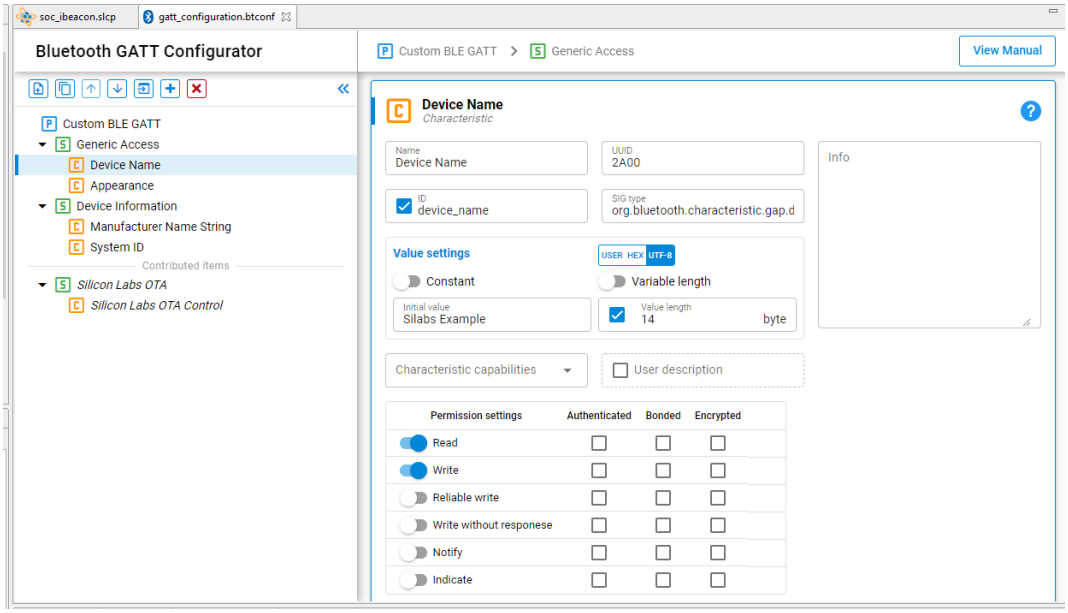
UG438: GATT Configurator User's Guide for *Bluetooth*® LE and Bluetooth Mesh

This guide provides the information needed to effectively use the Bluetooth GATT Configurator provided as a part of Simplicity Studio® 5 with Bluetooth SDK 3.x and Bluetooth Mesh SDK 2.x. 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 of the most common uses of the Configurator, and the usage of the Dynamic GATT Configurator.

KEY POINTS

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



1 GATT Configurator Overview

The GATT Configurator is a simple-to-use tool to help you build your own GATT database. A list of project Profiles/Services/Characteristics/Descriptors is shown on the left and details about the selected item is shown on the right.

The GATT Configurator is composed of a Custom GATT editor on the left, showing a list of project Profiles/Services/Characteristics/Descriptors, and a Settings editor on the right. A SIG selector allows you to add standard elements to the profile.

An options menu is provided at the top of the Custom GATT editor.

The Custom GATT editor is always visible, and the Settings editor opens by default.

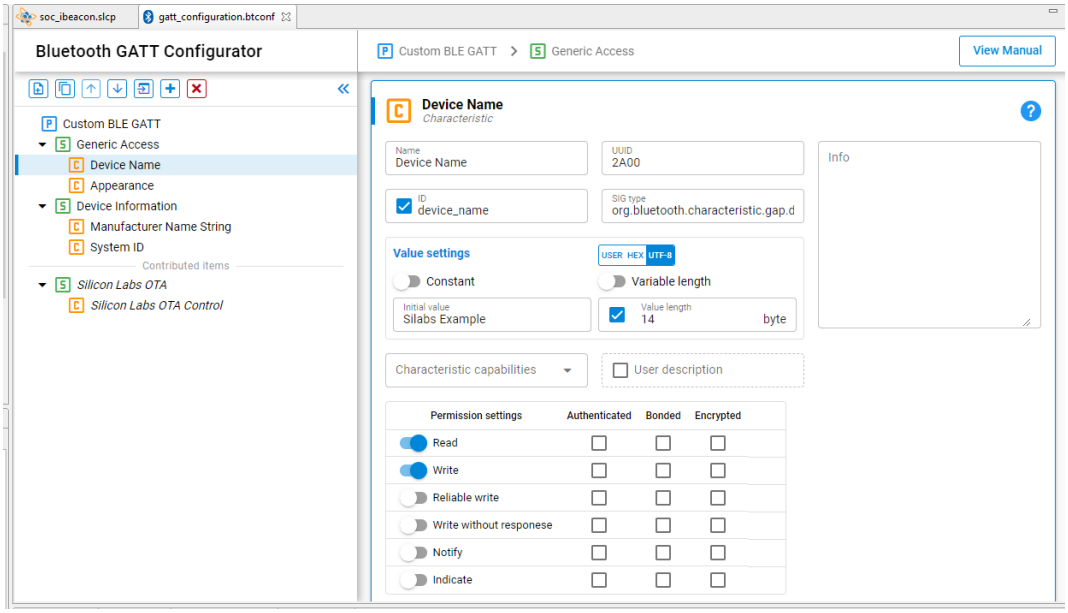
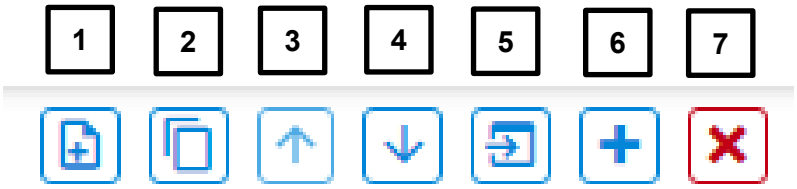


Figure 1.1: GATT Configurator with Settings Editor

The GATT Configurator menu is:



1) Add an item.
2) Duplicate the selected item.
3) Move the selected item up.
4) Move the selected item down.
5) Import a GATT database.
6) Add Predefined (opens the SIG editor).
7) Delete the selected item.

Click Add Predefined (6) to open the SIG selector.

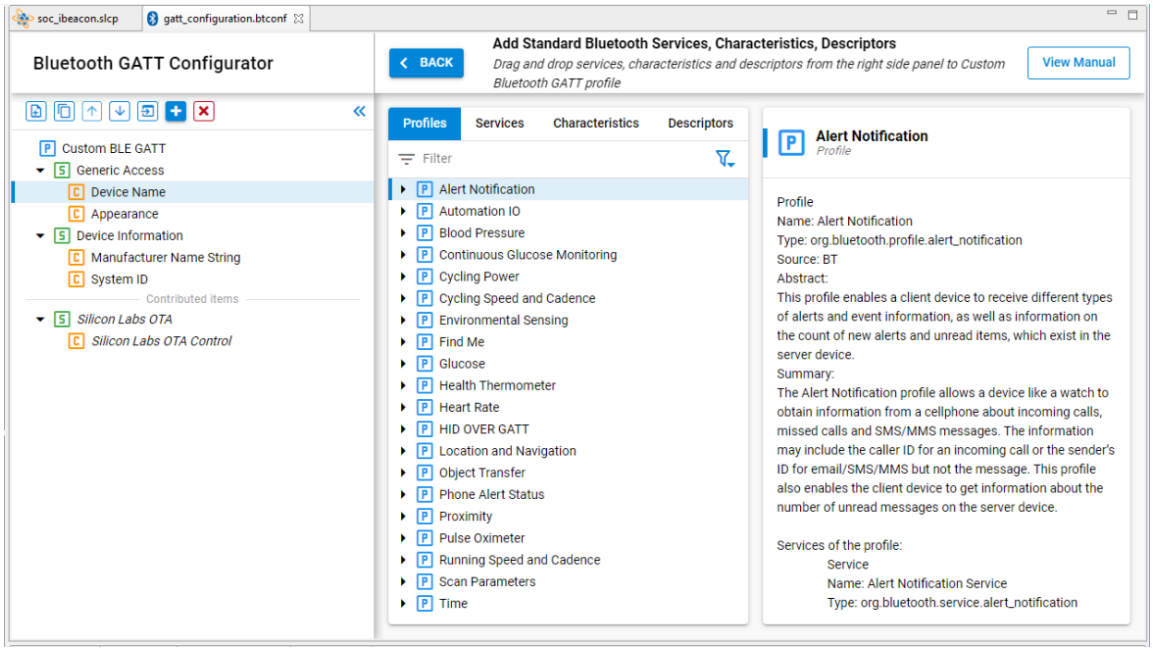


Figure 1.2: GATT Configurator with SIG Selector

1.1 SIG Selector

The SIG Selector displays a list of predefined Profiles, Services, Characteristics, and Descriptors. These items can be filtered, using the filter pane. Tabs allow you to switch between different lists. As shown in the following figure, the pane on the right side of the list displays textual information about the latest selection. To add an item to the Custom GATT editor, mouse over it and click + on the right. The item can then be edited in the Settings Section. The selected SIG service/characteristic/descriptor will be added under the highlighted profile/service/characteristic. Click **< BACK** to return to the Setting Editor.

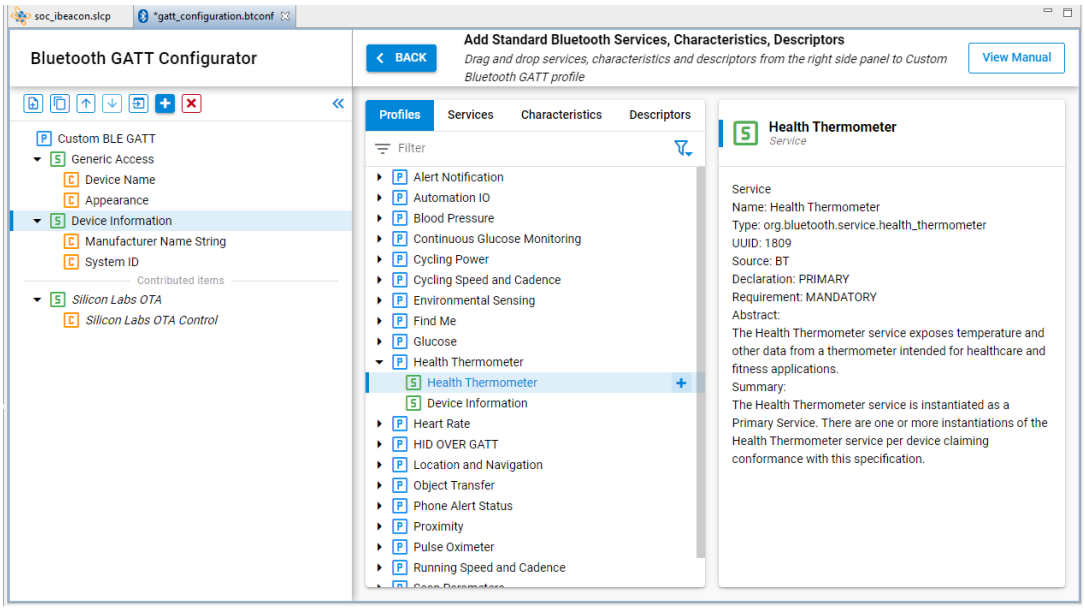


Figure 1.3: SIG Selector

1.2 Custom GATT Editor

The Custom GATT Editor 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 reflects the order in which they exist in the GATT database. When the Settings Editor is open, select an item to see its properties and configuration.

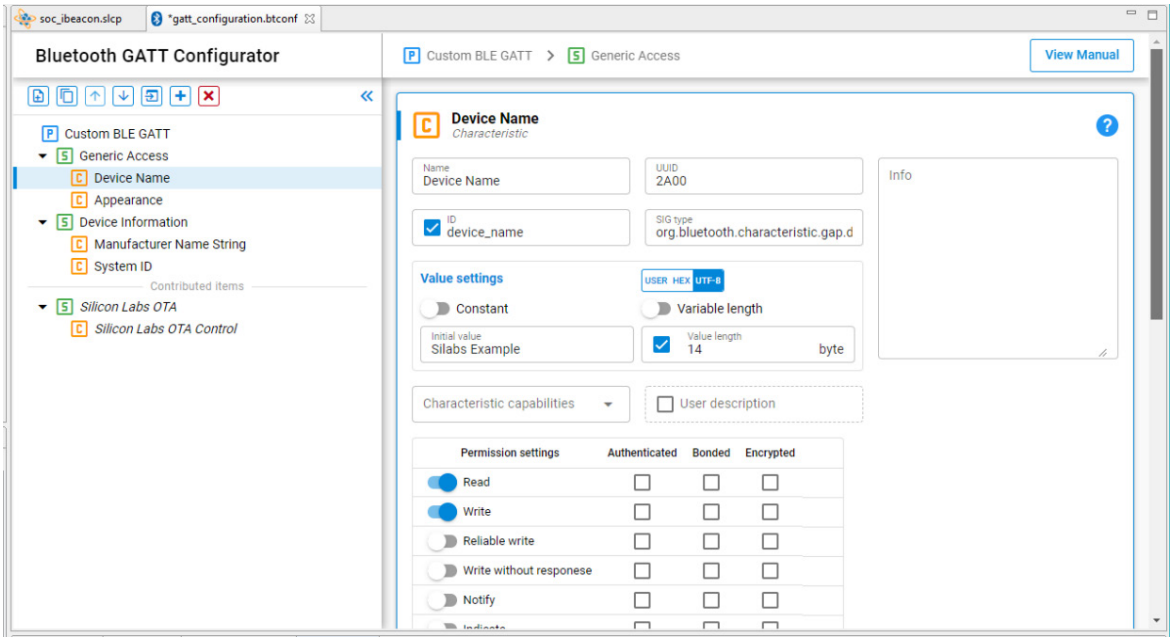


Figure 1.4: GATT Custom GATT Editor with Characteristic Selected

The * next to the configuration file name indicates unsaved changes in the configuration.

Note: The Generic Attribute Service is not listed in the Custom GATT database structure (on the left in the previous figure). This is a special service that is maintained by the stack, and can be added by enabling the **Generic Attribute Service** slider in the settings of the Custom BLE GATT profile. Once enabled, the service will be part of the database. It still will not appear in the Custom GATT database structure, nor on iOS devices as iOS hides this service, but you may see it on Android devices, for example.

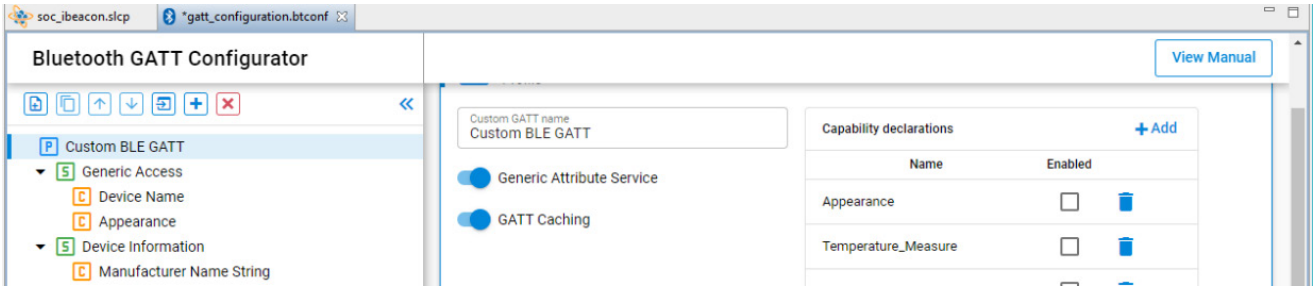


Figure 1.5. Generic Attribute Service Enabled

1.2.1 Contributed Items

Some services are listed in the configurator as "contributed items". This means that their content is defined in other components, and they cannot be edited from this view.

1.3 Settings Section

The Settings editor allows you to configure the properties of items such as Profiles, Services, Characteristics and Descriptors that are present in the Custom GATT editor. 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. You can minimize the Custom GATT editor while editing if you want. All Characteristics for a Service are included in the same Settings editor pane.

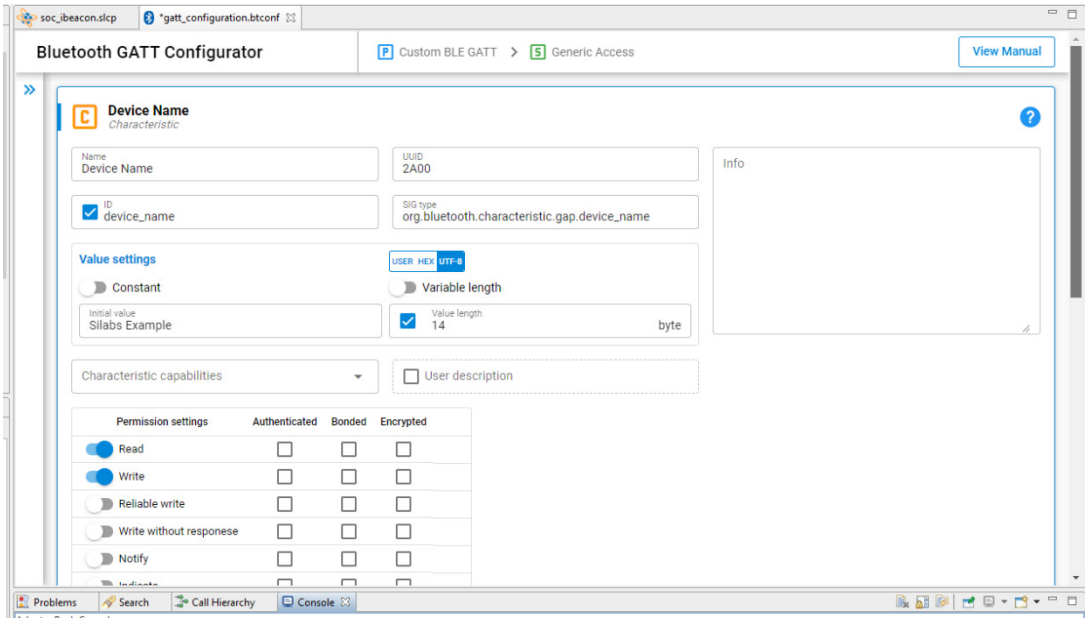


Figure 1.6: Settings Editor

1.4 Generating the GATT Database

Database generation happens automatically when the configuration is saved. The generated source files can be found in the directory named "autogen".

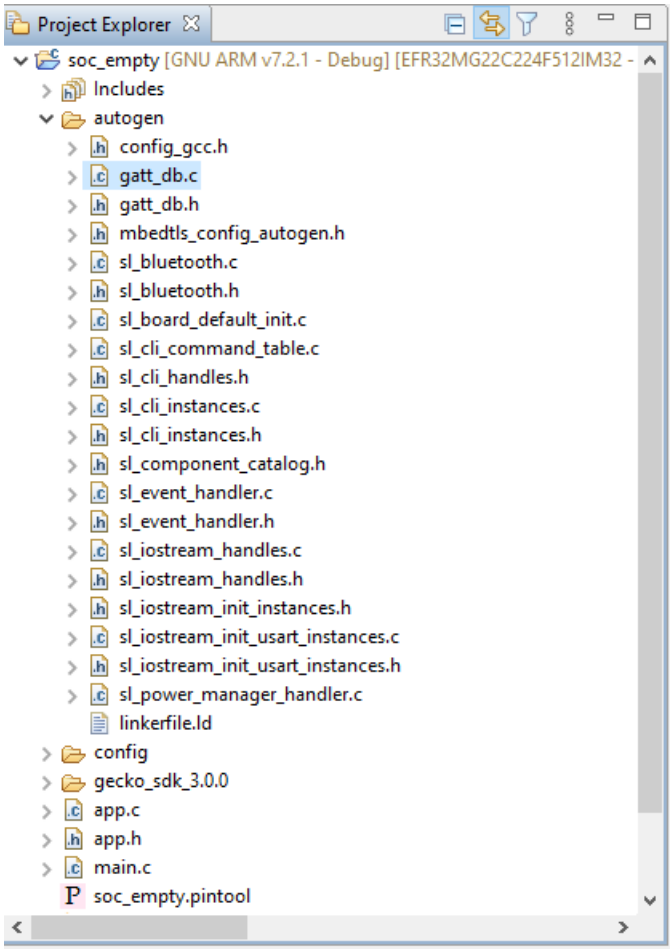


Figure 1.7: Generated Files

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.

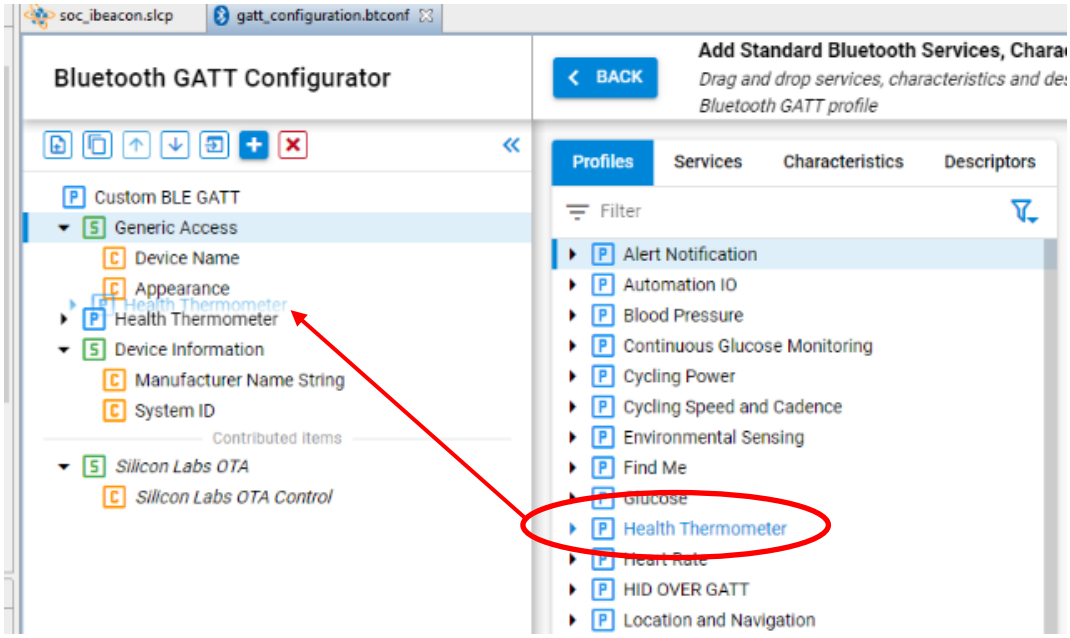


Figure 2.1: Drag and Drop a Service to Include all the Items

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

Use the Add an item (1) menu option to add a new item in the Custom GATT editor. If the profile is selected, a new Service will be created. If a Service is selected, a new Characteristics is created under the selected item. Descriptors can only be created when you have selected a Characteristic.

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.

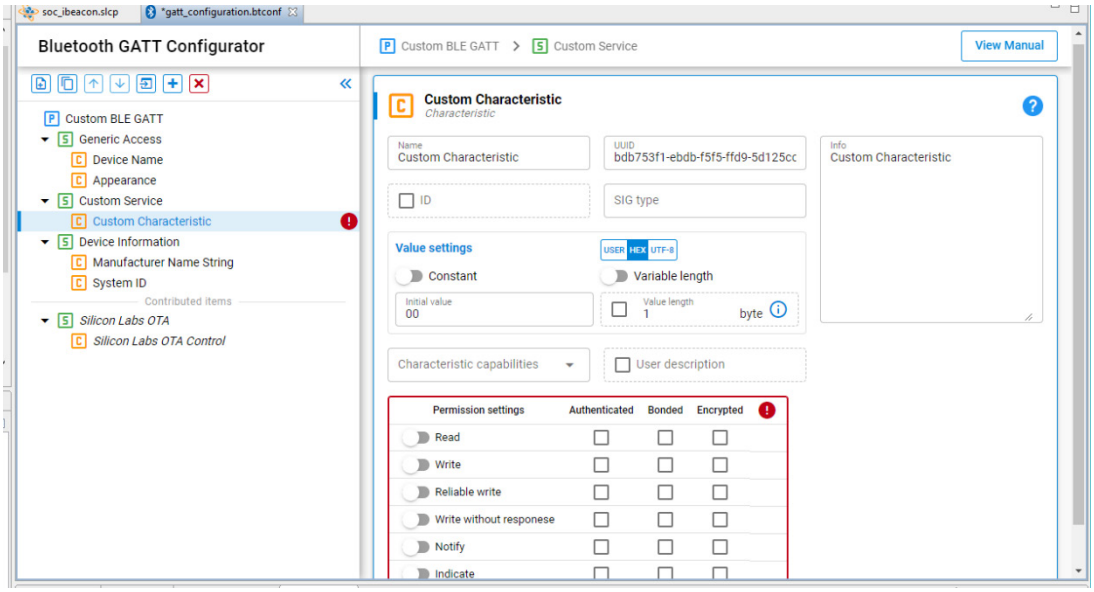


Figure 2.2: Default Values for a Newly Created Characteristic

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.

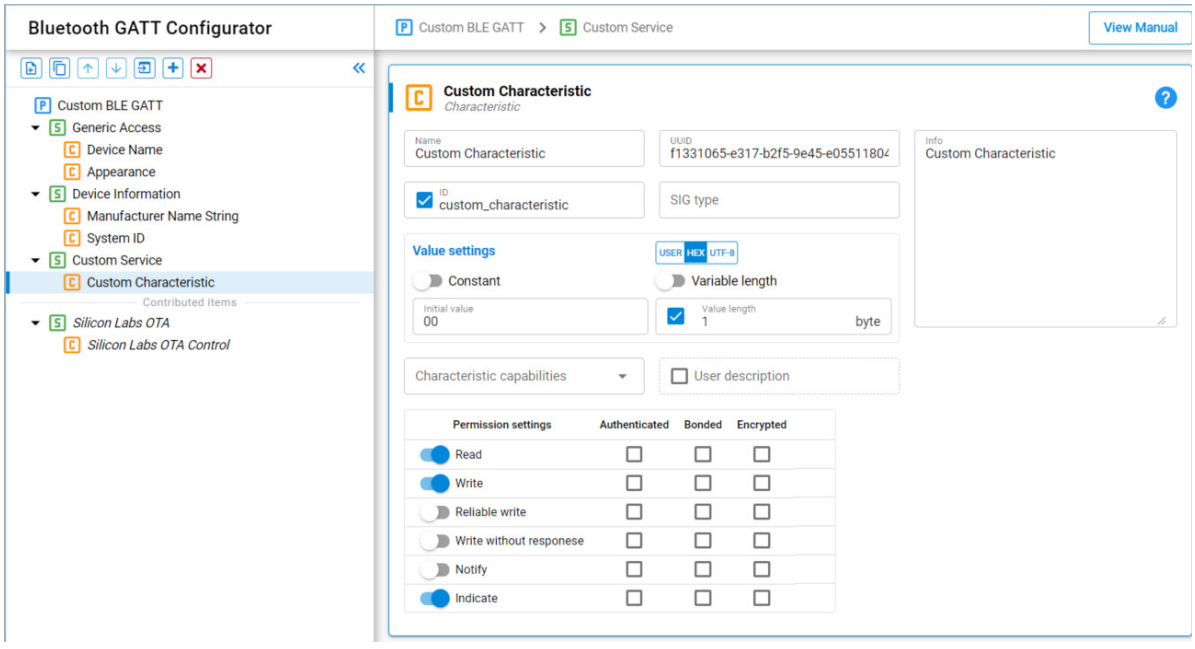


Figure 2.3: Characteristic ID Enabled

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_custom_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 Permissions

Permissions 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. For more information about access types and security requirements see the Properties section of *UG118: Blue Gecko Bluetooth® Profile Toolkit Developer's Guide*.

First the required access types can be enabled with the sliders, and then the security requirements can be selected with the checkboxes.

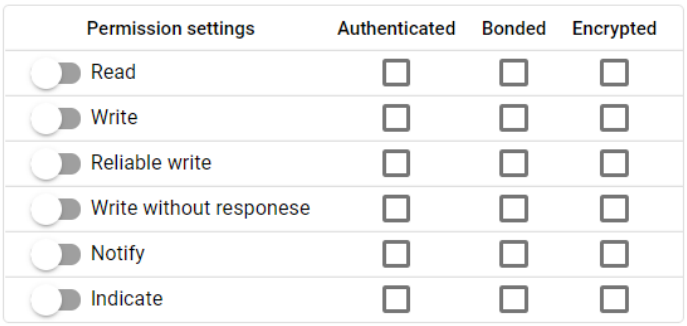


Figure 2.4: Setting Permission for a Characteristic or Descriptor

Note: The *notify* and *indicate* attribute is stored in the SIG defined Client Characteristic Configuration Descriptor (a descriptor with the UUID 0x2902, which will be autogenerated when notifications are enabled). If you manually add a CCCD to the characteristic, the descriptor's value will overwrite this setting. A warning will be displayed on the UI for this case.

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. Select the Custom BLE GATT profile, and click the + control in the "Capability declarations" table. After adding a capability, you can change the name and default value. For example, Appearance, Temperature_Measure and Tx_power are added to the profile as shown in the following figure.

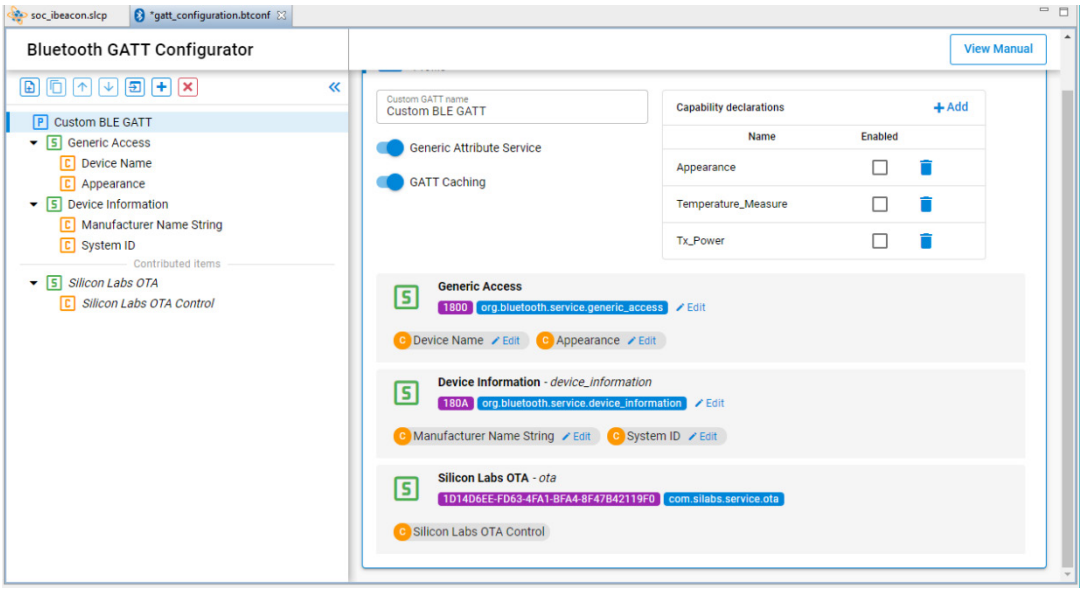


Figure 2.5: Declaring GATT-level Capabilities

Once those capabilities are added, they become available on each of the services and characteristics. They can be declared from the dropdown list in the Settings section, named "Characteristic capabilities".

On the Service and Characteristic level declared capabilities count as enabled, and the ones which were not selected are disabled.

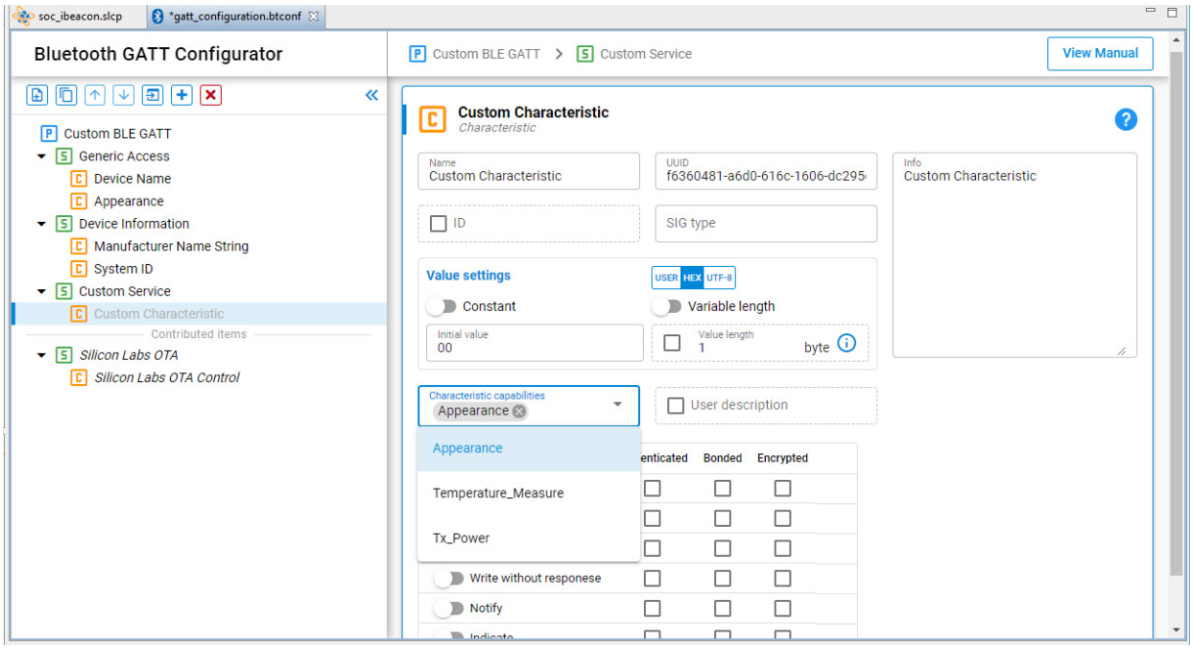


Figure 2.6: Including GATT-level Capabilities in a Characteristic

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 from the dropdown list, named "Service includes".

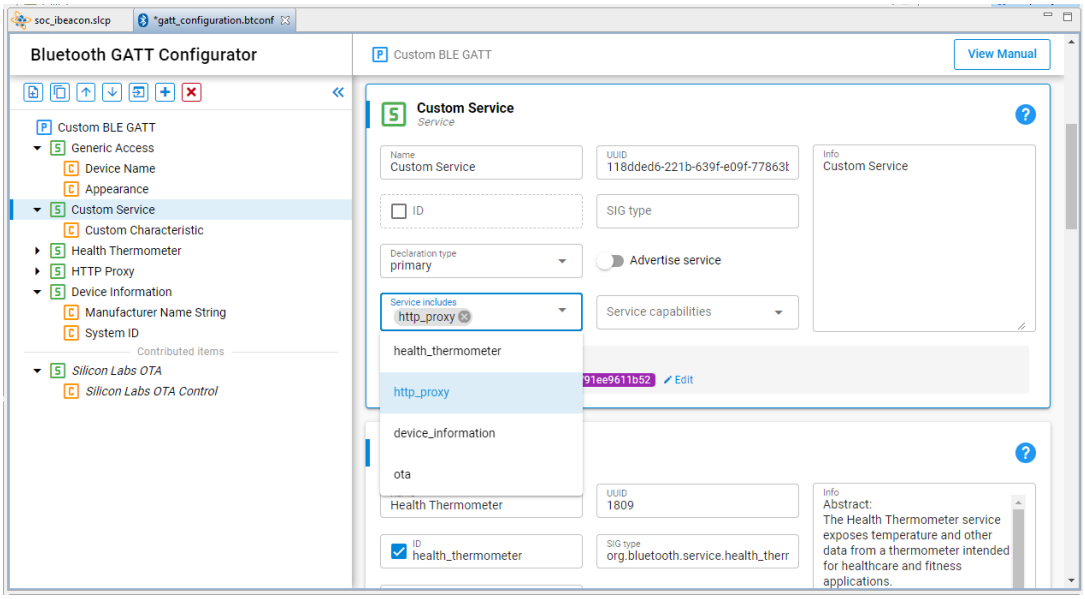


Figure 2.7: Referencing a Service using Service Includes

2.6 Import and Export a GATT Database

Import: The Import control in the Custom GATT toolbar allows you to import an existing GATT database, using a .btconf file. Note that this will overwrite the existing GATT data.

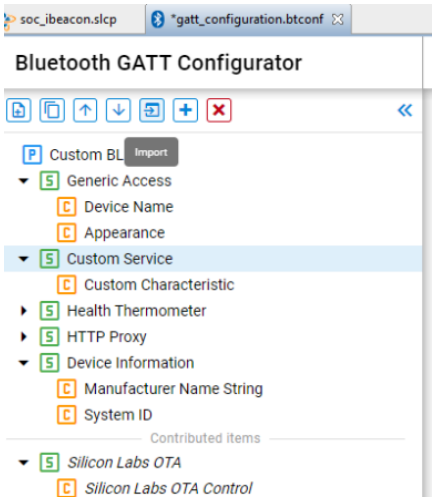


Figure 2.8: Importing a GATT database

Export: There is no separate export function. Another project can directly import the GATT database of this project (named gatt_configuration.btconf)

3 Dynamic GATT Configuration

3.1 Overview

Silicon Labs Bluetooth SDK v3.2 introduced the ability to create the GATT database dynamically with Bluetooth APIs. These dynamically-created GATT attributes can coexist with a static database generated from a GATT XML file. In this case, the attribute table of the static database is placed at the beginning of the database. When new services and characteristics are created dynamically, they are added into the attribute table after the attributes of the static database.

This feature is recommended for NCP projects. With this method, the target application where the GATT database is located does not need to be modified. Therefore, the database can be built from the host side with the APIs. In other cases, the GATT Configurator is the preferred solution.

3.2 Usage

The "Dynamic GATT Database" feature is not included in projects by default. It needs to be installed from the Software Components tab.

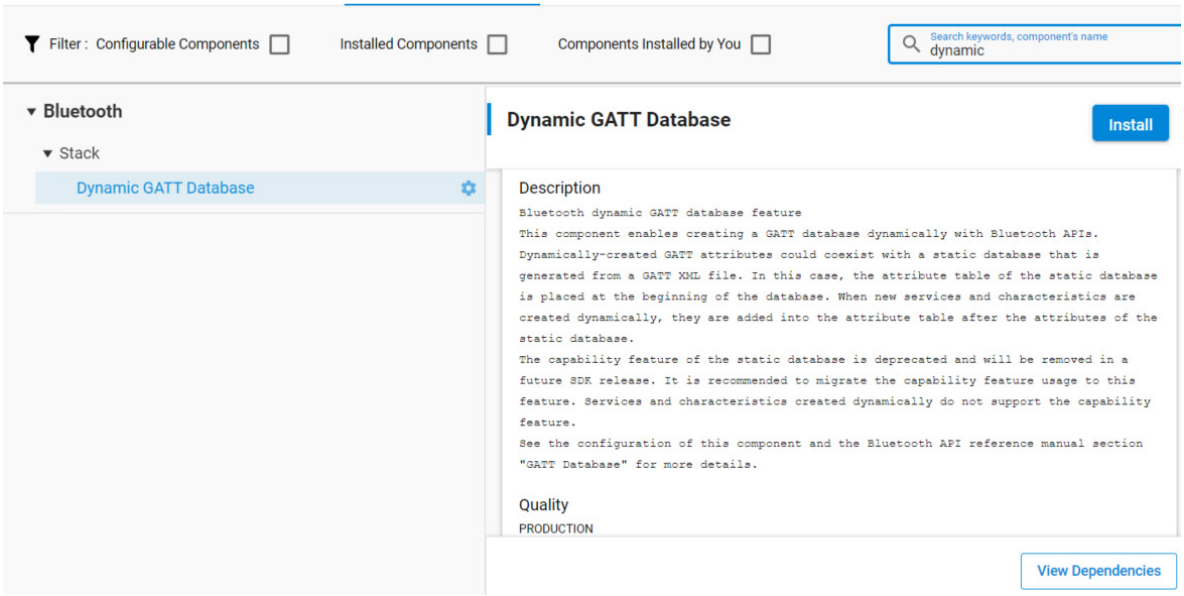


Figure 3.1: Installing the Dynamic GATT Database

Operations on the GATT database are done in a "session". The changes are saved when a session is finished by "committing" the changes. Unsaved changes are invisible to a connected remote GATT client. The modifications only takes effect when the session is finished.

Each added service and characteristic needs to be started with the appropriate API, or they will not be visible to the connected clients. This start and stop mechanism can be used to have a polymorphic GATT database, as the capability feature is not supported in the dynamic GATT databases.

The following code snippet shows how to add the Health Thermometer Service and the Temperature Measurement characteristic dynamically to the database. See the Bluetooth API reference manual section "GATT Database" for more details.

```
//create a session for the database update
sl_bt_gattdb_new_session(&session);
//add the Thermometer service (UUID: 0x1809) to the database, as an advertised primary service
sl_bt_gattdb_add_service(session, sl_bt_gattdb_primary_service,
                                  SL_BT_GATTDB_ADVERTISED_SERVICE,2,uuid_service, &service);
//add the Temperature measurement (UUID:0x2A1C) characteristic to the service
sl_bt_gattdb_add_uuid16_characteristic(session, service, SL_BT_GATTDB_CHARACTERISTIC_INDICATE,
                                       0, 0, uuid_characteristic, sl_bt_gattdb_fixed_length_value,
                                       1, 1, 0, &characteristic);
//activate the new service
sl_bt_gattdb_start_service(session, service);
//activate the new characteristic
sl_bt_gattdb_start_characteristic(session, characteristic);
//save changes and close the database editing session
sl_bt_gattdb_commit(session);
```

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 & Community
www.silabs.com/community

SILICON LABS

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

www.silabs.com