UG365: GATT Configurator User's Guide for *Bluetooth®* SDK v2.x

This guide provides the information needed to effectively use the Bluetooth GATT Configurator provided as a part of Simplicity Studio. The GATT Configurator is an intuitive interface providing access to all the Profiles, Services, Characteristics, and Descriptors as defined in the Bluetooth specification. It also supports creating, importing, or exporting custom GATT profiles for Bluetooth applications. This guide reviews the user interface and covers some most common uses of the Configurator.

KEY POINTS

- GATT Configuration user interface review.
- Using the GATT Configurator to create and configure the GATT database.
- Typical GATT Configurator Use Cases

1 GATT Configurator Overview

The GATT Configurator is composed of four different sections that allow you to work with the various components and aspects of the interface. The Source section, Custom GATT section, Settings section, and Tools section are shown in the following figure. This chapter reviews each section.
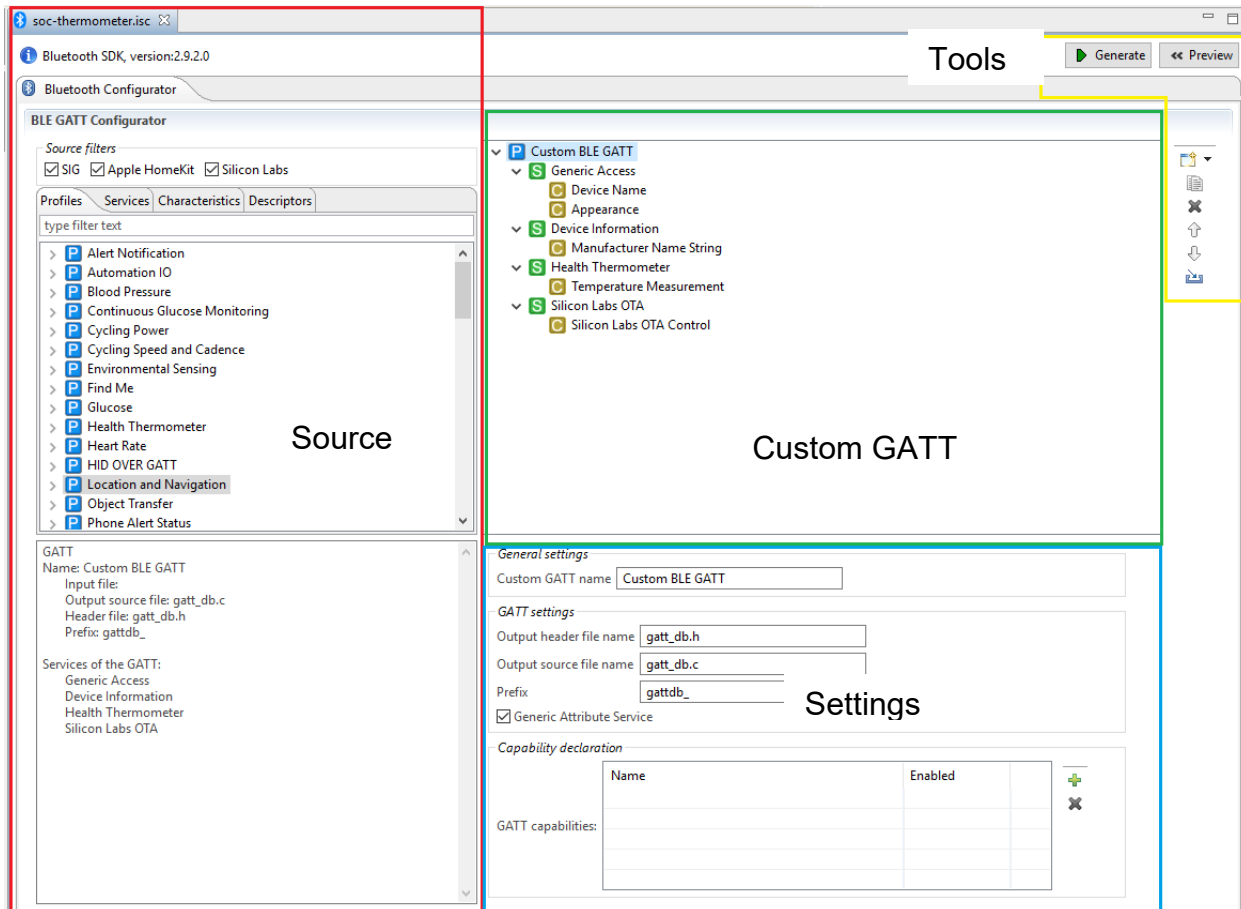

Figure 1-1. GATT Configurator Interface

1.1 Source Section

The Source section displays a list of predefined Profiles, Services, Characteristics, and Descriptors. These items can be filtered based on their source using the checkboxes. Tabs allow you to switch between different lists. As shown in the following figure, the pane below the list displays textual information about the latest selection.

Note: This pane is shared by the Source and the Custom GATT sections. It can be used to find more information about a selected item in the Source section or in the Custom GATT section.

The top of the Source section shows the name of the configurator file and the Bluetooth SDK version. An * in the configurator file name indicates there are unsaved changes. Clicking the ⓘ icon gives you more information about the current framework.
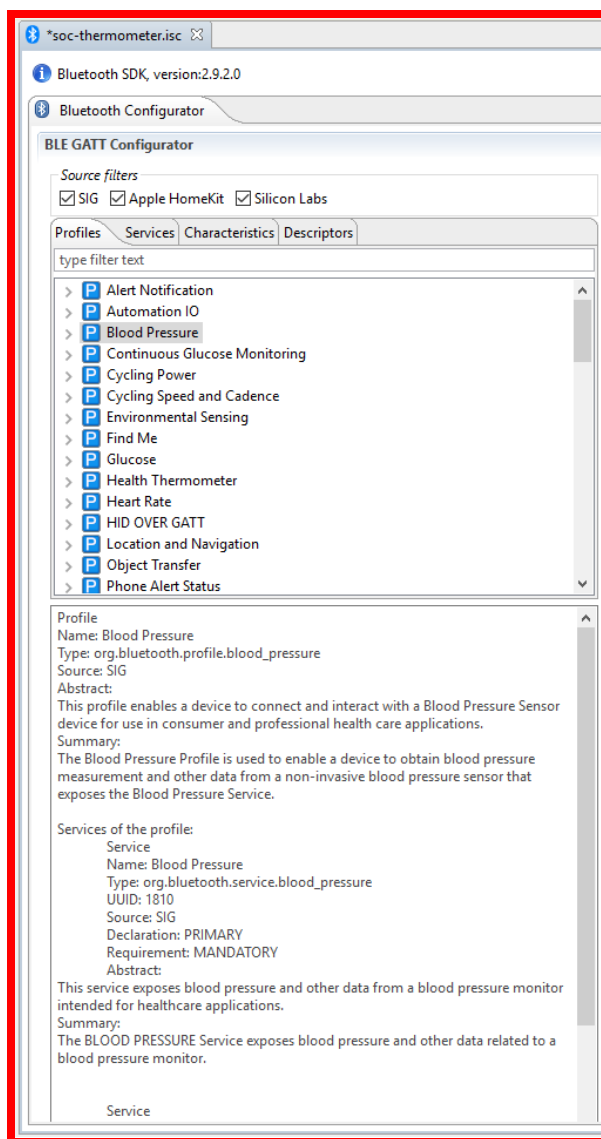

Figure 1-2. Source Section

1.2 Custom GATT Section

The Custom GATT section displays the items present in the current configuration file. This includes a Custom GATT Profile, Services, Characteristics, and Descriptors displayed as a hierarchical list. The order of items shown here reflect the order in which they exist in the GATT database. You can select an item here to see its properties and configuration.
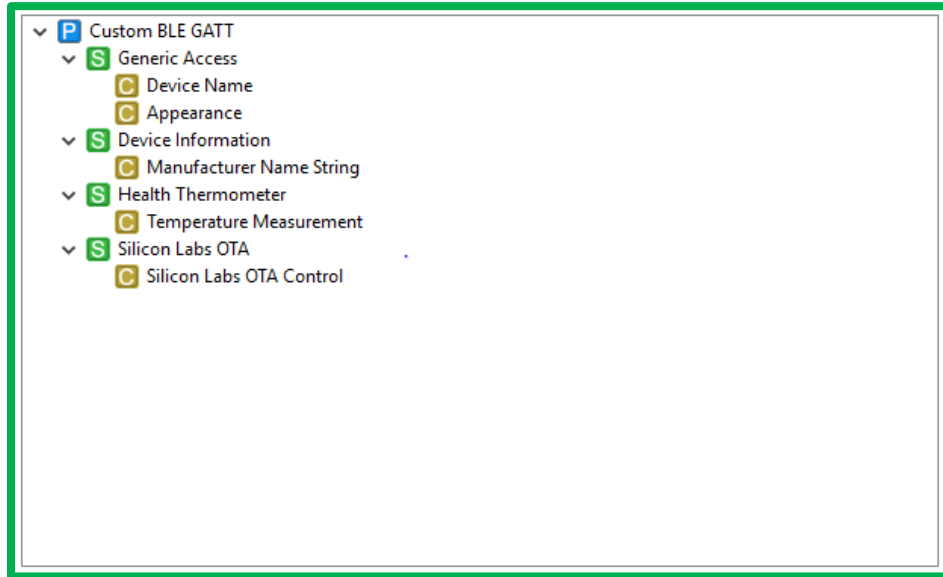

Figure 1-3. Custom GATT Section

Note: Unsaved changes will be shown in this section, so this view does not necessarily reflect the true state of the GATT database.

1.3 Settings Section

The Settings section allows you to configure the properties of items such as Profiles, Services, Characteristics and Descriptors that are present in the Custom GATT section. Selecting an item populates the relevant configuration options such as the name, ID, properties and capabilities. Any changes made in this section reflect immediately for the selected item.

Figure 1-4. Settings Section

Note: Unsaved changes will be shown in this view, so changes made in this view does not necessarily reflect the true state of the GATT database.

1.4 Tools Section

Tools allow you to create new items, duplicate, remove or reorder items, or import a new GATT database. Controls in this menu get activated when you select a relevant item in the Custom GATT view section. For example, you can only create a new Descriptor when a Characteristic is selected in the Custom GATT view. Similarly, reordering options will not work if there is only one item at that level. The purpose of the control appears when the cursor is hovered over it.

Figure 1-5. Tools Section

Generate is used to generate GATT database xml file (gatt.xml) as well as the associated source files (gatt_db.c/.h).

The **Import** control (🔼) imports an existing GATT database using an .xml file or a .bgproj file.

Note: Importing a gatt.xml file using the Import function will overwrite the current custom GATT data. This message also appears as a prompt when you click the **Import** control. The gatt.xml structure and notation is described in *UG118: Blue Gecko Bluetooth® Profile Toolkit Developer's Guide*.

1.4.1 Difference between Generate and Save

Save: This saves any changes made in the GATT Configurator to the ".isc" file of the project. Unsaved changes are indicated by an asterisk (*) next to the name of the .isc file. Changes can be saved by pressing "Ctrl + S" or clicking **Save** under the File menu. Use the Save function to save changes that you intend to keep but do not need to reflect yet in the GATT database or GATT source files.

Generate: When you click **Generate**, changes are made to the GATT database (gatt.xml) and source file (gatt_db.c/.h). Further generations will overwrite the changes to these files. Additionally, it can also create files and overwrite them if they have been modified (for example. init_mcu*.c). If you have made modifications to such files, make sure to uncheck the files so that they do not get overwritten. A prompt is displayed as shown below giving you the option of selecting specific files that need to be overwritten. Generating does not save the changes to the ".isc" file.
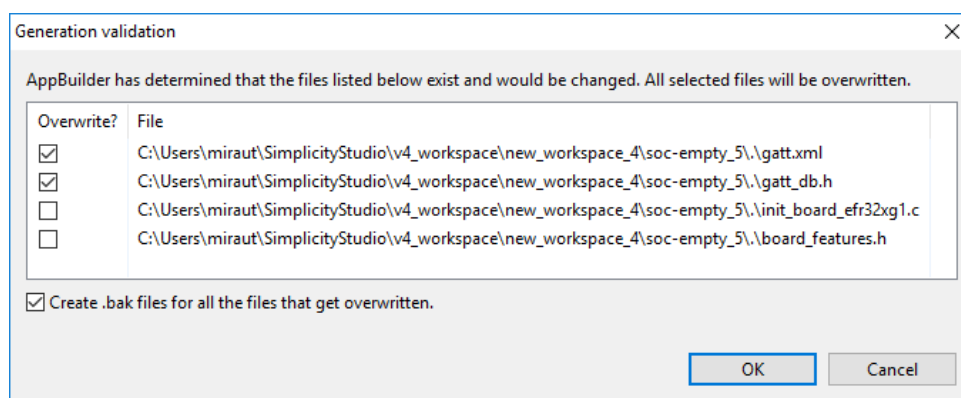

Figure 2-6. Overwrite prompt after Generation

As a best practice, always first save the changes and then generate.

2 Use Cases

This chapter describes common tasks performed with the GATT Configurator.

2.1 Drag and Drop

To include predefined items from the source list in your application, drag and drop the item from the Source Section to the Custom GATT section. When you drag and drop a profile or a service, all the Characteristics and Descriptors in the levels underneath get included automatically. Maintaining the hierarchical structure, Descriptors can only be included under Characteristics, which go under Services.
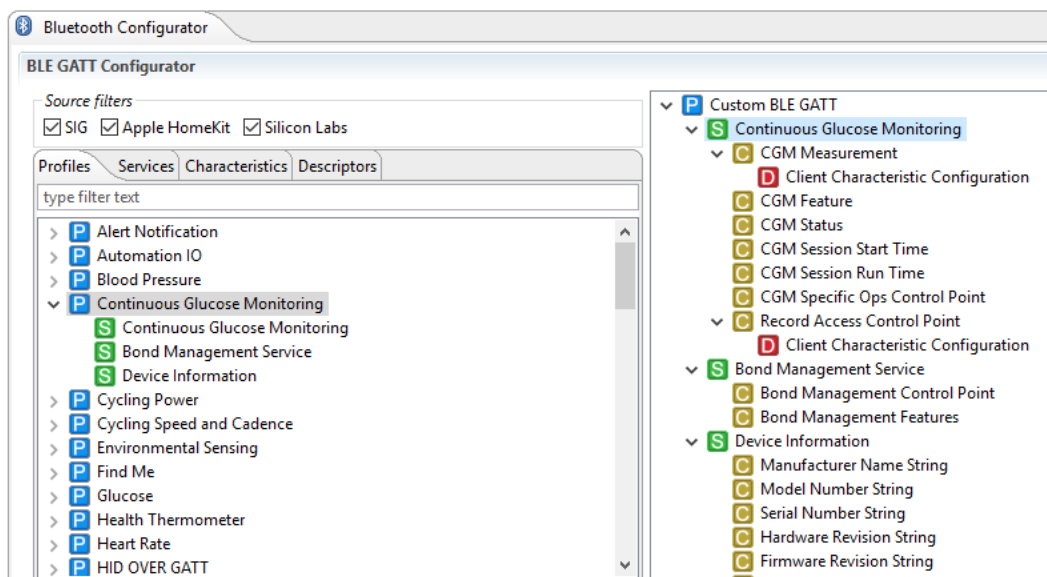

Figure 2-1. Drag and Drop a Profile to Include all the Items

Within the Custom GATT section, drag and drop can be used to reorder items. This saves the trouble of including and configuring the item again. Similarly, an item can be duplicated and moved around in the section.

2.2 Create New Item

New items can be created from the tools menu. New Services get created at the bottom whereas Characteristics and Descriptors are created under the selected item. Descriptors can only be created when you have selected a Characteristic. When you select the Profile and create a new Characteristic, it gets created under the first Service.
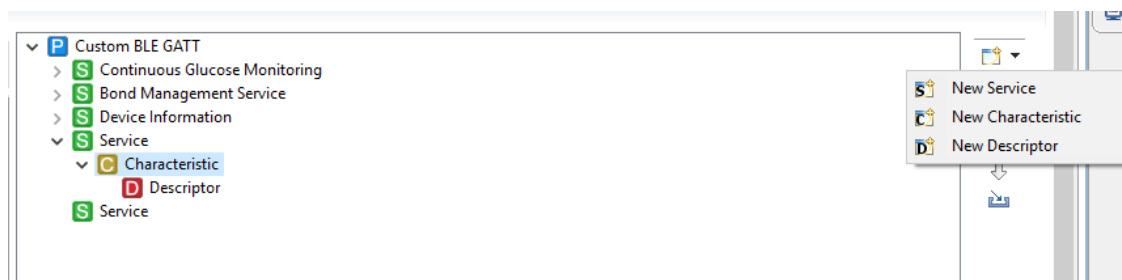
Figure 2-2. Create New Item

When a new item is selected, the Settings section displays the default properties of the item. Here the item can be configured as per the requirements.
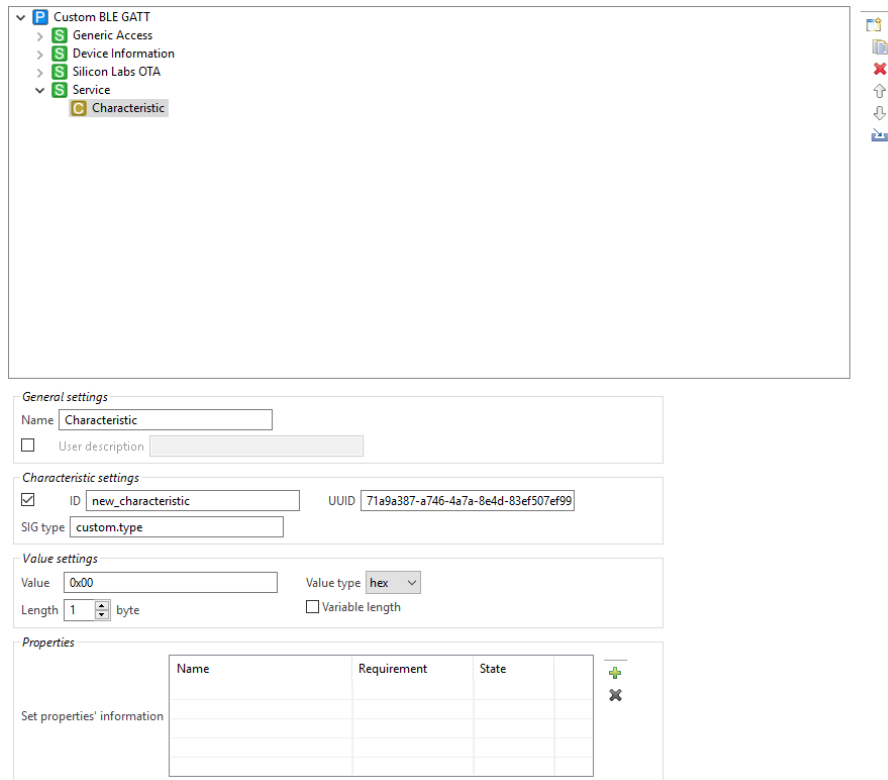
Figure 2-3. Default Values for a Newly Created Characteristic

The application gets local access to the GATT database using the characteristic ID. You can enter this by selecting the checkbox and entering a unique ID. Upon generation, this ID gets a macro in the `gatt_db.h` file as shown below.

```
extern const struct bg_gattdb_def bg_gattdb_data;

#define gattdb_service_changed_char          3
#define gattdb_device_name                   7
#define gattdb_ota_control                   21
#define gattdb_new_characteristic            24
```

UUID or Universally Unique identifier are numbers used to identify Services, Characteristics, and Descriptors uniquely. There are two types of UUID:

1. **16 bit:** These 16 bit UUIDs are predefined by the Bluetooth SIG. Being short they are energy and memory efficient. For example, the Blood Pressure Service has a UUID of 0x1810 whereas the Battery level Characteristic has a UUID of 0x2A19.

2. **128 bit:** This overcomes the limitation of running out of 16 bit UUIDs and gives the power to declare your own UUIDs for Custom Services and Characteristics. These randomly generated UUIDs in the GATT Configurator are of version 4 (random) variant 1. You can use any UUID for a custom Service or Characteristic if it does not overlap with Bluetooth base UUID: xxxxxxxx-0000-1000-8000-00805F9B34FB.

While there is no central authority ensuring other devices don't use the same UUID, there is very little chance (1 in 340 undecillion) that two devices end up with the same UUID.

2.3   Adding Properties

Properties define what actions can be performed for a given Characteristic or Descriptor. For example, in the Blood Pressure Profile, the Blood Pressure Feature has a Mandatory Read property. The green **+** control on the right can be used to add properties to an item. Similarly, the red **x** control can be used to delete the selected item. Clicking on the fields in the table allows you to select the available options for the requirements and the state fields for a given property.

Figure 2-4. Setting properties for a Characteristic or Descriptor

2.4   Adding Capabilities

Bluetooth SDK 2.4 introduced a new feature called Polymorphic GATT that can be used to dynamically show or hide GATT Services and Characteristics. The GATT Configurator implements this feature using GATT capabilities. This section describes how to do it.

To summarize how capabilities work, each Service/Characteristic can declare several capabilities and the state of the capabilities (enable/disable) determines the visibility of those Services/Characteristics as a bit-wise OR operation. For example, the Service/Characteristic is visible when at least one of its capabilities is enabled and it is not visible when all its capabilities are disabled.

Always start by declaring the GATT-level capabilities and defining their default value. That is done by selecting "Custom BLE GATT" and adding the capabilities in "Capability declaration". To add a capability just click the **+** control on the right-hand side and then change the capability name and default value. For example, Appearance, Temperature_measure and Tx_power are added to the profile as shown in the following figure.

Figure 2-5. Declaring GATT-level Capabilities

Once those capabilities are added they become available on each of the services and characteristics. They can be added through the **+** control but this time from the list of capabilities includes all those declared at the GATT level, as shown in the following figure.

Figure 2-6. Including GATT-level Capabilities in a Characteristic

Note: The capabilities state should not be changed during a connection, as that can cause misbehavior. The safest way is to change the capabilities when no devices are connected.

2.5 Including Services

In a Service definition, you can add one or more references to other services, using the Service includes feature. Include definitions consist of a single attribute (the include declaration) that contains all the details required for the client to reference the included service.

Included services can help avoid duplicating data in a GATT server. If a service will be referenced by other services, you can use this mechanism to save memory and simplify the layout of the GATT server.

Start by declaring an ID for each service that needs to be included. Services without an ID cannot be referenced. This is done by selecting the ID checkbox and providing an identifier text for the Service. Next, select the Service to be referenced. In the Settings -> Service includes section, click the **+** control on the right-hand side and click on the Service.
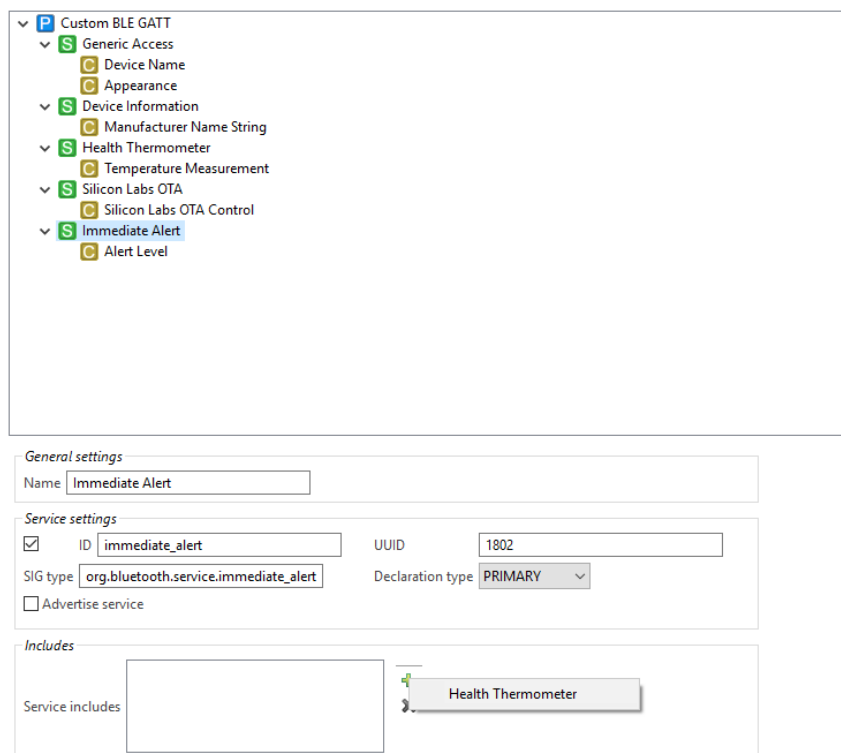
Figure 2-7. Referencing a Service using Service Includes

2.6 Import and Export a GATT Database

Import (⬆): The Import control in the Tools section allows you to import an existing GATT database using an .xml file or a .bgproj file. Note that this will overwrite the existing GATT data.

Export: The GATT database (usually named gatt.xml) gets updated whenever you generate. A validation prompt asks for permission to overwrite files, as shown in the image below. Here you can uncheck files that do not need to be overwritten. All the generated or modified files are in the project folder. The gatt.xml file generated here can be used for exporting the GATT database (File menu).
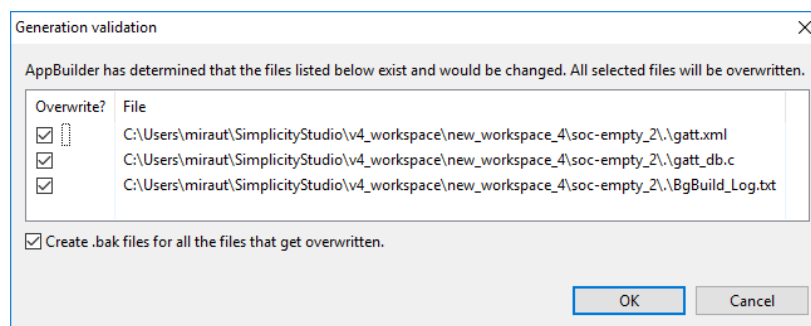
Figure 2-8. GATT Database Generation for Export

Generation also has a feature of making backup files before overwriting existing files. Backup files are saved in `file-name.extension.bak` format for each file. These files are also located in the same folder as the overwritten files.

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

http://www.silabs.com