QSG126: *Bluetooth* ® Developer Studio Quick-Start Guide

Bluetooth® Developer Studio (BTDS) is a graphical GATT-based development framework that facilitates building Bluetooth-enabled applications on EFR32 devices using the Silicon Labs Bluetooth stack.

BTDS will auto-generate the skeleton project for any standard Bluetooth profile configuration, which can be integrated into Silicon Labs Software Development environment and tools.

For most of the profiles the generated code will be ready for compilation and flashing, although generally the code should be customized further to fit user requirements.

KEY FEATURES OR KEY POINTS

- Getting started with BTDS.
- Generating a project from BTDS.
- Working with generated projects.

1. Introduction

The Bluetooth Developer Studio (BTDS) is a desktop development framework provided by the Bluetooth SIG. It facilitates 70% less development time by providing a library of all standard Bluetooth Profiles that can be quickly customized using a GUI (Graphical User Interface). This interactive development kit enables developers to learn Bluetooth development up to 50% faster by providing drag and drop functionality for a growing list of Bluetooth profiles.

BTDS Code Plugin Framework allows code generation based on the GATT configuration that was created in the GUI.

Silicon Labs provides two BTDS Plugins that auto-generate the skeleton project for various integration development environments for any standard Bluetooth GATT configuration.

Please note that the generated skeleton project is not always ready for compilation, as some Bluetooth Format Types are not supported. Customers are responsible for manually implementing TODOs in the generated code.

This Quick Start Guide describes how to generate a BTDS project based on the Silicon Labs Bluetooth stack and an EFR32 device.

Prerequisites:

- You are familiar with most important terms of the Bluetooth protocol. Terms are described in 1.2 Definitions.
- You plan to develop Bluetooth code on the Silicon Labs EFR32 microcontroller using the Silicon Labs Bluetooth stack.
- You are using the IAR toolchain for development and have an IAR license.
- You have downloaded and installed IAR Embedded Workbench or Simplicity Studio.
- You have installed the Silicon Labs Bluetooth SDK 2.0.0 or higher containing the BTDS Plugins.
- You have downloaded and installed BTDS from www.bluetooth.com.

Because automatic part detection is not possible using BTDS, all generated project files assume the EFR32BG1B232F256GM48 part (for example, BGM111 Blue Gecko Module kit). If you are working with a different part, change the project settings after generation.

An alternative tool to BTDS is Simplicity Studio's GATT Editor, which also enables graphical GATT table editing and basic code skeleton generation when you select an SOC-Empty project as a starting point. The project generated with that tool will support any Silicon Labs radio board and part. For more information please refer to *QSG139: Bluetooth Development with Simplicity Studio*.

1.1 Silicon Labs Plugins

The following Plugins are currently available.

C-Project generation:

- **Simplicity Studio™ Project Generator:** Auto-generates the skeleton for Simplicity Studio projects (IAR Toolchain). For more information about Simplicity Studio, see *AN0822: Simplicity Studio™ User Guide* and *QSG139: Bluetooth Development with Simplicity Studio*.

  Simplicity Studio also contains a handful of more sophisticated kit- and part-aware example applications implementing the most popular Bluetooth Profiles. These can be taken as a reference without using Bluetooth Developer Studio. If you are starting development immediately from Simplicity Studio, see the following documents:

  - *QSG108: Getting Started with Silicon Labs' Bluetooth® Software*
  - *UG136: Silicon Labs Bluetooth® C Application Developer's Guide*
- **IAR Project Generator:** Auto-generates the skeleton for IAR Embedded Workbench projects.

1.2 Definitions

In Bluetooth, the following terms are used:

- **GATT:** Generic Attribute Profile, High-level Stack layer in Bluetooth Architecture, which defines attributes as basic structures of transferrable data and organizes attributes into Profiles, Services, Characteristics, and Descriptors.
- **Characteristic:** A set of data including a value and properties such as declarations and descriptors.
- **Service:** A collection of characteristics and relationships to other services that encapsulate the behavior of a part of a device.
- **Profile:** A high-level definition that determines how services are used to enable an application or use case. Bluetooth profiles specify general behaviors that Bluetooth-enabled devices use to communicate with other Bluetooth devices. At a minimum, each Bluetooth profile contains information on the following topics:
  - Dependencies on other profiles
  - Suggested user interface formats
  - Specific parts of the Bluetooth protocol stack used by the profile

  To perform its task, each profile uses particular options and parameters at each layer of the stack.
- **Device:** A combination of hardware and software that operates according to a Bluetooth profile.

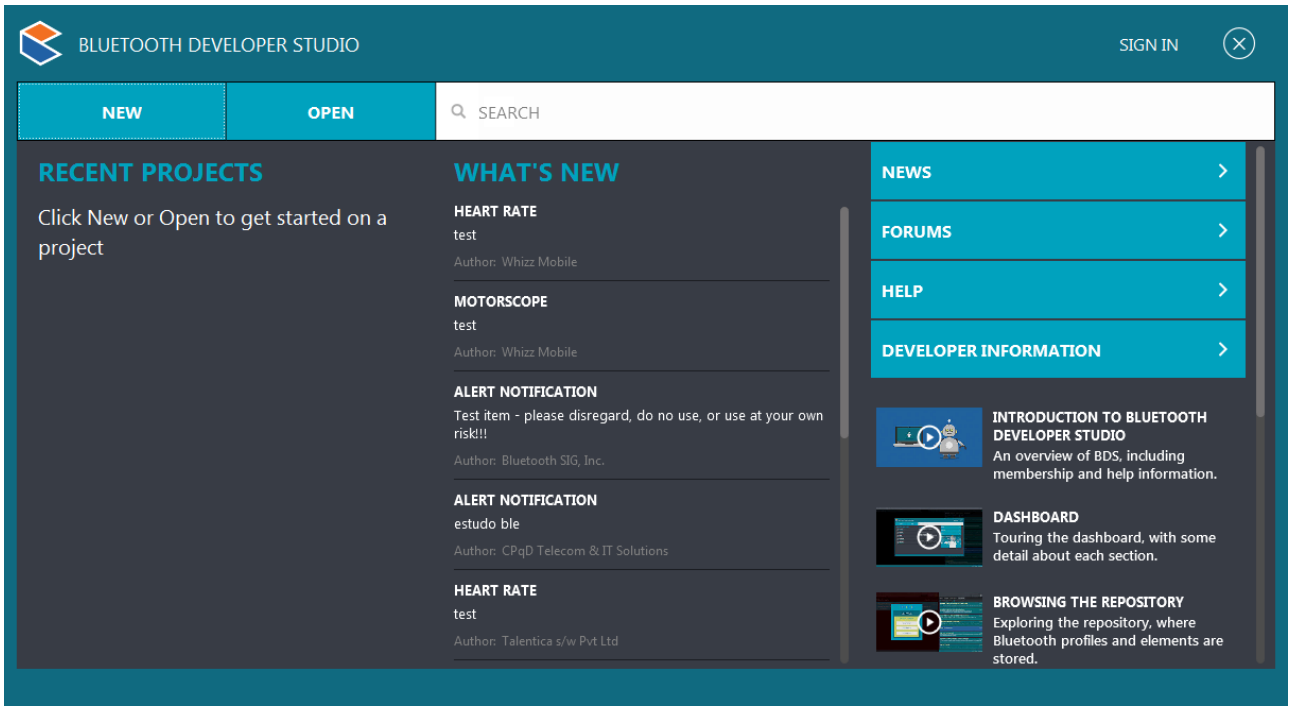
2. Generating a Project

2.1 Import the Silicon Labs Plugins into BTDS

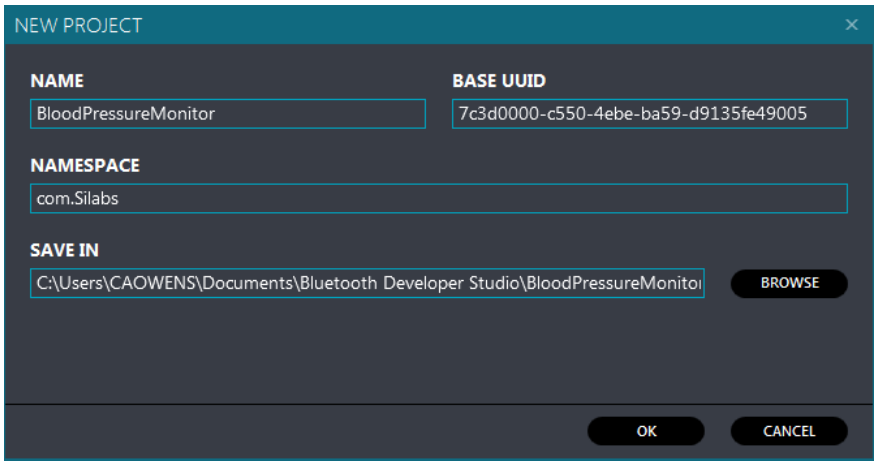
If you have not yet done so, copy the Plugin folder from app\<bluetooth_x.y>\btds\Plugins into the BTDS installation's Plugin folder (for example, c:\Program Files (x86)\Bluetooth SIG\Bluetooth Developer Studio\Plugins\). The Plugins are now ready to use from BTDS. Each Bluetooth SDK comes with its own Bluetooth Developer Studio plugins, as the plugins may need to be updated for each Bluetooth SDK. Make sure the Bluetooth developer Studio plugins you use correspond to your selected Bluetooth SDK.

2.2 Create a New Project

1. Open BTDS. The BTDS Dashboard should appear. If not, select **File > New**.

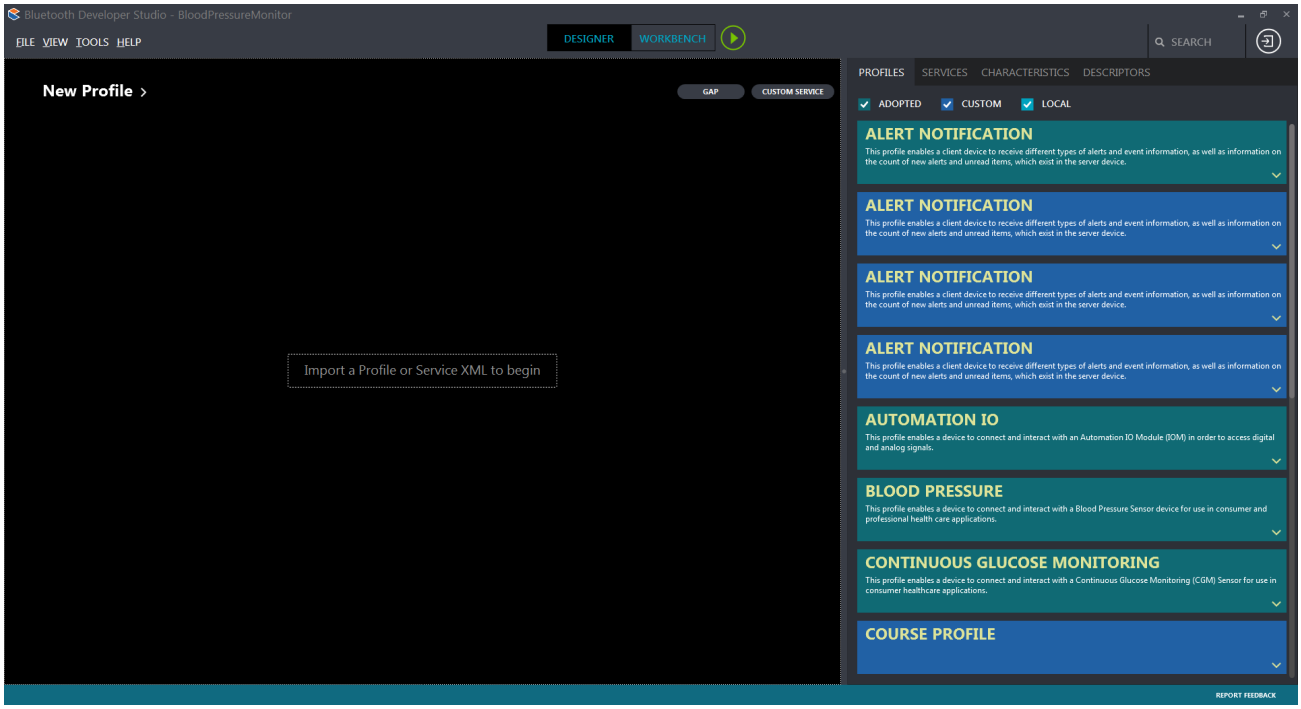


2. In the BTDS Dashboard, click [**New**]. Complete the form for a new project, and click [**OK**]. The Project View opens.

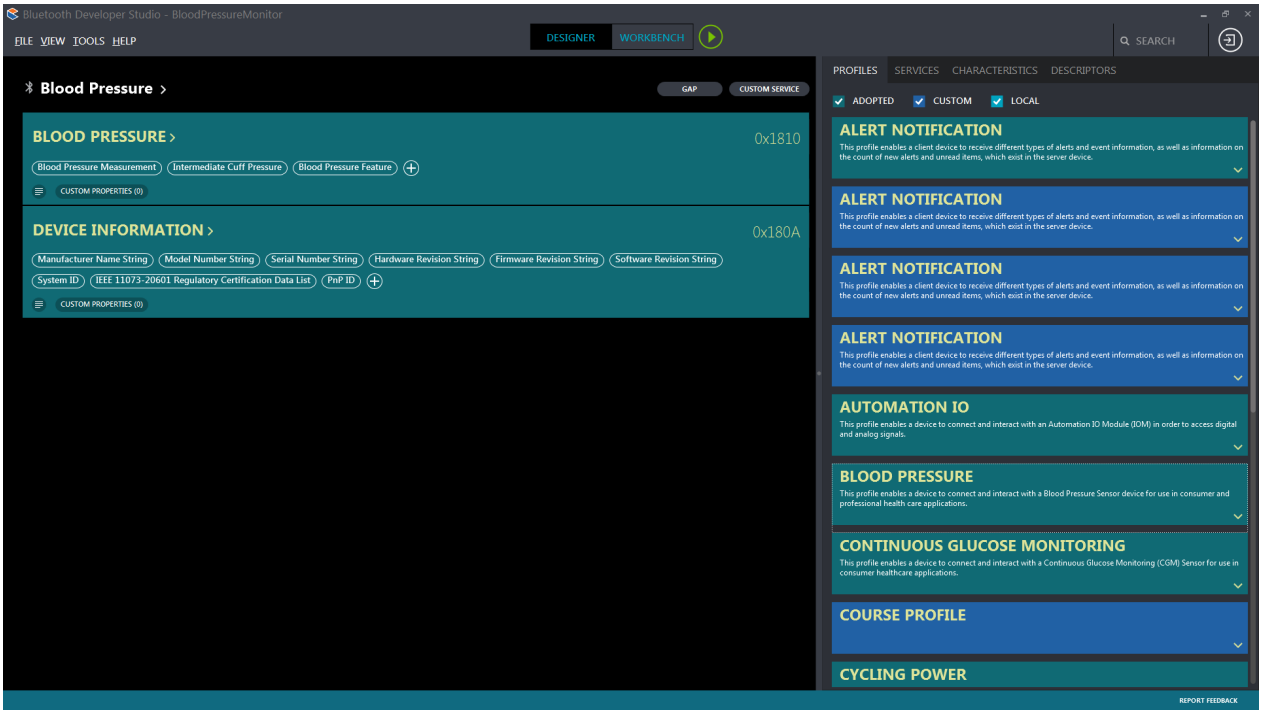


2.3 Create the GATT Database

1. In Project view, drag an element from one of the available Profiles/Services tabs on the left to the workspace on the right to create the visual representation of the GATT database. You can add arbitrary Characteristics to or remove them from these Services, and add arbitrary Descriptors to or remove them from these Characteristics.

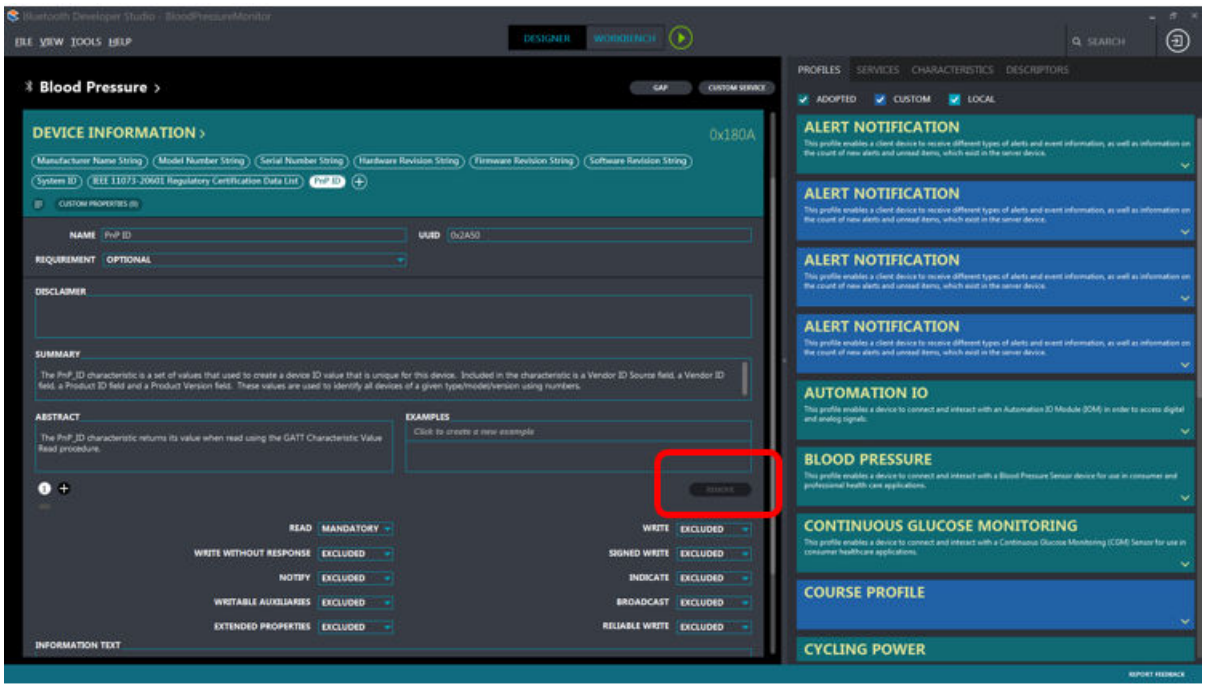


2. The workspace populates with the profile and other information.

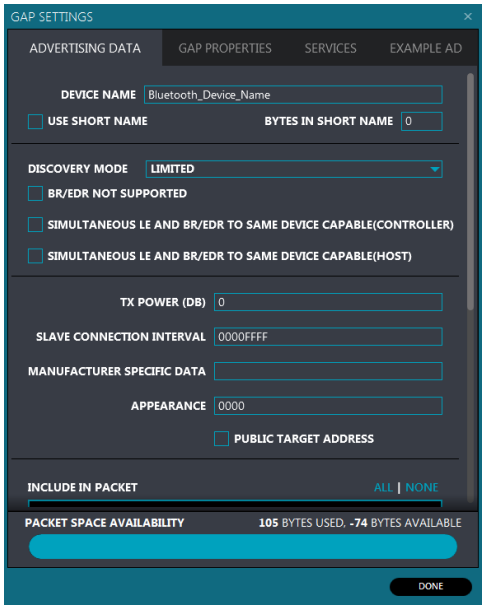


3. (optional) Click a characteristic name to open the Characteristic view. To remove an optional or unnecessary characteristic, click [**Remove**]. Please note that some Characteristics contain unsupported Bluetooth Format Types. In these cases it is recommended to either remove the Characteristic from the Service, or implement that specific Format Type in the generated code before

compilation. For example the Device Information Service's 'System ID' and 'IEEE 11073-20601 Regulatory Certification Data List' Characteristics are not supported by default, so they should be removed, if they are not needed.



4. (optional) All Characteristics and all Characteristic Properties are listed as either Mandatory, Excluded, Conditional, or Optional. By default, all Mandatory and Optional Characteristics and Characteristics Properties will be handled in the Generated Code. Optional features can be easily disabled by defining a Custom Property '*DisableOptionalFeatures*' with the value '1'.
5. (optional) Click [**GAP**] above the workspace to open the GAP Settings dialog. Here you can set the Device name and Appearance value, which appear in the GATT xml file. If you do not modify the default '*Bluetooth_Device_Name'* Device name, it is shortened to *'BLE Device'* in the GATT XML database to fit on most smart phone screens when testing.

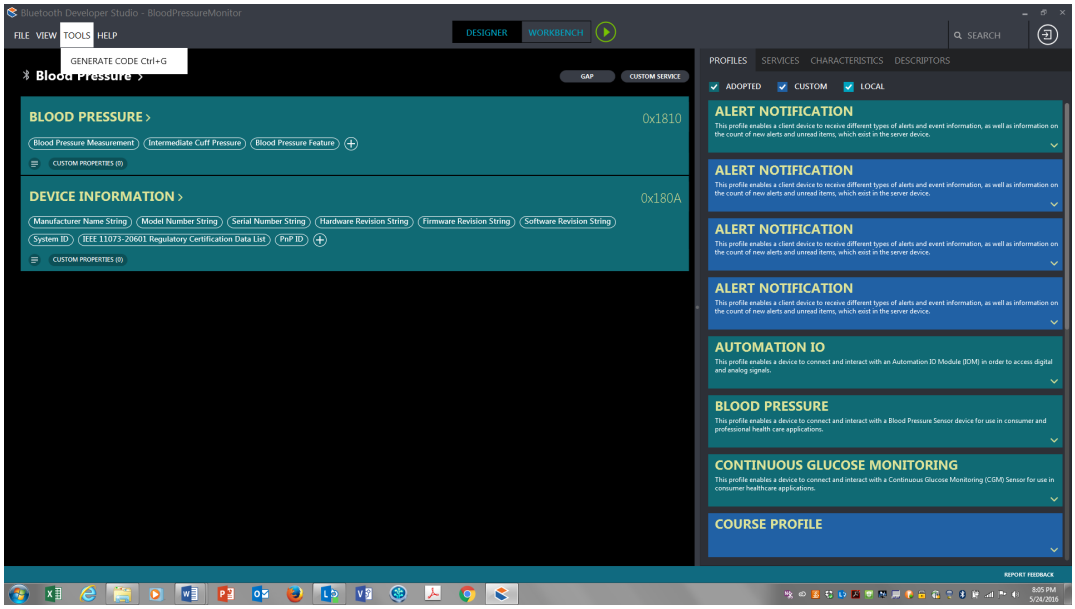


2.4 Generate Application Code

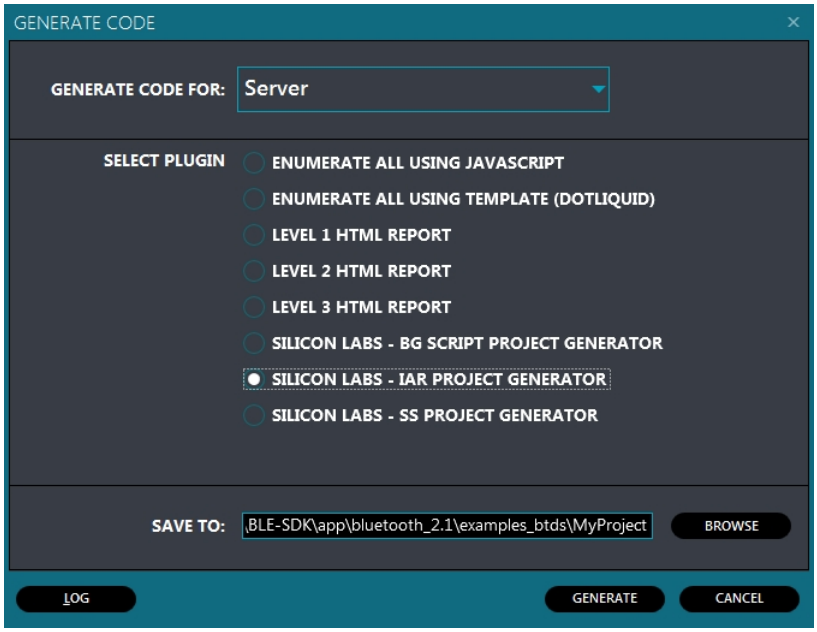
Silicon Labs provides three Plugins to generate code from BTDS. Select the one that corresponds to the type of project you want to create:

- Simplicity Studio Project Generator
- IAR Project Generator

1. In the Main Menu, select **Tools > Generate Code**.



2. In the Generate Code dialog window, choose the correct path for plugin output. Select the following path: app\<bluetooth_x.y>\examples_btds\<project_folder>\. This is important because the Bluetooth Stack's and tool's paths are generated relative to this root folder in the project files. Modify the Plugin generation folder, if necessary, and click [**Generate**].



2.5 Customize Application Code

After generation, Bluetooth Developer Studio plugins create projects that use the Silicon Labs Bluetooth stack to implement the designed GATT table. This can be tested with iOS® or Android™ applications right away for most of the profiles. Using standard Bluetooth Client smart phone apps, a connection can be made to the EFR Device, its attributes can be discovered, read, written, and so on (if characteristic properties allow for the specific operation).

The generated project's source and XML files contain a small number of TODOs, which you should replace with your code to customize the project. Handling of notifications, indications, and user-specific read and write operations should be done in the <service>.c/<service>.bgs files.

Note: The Bluetooth stack automatically handles characteristic Read and Write operations without special handling in the <service>.c file. To add a user-specific Read or Write action, the service must be looked up in the gatt.xml database and a type=user tag must be added to that service.

3. Working with Generated Projects

3.1 Simplicity Studio Project Generator

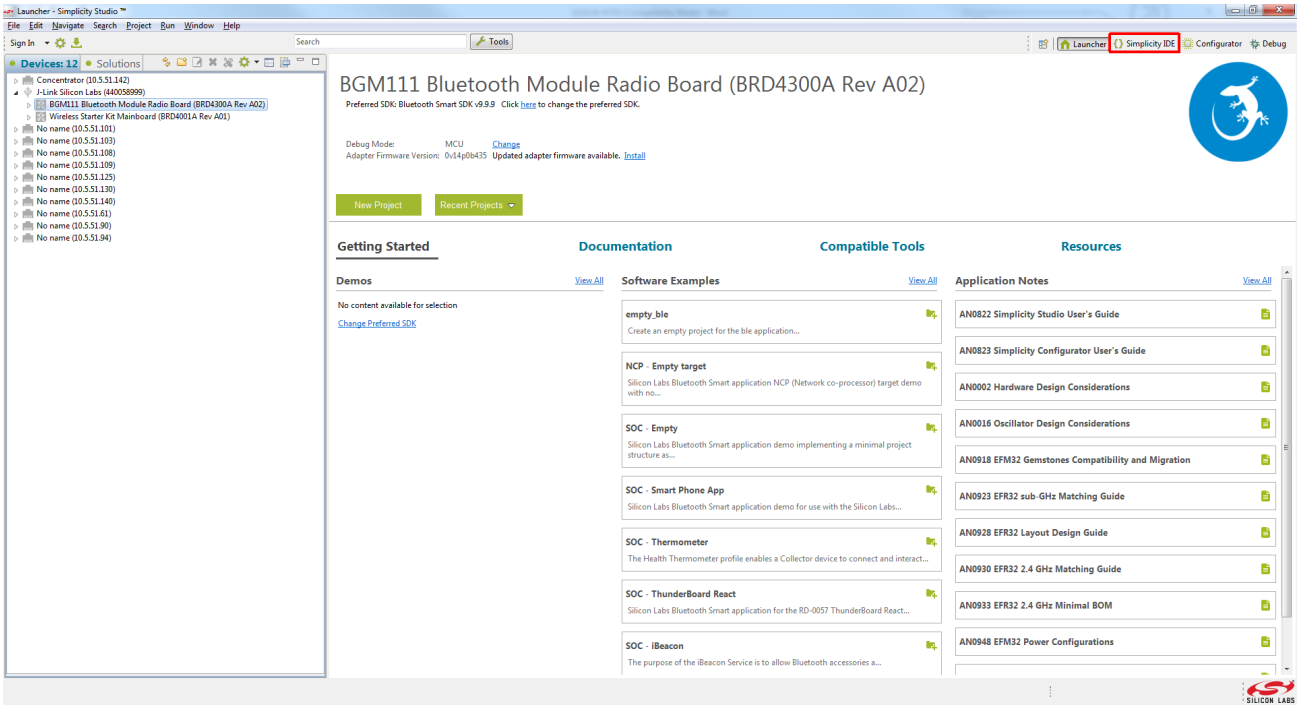
When you use the Simplicity Studio Project Generator Plugin to generate code, an .slsproj file is generated along with the source files. This allows easy integration of the project into Simplicity Studio. For more details about the Bluetooth C-SDK and Simplicity Studio, see *QSG108: Getting Started with Silicon Labs' Bluetooth® Software, QSG139: Bluetooth Development with Simplicity Studio,* and *UG136: Silicon Labs Bluetooth C Application Developer's Guide*.

Please be advised, that the generated project will support BGM111 only, and no Hardware Configurator will be available to support migration of project to a different board. If you want to take advantage of Simplicity Studio's hardware detection, Hardware Configurator, and Gatt Editor please refer to QSG139.

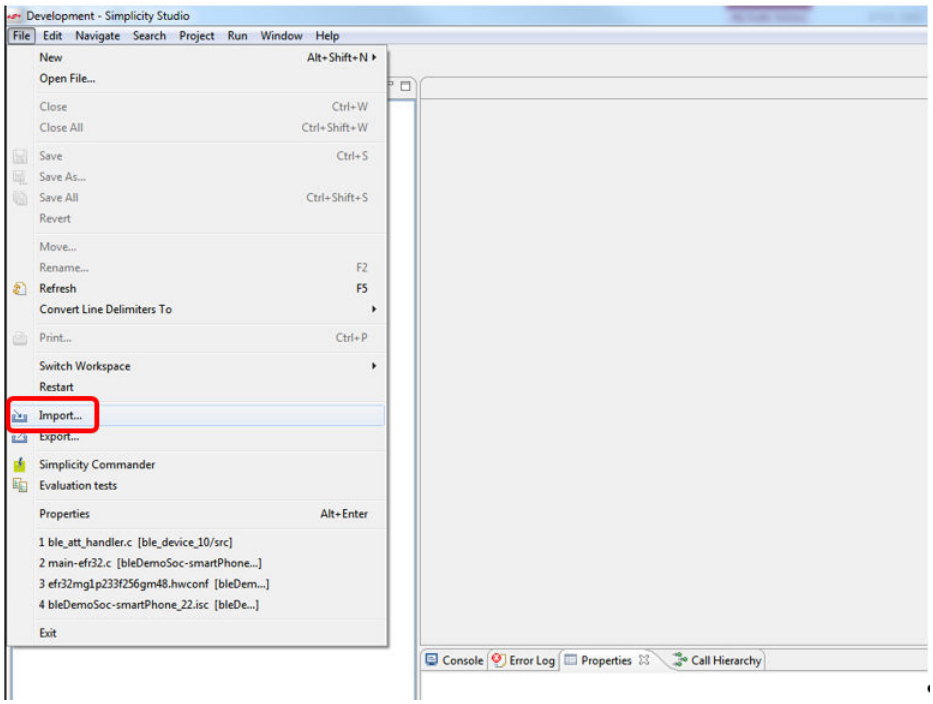
Generation creates the following files in the specified Plugin output location:

ble_device.slsproj	Simplicity Studio project file.
bleDemo.icf	IAR linker configuration file.
BGBUILD/	
bgbuild/gatt.bgproj	Contains the GATT database.
bgbuild/gatt.xml gatt_db.c gatt_db.h	C code for the GATT database is created from this XML-based description before compilation as a prebuild step in gatt_db.c and gatt_db.h. This XML and .bgproj file are automatically populated by BTDS, but it is also possible to modify them manually according to the instructions in *UG118: Blue Gecko Bluetooth® Profile Toolkit Developer User's Guide*, to add more services and characteristics.
SRC/	
main.c	Main application file, which contains a bit switch-case structure handling all anticipated stack events.
InitDevice.c	Device initialization file, which contains initialization code for all peripherals.
InitDevice.h	Header file for InitDevice.c.
ble_att_handler.c	Handles long characteristic reads and writes.
ble_att_handler.h	Header file for ble_att_handler.c.
<service_name>.c	Implementation of a standard GATT service.
<service_name>.h	Header file for <service_name>.c

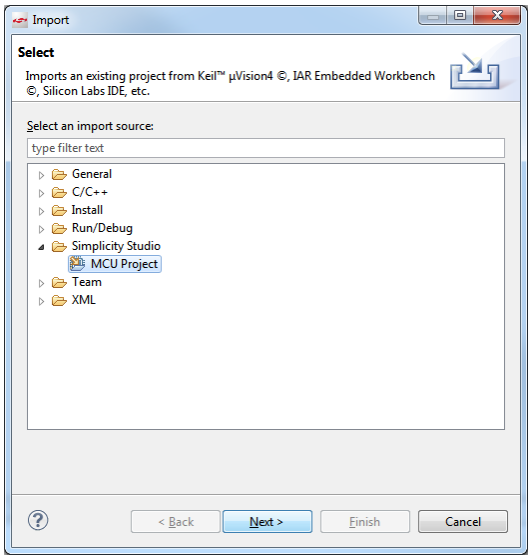
1. Open Simplicity Studio. If you are using a Silicon Labs Kit, connect it to your PC by an USB cable and wait until Simplicity Studio detects it. If this is a new installation, follow the instructions in *QSG108: Getting Started with Silicon Labs' Bluetooth® Software* until the Launcher is displayed.
2. Open the **Simplicity IDE**. If the control is not displayed in the upper right, select it from the Tools menu.



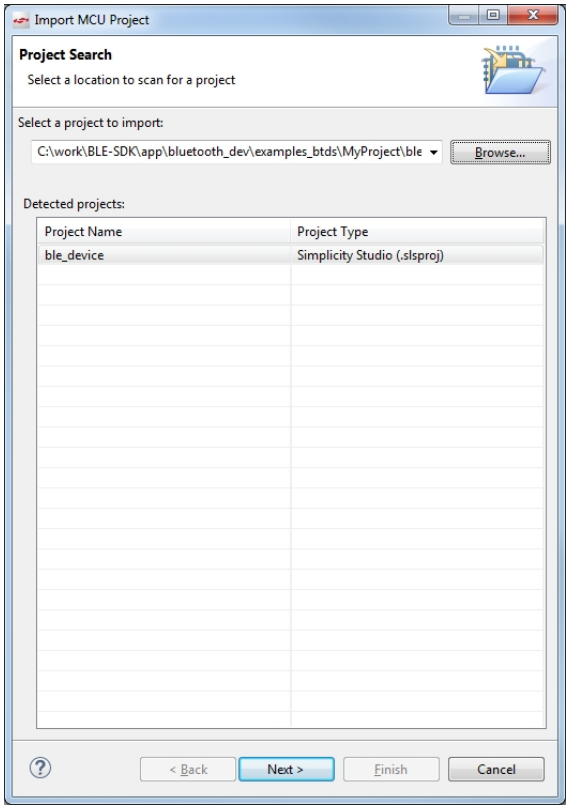
3. Select **File > Import**.



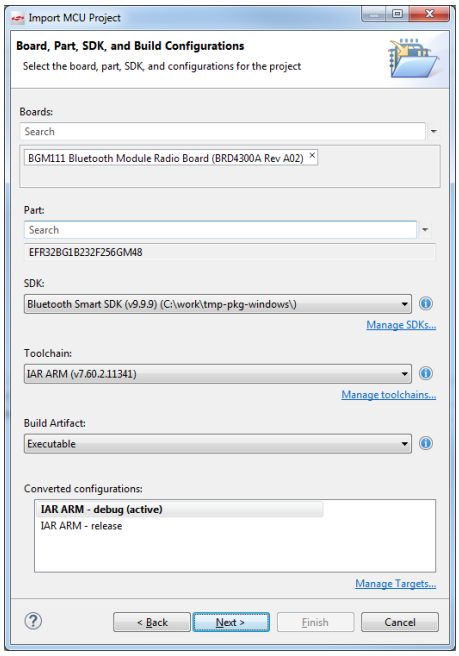
4. Select **Simplicity Studio > MCU Project.**



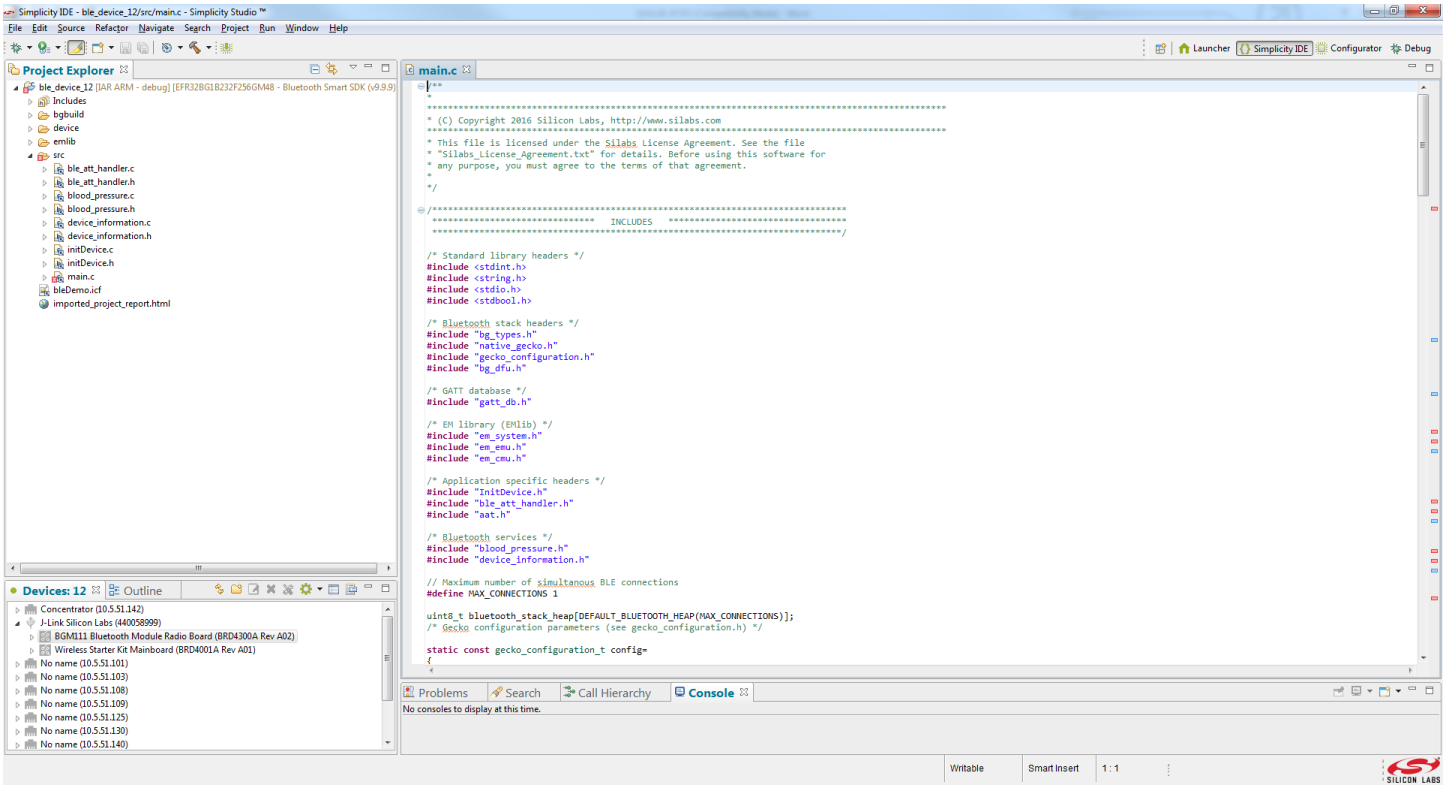
5. Navigate to the generated .slsproj file (ignore the warning about editing files in the SDK).



6. Select the correct Kit, Part, and SDK. Click [**Next >**] and then [**Finish**]. The project is now imported into Simplicity Studio.



You can now modify, build, and debug the project from Simplicity Studio.



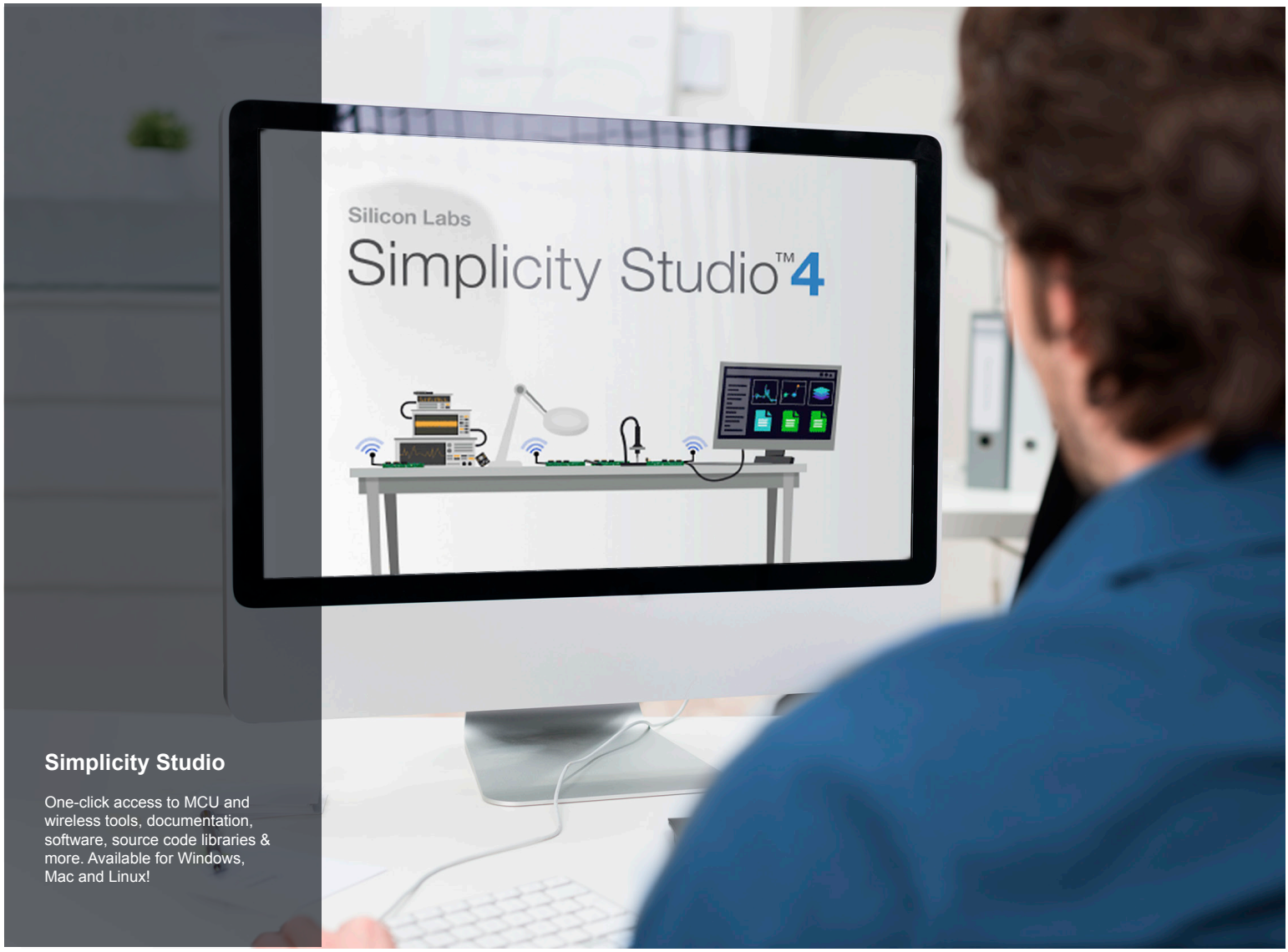
3.2 IAR Project Generator

This Plugin generates an IAR Embedded Workbench project using the Silicon Labs Bluetooth SDK. The project links the stack as IAR precompiled libraries to the application code, which can be customized, debugged, and flashed on the EFR32 device right away for most profiles.

The .eww workspace file is located in <project_name>/iar/ble_workspace.eww.

Generation creates the following files in the specified Plugin output location:

BGBUILD:	
bgbuild/gatt.bgproj	Contains the GATT database.
bgbuild/gatt.xml	C code for the GATT database is created from this XML-based description before compilation as a prebuild step in gatt_db.c and gatt_db.h. This XML and .bgproj file are automatically populated by BTDS, but it is also possible to modify them manually according to the instructions in *UG118: Blue Gecko Bluetooth® Profile Toolkit Developer User's Guide*, to add more services and characteristics.
IAR/	
ble_device.ewd	IAREW Debugger settings file.
ble_device.ewp	IAREW Project file.
ble_device.eww	IAREW Workspace file.
bleDemo.icf	IAREW linker configuration file.
SRC/	
main.c	Main application file, which contains a bit switch-case structure handling all anticipated stack events.
InitDevice.c	Device initialization file, which contains initialization code for all peripherals.
InitDevice.h	Header file for InitDevice.c.
ble_att_handler.c	Handles long characteristic reads and writes.
ble_att_handler.h	Header file for ble_att_handler.c.
<service_name>.c	A skeleton file is created for each service where a custom Property GENERATECODE was added with a value of 1.
<service_name>.h	Header file for <service_name>.c.



Simplicity Studio

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

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

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

http://www.silabs.com