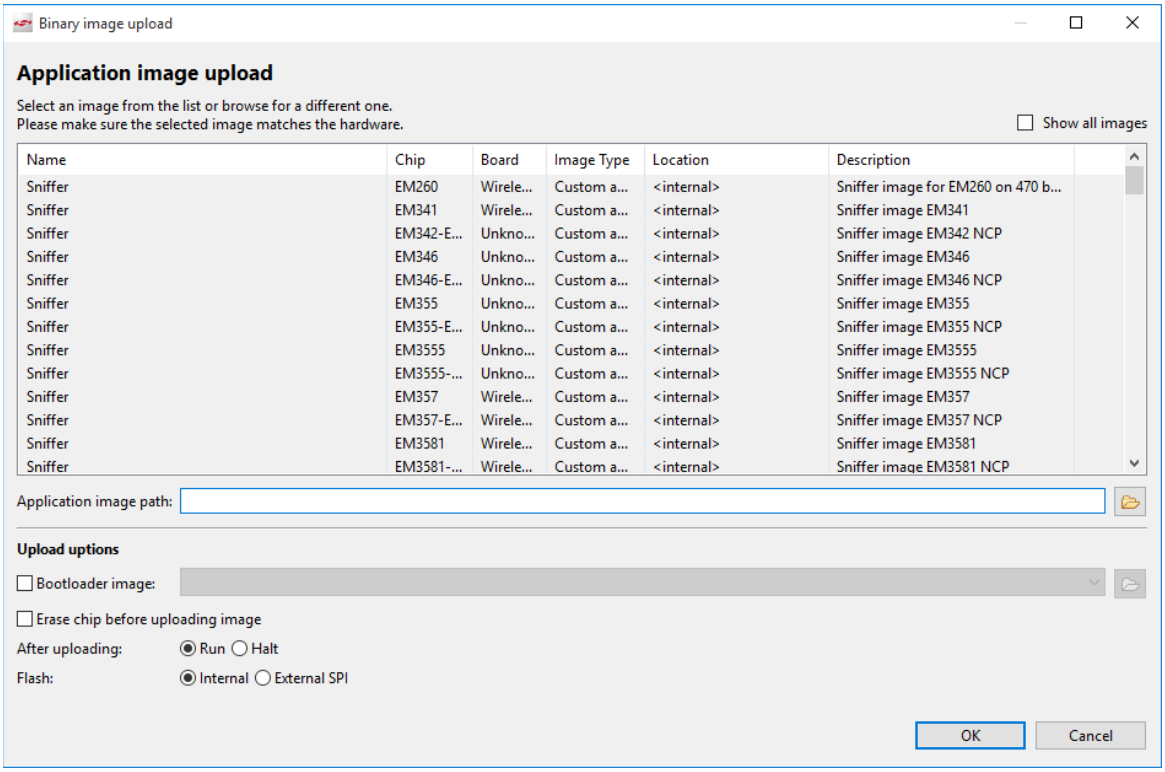


The build should complete with no errors. If any errors occur they are highlighted in the console. Contact technical support for assistance.

2. To load the application and the bootloader images, first make sure your hardware is displayed in the Device perspective. Expand the radio board to show the part number, as you will need that to find the correct bootloader file. Note that the folder in which the example was generated is displayed on the General tab.

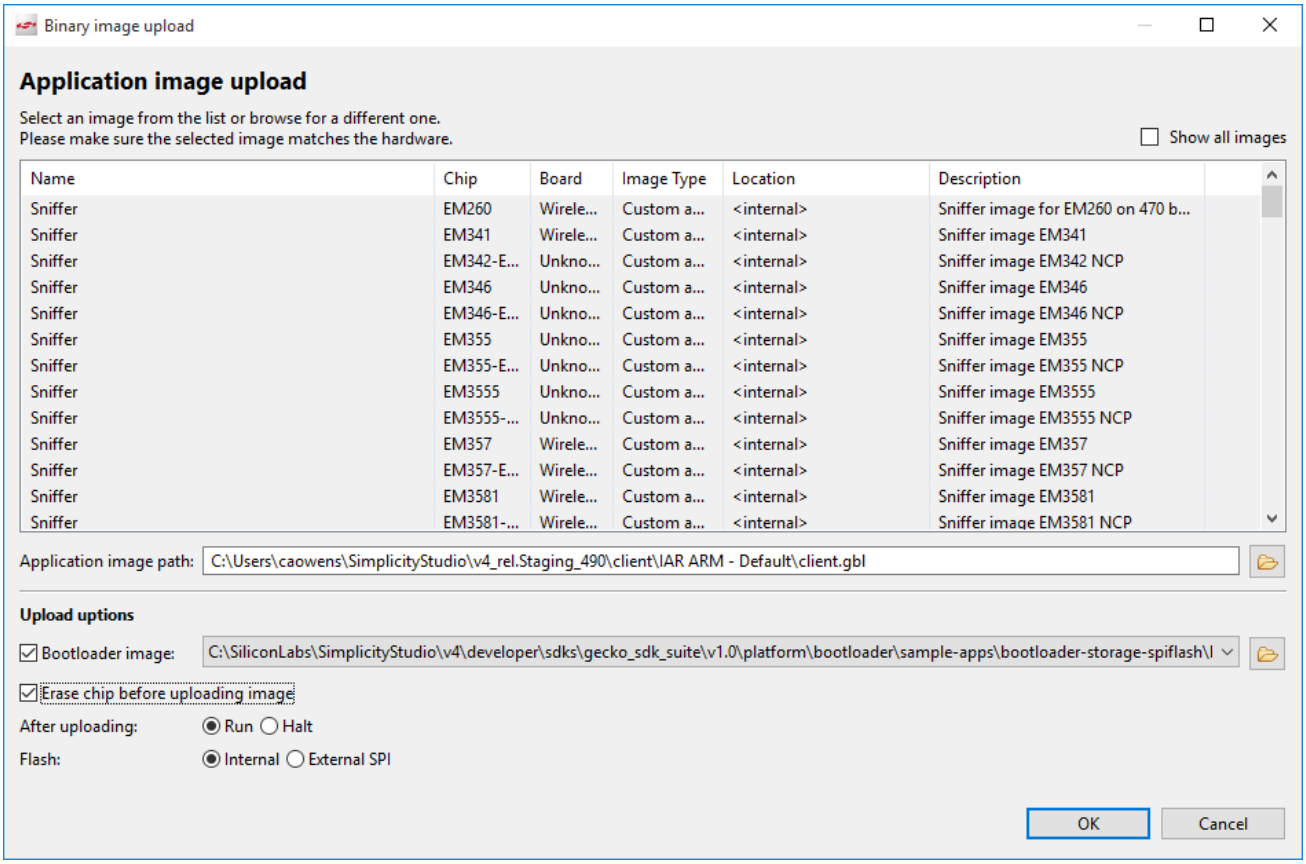


3. Only if you are connected by Ethernet (typical when using an EM3x development kit), right-click on your device and select **Connect**. Then go to the next step.
4. Right-click on the device and select **Upload Application**. The Application Image Upload dialog is displayed. Your list may differ from the following example.

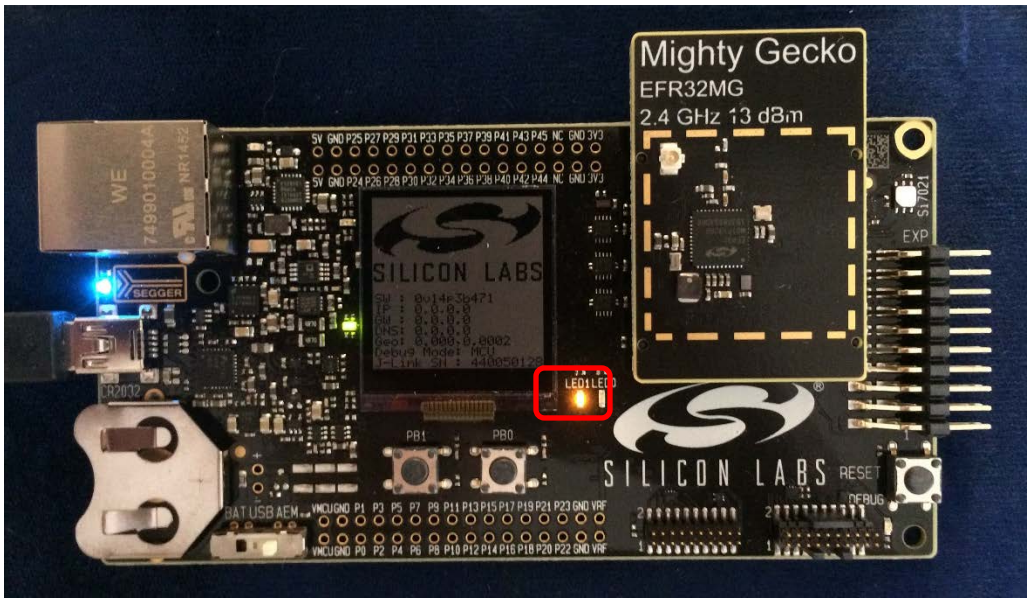


5. Browse to the folder with your compiled application (`<folder on General tab>\IAR ARM - <qualifier>\`) and select the .gbl or.ebl file.
6. If you have not already loaded a bootloader, check **Bootloader image**, then browse to the folder containing a prebuilt bootloader image corresponding to the part number for your radio board.
   - **For legacy bootloaders**: Images are located in the stack folder, which by default is installed under Simplicity Studio (for example, `C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\<version>\protocol\thread_<version>\`**)**. Browse to the Tool folder, then to the bootloader folder that is specific to your radio board part number. Open the app-bootloader-spiflash folder and select the .s37 file.
   - **For the Gecko bootloader:** Images are located in the Simplicity Studio bootloader folder under platform (for example: `C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\<version>\platform\bootloader\`). Browse to sample-apps and the bootloader-storage-spiflash folder. Select the .s37 file corresponding to the radio board part number, for example 'bootloader-storage-spiflash-efr32mg12p432f1024gl125.s37'.

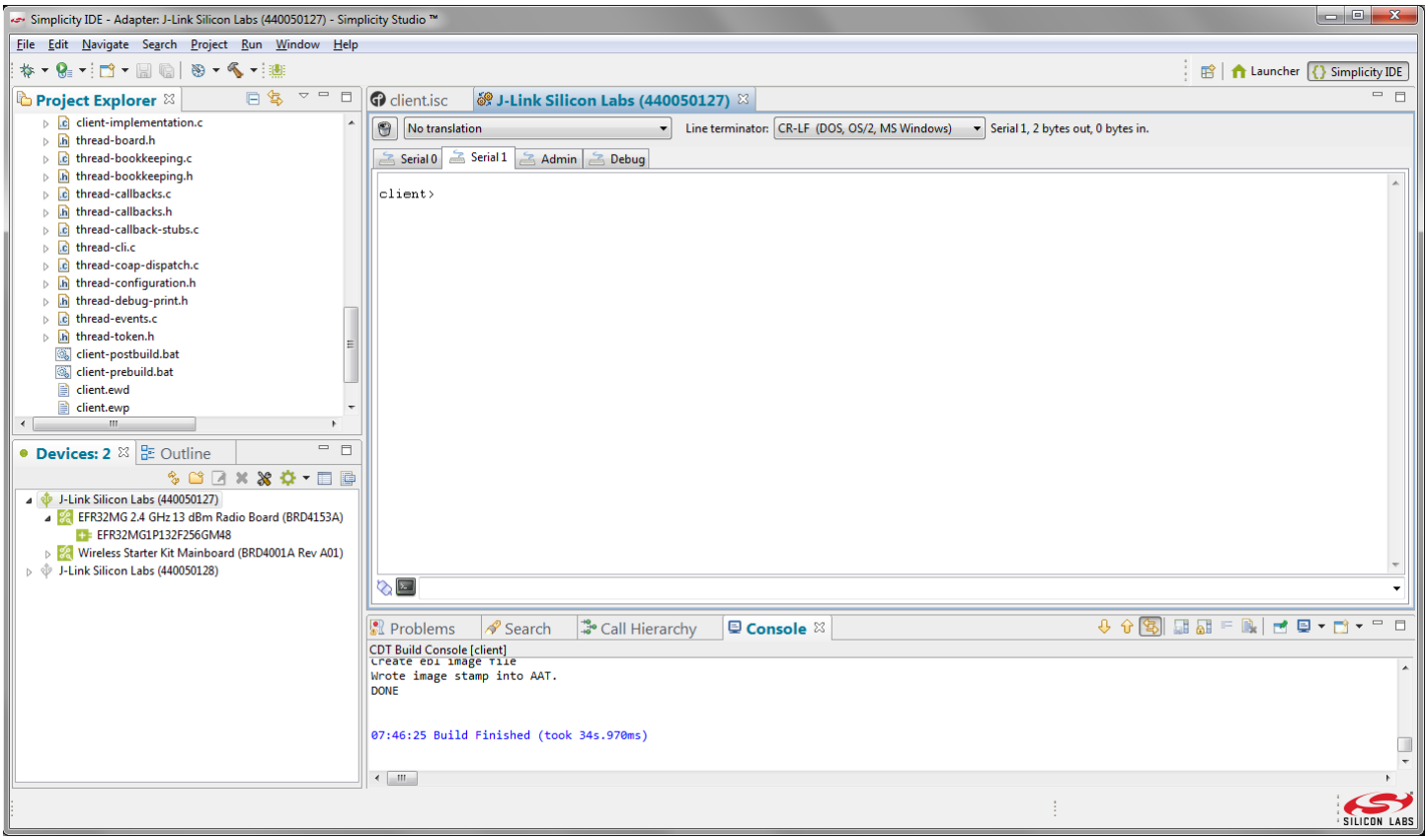
7. Check **Erase Chip**, to make sure that the main flash block is erased before your new image is uploaded. New users will typically always check this. Your completed dialog should resemble the following:



8. The **After uploading** options are **Run** (begin executing the code immediately) and **Halt** (wait for an event, such as a debugger to connect or manual initiation of a boot sequence). During initial development you will typically leave this set to **Run**.
9. The Flash options determine the storage location, and are **Internal** and **External SPI**. Leave the option set to **Internal**.
10. Click **OK.** Load progress is shown in the lower right. When the load progress clears, on the WSTK you know that the application has loaded if LED1 is flashing a heartbeat.



11. You can also right-click on the device and select **Launch Console**. In the console window, click the **Serial 1** tab, and press enter. You should see a prompt that corresponds to the project name. Note that the icons next to the device are now green, indicating a serial connection to the console.



Now repeat the procedure for the Server example, by clicking Launcher in the upper right, and following the steps beginning in section **Selecting an Example Application**. Like the Client example, the Server example displays a heartbeat on LED1, and you can see the name of the project in a prompt on the Serial1 tab of a connected console.

Note: Before you can load a different application, you must disconnect from the console. Right-click the device and select **Disconnect**.

5 Creating a Network

Once you have downloaded both the client and server applications you can create a network.

At startup, both the Client and the Server automatically start network operations.

The first time the Server starts, it forms a new network called "client/server." The server sends advertisement messages to the network at regular intervals using multicast transmission. These inform clients in the network about the presence of the server. The server console displays messages such as:

```
Advertising to ff33:40:fdbb:d793:d131::1
```

When the Client starts, it searches for a network called "client/server." When attempting to join the network, the client node prints its unique join key to the console, for example:

```
Joining network "client/server" with EUI64 >000B57FFFE07AFAB and join key "LP7TUCB0"
```

In order for the client to join the network, the server must have the join key. Enter `expect “<join key>”` on the server's console, where <join key> is your client's join key, for example:

```
expect "LP7TUCB0"
```

The server responds:

```
Sent steering data
server>
```

Once the server has received the join key, the client can join the network and, once it attaches to the server, can begin sending data.

```
Joining network "client/server" with EUI64 >000B57FFFE07AFAB and join key "LP7TUCB0"
Joined network "client/server"
Waiting for an advertisement from a server
Bound to fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Attached to server at fdbb:d793:d131:0:3e4e:5d70:f141:6445
Reporting 32000 to server at fdbb:d793:d131:0:3e4e:5d70:f141:6445
```

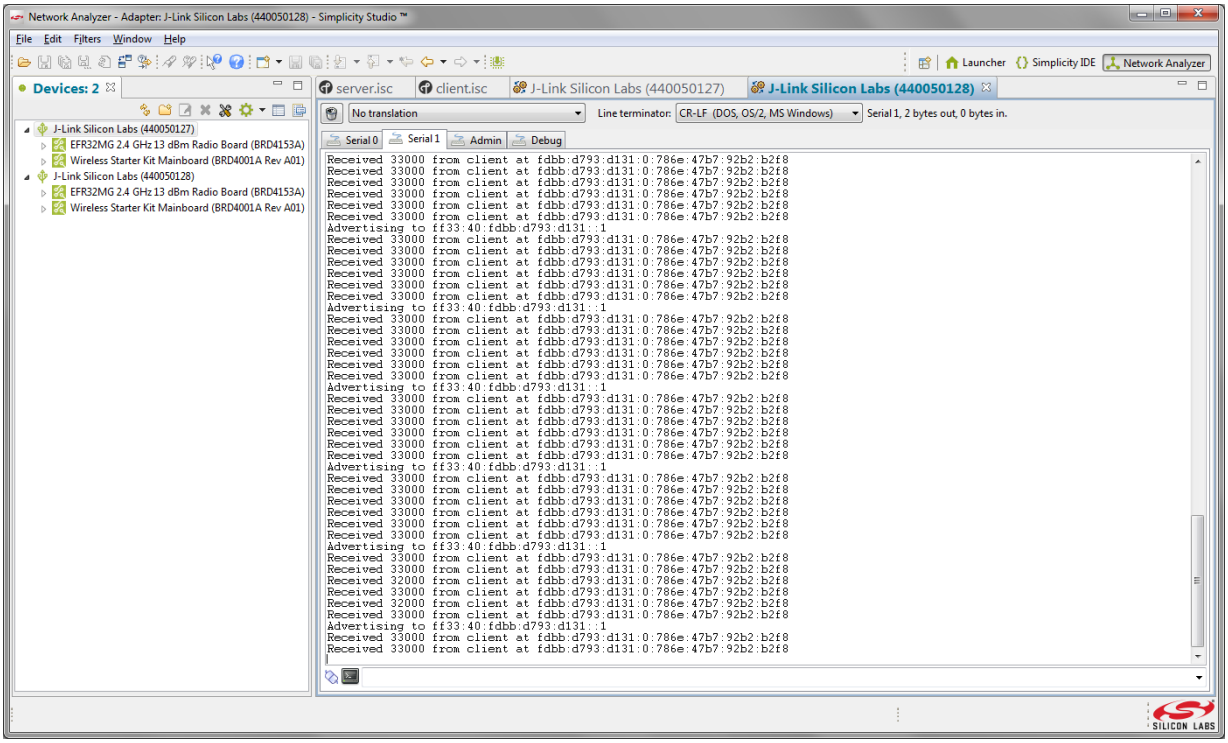
The server receives data and periodically advertises:

```
Advertising to ff33:40:fdbb:d793:d131::1
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Advertising to ff33:40:fdbb:d793:d131::1
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
Received 32000 from client at fdbb:d793:d131:0:786e:47b7:92b2:b2f8
```

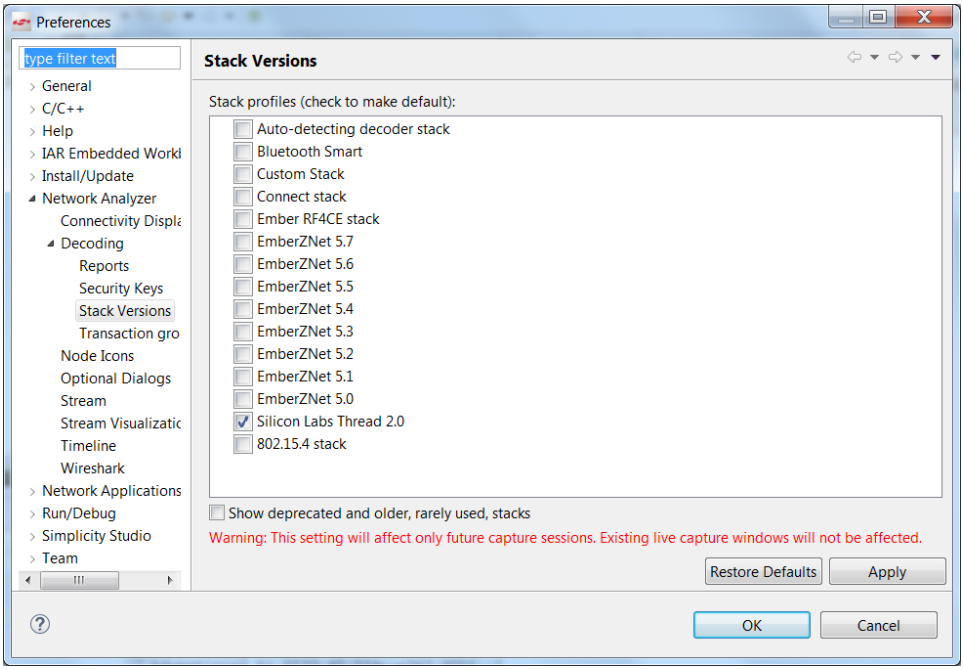
6 Using the Network Analyzer

Now that your network is functioning, you can evaluate the data being transmitted using the Network Analyzer tool.

1. Click the Launcher button in the upper right, and select Network Analyzer from the Tools menu. The Network Analyzer opens with your console window(s) still displaying data.

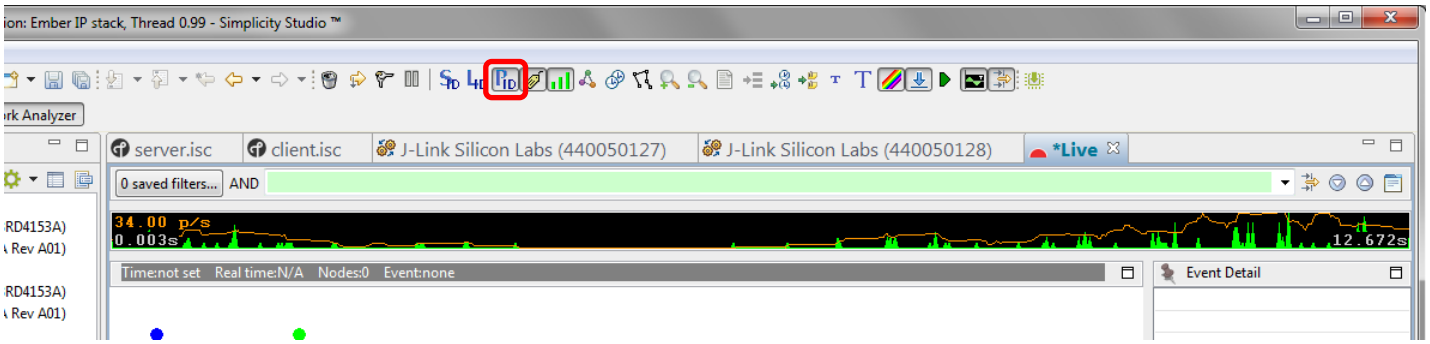


2. Make sure that Network Analyzer is set to decode the correct protocol. Select **File > Preferences > Stack**, verify it is set correctly. If you need to change it, click the correct stack, click **Apply**, then **OK**.



3. Right-click on the Sensor device, and select **Start Capture**. Do the same for the Client device.
4. If you are in an environment with a number of wireless devices, you may have a very noisy Network Analyzer environment, as reflected both in the event traffic and in the map. To show additional information in the map, click on the map.

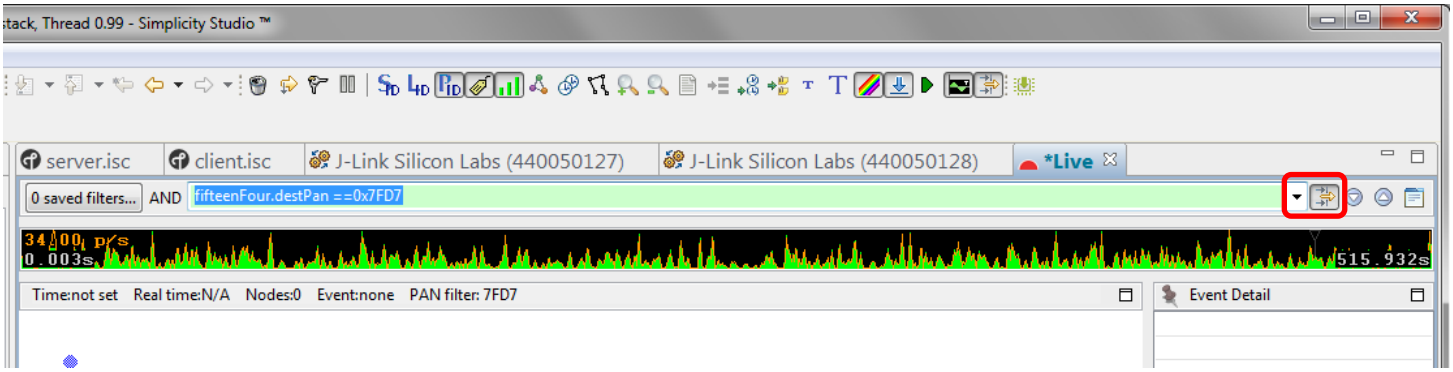
On the toolbar, click the PAN ID button: 



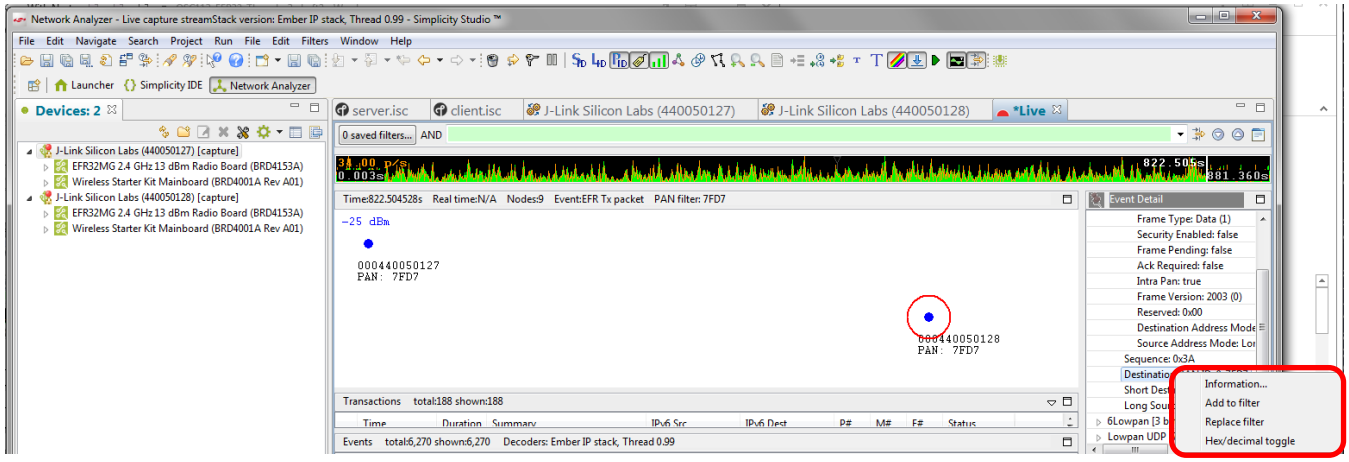
5. Right-click on the representation of your Server device and select **Show only this PAN**.
6. To filter the event traffic, in the green filter bar enter:

```
fifteenFour.destPan == 0x<PANID>
```

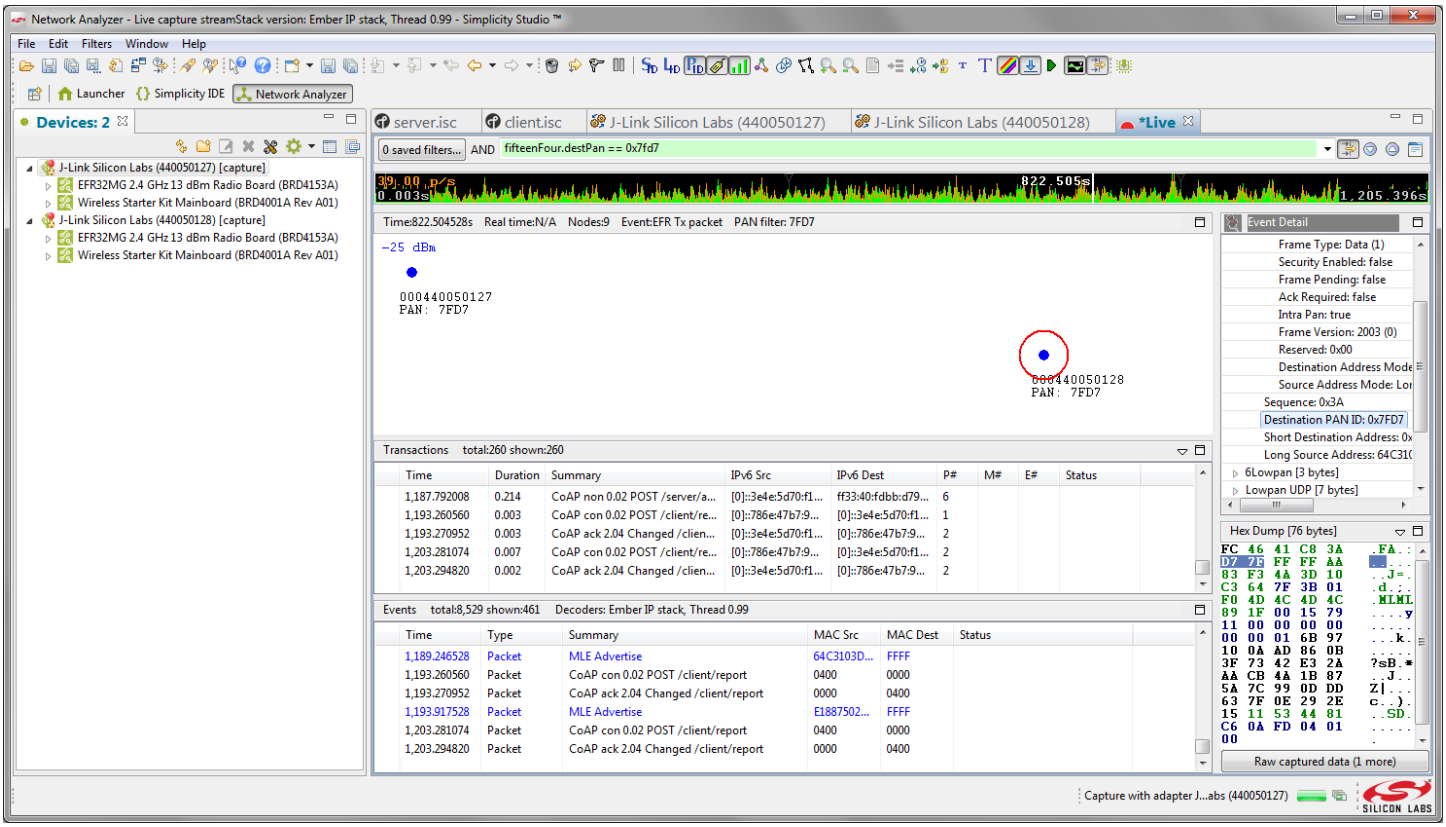
Click the Apply Filter button. 



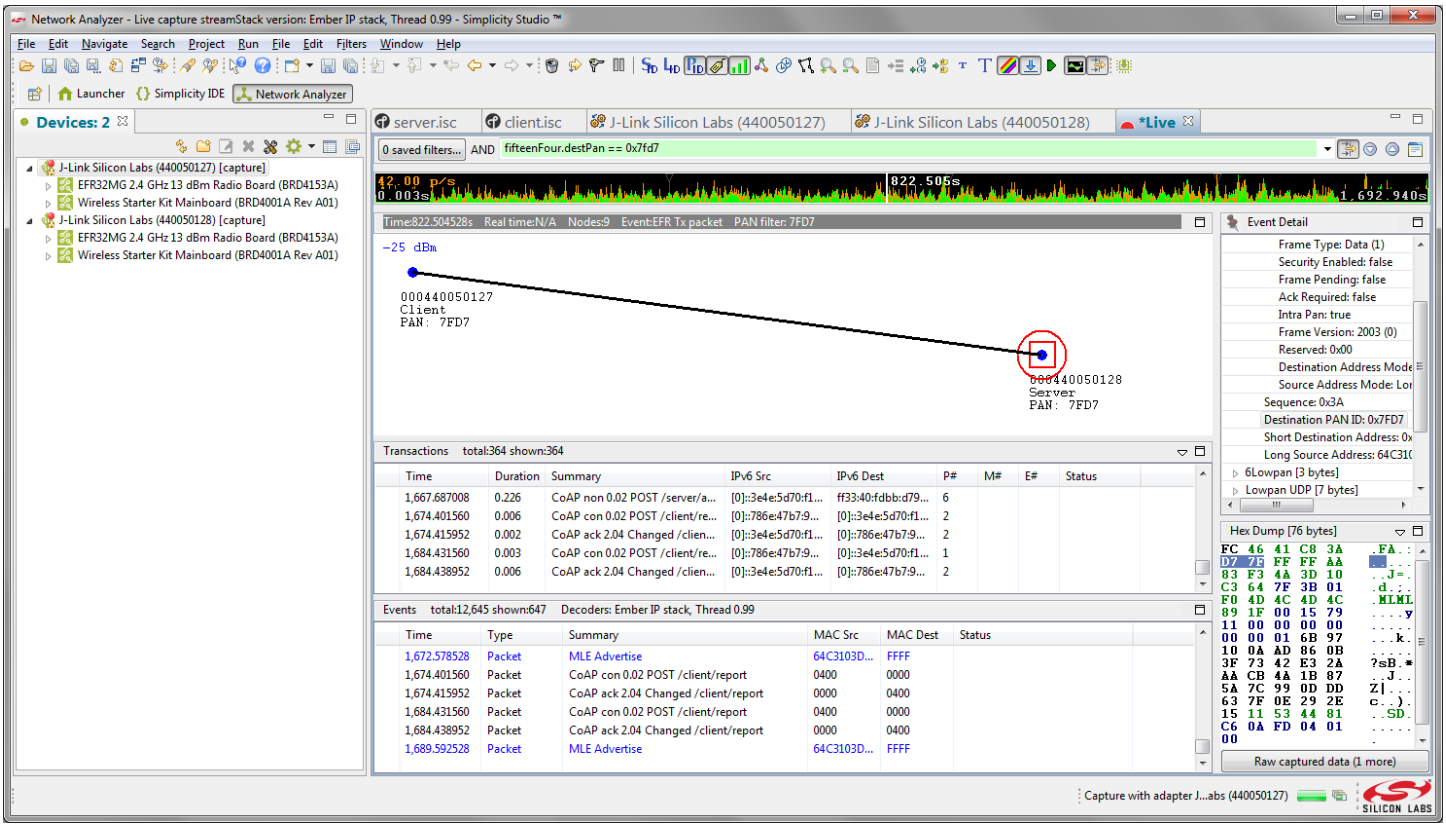
7. (optional) Alternatively, in the Events frame, click one of the MLE Advertise events (in blue). In the Event Detail, scroll down until you see the Destination PAN ID, right-click on it, and select **Add to filter**. Click  .

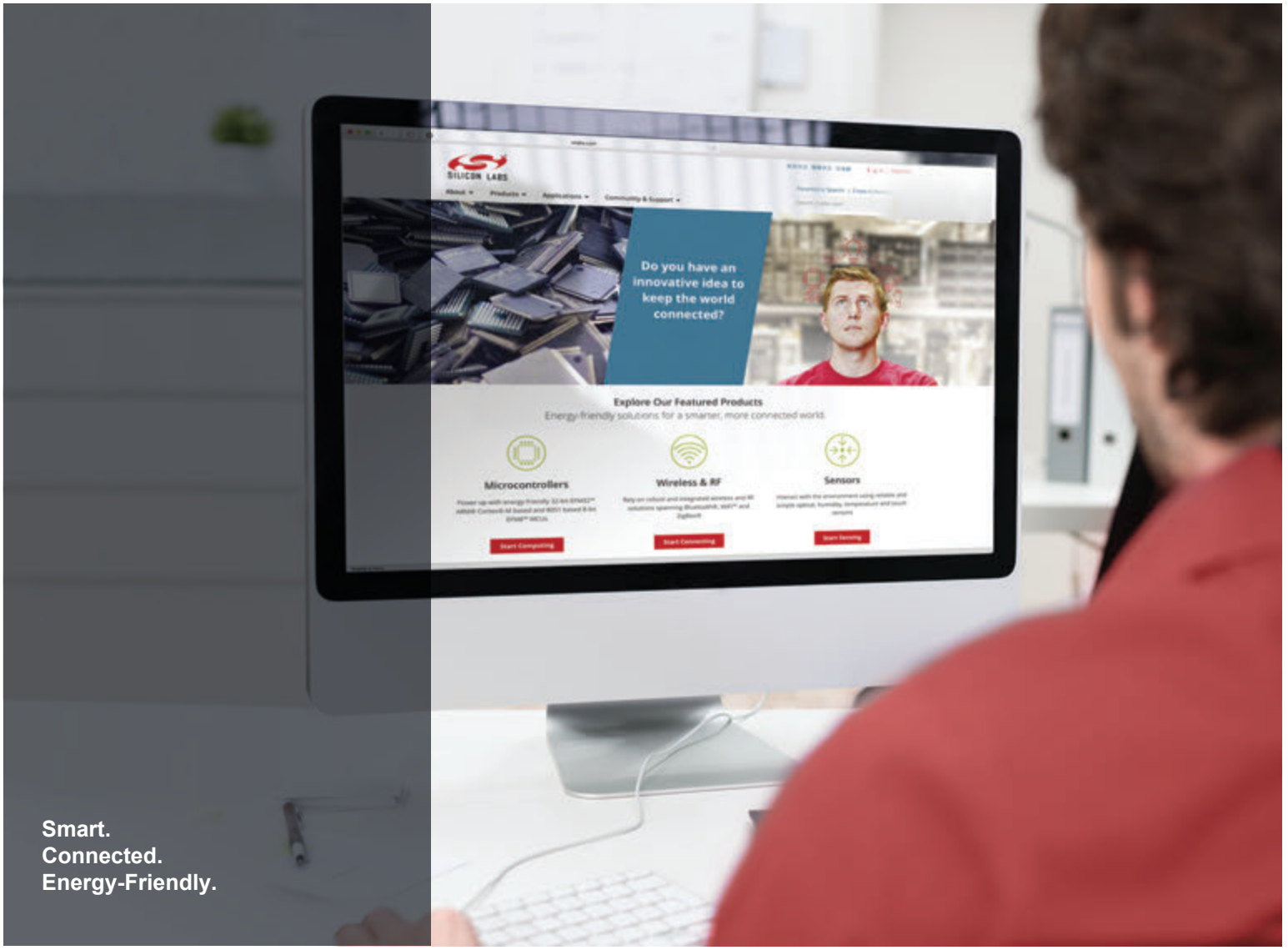


8. You should see the over-the-air message being captured live in the Transactions window:



When analyzing more complex networks, you can drag and reposition the items shown in the map. By right-clicking on a device, you can also show connectivity and add labels.


Network Analyzer - Live capture streamStack version: Ember IP stack, Thread 0.99 - Simplicity Studio ™
File Edit Navigate Search Project Run File Edit Filters Window Help
Launcher
Simplicity IDE
Network Analyzer
Devices: 2
J-Link Silicon Labs (440050127) [capture]
EFR32MG 2.4 GHz 13 dBm Radio Board (BRD4153A)
Wireless Starter Kit Mainboard (BRD4001A Rev A01)
J-Link Silicon Labs (440050128) [capture]
EFR32MG 2.4 GHz 13 dBm Radio Board (BRD4153A)
Wireless Starter Kit Mainboard (BRD4001A Rev A01)
server.isc
client.isc
J-Link Silicon Labs (440050127)
J-Link Silicon Labs (440050128)
*Live
0 saved filters...
AND
fifteenFour.destPan == 0x7fd7
42.00 p/s
822.505s
1,692.940s
Time:822.504528s Real time:N/A Nodes:9 Event:EFR Tx packet PAN filter: 7FD7
-25 dBm
000440050127
Client
PAN: 7FD7
000440050128
Server
PAN: 7FD7
Event Detail
Frame Type: Data (1)
Security Enabled: false
Frame Pending: false
Ack Required: false
Intra Pan: true
Frame Version: 2003 (0)
Reserved: 0x00
Sequence: 0x3A
Destination PAN ID: 0x7FD7
6Lowpan [3 bytes]
Lowpan UDP [7 bytes]
Transactions total:364 shown:364
Time Duration Summary IPv6 Src IPv6 Dest P# M# E# Status
1,667.687008 0.226 CoAP non 0.02 POST /server/a... [0]::3e4e:5d70:f1... ff33:40:fdbb:d79... 6
1,674.401560 0.006 CoAP con 0.02 POST /client/re... [0]::786e:47b7:9... [0]::3e4e:5d70:f1... 2
1,674.415952 0.002 CoAP ack 2.04 Changed /clien... [0]::3e4e:5d70:f1... [0]::786e:47b7:9... 2
1,684.431560 0.003 CoAP con 0.02 POST /client/re... [0]::786e:47b7:9... [0]::3e4e:5d70:f1... 1
1,684.438952 0.006 CoAP ack 2.04 Changed /clien... [0]::3e4e:5d70:f1... [0]::786e:47b7:9... 2
Events total:12,645 shown:647 Decoders: Ember IP stack, Thread 0.99
Time Type Summary MAC Src MAC Dest Status
1,672.578528 Packet MLE Advertise 64C3103D... FFFF
1,674.401560 Packet CoAP con 0.02 POST /client/report 0400 0000
1,674.415952 Packet CoAP ack 2.04 Changed /client/report 0000 0400
1,684.431560 Packet CoAP con 0.02 POST /client/report 0400 0000
1,684.438952 Packet CoAP ack 2.04 Changed /client/report 0000 0400
1,689.592528 Packet MLE Advertise 64C3103D... FFFF
Hex Dump [76 bytes]
FC 46 41 C8 3A .FA.:
D7 7F FF FF AA ....
83 F3 4A 3D 10 ..J=.
C3 64 7F 3B 01 .d.;.
F0 4D 4C 4D 4C .MLML
89 1F 00 15 79 ....y
11 00 00 00 00 .....
00 00 01 6B 97 ...k.
10 0A AD 86 0B .....
3F 73 42 E3 2A ?sB.*
AA CB 4A 1B 87 ..J..
5A 7C 99 0D DD Z|...
63 7F 0E 29 2E c..).
15 11 53 44 81 ..SD.
C6 0A FD 04 01 .....
00 .
Raw captured data (1 more)
Capture with adapter J...abs (440050127)
SILICON LABS



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

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

http://www.silabs.com