Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

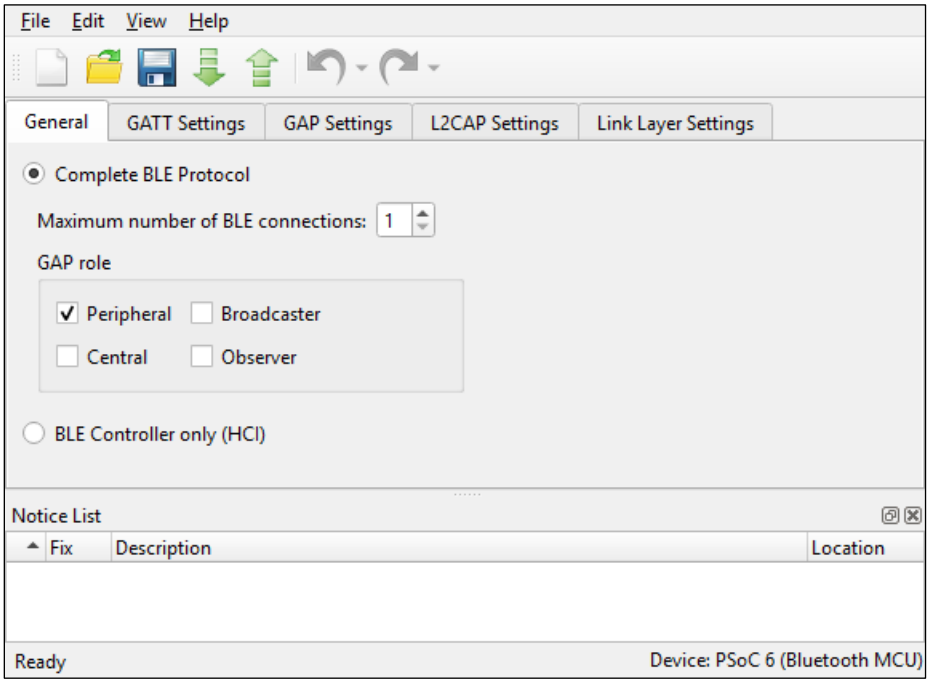
Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

ModusToolbox® Bluetooth Configurator Guide

Version 2.21

Overview

The Bluetooth Configurator is a stand-alone graphical tool included with the ModusToolbox software. This configurator helps generate code for Bluetooth applications, including the Generic Attribute Profile (GATT) database, Service Discovery Protocol (SDP) database (provided at Beta-level support), Generic Access Profile (GAP) configuration, Logical Link Control and Adaption Protocol (L2CAP), and Link Layer parameters.



Supported Devices and Libraries

The Bluetooth Configurator supports various devices and libraries (also known as middleware), and the following table summarizes these options. The Configurator generates different code for each of the supported devices.

Device Type	Library Name	Link
20xxx (Bluetooth SoC)	BT SDK	https://cypresssemiconductorco.github.io/btsdk-docs/BT-SDK/index.html
PSoC 6 (Bluetooth MCU)	PSoC 6 BLESS	https://github.com/cypresssemiconductorco/bless
PSoC 6 MCU with 43xxx Connectivity	AnyCloud SDK	https://github.com/cypresssemiconductorco/bluetooth-freertos https://github.com/cypresssemiconductorco/btstack

The 20xxx (Bluetooth SoC), PSoC 6 (Bluetooth MCU), PSoC 6 MCU with 43xxx Connectivity devices are hereinafter referred to as **20xxx**, **Bluetooth MCU**, and **43xxx** respectively.

SIG-Adopted Profiles and Services

The Bluetooth Configurator supports numerous SIG-adopted Profiles and Services. Also, the Bluetooth Configurator generates all the necessary code for a particular Profile/Service operation in accordance with its configuration.

Custom Profiles

You can create custom BLE Profiles that use existing Services, and you can create custom Services with custom Characteristics and Descriptors.

Launch the Bluetooth Configurator

You can launch the Bluetooth Configurator various ways: as a stand-alone tool, from the Eclipse IDE for ModusToolbox, or from the command line. Then, you can either use the generated source with an Eclipse IDE application, or use it in any software environment you choose.

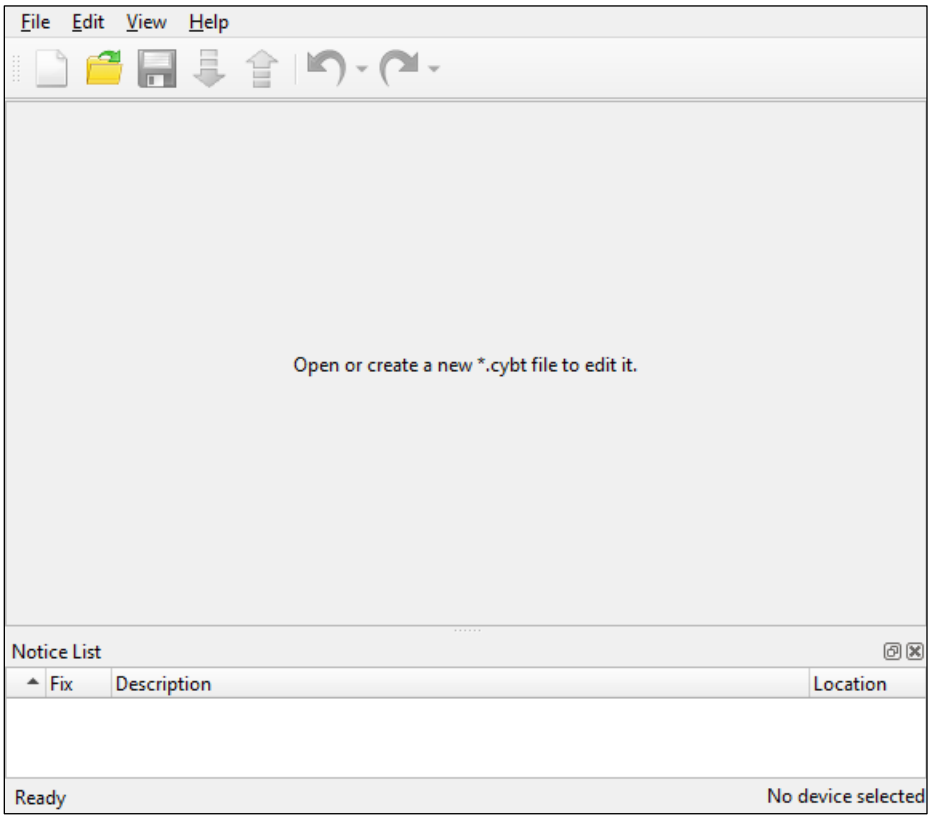
As a Stand-Alone Tool

You can launch the Bluetooth Configurator as a stand-alone tool. By default, it is installed here:

<install_dir>/ModusToolbox/tools_<version>/bt-configurator<version>

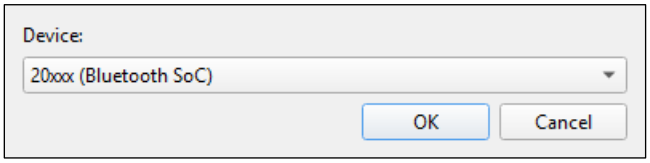
On Windows, launch the tool from the **Start** menu. For other operating systems, navigate to the install location and run the executable.

When opened this way, the Bluetooth Configurator GUI opens without any information. You must open an existing **.cybt* file or create a new one for the application in which you want to configure Bluetooth.



Create New Configuration File

To create a new configuration file, select **File > New** ([**Ctrl**]+[**N**]). The Select Device dialog displays as follows:



The drop-down menu displays the following options:

- **20xxx (Bluetooth SoC)**
- **PSoC 6 (Bluetooth MCU)**
- **PSoC 6 MCU with 43xxx Connectivity**.

Various parameter configuration options depend on the selection, so select the device you want to generate code for.

Open Existing Configuration File

To open an existing configuration file, select **File > Open** ([**Ctrl**]+[**O**]). On the Open Configuration File dialog, navigate to the appropriate directory, and select the desired configuration file (**.cybt*).

From the Eclipse IDE

If there is a **.cybt* file in the application folder, you can launch the Bluetooth Configurator GUI directly from the Eclipse IDE using any of the following methods:

- Double-click on the **.cybt* file in the application.
- Right-click on the top-level application folder, and select **ModusToolbox > Bluetooth Configurator**.
- Click on the "Bluetooth Configurator" link in the IDE Quick Panel after selecting the appropriate project.

Refer to the Eclipse IDE for ModusToolbox User Guide for more details.

If there is no **.cybt* file in the application folder, the options from the menu and Quick Panel read **Bluetooth Configurator (new configuration)**. Select either option, and the Bluetooth Configurator GUI opens with a default configuration (**.cybt*) that will be saved to the *design.cybt* file in the application folder.

From the Command Line

You can run the bt-configurator executable from the command line. However, there are only a few reasons to do this in practice. There is also a bt-configurator-cli executable which re-generates source code based on the latest configuration settings from a command-line prompt or from within batch files or shell scripts. The exit code for the bt-configurator-cli executable is zero if the operation is successful, or non-zero if the operation encounters an error. In order to use the bt-configurator-cli executable, you must provide at least the `--config` argument with a path to the configuration file.

For more information about command-line options, run the bt-configurator or bt-configurator-cli executable using the `-h` option.

Quick Start

1. Launch the Bluetooth Configurator.
2. Use the Bluetooth Configurator to configure the application (GATT, GAP, L2CAP, LL, SDP, etc). Refer to the Parameter Configuration section for more details.

3. Save the configuration file and generated source code.

Code Generation

The Bluetooth Configurator generates code into a *GeneratedSource* directory in your Eclipse IDE application, or into the specified location for non-Eclipse IDE applications. That directory contains the necessary source (.c) and header (.h) files for the generated firmware which uses the relevant driver APIs to configure hardware. The user configuration is saved to the *.cybt file.

20xxx

- *design.cybt* – Contains an XML representation of the configuration. You may specify any file name.
- *cycfg_gatt_db.h* and *cycfg_gatt_db.c* – Contains the GATT database code. Include *cycfg_gatt_db.h* in your application.
- *cycfg_sdp_db.h* and *cycfg_sdp_db.c* – Contains the SDP database code. Include *cycfg_sdp_db.h* in your application.

Bluetooth MCU

- *design.cybt* – Contains an XML representation of the configuration. You may specify any file name.
- *cycfg_ble.h* and *cycfg_ble.c* – Contain the generated C code. Include the header file in your application. Use the generated structures as input parameters for the Cy_BLE_Init() function. Refer to the "Configuration Considerations" section in the *Bluetooth Low Energy (BLE) Middleware API Reference Guide* for more details about how to initialize and enable the Bluetooth Middleware.

43xxx

- *design.cybt* – Contains an XML representation of the configuration. You may specify any file name.
- *cycfg_gatt_db.h* and *cycfg_gatt_db.c* – Contains the GATT database code. Include *cycfg_gatt_db.h* in your application.
- *cycfg_gap.h* and *cycfg_gap.c* – Contains the GAP configuration code.
- *cycfg_bt_settings.h* and *cycfg_bt_settings.c* – Contains the *wiced_bt_cfg_settings_t* initialization structure. Include *cycfg_bt_settings.h* in your application.

GUI Description

The Bluetooth Configurator GUI contains menus and tabs to configure Bluetooth settings, as well as a Notice List and Status Bar to provide indications. The tabs are described under Parameter Configuration.

Menus

File

- **New** – Creates a new configuration file.
- **Open** – Opens and loads an existing configuration file.
- **Save** – Saves changes to the file. If the file does not exist, the **Save** file dialog opens. When saving, it also generates code files.
- **Save As** – Saves changes to a new file location.

- **Import** – Imports a specified configuration file.
- **Export** – Exports the current configuration file into a specified file.
- **Close** – Closes the configuration file. If there are unsaved changes, a dialog opens asking to save or not.
- **Exit** – Closes the configurator.

Edit

- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

View

- **Notice List** – Hides or shows the Notice List pane. The pane is shown by default.
- **Toolbar** – Hides or shows the Toolbar.
- **Reset View** – Resets the view to the default.

Help

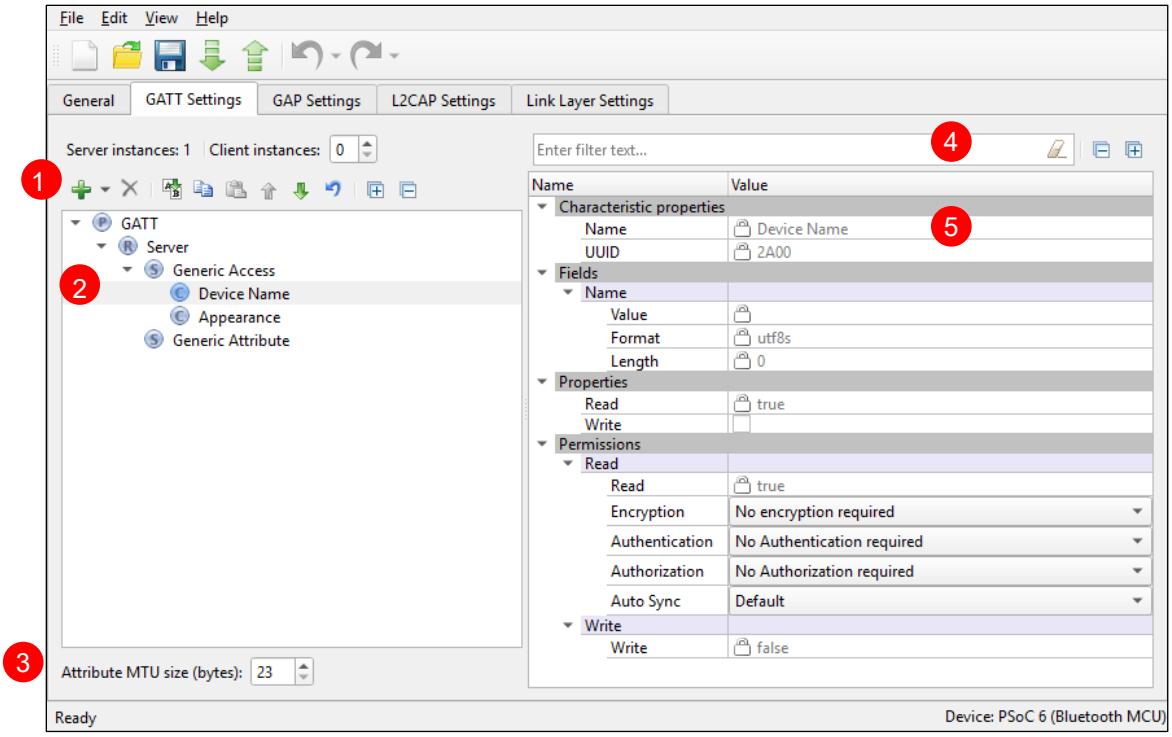
- **View Help** – Opens this document.
- **About** – Opens the About box for version information.

Toolbar

The toolbar contains common commands from the **File** and **Edit** menus, such as New, Open, Save, etc. Use the check box under the **View** menu to show or hide the toolbar.

Tab Components

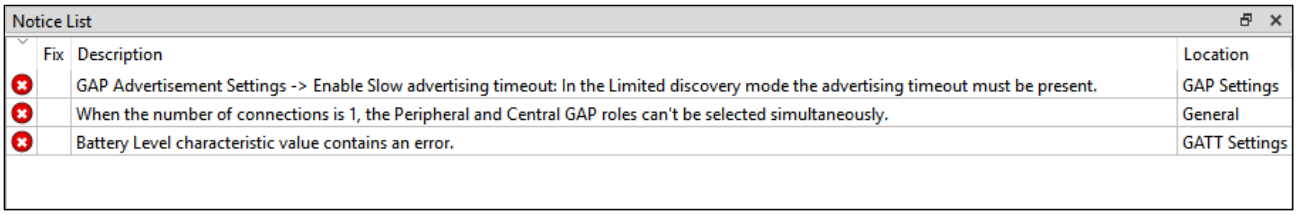
For the different devices, the GUI contains one or more tabs to update various settings. The following shows a typical tab and identifies the key components:



1. The tab toolbar used for commands for a given tab.
2. The configuration tree to add nodes and define parameters on a given node.
3. A typical field to enter data.
4. The toolbar to filter, expand, and collapse parameters.
5. Node configuration settings.

Notice List

The Notice List pane combines notices (errors, warnings, tasks, and notes) from many places in the configuration into a centralized list. If a notice shows a location, you can double-click the entry to show the error or warning.



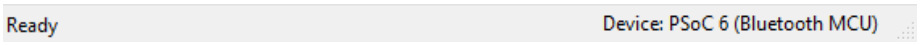
The Notice List pane contains the following columns:

- **Icon** – Displays the icons for the error, warning, task, or note.
- **Fix** – This may display a wrench icon that can be used to automatically address the required notice.
- **Description** – Displays a brief description of the notice.
- **Location** – Displays the specific tab of the message, when applicable.

For more information about the Notice List, refer to the Device Configurator Guide.

Status Bar

The Status Bar displays various information, including a file operation status, hints, input field ranges, etc. On the right side, it displays the selected device.

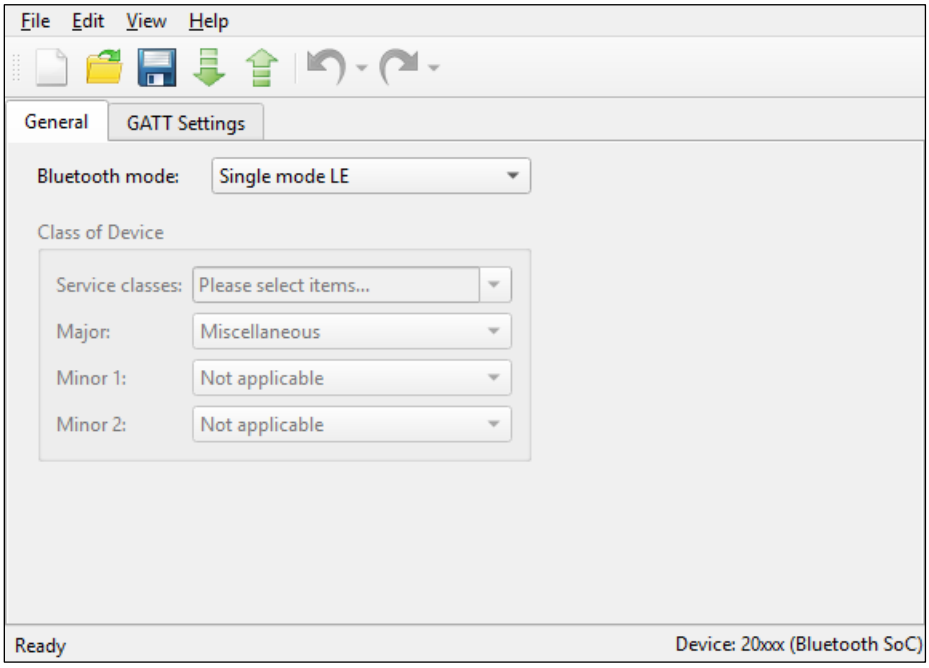


Parameter Configuration

The Bluetooth Configurator contains several tabs in which to configure parameters. The set of active tabs depends on the selected device and other parameter values. See Create New Configuration File.

General Tab (20xxx)

The **General** tab allows general configuration of the Bluetooth resource. For **20xxx** (Bluetooth SoC), the tab looks as follows, depending on the mode selected:



The **General** tab contains the following settings:

Bluetooth Mode

The **Bluetooth mode** parameter supports the following functionality in different modes:

- **Single mode LE** – Only the Low Energy
- **Single mode BR/EDR** – Only the BR/EDR or “classic” Bluetooth
- **Dual mode** – Both the Low Energy and BR/EDR.

Class of Device

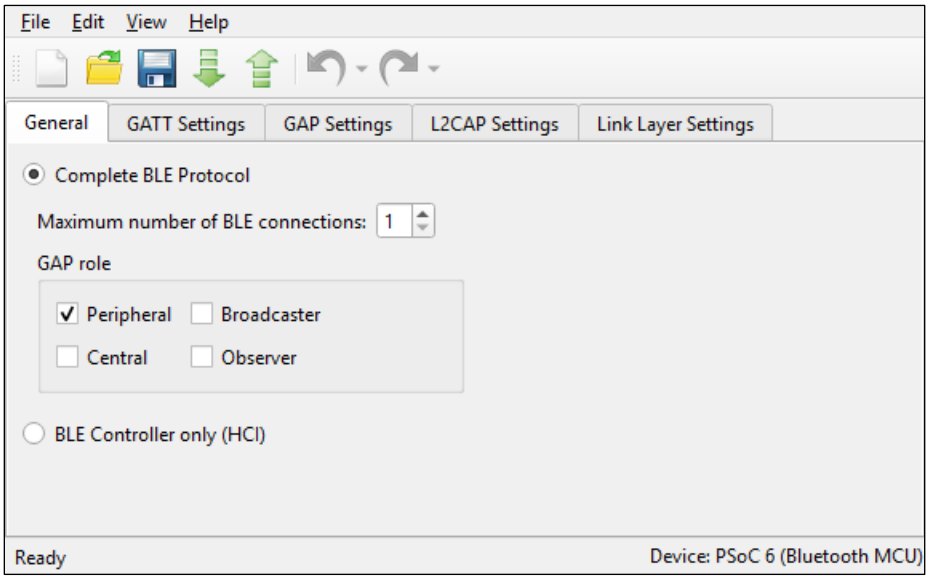
The **Class of Device** parameters group is available in **Single mode BR/EDR** or **Dual mode**. It indicates the type of a device and includes the following parameters:

- **Service classes** – Defines the general categories of devices your application will be associated with.

- **Major** – Major device class. Defines the main device's function.
- **Minor 1** – Minor device class. Extends the **Major** device class value.
- **Minor 2** – Additional minor device class. Extends the **Major** device class value.

General Tab (Bluetooth MCU)

The **Bluetooth MCU General** tab allows general configuration of the Bluetooth resource. The tab displays as follows, depending on the mode selected:



The **Bluetooth MCU General** tab settings:

Complete BLE Protocol

The Complete BLE Protocol mode enables both BT Host and Controller. All GAP roles are exposed for configuration.

Maximum Number of BLE connections

This parameter displays how many BLE connections (both Central and Peripheral) are allowed. The valid range is from 1 to 4.

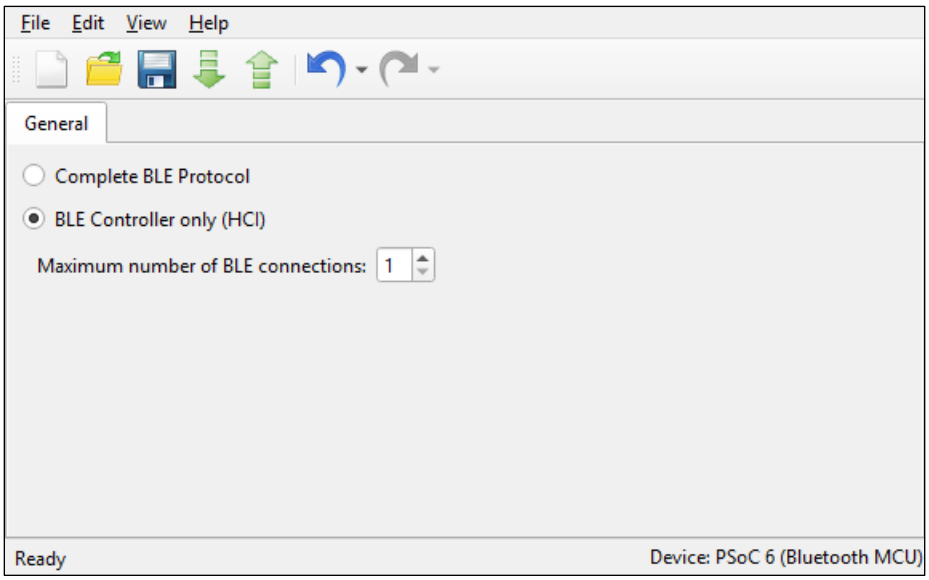
GAP Role

The **GAP role** parameter can take the following values:

- **Peripheral** – Defines a device that advertises using connectable advertising packets and so becomes a slave once connected. Peripheral devices need a Central device, as the Central device initiates connections. Through the advertisement data, a Peripheral device can broadcast the general information about the device.
- **Central** – Defines a device that initiates connections to peripherals and will therefore become a master when connected. Peripheral devices need a Central device, as the Central device initiates connections.
- **Broadcaster** – Defines a device that sends advertising data. It is similar to the Peripheral role. However, Broadcaster does not support connections and can only send data but not receive it.
- **Observer** – Defines a device that scans for Broadcasters and reports the received information to an application. The Observer role does not allow transmissions.

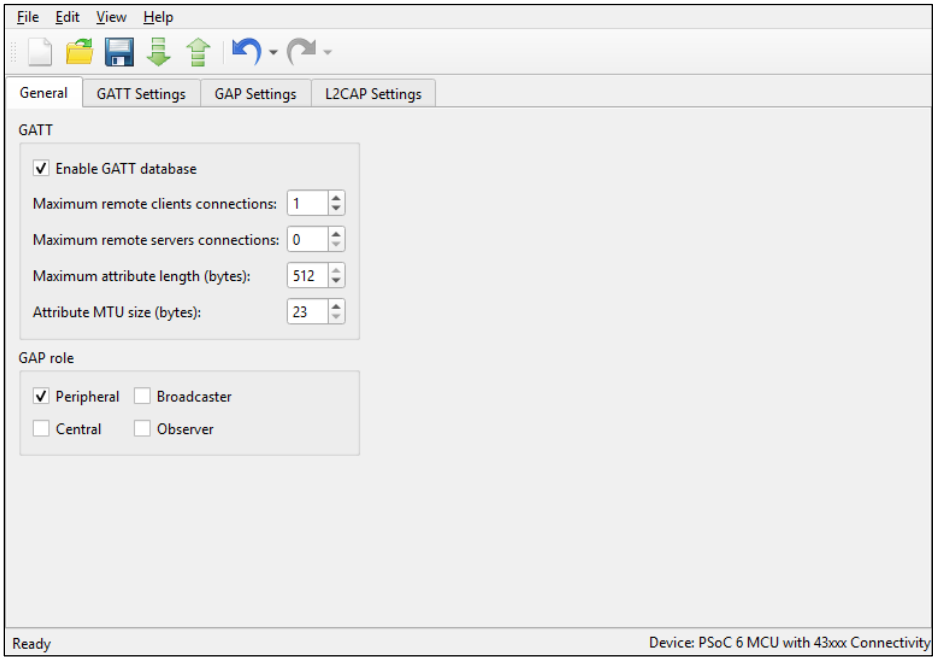
BLE controller only (HCI)

HCI mode enables the use of the device as a BLE controller. Selecting this mode makes all other tabs unavailable.



General Tab (43xxx)

The **43xxx General** tab allows general configuration of the Bluetooth resource. The tab displays as follows:



The **43xxx General** tab settings:

Enable GATT database

This parameter enables the GATT database. When not enabled, the **GATT Settings** tab hides and files *cycfg_gatt_db.h* and *cycfg_gatt_db.c* are not generated.

Maximum remote clients connections

This parameter displays the maximum number of clients that a local server can connect to.

Maximum remote servers connections

This parameter displays the maximum number of servers that a local client can connect to.

Maximum attribute length

This value is the maximum length of an attribute in the design. The valid range is from 23 to 512 bytes.

Attribute MTU size

This value is the Maximum Transmission Unit size (bytes) of an attribute in the design. The valid range is from 23 to 517 bytes. This value is used to respond to an Exchange MTU request from the GATT Client.

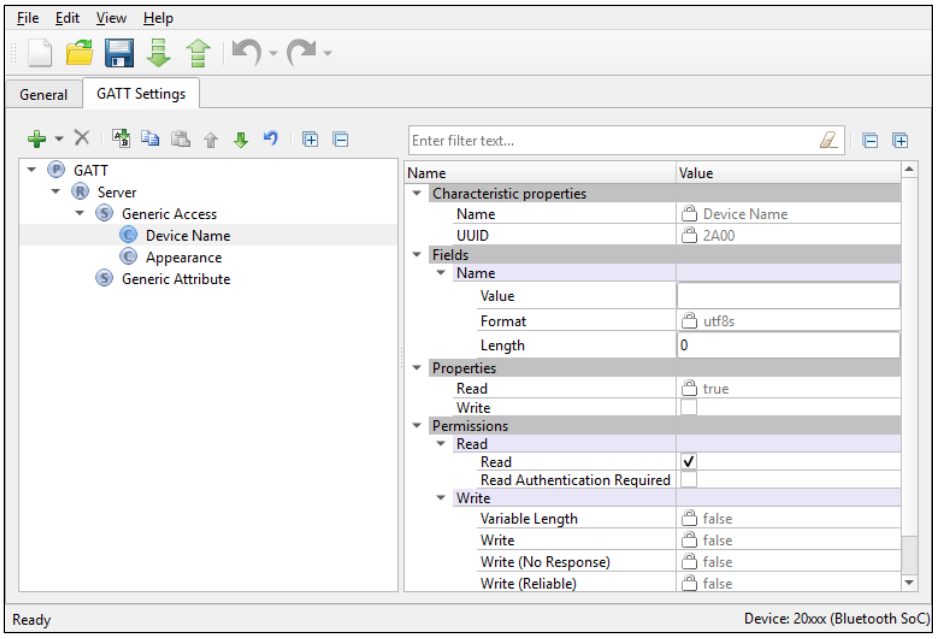
GAP Role

See GAP Role

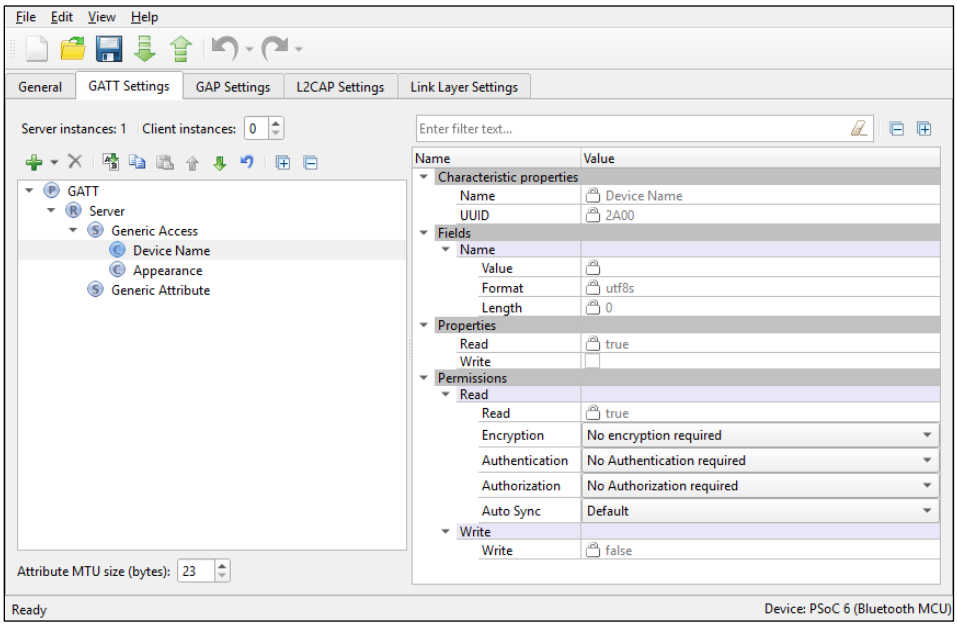
GATT Settings Tab (All Devices)

The **GATT Settings** tab is used to configure Profile-specific parameters. The **GATT Settings** tab has three areas: toolbars, a Profiles tree, and a parameters configuration section. Depending on the selected device, some parameters may be unavailable.

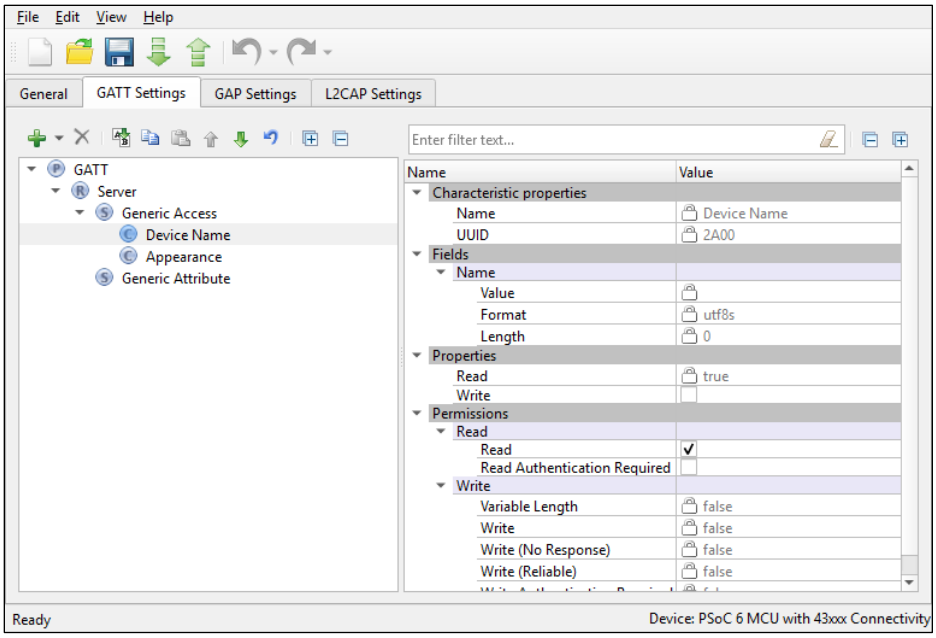
For **20xxx,** the tab displays as follows:



For **Bluetooth MCU**, the tab displays as follows:



For **43xxx**, the tab displays as follows:



Toolbars

The toolbars contain navigation options and various commands to add or delete Services, Characteristics, and Descriptors.

- **Server instances** – The number of GATT Server instances. The Bluetooth resource supports a single instance of a GATT Server (single GATT database). You can add additional Services or complete Profiles to the existing Server Profiles tree to build the GATT database. This single GATT database will be re-used across all BLE connections.

  Note The CCCD values for each of active connections will be unique.

Note Applicable for **Bluetooth MCU** only.

- **Client instances** – The number of GATT Client instances. One GATT Client instance exists per connection. You can configure up to four GATT Client instances. All GATT Client instances have one common Client Profiles tree configuration.

  Note Applicable for **Bluetooth MCU** only.

- **Add Profile** – Available when the GATT node is highlighted in the Profile tree. It allows adding a whole Profile to the Profiles tree. This option does not remove existing Services from the tree. Several Profiles can exist in the tree simultaneously.
- **Add Service** – Available when the **Profile Role** is highlighted in the Profile tree. Allows loading of Services in the selected **Profile Role**. In the GATT server configuration, adds the selected service data to the server GATT database and enables service-specific APIs.

  Not applicable to 20xxx and 43xxx devices, in the GATT client configuration, the data structures for auto discovery of this service are created. If services not populated in the GUI are discovered during auto discovery, this option ignores those services and the application is responsible for discovering the details of such services. Refer to the Profiles section for the available Services.
- **Add Characteristic** – Available when a Service is highlighted in the Profile tree. The Characteristic options are unique to each Service and are loaded automatically when a Service is added to the design. The **Add Characteristic** button can be used to manually add a new Characteristic to the Service. All Characteristics for the above-mentioned Services plus Custom Characteristic are available for selection.
- **Add Descriptor** – Available when a Characteristic is highlighted in the Profile tree. Similar to the Characteristic options, the Descriptor options are unique to a Characteristic and are all automatically loaded when a Characteristic is added to the design. For more information about BLE Characteristic Descriptors, refer to developer.bluetooth.org.

  Note You should be a member of Bluetooth SIG to have full access to this site.
- **Delete** – Deletes the selected Service, Characteristic, or Descriptor.
- **Rename** – Renames the selected item in the Profiles tree.
- **Copy/Paste** – Copies/pastes items in the Profiles tree.
- **Move Up/Down** – Moves the selected item up or down in the Profiles tree.
- **Reset branch to default** – Resets the selected item with child items in the Profiles tree to the default.
- **Expand All** – Expands all items in the Profiles tree.
- **Collapse all Services** – Collapses all Services in the Profiles tree.

Profiles Tree

The Profiles tree is used to view GATT Services, Characteristics, and Descriptors of the GATT Server and Client roles (GATT Client role is supported for **Bluetooth MCU** only). By navigating through the tree, you can quickly add, delete, or modify Services, Characteristics, and Descriptors using the toolbar buttons or the context menu. You can configure the parameters by clicking an item on the tree. These parameters will show in the Parameters Configuration section.

The tree may contain the following nodes:

- Profile
- Profile Role
- Service
- Characteristic
- Descriptor

In addition, nodes may include colors and icons as follows:

- Shaded icon – The node is mandatory.
- White icon– The node is optional.
- A '+' sign – The Service includes other Services.
- Error icon– The node parameters have errors.

Parameters Configuration

The Parameters Configuration section allows you to configure a Profile, Service, or Characteristic by selecting the type of Service or Characteristic in the tree. The filter box above the list of parameters allows limiting the items shown in the pane.

Attribute MTU Size

The Maximum Transmission Unit size (bytes) of an attribute to be used in the design. The valid range is from 23 to 512 bytes. This value is used to respond to an Exchange MTU request from the GATT Client.

Note Applicable for **Bluetooth MCU**.

Profiles

You can add a whole Profile to the Profiles tree from a list of supported Profiles.

Notes

- All Profiles must have a **Generic Access Service** and a **Generic Attribute Service**.
- For **20xxx** and **43xxx** devices, only the **GATT Server** role is configurable.
- The Service Characteristics are configurable only if they belong to a GATT Server node.
- The security settings located in the **GAP Settings** tab are applied globally. In addition to this, you may manually configure the security of each Characteristic/Descriptor.

The following Profiles are available for selection:

Alert Notification

This Profile enables a GATT Client device to receive different types of alerts and event information, as well as information on the count of new alerts and unread items, which exist in the GATT Server device.

- **Alert Notification Server** Profile role – Specified as a GATT Server. Requires the following Service: **Alert Notification Service**.
- **Alert Notification Client** Profile role – Specified as a GATT Client.

Refer to the Alert Notification Profile Specification for detailed information about the Alert Notification Profile.

Automation IO

This Profile enables a device to connect and interact with an Automation IO Module (IOM) in order to access digital and analog signals.

- **Automation IO Server** Profile role – Specified as a GATT Server. Requires the following Service: **Automation IO Service**.
- **Automation IO Client** Profile role – Specified as a GATT Client.

Refer to the Automation IO Profile Specification for detailed information about the Automation IO Profile.

Blood Pressure

This Profile enables a device to connect and interact with a Blood Pressure Sensor device for use in consumer and professional health care applications.

- **Blood Pressure Sensor** Profile role – Specified as a GATT Server. Requires the following Services: **Blood Pressure Service**, **Device Information Service**.
- **Blood Pressure Collector** Profile role – Specified as a GATT Client. Requires support of the following Services: **Blood Pressure Service**. Support of **Device Information Service** is optional.

Refer to Blood Pressure Profile Specification for detailed information about the Blood Pressure Profile.

Continuous Glucose Monitoring

This Profile enables a device to connect and interact with a Continuous Glucose Monitoring Sensor device for use in consumer healthcare applications.

- **Continuous Glucose Monitoring Sensor** Profile role – Specified as a GATT Server. Requires the following Services: **Continuous Glucose Monitoring Service**, **Device Information Service**. Optionally may include **Bond Management Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Services: **Continuous Glucose Monitoring Service**. Support of **Bond Management Service** and **Device Information Service** is optional.

Refer to Continuous Glucose Monitoring Profile Specification for detailed information about the Continuous Glucose Monitoring Profile.

Cycling Power

This Profile enables a Collector device to connect and interact with a Cycling Power Sensor for use in sports and fitness applications.

- **Cycling Power Sensor** Profile role – Specified as a GATT Server. Requires the following Service: **Cycling Power Service**. Optionally may include **Device Information Service** and **Battery Service**.
- **Cycling Power Sensor and Broadcaster** Profile role. Requires the following Service: **Cycling Power Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Service: **Cycling Power Service**. Support of **Device Information Service** and **Battery Service** is optional.

Refer to Cycling Power Profile Specification for detailed information about the Cycling Power Profile.

Cycling Speed and Cadence

This Profile enables a Collector device to connect and interact with a Cycling Speed and Cadence Sensor for use in sports and fitness applications.

- **Cycling Speed and Cadence Sensor** Profile role – Specified as a GATT Server. Requires the following Service: **Cycling Speed and Cadence Service**. Optionally may include **Device Information Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Service: **Cycling Speed and Cadence Service**. Support of **Device Information Service** is optional.

Refer to Cycling Speed and Cadence Profile Specification for detailed information about the Cycling Speed and Cadence Profile.

Environmental Sensing Profile

This Profile enables a Collector device to connect and interact with an Environmental Sensor for use in outdoor activity applications.

- **Environmental Sensor** Profile role – Specified as a GATT Server. Requires the following Service: **Environmental Sensing Service**. Optionally may include **Device Information Service** and **Battery Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Service: **Environmental Sensing Service**. Support of **Device Information Service** and **Battery Service** is optional.

Refer to Environmental Sensing Profile Specification for detailed information about the Environmental Sensing Profile.

Find Me

This Profile defines the behavior when a button is pressed on one device to cause an alerting signal on a peer device.

- **Find Me Target** Profile role – Specified as a GATT Server. Requires the following Service: **Immediate Alert Service**.
- **Find Me Locator** Profile role – Specified as a GATT Client. Requires support of the following Service: **Immediate Alert Service**.

Refer to Find Me Profile Specification for detailed information about the Find Me Profile.

Glucose

This Profile enables a device to connect and interact with a Glucose Sensor for use in consumer healthcare applications.

- **Glucose Sensor** Profile role – Specified as a GATT Server. Requires the following Services: **Glucose Service**, **Device Information Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Service: **Glucose Service**. Support of **Device Information Service** is optional.

Refer to Glucose Profile Specification for detailed information about the Glucose Profile.

Health Thermometer

This Profile enables a Collector device to connect and interact with a Thermometer sensor for use in healthcare applications.

- **Thermometer** Profile role – Specified as a GATT Server. Requires the following Services: **Health Thermometer Service**, **Device Information Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Service: **Health Thermometer Service**. Support of **Device Information Service** is optional.

Refer to Health Thermometer Profile Specification for detailed information about the Health Thermometer Profile.

HTTP Proxy

This Service allows a Client device, typically a sensor, to communicate with a Web Server through a gateway device. HTTP Proxy Service is not available in the **Add Profile** drop-down list. It can be added as a separate Service.

Refer to HTTP Proxy Service Specification for detailed information about the HTTP Proxy Service.

Heart Rate

This Profile enables a Collector device to connect and interact with a Heart Rate Sensor for use in fitness applications.

- **Heart Rate Sensor** Profile role – Specified as a GATT Server. Requires the following Services: **Heart Rate Service**, **Device Information Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the **Heart Rate Service**. Support of **Device Information Service** is optional.

Refer to Heart Rate Profile Specification for detailed information about the Heart Rate Profile.

HID over GATT

This Profile defines how a device with BLE wireless communications can support HID Services over the BLE protocol stack using the Generic Attribute Profile.

- **HID Device** Profile role – Specified as a GATT Server. Requires the following Services: **HID Service**, **Battery Service**, and **Device Information Service**. Optionally may include **Scan Parameters Service** as part of the **Scan Server** role of the **Scan Parameters** Profile. **HID Device** supports multiple instances of **HID Service** and **Battery Service** and may include any other optional Services.
- **Boot Host** Profile role – Specified as a GATT Client. Requires support of the following Service: **HID Service**. Support of **Battery Service** and **Device Information Service** is optional.
- **Report Host** Profile role – Specified as a GATT Client. Requires support of the following Services: **HID Service**, **Battery Service**, **Device Information Service**. Support of **Scan Client** role of the **Scan Parameters** is optional.
- **Report and Boot Host** Profile role – Specified as a GATT Client. Requires support of the following Services: **HID Service**, **Battery Service**, **Device Information Service**. Support of **Scan Client** role of the **Scan Parameters** is optional.

Refer to HID over GATT Profile Specification for detailed information about the HID over GATT Profile.

Indoor Positioning

This Service exposes location information to support mobile devices to position themselves in an environment where GNSS signals are not available. For example, on indoor premises. The location information is mainly exposed via advertising and the GATT-based service is primarily intended for configuration.

The Indoor Positioning Service is not available in the Profile drop-down list. It can be added as a separate Service.

Refer to Indoor Positioning Service Specification for detailed information about the Indoor Positioning Service.

Internet Protocol Support

This Profile provides the support of exchanging IPv6 packets between devices over the Bluetooth Low Energy transport. The IPSP defines two roles – Node role and Router role. A device may support both Node role and Router role. A device supporting the Node role can be a sensor or actuator. A device supporting the Router role can be an Access Point (such as home router, mobile phone, or similar).

- **Node** Profile role – Specified as a GATT Server. Requires the following Service: **Internet Protocol Support Service**.
- **Router** Profile role – Specified as a GATT Client. Requires support of the following Services: **Internet Protocol Support Service**.

Refer to Internet Protocol Support Profile Specification for detailed information about IPSP.

Location and Navigation

This Profile enables devices to communicate with a Location and Navigation Sensor for use in outdoor activity applications.

- **Location and Navigation Sensor** Profile role – Specified as a GATT Server. Requires the following Service: **Location and Navigation Service**. Optionally may include **Device Information Service** and **Battery Service**.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Services: **Location and Navigation Service**. Support of **Device Information Service** and **Battery Service** is optional.

Refer to Location and Navigation Profile Specification for detailed information about the Location and Navigation Profile.

Phone Alert Status

This Profile enables a device to alert its user about the alert status of a phone connected to the device.

- **Phone Alert Server** Profile role – Specified as a GATT Server. Requires the following Services: **Phone Alert Status Service**.
- **Phone Alert Client** Profile role – Specified as a GATT Client. Requires support of the following Service: **Phone Alert Service**.

Refer to Phone Alert Status Profile Specification for detailed information about the Phone Alert Status Profile.

Proximity

This Profile enables the proximity monitoring between two devices.

- **Proximity Reporter** Profile role – Specified as a GATT Server. Requires the following Service: **Link Loss Service**. Optionally may include **Immediate Alert Service** and **Tx Power Service** if both are used. Using only one of the optional Services is not allowed.
- **Proximity Monitor** Profile role – Specified as a GATT Client. Requires support of the following Services: **Link Loss Service**. Support of **Immediate Alert Service** and **Tx Power Service** is optional. Same restrictions apply as to **Proximity Reporter**.

Refer to Proximity Profile Specification for detailed information about the Proximity Profile.

Pulse Oximeter

This Profile enables a device to connect and interact with a Pulse Oximeter device for use in consumer and professional health care applications.

- **Pulse Oximeter Sensor** Profile role – Specified as a GATT Server. Requires the following Services: **Pulse Oximeter Service**, **Device Information Service**. Optionally may include Bond Management Service, Current Time Service and Battery Service.
- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Services: **Pulse Oximeter** and **Device Information Service.** Support of Bond Management Service, Current Time Service and Battery Service are optional.

Refer to Pulse Oximeter Profile Specification for detailed information about the Pulse Oximeter Profile.

Running Speed and Cadence

This Profile enables a Collector device to connect and interact with a Running Speed and Cadence Sensor for use in sports and fitness applications.

- **Running Speed and Cadence Sensor** Profile role – Specified as a GATT Server. Requires the following Service: **Running Speed and Cadence Service**. Optionally may include **Device Information Service**.

- **Collector** Profile role – Specified as a GATT Client. Requires support of the following Services: **Running Speed and Cadence Service**. Support of **Device Information Service** is optional.

Refer to Running Speed and Cadence Profile Specification for detailed information about the Running Speed and Cadence Profile.

Scan Parameters

This Profile defines how a Scan Client device with BLE wireless communications can write its scanning behavior to a Scan Server, and how a Scan Server can request updates of the Scan Client scanning behavior.

- **Scan Server** Profile role – Specified as a GATT Server. Requires the following Service: **Scan Parameters Service**.
- **Scan Client** Profile role – Specified as a GATT Client. Required support of the following Service: **Scan Parameters Service**.

Refer to Scan Parameters Profile Specification for detailed information about the Scan Parameters Profile.

Time

This Profile enables the device to get the date, time, time zone, and DST information and control the functions related to time.

- **Time Server** Profile role – Specified as a GATT Server. Requires the following Service: **Current Time Service**. Optionally may include **Next DST Change Service** and **Reference Time Update Service**.
- **Time Client** Profile role – Specified as a GATT Client. Requires support of the following Service: **Current Time Service**. Support of **Next DST Change Service** and **Reference Time Update Service** is optional.

Refer to Time Profile Specification for detailed information about the Time Profile.

Weight Scale

This Profile is used to enable a data collection device to obtain data from a Weight Scale that exposes the Weight Scale Service.

- **Weight Scale** Profile role – Specified as a GATT Server.

  Requires the following Services: **Weight Scale Service** and **Device Information Service**.

  Optionally may include: **User Data Service**, **Body Composition Service**, **Battery Service** and **Current Time Service**.
- **Collector** Profile role – Specified as a GATT Client.

  Required support of the following Service: **Weight Scale Service** and **Device Information Service**.

  Support of **User Data Service**, **Body Composition Service**, **Battery Service** and **Current Time Service** is optional.

Refer to Weight Scale Profile Specification for detailed information about the Weight Scale Profile.

Wireless Power Transfer

The Wireless Power Transfer Profile (A4WP) enables the communication between Power Receiver Unit and Power Transmitter Unit in Wireless Power Transfer systems.

- **Power Receiver Unit** Profile role – Specified as a GATT Server. Requires the following Service: **Wireless Power Transfer**.

- **Power Transmitter Unit** Profile role – Specified as a GATT Client. Requires support of the following Service: **Wireless Power Transfer**.

Wireless Power Transfer Profile is a custom service defined by the Alliance for Wireless Power (A4WP). Refer to the AirFuel Alliance web site for detailed information about the Wireless Power Transfer Profile.

Bootloader Profile

Note Available for **Bluetooth MCU** only.

The Bluetooth Configurator supports the Bootloader Profile and Bootloader Service, which allow a Bootloader to update the existing firmware on the Cypress BLE device. The Bootloader Service uses the Bluetooth Low Energy interface as a communication interface. It can be added to any of the profiles if the design requires updating the firmware Over-the-Air (OTA).

Refer to the "Bootloader Service Configuration" section for detailed information about the Bootloader Service.

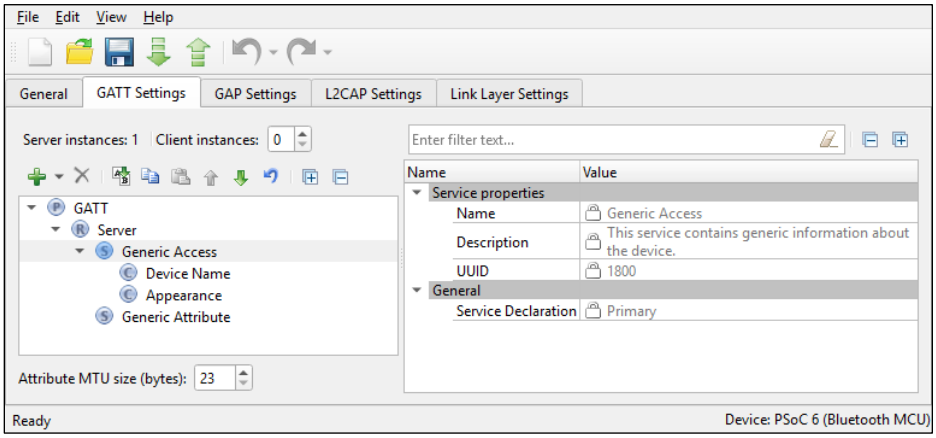
OTA Upgrade and OTA Secure Upgrade Services

Note Available for **20xxx** devices only.

The Bluetooth resource provides the firmware upgrade feature. It supports secure and non-secure versions of the upgrade. The OTA Upgrade Service supports the non-secure upgrade, and the OTA Secure Upgrade Service supports the secure upgrade. They can be added to any of the profiles if the design requires updating the firmware OTA.

Refer to the 20xxx Firmware Upgrade Library documentation for detailed information about the OTA Upgrade and OTA Secure Upgrade Services.

Generic Access Service



This Service is used to define the basic Bluetooth connection and discovery parameters. Click on the Characteristic under the **Generic Access Service** to view those particular Characteristic settings.

Not applicable to 20xxx devices, you perform the actual Characteristics configuration in the **General** options located in the **GAP Settings** tab.

- **Device Name** – This is the name of your device. It has a Read (without authentication/authorization) property associated with it by default. This parameter can be up to 248 bytes. For **Bluetooth MCU** and **43xxx**, the value comes from the **Device Name** field on the GAP Settings tab, under General.
- **Appearance** – The device's logo or appearance, which is a SIG defined 2-byte value. It has a Read (without authentication/authorization) property associated with it by default. For **Bluetooth MCU** and **43xxx**, the value comes from the **Appearance** field on the GAP Settings tab, under General.

- **Peripheral Preferred Connection** – A device in the peripheral role can convey its preferred connection parameter to the peer device. This parameter is 8 bytes in total and is composed of the following sub-parameters.

  Note This parameter will only be available when the device supports a Peripheral role. Refer to the GAP Settings Tab – connection parameters section for more information.

  - **Minimum Connection Interval** – A 2-byte parameter that denotes the minimum permissible connection time.
  - **Maximum Connection Interval** – A 2-byte parameter that denotes the maximum permissible connection time.
  - **Slave Latency** – A 2-byte value and defines the latency between consecutive connection events.
  - **Connection Supervision Timeout Multiplier** – A 2-byte value that denotes the LE link supervision timeout interval. It defines the timeout duration for which an LE link needs to be sustained in case of no response from the peer device over the LE link.

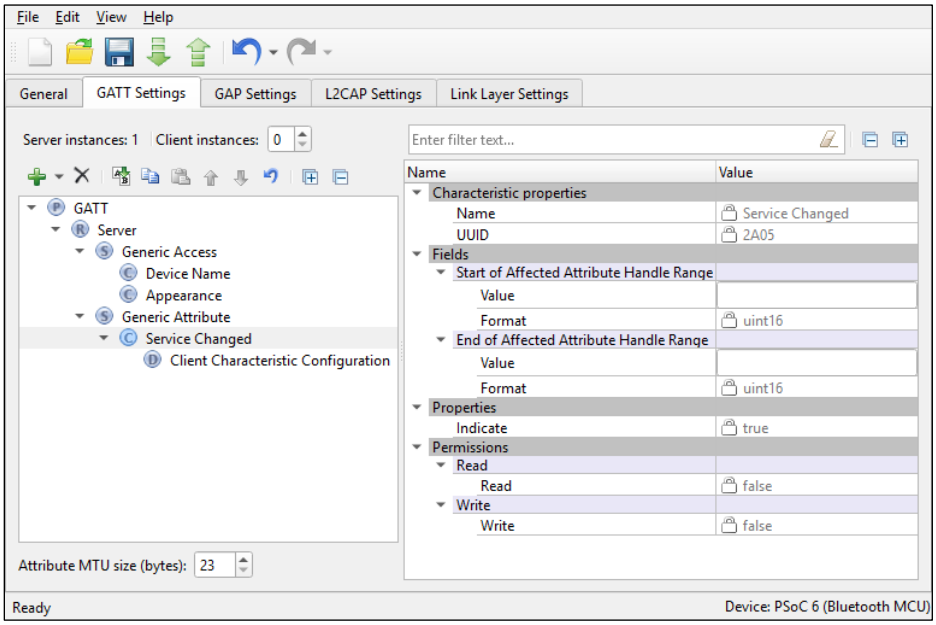
    Note For proper operation, the Connection Supervision Timeout must be larger than **(1 + Slave latency) * Connection Interval * 2** (ms). Refer to Bluetooth Core Specification Volume 6, Part B, Chapter 4.5.2 for more information on Connection Supervision Timeout.

  Note The above parameters are used for the connection parameters update procedure over L2CAP if a GAP central device does not use the peripheral preferred connection parameters. For example, iOS7 ignores peripheral preferred connection parameter Characteristics and establishes a connection with a default 30 ms connection interval. The peripheral device requests a connection parameter update by sending an L2CAP connection parameter update request at an appropriate time.

  Typical peripheral implementation initiates the L2CAP connection parameter update procedure once any Characteristic is configured for periodic notification or indication.

- **Central address resolution** – A device in the central role can convey whether it supports privacy with address resolution. The Peripheral shall check if the peer device supports address resolution by reading the Central Address Resolution characteristic before using directed advertisement where the initiator address is set to a Resolvable Private Address (RPA).
- **Resolvable Private Address Only** – Defines whether the device will only use Resolvable Private Addresses (RPAs) as local addresses.

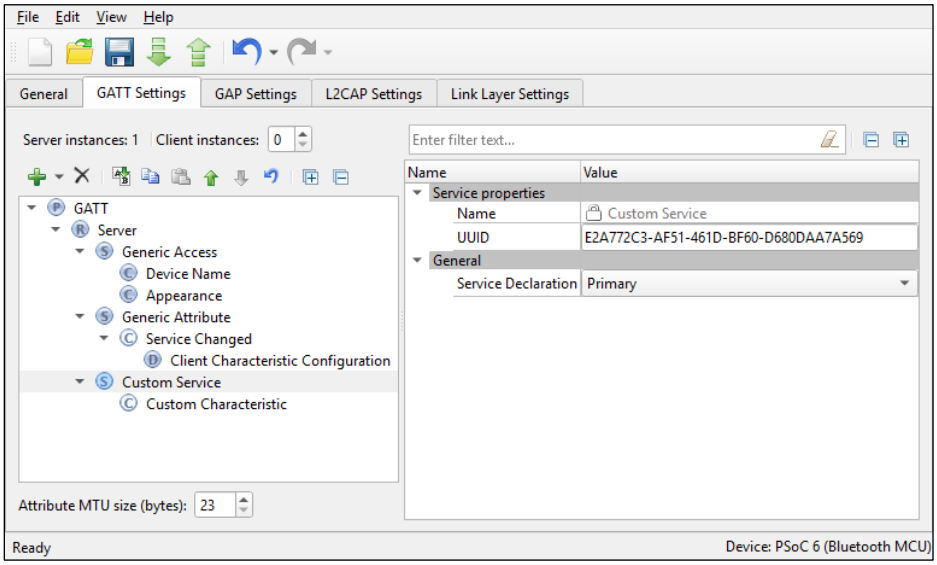
Generic Attribute Service



By default, this Service does not have any Characteristics. Add the optional Service Changed Characteristic under the Generic Attribute Service if needed.

- **Service Changed** – This Characteristic is used to indicate to the connected devices that a Service has changed (i.e., added, removed, or modified). It is used to indicate to GATT Clients that have a trusted relationship (i.e., bond) with the GATT Server when GATT based Services have changed when they re-connect to the GATT Server. It is mandatory for the device in the GATT Client role. For the device in the GATT Server role, the Characteristic is mandatory if the GATT Server changes the supported Services in the device.

Custom Service Configuration



UUID

A universally unique identifier of the service. This field is editable for Custom Services. By default, it is initialized with a random 128-bit UUID.

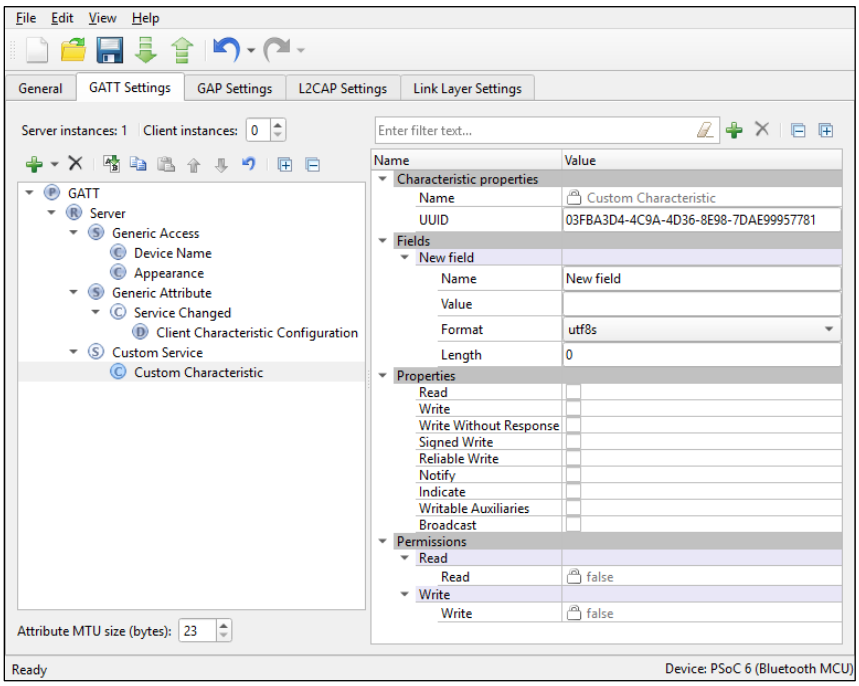
Service type

- **Primary** – Represents the primary functionality of the device.
- **Secondary** – Represents additional functionality of the device. The secondary service must be included in another service.

Included services

- The list of the Services that can be included in the selected Service. Each Service may have one or more included Services. The included Services provide additional functionality for the Service.

Custom Characteristic Configuration



UUID

A universally unique identifier of the Characteristic. This field is editable for Custom Characteristics. By default, it is initialized with a random 128-bit UUID.

Fields

Fields represent a Characteristic value. The default value for each field can be set in the **Value** property. The fields are customizable for the Custom Characteristic. You can add or delete fields using the tool buttons located above the Properties editor.

Properties

The Characteristic properties define how the Characteristic value can be used. Some properties (Broadcast, Notify, Indicate, Reliable Write, Writable Auxiliaries) require the presence of a corresponding Characteristic

Descriptor. For details, see Bluetooth Core Specification Vol.3, part G (GATT), section 3.3.1.1 “Characteristic Properties”.

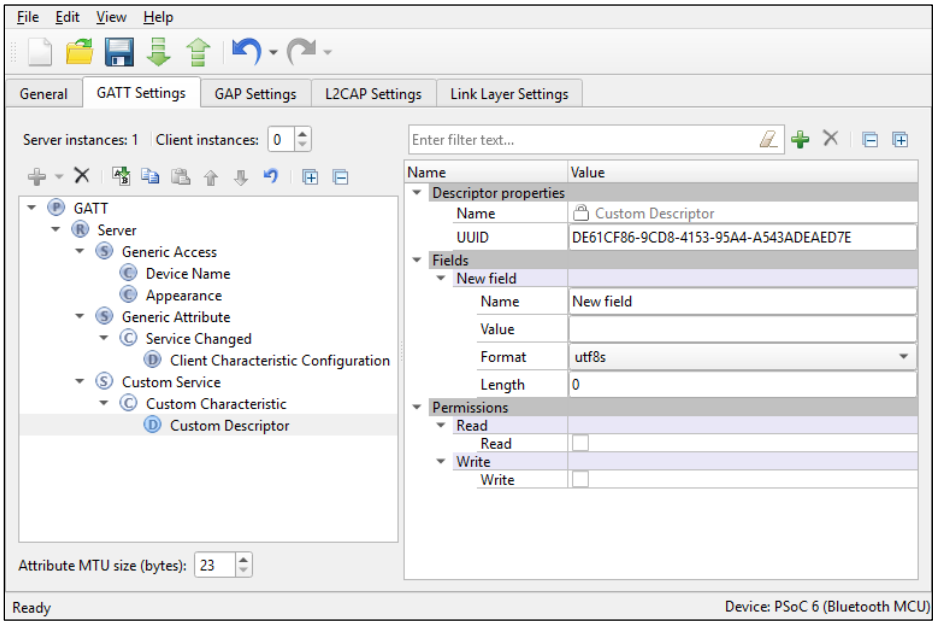
Permissions

Characteristic permissions define how the Characteristic Value attribute can be accessed and the security level required for this access. Access permissions are set based on the Characteristic properties.

Permissions parameters differ depending on the selected device.

For **Bluetooth MCU**, the **Auto sync** property is available which determines if the Security permissions are automatically updated when the **Security Mode** or **Security Level** parameters are changed in Security Configuration 0 on the **GAP Settings** tab. Additional Security configurations do not affect attribute permissions.

Custom Descriptor Configuration



UUID

A universally unique identifier of the Descriptor. This field is editable for Custom Descriptors. By default, it is initialized with a random 128-bit UUID.

Fields

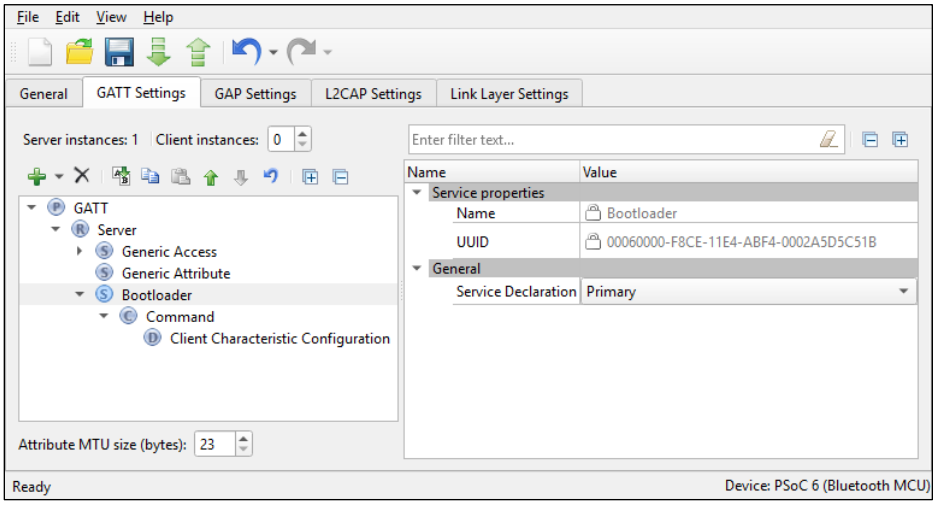
Fields represent a Descriptor value. The default value for each field can be set in the **Value** property. In case of the Custom Descriptor, the fields are customizable. You can add or delete fields using the tool buttons located above the Properties editor.

Permissions

Descriptor permissions define how the Descriptor attribute can be accessed and the security level required for this access. Permissions parameters differ depending on the selected device.

Bootloader Service Configuration

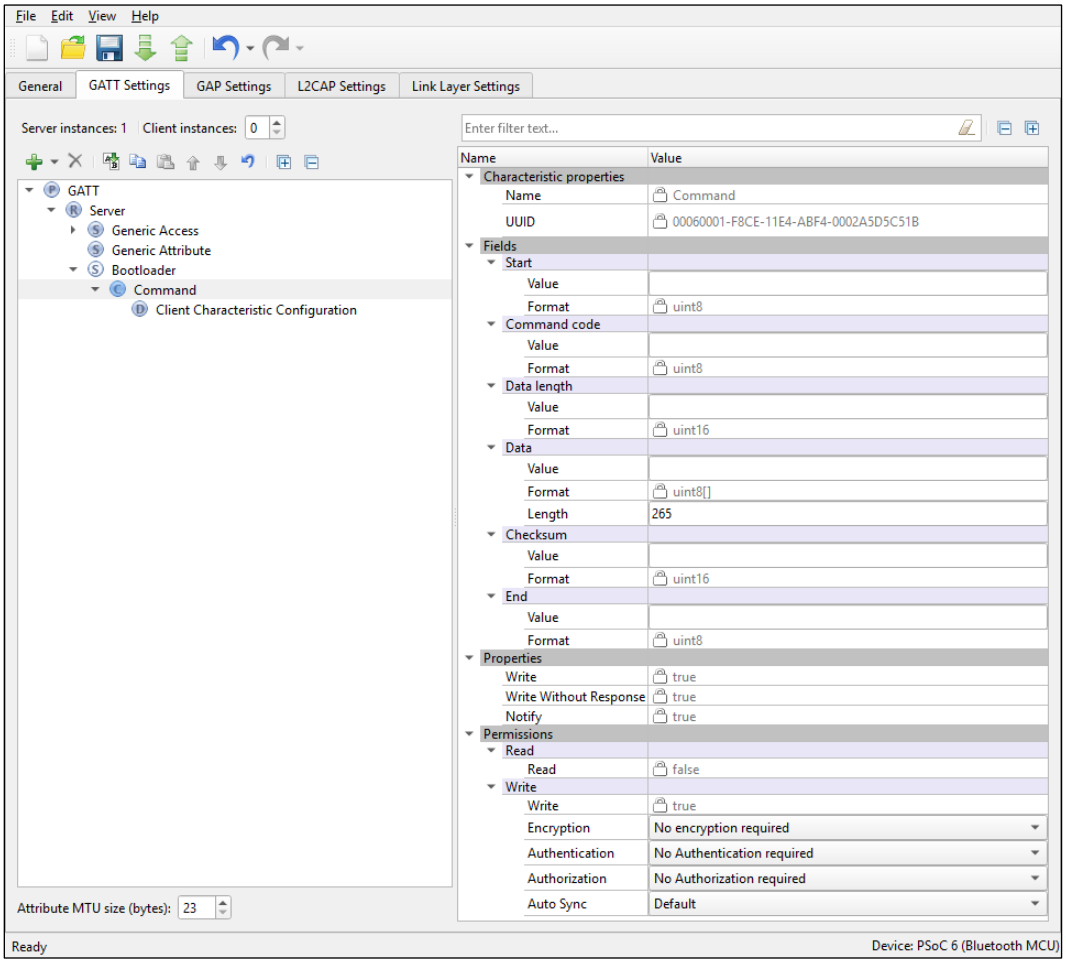
Note Available for **Bluetooth MCU** only.



UUID

A universally unique identifier of the service. The UUID is set to 00060000-F8CE-11E4-ABF4-0002A5D5C51B.

Command Characteristic Configuration



UUID

A universally unique identifier of the Characteristic. The UUID is set to 00060001-F8CE-11E4-ABF4-0002A5D5C51B.

Fields

The fields represent Command Characteristic values that define:

- Start of packet – The start of a bootloader packet.
- Command – A bootloader command.
- Status Code – Command status code.
- Data Length – The length of a bootloader command/response to be set to the maximum command data length that can be used in the design.

  Per BLE protocol, if the command requires a response larger than 20 bytes, the attribute MTU size should be increased. To support responses with a data length set to 56 (response for **Get Metadata** command), the attribute MTU size is set to 66 according to the equation:

  MTU size = Data Length + Bootloader command overhead + notification parameters overhead

where:

- *Data Length* = the response data length
- *Bootloader command overhead* = 7
- *Notification parameters overhead* = 3

Not following the requirement will result in the Bluetooth resource failing to send a response to the requested command.

- Data – Bootloader command data. The length of this field is specified by the Data Length field.
- Checksum – A checksum computed for the entire packet except for the Checksum and End of Packet fields.
- End of Packet – The end of a bootloader packet.

GAP Settings Tab (Bluetooth MCU and 43xxx)

The GAP parameters define the general connection settings required when connecting Bluetooth devices. The tab contains various sections of parameters based on the item you select in the tree.

The **GAP Settings** tab displays the settings possible based on the GAP role selected in the **General** tab. This tab allows the default settings of the active tree item to be restored by using the **Restore Defaults** button.

Note Many parameter settings also contain an **Enable** <parameter> check box above them to enable or disable the parameter, in addition to setting that parameter's value.

The following sections show the different categories of parameters based on what item you select in the tree.

Toolbar

The toolbar contains a means to add or delete GAP role configurations and Security configurations.

- **Add** – Allows adding for **Bluetooth MCU**: Peripheral, Central, Broadcaster, Observer or Security configurations.

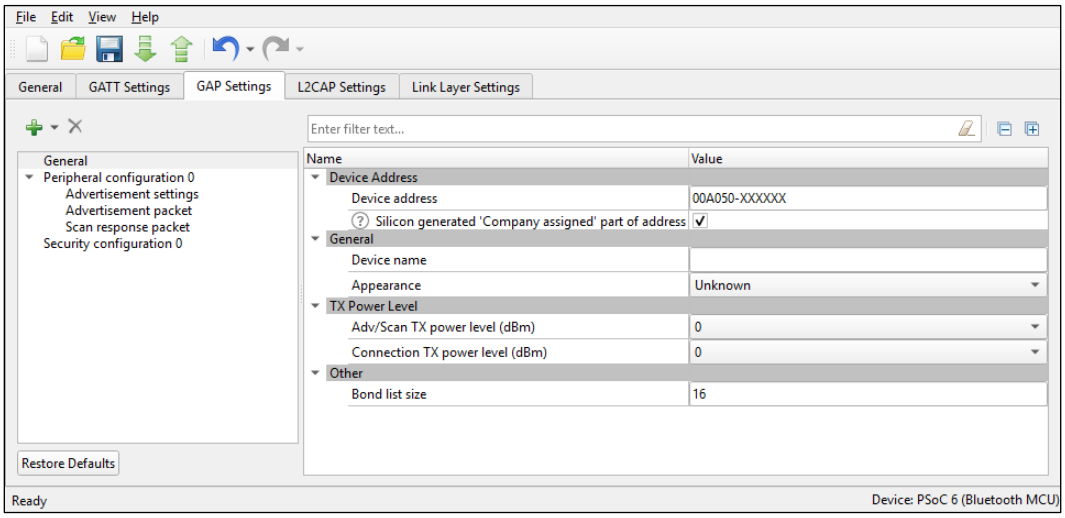
  Note (Bluetooth MCU) The available options depend on the GAP role selected in the **General** tab. You can add several configurations for one GAP role and switch between them using the firmware.

- **Delete** (**Bluetooth MCU**) – Deletes the selected Configuration.

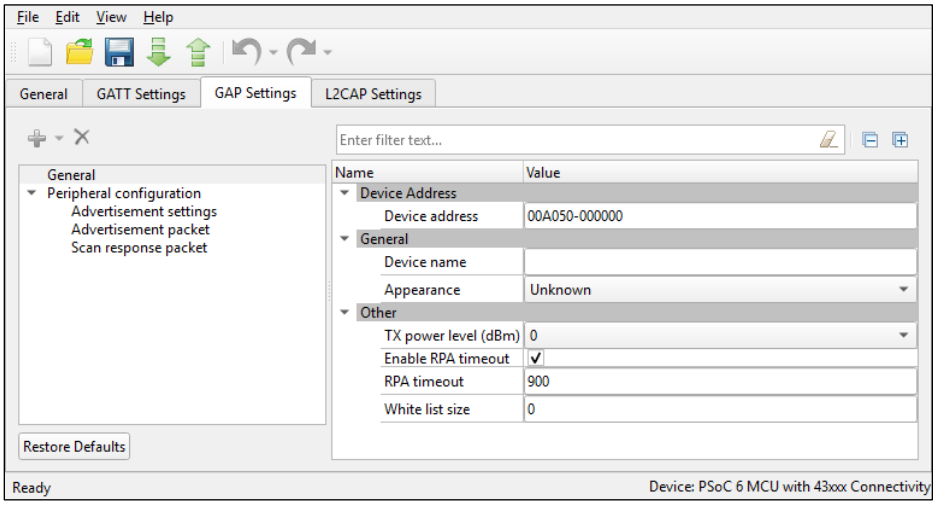
GAP Settings Tab – General

This section contains general GAP parameters. **Bluetooth MCU** and **43xxx** have different sets of parameters.

Bluetooth MCU



43xxx



Device address (Bluetooth MCU and 43xxx)

This is a unique 48-bit Bluetooth public address used to identify the device. It is divided into two parts:

- **Company ID** – The 24 most significant bits: a 24-bit Organization Unique Identifier (OUI) address assigned by IEEE.
- **Company assigned** – The 24 least-significant bits.

The address configured here is static and designed for development purposes only.

Note (Bluetooth MCU)

During production, the device address is programmed into the user area of the supervisory flash (Sflash) location for the device address (Row 0) via the SWD interface. Normally, this address must be programmed only once

during mass production, and then never changed in-field. However, application flash can be reprogrammed in-field many times.

During prototyping (FW design), the device address can be programmed into the user area of the Sflash location using the Cypress Programmer OpenOCD CLI. Refer to *Cypress Programmer 2.0 OpenOCD CLI User Guide* for more details.

The following command is an example of using the CLI to update the device address structure (type of cy_stc_ble_gap_bd_addr_t) with the data: 11 00 00 50 A0 00 00 (first 6 bytes are address with LSB first and last byte is the address type) in the Row 0 (0x1600 0800) of user area of the Sflash.

```
openocd.exe -s ../scripts -f interface/kitprog3.cfg -f target/psoc6.cfg -c "
init; reset init; psoc6 allow_unsafe_sflash on; flash rmw 0x16000800
11000050A00000; reset; exit"
```

```
Warn : Writes to 'unsafe' Supervisory Flash rows is now ALLOWED
[100%] [##################################] [ Programming ]
[100%] [##################################] [ Programming ]
modified 7 byte(s) in 512 byte region at 0x16000800 in 0.088930s (5.622 KiB/s)
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hard
ware srst support.
```

Row 1, Row 2, and Row 3 of the user area of the Sflash are not used by BLE and available for the user information storage.

Silicon generated 'Company assigned' part of address (Bluetooth MCU)

When checked, the "'Company assigned' part of address" is generated using the factory programmed die X/Y location, wafer ID and lot ID of the silicon.

Note The "Silicon generated 'Company assigned' part of address" option does not guarantee a unique device address. For mass production, Cypress strongly suggests that the device address be programmed into the user area (Row 0) of the Sflash location via the SWD interface.

Device Name (Bluetooth MCU and 43xxx)

The device name to be displayed on the peer side. It has a Read (without authentication/authorization) property associated with it by default. This parameter can be up to 248 bytes.

Note This parameter configures the **GAP Service Device Name** Characteristic located in the **Profile Tree**. It is available for modification only when the device is a GATT Server.

Appearance (Bluetooth MCU and 43xxx)

The device's logo or appearance is a SIG-defined 2-byte value. It has a Read (without authentication/authorization) property associated with it by default.

Note This parameter configures the **GAP Service Appearance** Characteristic located in the **Profile Tree**, available for modification only when the device is a GATT Server.

Adv/Scan TX power level (Bluetooth MCU)

The initial transmitter power level (dBm) of the advertisement or scan channels upon startup. Default: 0 dBm. Possible values: -20 dBm, -16 dBm, -12 dBm, -6 dBm, 0 dBm, 4 dBm.

Connection TX power level (Bluetooth MCU)

The initial transmitter power level (dBm) of the connection channels upon startup. Default: 0 dBm. Possible values: -20 dBm, -16 dBm, -12 dBm, -6 dBm, 0 dBm, 4 dBm.

TX power level (43xxx)

The initial transmitter power level (dBm) upon startup. Default: 0 dBm. Possible values: -16 dBm, -12 dBm, -8 dBm, -4 dBm, 0 dBm, 4 dBm, 8 dBm, 12 dBm.

Bond list size (Bluetooth MCU)

The maximum number of bonded devices supported by this device. The valid range is from 1 to 128. Default: 16.

Note The maximum number of bonded devices is also limited by the size of an available application flash (emulated EEPROM area) size to be consumed for data storage. The consumed application flash size is calculated as the multiple of the number of supported services and the multiple of the number of supported bonded devices.

RPA timeout (43xxx)

The interval of random address refreshing (seconds). The valid range is from 1 to 41400. Default: 900.

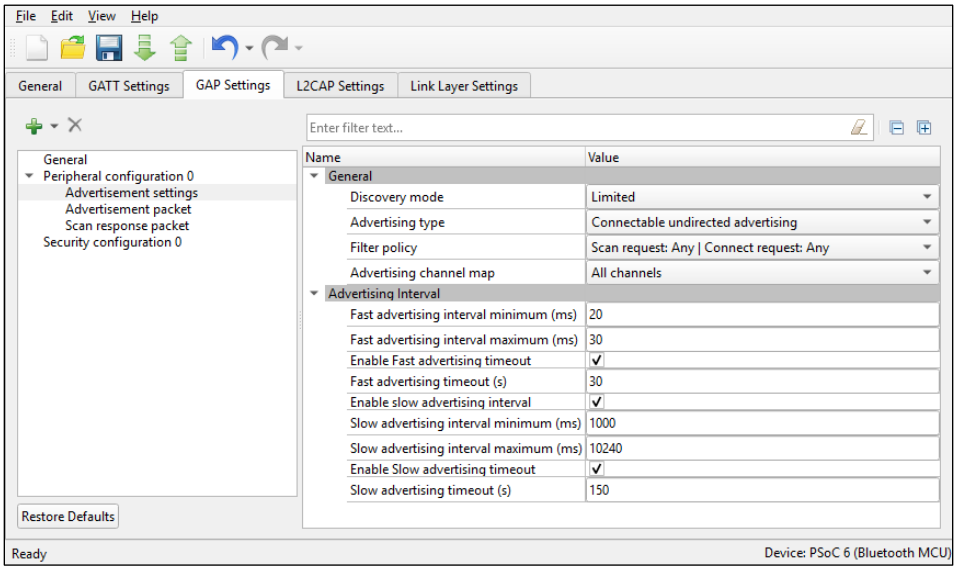
White list size (43xxx)

The maximum number of devices that can be added to the White list. The valid range is from 0 to 128. Default: 0.

GAP Settings Tab – Advertisement Settings

Bluetooth MCU

These parameters are available when the device is configured to contain a Peripheral or Broadcaster GAP role.



- **Discovery mode**
  - **Non-discoverable** – The device cannot be discovered by a Central device.
  - **Limited** – Enables a device to be discoverable only for a limited period of time, during temporary conditions, or for a specific event. The device advertising in Limited Discoverable mode can connect to the Central device that is performing the Limited Discovery procedure. The timeout duration is defined by the applicable advertising timeout parameter.

- **General** – The device is used by devices to be discoverable continuously or for no specific condition. The device advertising in General Discoverable mode can connect to the Central device that is performing the General Discovery procedure.

- **Advertising type**

  This parameter defines the advertising type to be used by the LL for appropriate **Discovery mode**.

  - **Connectable undirected advertising** – Advertises advertising and scan response data. It allows any other device to connect to this device.
  - **Scannable undirected advertising** – Broadcasts advertising data and scan response data to active scanners.
  - Non-connectable undirected advertising – Broadcasts advertising data.

- **Filter policy**

  This parameter defines how the scan and connection requests are filtered.

  - **Scan request** – **Any | Connect request: Any** – Processes scan and connect requests from all devices.
  - **Scan request** – **White list | Connect request: Any** – Processes scan requests only from the devices in the White list and connect requests from all devices.
  - **Scan request** – **Any | Connect request: White list** – Processes scan requests from all devices and connect requests only from the devices in the White List.
  - **Scan request** – **White list | Connect request: White list** – Processes scan and connect requests only from the devices in the White list.

- **Advertising channel map**

  This parameter is used to enable a specific advertisement channel.

  - **Channel 37** – Enables advertisement channel #37.
  - **Channel 38** – Enables advertisement channel #38.
  - **Channel 39** – Enables advertisement channel #39.
  - **Channels 37 and 38** – Enables advertisement channels #37 and #38.
  - **Channel 37 and 39** – Enables advertisement channels #37 and #39.
  - **Channels 38 and 39** – Enables advertisement channels #38 and #39.
  - **All channels** – Enables all three advertisement channels.

- **Advertising interval**

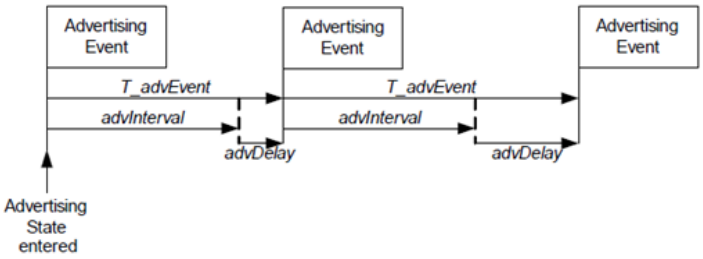
  This parameter defines the interval between two advertising events. Set the permissible minimum and maximum values of two Advertisement interval types: **Fast advertising interval** and **Slow advertising interval**. Typically, after the device initialization, a peripheral device uses the Fast advertising interval. After the **Fast advertising interval timeout** value expires, and if a connection with a Central device is not established, then the Profile switches to Slow advertising interval to save the battery life. After the **Slow advertising interval timeout** value expires, CY_BLE_EVT_GAPP_ADVERTISEMENT_START_STOP event is generated.

Note The Advertising interval needs to be aligned with the selected Profile specification.

- **Fast advertising interval** – This advertisement interval results in faster LE Connection. The Bluetooth resource uses this interval value when the connection time is between the specified minimum and maximum values of the interval.
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 20 ms to 10240 ms.

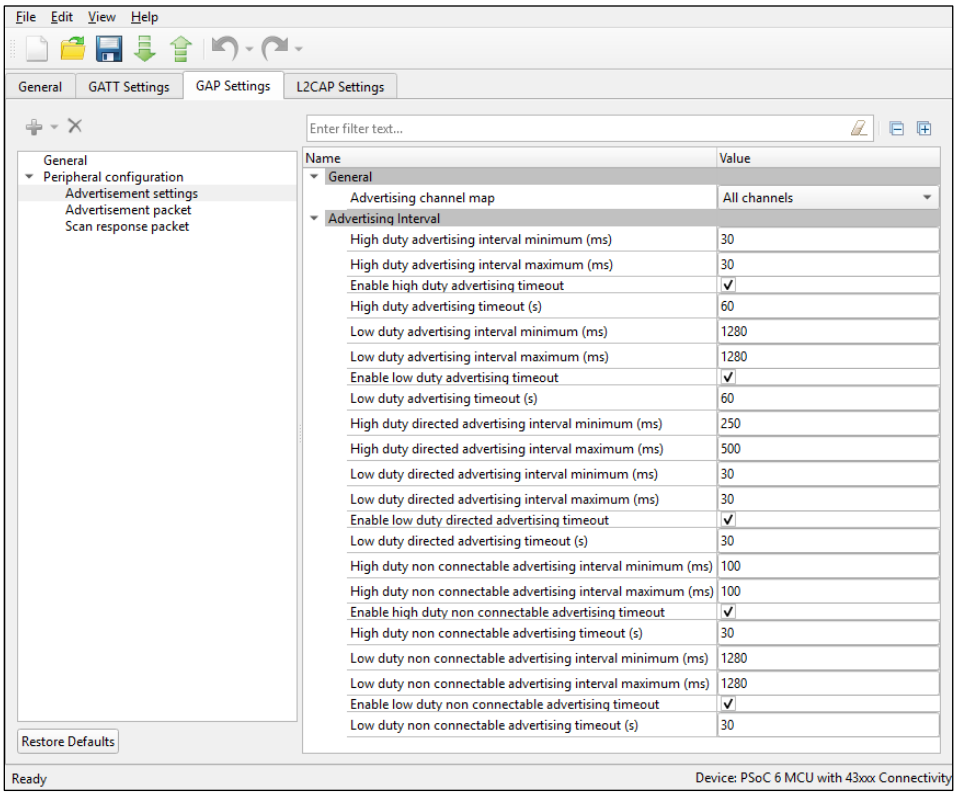
- Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 20 ms to 10240 ms.
- Timeout – The timeout value of advertising with fast advertising interval parameters. When equals 0, the device is advertising continuously and slow advertising settings become unavailable. A timeout cannot occur before the advertising interval expires. So, if a timeout value is smaller than the fast advertising interval minimum value, a warning displays.

- **Slow advertising interval** – Defines the advertising interval for slow advertising. This is an optional parameter to implement advertising with a lower duty cycle to save battery life. The Slow advertising interval parameters are applied to the device after an internal fast advertising interval timeout occurs. The minimum and maximum values defined using this parameter allow the BLE Stack to expect advertising to happen within these intervals. This parameter is not applicable in **General discovery** mode.
  - Minimum – The minimum interval for advertising the data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 1000 ms to 10240 ms.
  - Maximum – The maximum interval for advertising the data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 1000 ms to 10240 ms.
  - Timeout – A timeout value of advertising with slow advertising interval parameters. When equals 0, the device is advertising continuously. A timeout cannot occur before the advertising interval expires, that is why if the timeout value is smaller than the slow advertising interval minimum value, a warning is displayed.



  - AdvDelay – A pseudo-random delay of 0-10 ms.
  - Complete Advertising Event – Consists of one advertising PDU sent into each of the used advertising channels.

43xxx



- **Advertising channel map**

This parameter is used to enable a specific advertisement channel:

  - **Channel 37** – Enables advertisement channel #37.
  - **Channel 38** – Enables advertisement channel #38.
  - **Channel 39** – Enables advertisement channel #39.
  - **Channels 37 and 38** – Enables advertisement channels #37 and #38.
  - **Channel 37 and 39** – Enables advertisement channels #37 and #39.
  - **Channels 38 and 39** – Enables advertisement channels #38 and #39.
  - **All channels** – Enables all three advertisement channels.

- **Advertising interval**

This parameter defines the interval between two advertising events.

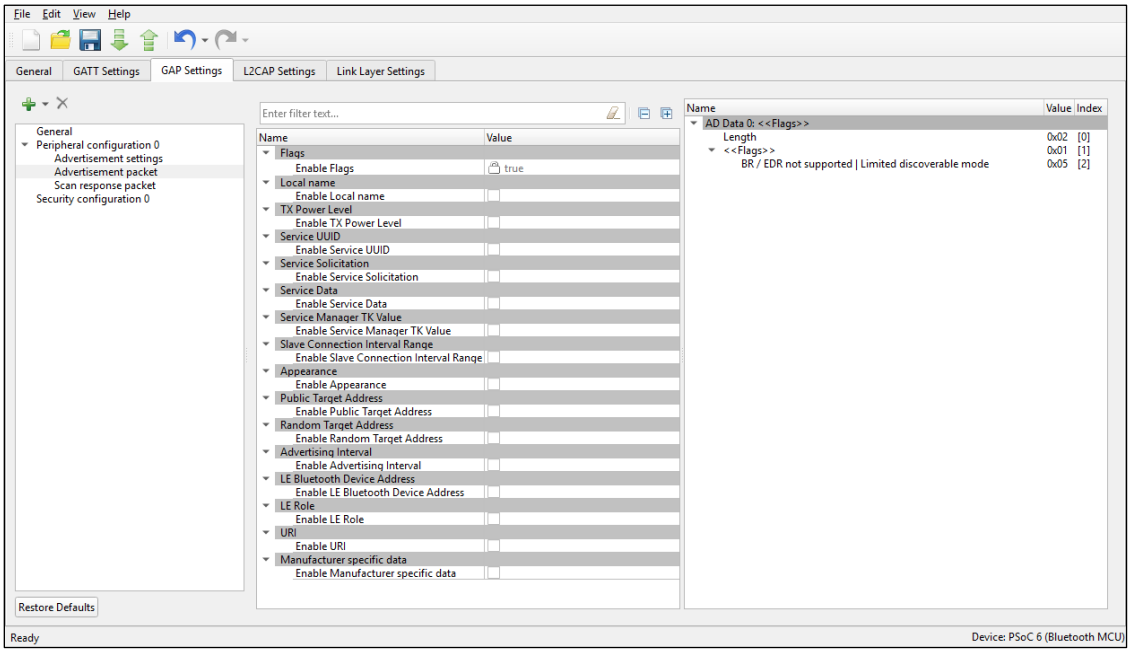
  - High-duty advertising interval
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Duration – The duration of advertising with high-duty advertising interval parameters. When not enabled, the device is advertising continuously.

- **Low duty advertising interval**
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Duration – The duration of advertising with low-duty advertising interval parameters. When not enabled, the device is advertising continuously.
- **High duty directed advertising interval**
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
- **Low duty directed advertising interval**
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Duration – The duration of advertising with low-duty directed advertising interval parameters. When not enabled, the device is advertising continuously.
- **High duty non connectable advertising interval**
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Duration – The duration of advertising with high-duty non-connectable advertising interval parameters. When not enabled, the device is advertising continuously.
- **Low duty non connectable advertising interval**
  - Minimum – The minimum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Maximum – The maximum interval for advertising data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms.
  - Duration – The duration of advertising with low-duty non-connectable advertising interval parameters. When not enabled, the device is advertising continuously.

GAP Settings Tab – Advertisement packet

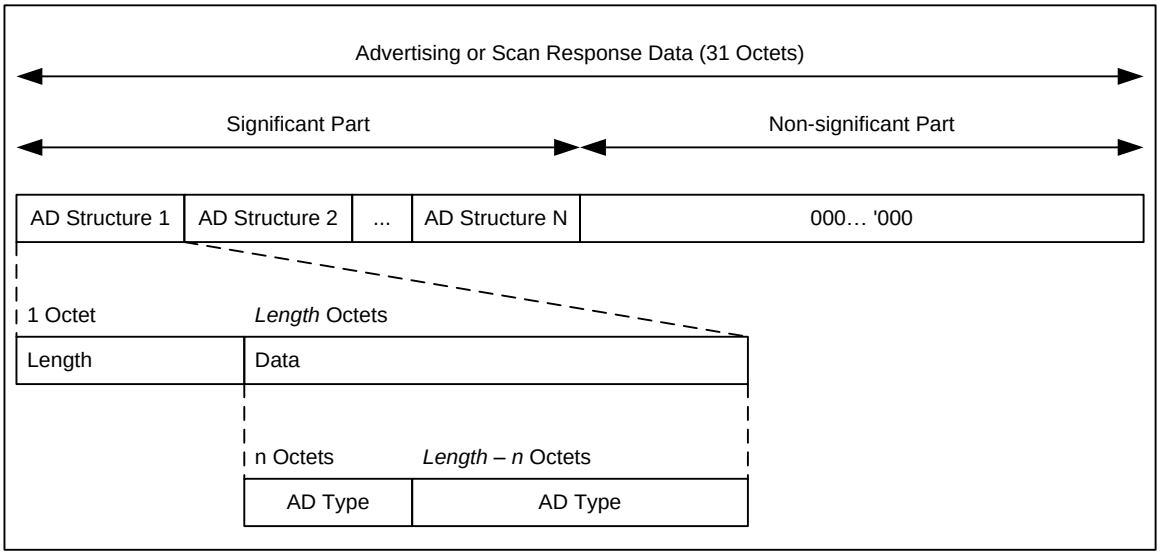
This section displays when the device is configured to contain a Peripheral or Broadcaster GAP role. It is used to configure **Advertisement data** to be used in device advertisements.

Bluetooth MCU and 43xxx



- **Advertisement / Scan response data settings**

An **Advertisement (AD)** or **Scan response data** packet is a 31-byte payload used to declare the device's BLE capability and its connection parameters. The structure of this data is shown below as specified in the Bluetooth specification.



The data packet can contain a number of AD structures. Each of these structures is composed of the following parameters.

- **AD Length** – The size of the **AD Type** and **AD Data** in bytes.

- **AD Type** – The type of an advertisement within the AD structure.
- **AD Data** – Data associated with the **AD Type**.

The total length of a complete Advertising packet cannot exceed 31 bytes.

An example structure for **Advertisement data** or **Scan response data** is as follows:

- AD Structure Element Definition:
  - **AD Length** – Size of **AD Type** and associated **AD Data** = 5 bytes
  - **AD Type** (1 byte) – 0x03 (Service UUID)
  - **AD Data** (4 bytes) – 0x180D, 0x180A (Heart Rate Service, Device Information Service)

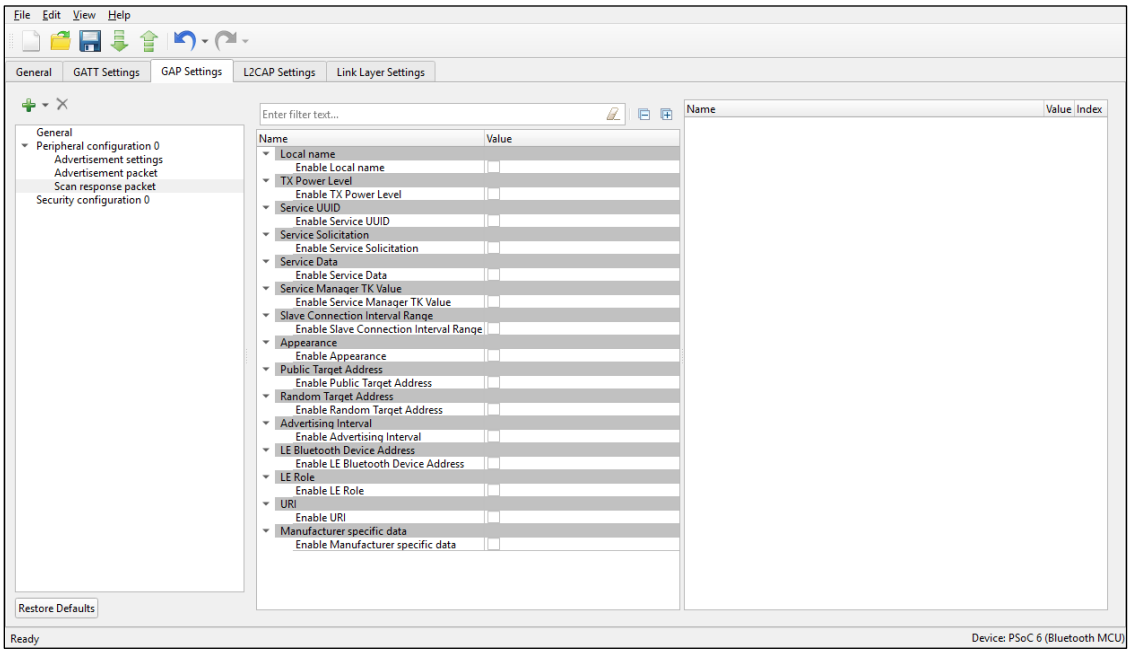
The following table shows the **AD Types**:

AD Type	Description
Flags	Flags to broadcast the underlying BLE transport capability such as Discoverable mode, LE only, etc.
Local Name	Device Name (complete of shortened). The device name value comes from the **Device name** field on the **GAP Settings** tab, under **General**.
TX Power Level	Transmit Power Level. Taken from the **Adv/Scan TX power level** (**Bluetooth MCU**) or **TX power level** (**43xxx**) field on the **GAP Settings** tab, under **General.**
Slave Connection Interval Range	The preferred connection interval range for the device. Not available in the **Broadcaster** GAP role.
Service UUID	The list of Service UUIDs to be broadcasted that the device has implemented. There are different AD Type values to advertise 16-bit, 32-bit, and 128-bit Service UUIDs. 16-bit and 32-bit Service UUIDs are used if they are assigned by the Bluetooth SIG.
Service Solicitation	The list of Service UUIDs from the central device to be used by the peripheral device. There are different AD Type values to advertise 16-bit, 32-bit, and 128-bit Service UUIDs.
Service Data	2/4/16-byte Service UUID, followed by additional Service data.
Security Manager TK value	A temporal key to use at the time of pairing. Not available in the **Broadcaster** GAP role.
Appearance	The external appearance of the device. The value comes from the **Appearance** field on the **GAP Settings** tab, under **General**.
Public Target Address	The public device address of intended recipients.
Random Target Address	The random device address of intended recipients.
Advertising Interval	For the Bluetooth MCU, calculated as an average of Fast advertising interval minimum and maximum values configured on the **GAP Settings** tab, under **Advertisement Settings**.
LE Bluetooth Device Address	The device address of the local device. The value comes from the **Public device address** field on the **GAP Settings** tab, under **General**.
LE Role	Supported LE roles. Not available in the **Broadcaster** GAP role.
URI	URI, as defined in the IETF STD 66.
Manufacturer Specific Data	2 bytes company identifier followed by manufacturer specific data.
Indoor Positioning	Data specified in the Indoor Positioning Service Specification. Available when the Indoor Positioning Service is present in the Profile.

GAP Settings Tab – Scan Response Packet

This section displays when the device is configured to contain a Peripheral or Broadcaster GAP role. It is used to configure a Scan response data packet to be used in response to device scanning performed by a GATT Client device.

Bluetooth MCU and 43xxx

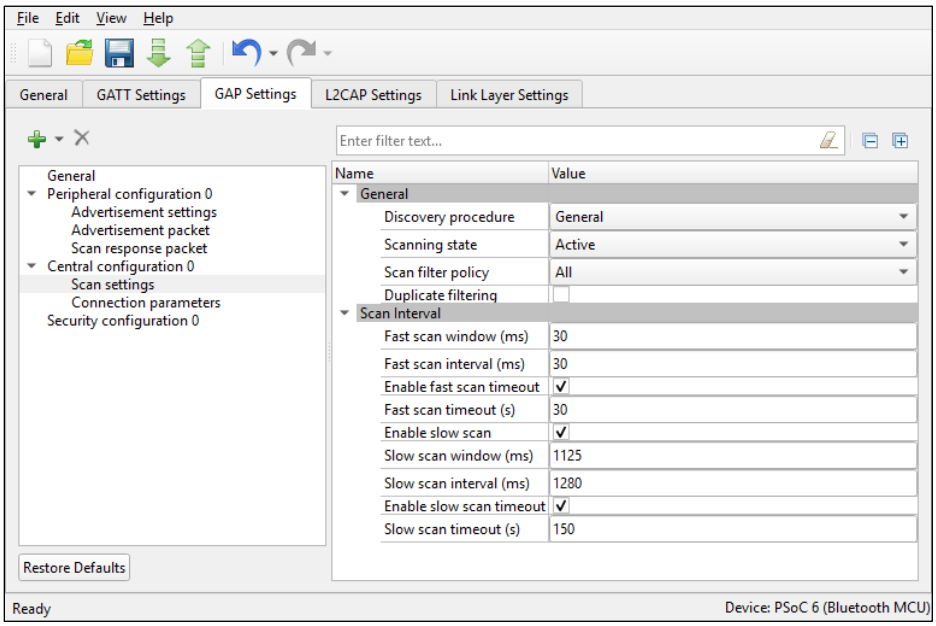


The packet structure of a Scan response packet is the same as an Advertisement packet. See Advertisement / Scan response data settings for information on configuring the Scan response packet.

GAP Settings Tab – Scan Settings

Bluetooth MCU

These parameters are available when the device is configured to contain the Central or Observer GAP role. Typically, during a device discovery, the GATT Client device initiates the scan procedure. It uses the **Fast scan parameters** for a period of time, approximately 30 to 60 seconds, and then it reduces the scan frequency using the **Slow scan parameters**.



Note The scan interval needs to be aligned with the user-selected Profile specification.

- **Discovery procedure**
  - **Limited** – A device that performs this procedure will discover a device that does limited Discovery mode advertising only.
  - **General** – A device that performs this procedure will discover a device that does general and limited Discovery advertising.
- **Scanning state**
  - **Passive** – A device can only listen to advertisement packets.
  - **Active** – A device may ask the advertiser for additional information.
- **Filter policy**

This parameter defines how the advertisement packets are filtered.

  - **All** – All advertisement packets are processed.
  - **White list only** – Only advertisement packets from the devices in the White list are processed.

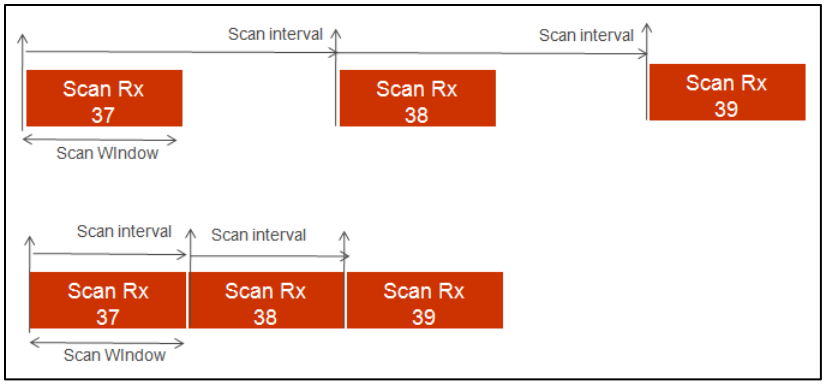
- **Duplicate filtering**

When enabled, this activates filtering of duplicated advertisement data. If disabled, the BLE stack will not perform filtering of advertisement data.

Note The controller firmware has eight address locations reserved to cache the previously seen advertiser devices and filter duplicate packets from them. If there are more than eight advertising devices in the proximity of a scanner during the scan period, the address storing buffer is exhausted. The firmware algorithm for overwriting the address cache buffer is implemented in the FIFO fashion. When the scanner sees more than eight advertisers, the ninth advertiser replaces the first one, the tenth advertiser replaces the second one, and so on, in the address cache. After flushing the first advertiser from the address cache, if the scanner sees the first advertiser's ADV packet again, it thinks that it is a new device (as the first advertiser is no longer in the address cache) resulting in sending an ADV packet to the host.

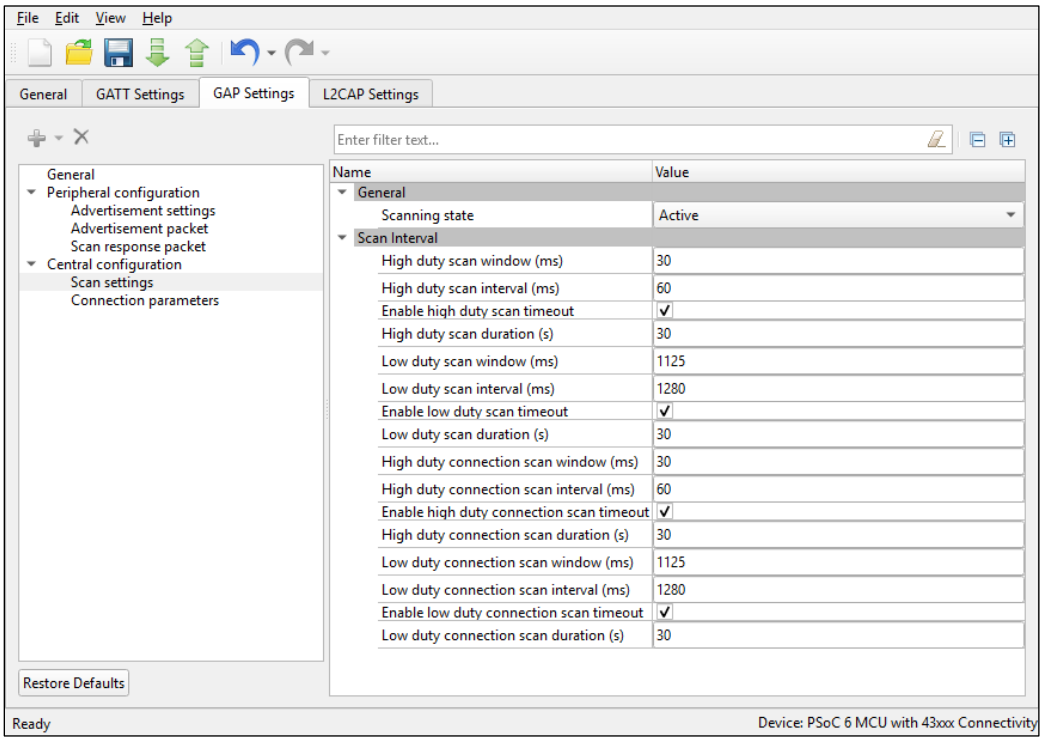
- **Scan parameters**

These parameters define the scanning time and interval between scanning events. Two different sets of Scan parameters are used: **Fast scan parameters** and **Slow scan parameters**. Typically, after the device initialization, a central device uses the Fast scan parameters. After the **Fast scan timeout** value expires, and if a connection with a Peripheral device is not set, then the Profile switches to the Slow scan parameters to save the battery life. After the **Slow scan timeout** value expires, the 'CY_BLE_EVT_GAPC_SCAN_START_STOP ' event is generated. See API documentation.



- **Fast scan parameters** – This connection type results in a faster connection between the GATT Client and Server devices than it is possible using a normal connection.
  - **Scan Window** – This parameter defines the scan window when operating in **Fast connection**. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 2.5 ms to 10240 ms. **Scan Window** must be less than the **Scan Interval**. Default: 30 ms.
  - **Scan Interval** – This parameter defines the scan interval when operating in **Fast connection**. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 2.5 ms to 10240 ms. Default: 30 ms.
  - **Scan Timeout** – The timeout value of scanning with fast scan parameters. Default: 30 s. When not enabled, the device is scanning continuously. A timeout cannot occur before the scanning interval is expired, that is why if a timeout value is smaller than the slow scanning interval minimum value, a warning displays.
- **Slow scan parameters** – This connection results in a slower than possible connection between the GATT Client and GATT Server devices that use a normal connection. However, this method consumes less power.
  - **Scan Window** – This parameter defines the scan window when operating in **Slow Connection**. The parameter is configured to increment in multiples of 0.625ms. The valid range is from 2.5 ms to 10240 ms. **Scan Window** must be less than the **Scan Interval**. Default: 1125 ms.
  - **Scan Interval** – This parameter defines the scan interval when operating in **Slow Connection**. The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 2.5 ms to 10240 ms. Default: 1280 ms.
  - **Scan Timeout** – The timeout value of scanning with slow scan parameters. Default: 150 s. When not enabled, the device is scanning continuously. A timeout cannot occur before the scanning interval expires, so if a timeout value is smaller than the slow scanning interval minimum value, a warning displays.

43xxx

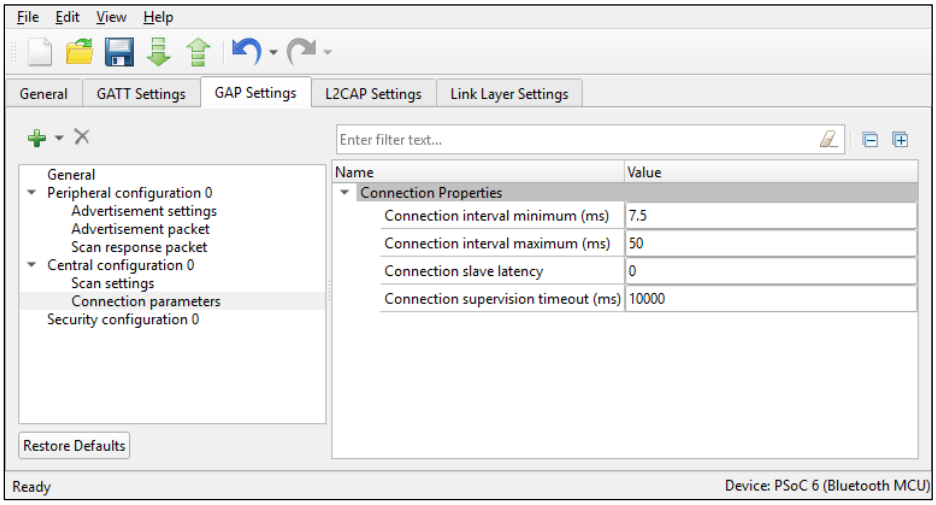


- **Scanning state**
  - **Passive** – A device can only listen to advertisement packets.
  - **Active** – A device may ask the advertiser for additional information.
- **Scan parameters**
  - **High duty scan parameters**
    - **Scan Window** – The parameter is configured to increment in multiples of 0.625 ms. **Scan Window** must be less than **Scan Interval**.
    - **Scan Interval** – The parameter is configured to increment in multiples of 0.625 ms.
    - **Scan Duration** – The timeout value of scanning with high-duty scan parameters. Default: 30 s. When not enabled, the device is scanning continuously.
  - **Low duty scan parameters**
    - **Scan Window** – The parameter is configured to increment in multiples of 0.625 ms. The valid range is from 2.5 ms to 10240 ms. **Scan Window** must be less than **Scan Interval**.
    - **Scan Interval** – The parameter is configured to increment in multiples of 0.625 ms.
    - **Scan Duration** –– The timeout value of scanning with low-duty scan parameters. Default: 30 s. When not enabled, the device is scanning continuously.
- **High duty connection scan parameters**
  - **Scan Window** – The parameter is configured to increment in multiples of 0.625 ms. **Scan Window** must be less than **Scan Interval**.
  - **Scan Interval** – The parameter is configured to increment in multiples of 0.625 ms.
  - **Scan Duration** – The timeout value of scanning with high duty connection scan parameters. Default: 30 s. When not enabled, the device is scanning continuously.

- **Low duty connection scan parameters**
  - **Scan Window** – The parameter is configured to increment in multiples of 0.625 ms. **Scan Window** must be less than **Scan Interval**.
  - **Scan Interval** – The parameter is configured to increment in multiples of 0.625 ms.
  - **Scan Duration** – The timeout value of scanning with low-duty connection scan parameters. Default: 30 s. When not enabled, the device is scanning continuously.

GAP Settings Tab – Connection Parameters

These parameters define the preferred BLE interface connection settings of the Central.



Note The scaled values of these parameters are used internally by the BLE stack. These are the actual values sent over the air.

- **Connection interval –** The Central device connecting to a Peripheral device needs to define the time interval for a connection to happen.
  - Minimum (ms) – This parameter is the minimum permissible connection time value to be used during a connection event. It is configured in steps of 1.25 ms. The valid range is from 7.5 ms to 4000 ms.
  - Maximum (ms) – This parameter is the maximum permissible connection time value to be used during a connection event. It is configured in steps of 1.25 ms. The valid range is from 7.5 ms to 4000 ms.

  Note In the multi-connection use case, the recommended minimum connection interval per connection should be greater than N * Max Time taken by individual connections to complete a Bluetooth Connection Event (CE).

  *Min_CI = N * Average Time Per CE*

  The average time for each CE is the amount of time taken to complete one BLE Tx and Rx transaction. This time varies depending on the Link Layer Data Length Extension (DLE) and BLE data rate (1 Mbps or 2 Mbps) configuration. The application can use the following timing lookup table for the CE value:
  - If DLE is enabled and data rate is 1Mbps, Average time = 6ms.
  - If DLE is enabled and data rate is 2Mbps, Average time = 3.5ms.
  - If DLE is disabled and data rate is 1Mbps, Average time = 2ms.
  - If DLE is disabled and data rate is 2Mbps, Average time = 1.6ms.

For example, if an application supports 4 BLE connections with DLE and 1-Mbps data rate, then the recommended minimum connection interval for each of the connections is:

*4 * 6 = 24 ms*

Note Connection intervals shorter than this value will still work, but under certain conditions, real-time control procedures (connection update, channel map update etc.) with a shorter update instance might result in a link disconnection.

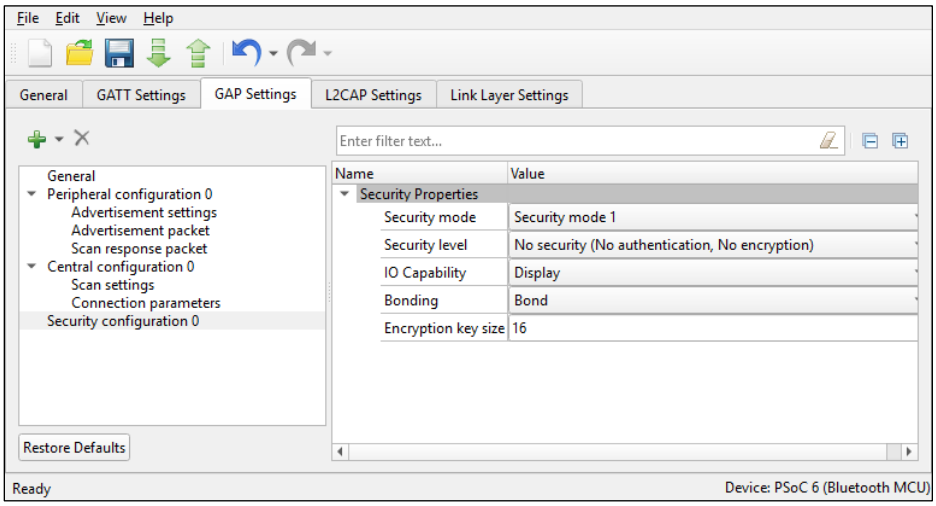
- **Slave Latency** – Defines the latency of the slave by responding to a connection event in consecutive connection events. This is expressed in terms of multiples of connection intervals, where only one connection event is allowed per interval. The valid range is from 0 to 499 events.
- **Connection Supervision Timeout** – This parameter defines the LE link supervision timeout interval. It defines the timeout duration for which an LE link needs to be sustained if there is no response from the peer device over the LE link. The time interval is configured in multiples of 10 ms. The valid range is from 100 ms to 32000 ms.

  Note For proper operation, the Connection Supervision Timeout must be larger than **(1 + Slave latency) * Connection Interval * 2** (ms). Refer to Bluetooth Core Specification Volume 6, Part B, Chapter 4.5.2 for more information on Connection Supervision Timeout.

GAP Settings Tab – Security (Bluetooth MCU)

This section contains several parameters to configure the global security options.

These parameters are configurable only if a connectable GAP role, Peripheral or Central, is selected. You can optionally set each Characteristic using its own unique security setting in the **Profile Tree**.



- **Security mode**

Defines the GAP security modes. Both available modes may support authentication.

  - Mode 1 – For designs where data encryption is required.
  - Mode 2 – For designs where data signing is required.

- **Security level**

Enables different levels of security depending on the selected **Security mode**:

  - Mode1
    - No security – The device does not use encryption or authentication.
    - Unauthenticated pairing with encryption – The device sends encrypted data after establishing a connection with the remote device.

- Authenticated pairing with encryption – The device sends encrypted data after establishing a connection with the remote device. To establish a connection, the devices perform the authenticated paring procedure.
- Authenticated LE Secure Connections pairing with encryption – The device uses an algorithm called Elliptic curve Diffie–Hellman (ECDH) for key generation, and a new pairing procedure for the key exchange. It also provides a new protection method from Man-In-The-Middle (MITM) attacks - Numeric Comparison.

- Mode 2
  - Unauthenticated pairing with data signing – The device performs data signing prior to sending it to the remote device after a connection is set up.
  - Authenticated pairing with data signing – The device performs data signing prior to sending it to the remote device after a connection is set up. To establish a connection, the device performs the authenticated paring procedure.

- **Keypress notifications**

Provides an option for a keyboard device during the LE secure pairing process to send key press notifications when the user enters or deletes a key. This option is available when the **Security level** is set to Authenticated LE Secure Connections pairing with encryption and **I/O capabilities** option is set to either Keyboard or Keyboard and Display.

- **I/O capabilities**

This parameter refers to the device's input and output capability that can enable or restrict a particular pairing method or security level.

- Display – Used in devices with the display capability and may display authentication data. GAP authentication is required.
- Display Yes/No – Used in devices with a display and at least two input keys for Yes/No action. GAP authentication is required.
- Keyboard – Used in devices with a numeric keypad. GAP authentication is required.
- No Input No Output – Used in devices without any capability to enter or display the authentication key data to the user. Used in mouse-like devices. No GAP authentication is required.
- Keyboard and Display – Used in devices like PCs and tablets. GAP authentication is required.

- **Bonding Requirement**

This parameter is used to configure the bonding requirements. The purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication. The maximum number of remote devices that can be bonded is 128.

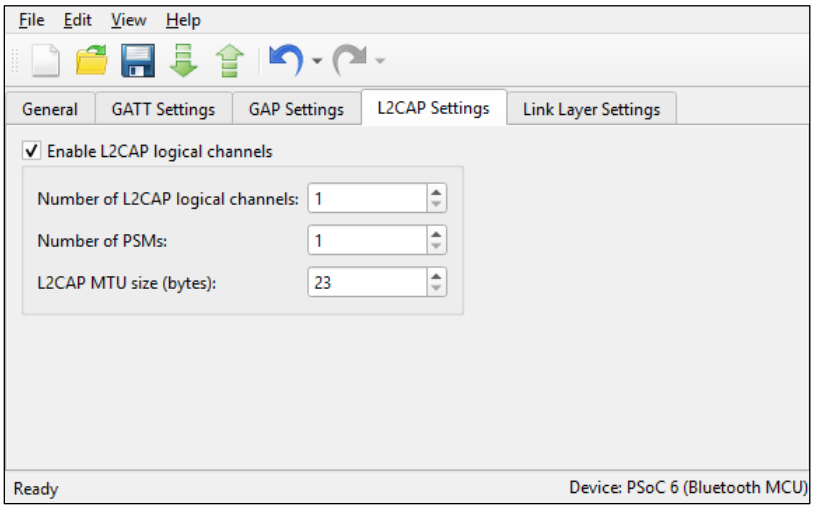
- **Bonding** – The device stores the link key of a connection after paring with the remote device in the application flash memory. If a connection is lost and re-established, the device uses the previously stored key for the connection.
- **Note** Bonding information is stored in RAM and should be written to application flash if it needs to be retained during a shutdown.
- **No Bonding** – The pairing process is performed on each connection establishment.

- **Encryption Key Size**

This parameter defines the encryption key size based on the Profile requirement. The valid values of an encryption key size are 7 to 16 bytes.

L2CAP Settings Tab (Bluetooth MCU and 43xxx)

The L2CAP settings define parameters for L2CAP connection oriented channel configuration.



Enable L2CAP Logical Channels

This parameter enables configuration of the L2CAP logical channels.

The default value for Bluetooth MCU is true, for 43xxx – false.

Number of L2CAP Logical Channels

This parameter defines the number of LE L2CAP connection oriented logical channels required by the application. The valid range is from 1 to 255. Default: 1.

Number of PSMs

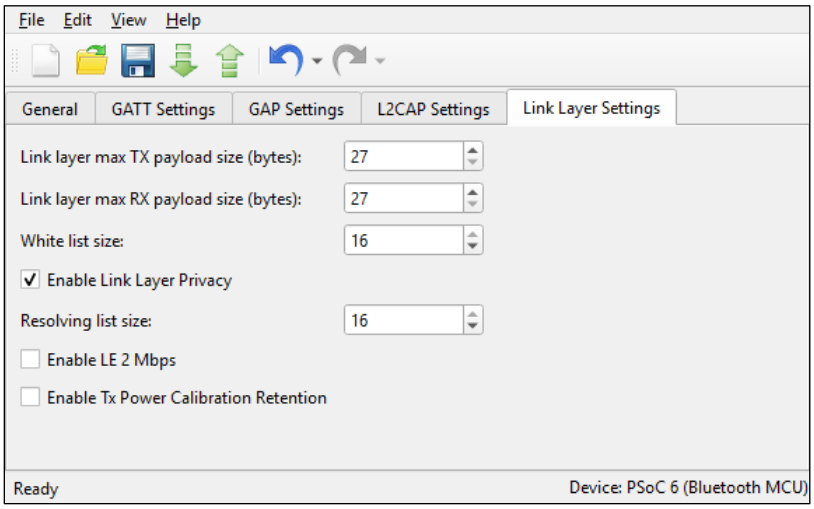
This parameter defines the number of PSMs required by the application. The valid range is from 1 to 255. Default: 1.

L2CAP MTU Size

This parameter defines the maximum SDU size of an L2CAP packet. The valid range is from 23 to 65488 bytes. Default: 1280 bytes when **Internet Protocol Support Service** is supported and 23 bytes otherwise.

Link Layer Settings (Bluetooth MCU)

The Link Layer settings parameters for the Link Layer.



Link Layer Max TX Payload Size

The maximum link layer transmits a payload size to be used in the design. The actual size of the link-layer transmit packet is decided based on the peer device's link-layer receive packet size during Data Length Update Procedure and will be informed through the CY_BLE_EVT_GAP_DATA_LENGTH_CHANGE event. The valid range is from 27 to 251 bytes.

Link Layer Max RX Payload Size

The maximum link layer receives a payload size to be used in the design. The actual size of the link-layer receive packet is decided based on the peer device's link-layer transmit packet size during Data Length Update Procedure and will be informed through 'CY_BLE_EVT_GAP_DATA_LENGTH_CHANGE' event. The valid range is from 27 to 251 bytes.

Setting Link Layer Max TX Payload Size or Link Layer Max Rx Payload Size to the value greater than 27 enables the LE Data Length Extension feature.

White List Size

The maximum number of devices that can be added to the White list. The valid range is from 1 to 16. Default: 16.

Enable Link Layer Privacy

Enables LL Privacy 1.2 feature of Bluetooth 4.2 and enables generation of CY_BLE_EVT_GAP_ENHANCE_CONN_COMPLETE and CY_BLE_EVT_GAPC_DIRECT_ADV_REPORT events.

Note that CY_BLE_EVT_GAP_DEVICE_CONNECTED event is not generated when this feature is enabled.

Resolving List Size

The maximum number of peer devices whose addresses should be resolved by this device. This parameter is applicable when the **Enable Link Layer Privacy** feature is enabled. The valid range is from 1 to 16. Default:16.

Enable LE 2 Mbps

Enables LE 2 Mbps feature of Bluetooth 5.0.

The 2 Mbps feature enables a new Physical (PHY) modulation scheme that allows increasing data throughput between the two devices which support this feature. Refer to Bluetooth Core Specification v5.0 for more details about this feature.

Use the Cy_BLE_SetDefaultPhy() API after CY_BLE_EVT_STACK_ON event to set the preferred default PHY for all connections, or Cy_BLE_SetPhy() API to set PHY for the current connection.

The CY_BLE_EVT_PHY_UPDATE_COMPLETE event will indicate when controller has changed the transmitter PHY or receiver PHY in use.

Enable TX Power Calibration Retention

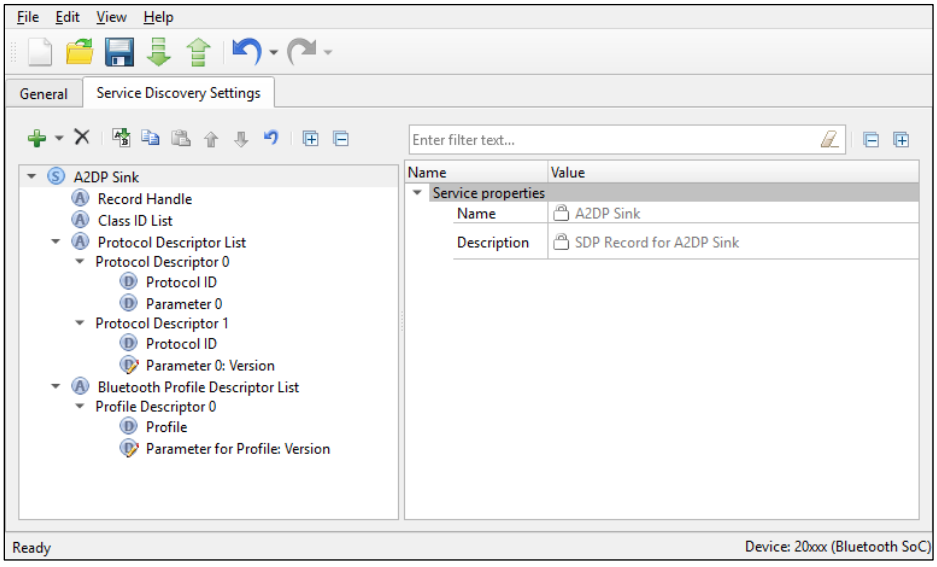
When enabled, the BLE radio TX power calibration is performed only once after programming and the calibration values are retained in the Sflash location. This retained value is reloaded to radio power calibration registers during consecutive device reboots. This reduces the BLE stack boot-up time significantly.

Notes

- In BLE dual-core mode, ensure to call the Cy_SysDisableCM4() function before enabling the BLE controller [that is, before calling Cy_BLE_Enable() on the controller core].
- The calibration values are retained in user's row 0 (after BLE_DEVICE_ADDRESS) of the Sflash location.

Service Discovery Settings Tab (20xxx) (Beta)

The **Service Discovery Settings** tab is used to configure SDP Services. This tab is available for Bluetooth modes **Single mode BR/EDR** or **Dual mode**. The tab has three areas: toolbars, the Services tree, and the parameters configuration section.



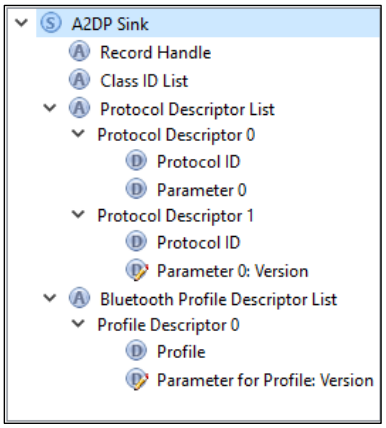
Toolbars

The toolbars contain navigation options and means to add or delete Services and Attributes.

- **Add** – Allows adding a Service or an Attribute to the Services tree. To add an Attribute, the parent Service should be highlighted in the Services tree. Refer to the Profiles section for the available Services.
- **Delete** – Deletes the selected Service or Attribute.
- **Rename** – Renames the selected item in the Services tree.
- **Copy/Paste** – Copies/pastes items in the Services tree.
- **Move Up/Down** – Moves the selected item up or down in the Services tree.
- **Reset branch to default** – Resets the selected item with child items in the Services tree to the default.
- **Expand All** – Expands all items in the Services tree.
- **Collapse All** – Collapses all items in the Services tree.

Services Tree

The Services tree is used to view SDP Services, Attributes, Data element groups, and Data elements. By navigating through the tree, you can quickly add, delete, or modify Services and Attributes using the toolbar buttons or the context menu.



You can configure the parameters by clicking an item on the tree. These parameters will show in the Parameters Configuration section.

The tree may contain the following nodes:

- Service
- Service Attribute
- v Data element group
- Data element. means that the Data element has editable fields.

In addition, nodes may have the following colors:

- Shaded icon – The node is mandatory.
- White icon – The node is optional.

Parameters Configuration

The Parameters Configuration section allows you to view and configure a Service, Attribute, or Data element by selecting them in the tree. Attribute values are mainly hard-coded (and thus editing is disabled) to supply the required values according to the Bluetooth Core Specification. Click through each Attribute and Data element under each Service to identify configurable values, and perform any additional configuration needed.

SDP Services

The following SDP Services are available for selection:

- Advanced Audio Distribution Profile (A2DP) Sink
- Advanced Audio Distribution Profile (A2DP) Source
- A/V Remote Control (AVRC) Controller
- A/V Remote Control (AVRC) Target
- Device ID
- Hands-Free Audio Gateway
- Hands-Free Unit
- Human Interface Device
- Serial Port

Migration of Configuration File Format

The versions of the Bluetooth Configurator prior to 2.0 use a C header file to store the configuration as a comment in the XML format. For versions 2.0 and later, the configuration is stored in a separate .cybt file in the XML format. Use the following instructions to migrate to the .cybt file as appropriate:

1. Launch the Bluetooth Configurator.
2. Click **Open**.
3. Select the **Obsolete configurator files (*.h)** file extension and choose a header file.
4. After the configuration is loaded, click **Save** to create a .cybt file and update the C file.

Notes

- The .cybt file is located one directory up related to the header file.
- The command-line argument `--config` does not accept obsolete configuration files (*.h).

References

Refer to the following documents for more information, as needed:

- Eclipse IDE for ModusToolbox User Guide
- Device Configurator Guide
- BT SDK Documentation
- PSoC 6 BLE Middleware API Reference Guide
- Device Datasheets
- Device Technical Reference Manuals

Version Changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.1	Added 20xxx devices support.
	Fixed the issue with code generation for GAP Central role connection parameters. Their values were always generated as default.
2.0	Added the Bluetooth BR/EDR support for 20xxx devices (Beta version). Added the General and Service Discovery Settings tabs for 20xxx devices.
	Changed the user configuration storage location. Previously the configuration was stored in the header file as a comment in XML format. Now it is stored in the *.cybt file in XML format.
	Added Custom OTA Firmware Upgrade Service for 20xxx devices.
	Minor changes in the generated code for 20xxx devices: ‘u’ suffix not added to unsigned integer values.
	Changed BLE Stack RAM memory allocation (PSoC 6 devices).
	Changed the names of the advertisement packet defines (PSoC 6 devices). Before: CY_BLE_ADV_PKT_<INDEX>_INDEX<NAME>; After: CY_BLE_ADV_PKT_<INDEX>_INDEX_<NAME>.
	Added New, Close, Import, Export, and Reset View commands. User interface details were improved.
	Removed the functionality to launch the Bluetooth Configurator from the Device Configurator.
2.1	Added menu Edit with two items Undo/Redo.
	Fixed minor issues in the representation of the GATT-profiles properties.
	Modified the default selection of the GATT-characteristic permissions behavior for 20xxx devices.
2.20	Added the PSoC 6 MCU with 43xxx Connectivity support.
	Added Copy feature to the Notice List.
	Fixed issues in the HID report editor.
	Fixed issues in the Advertisement packet editor.
	Improved validation of fields.
	Removed the Profile role Weight Scale and Collector from the Add Profles menu.
2.21	Added a General tab for 43xxx devices.
	Added a possibility to disable the GATT database for 43xxx devices.
	Added new parameters for 43xxx devices: Maximum remote servers and clients connections, and Maximum attribute length.
	Changed the maximum limit of the MTU Size parameter from 512 to 517 for 43xxx devices.
	Fixed the issue with the CY_BT_L2CAP_MTU_SIZE value for 43xxx devices. In case when L2CAP logical channels were disabled, the default value generated for this define was 23 instead of 512 if the configurator’s output files were generated by the build script (not GUI itself).
	Fixed the issue that the Peripheral GAP role was restored after the configurator reopening even when previously removed (43xxx devices).
	Advertisement packet configuration: in case when the GATT Settings tab is disabled, the GATT services configured previously on the GATT Setting tab are not available for selection for the Advertisement packet Service UUID and Service Data AD types.
	The configurations with characteristics of length 0 produced empty arrays in the generated code for 20xxx and 43xxx devices. This could cause build warnings on some compilers. Added an extra ‘0’ element to the empty arrays to avoid that.
	Removed duplicated CY_BT_HIGH_DUTY_CONN_SCAN_WINDOW define.

Version	Change Descriptions
	GAP scan connection parameters: removed possibility to disable Connection interval minimum/maximum and Connection supervision timeout.
	GAP scan interval settings for 43xxx: corrected values of the following defines: CY_BT_HIGH_DUTY_SCAN_DURATION, CY_BT_LOW_DUTY_SCAN_DURATION, CY_BT_HIGH_DUTY_CONN_SCAN_DURATION, CY_BT_LOW_DUTY_CONN_SCAN_DURATION.
	When the GAP role is non-connectable (Broadcaster or Observer), the server_max_links field in the wiced_bt_cfg_settings struct is set to 0 instead of 1.