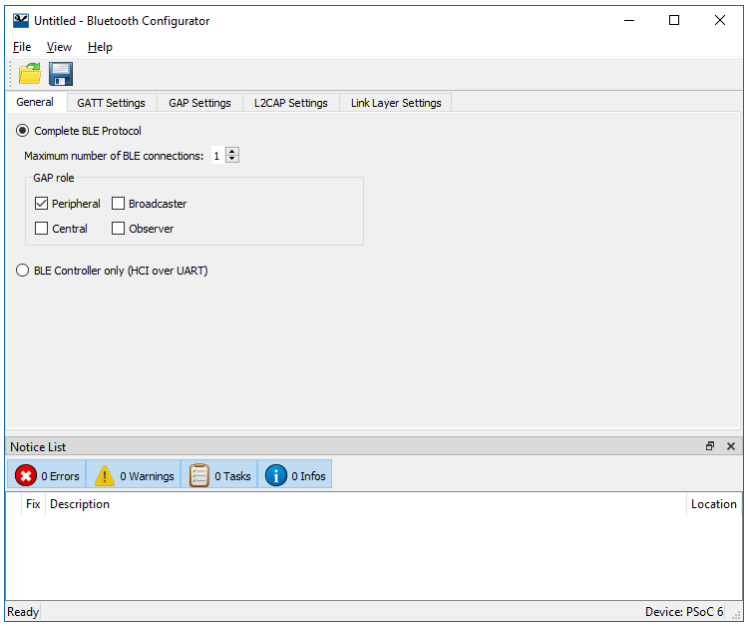
ModusToolbox™ Bluetooth Configurator Guide

Version 1.1

Overview

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



Supported Devices

The Bluetooth Configurator can be used with two families of Cypress devices: 20xxx and PSoC 6. The Configurator generates different code for each device family.

SIG adopted Profiles and Services

The Bluetooth Configurator supports numerous SIG-adopted GATT-based Profiles and Services. Each of these can be configured for either a GATT Client or GATT Server. It generates all the necessary code for a particular Profile/Service operation, as configured in the Bluetooth Configurator.

Custom Profiles

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

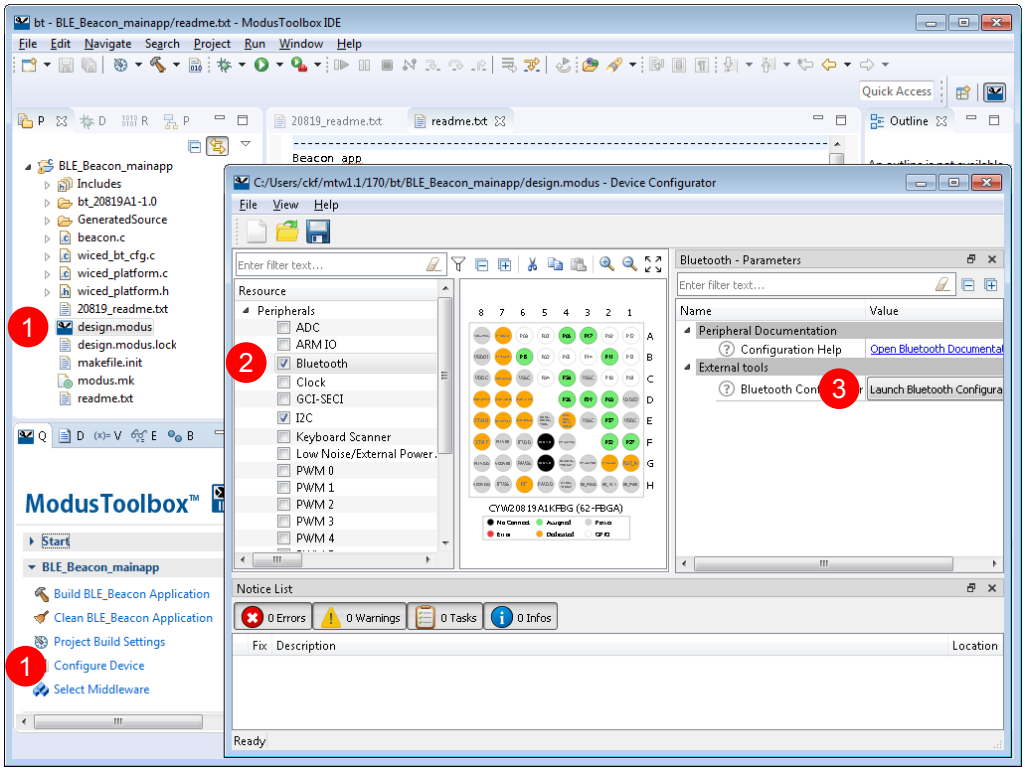
Launch the Bluetooth Configurator

The Bluetooth Configurator is a stand-alone tool that contains menus and tabs to configure Bluetooth settings. You can run the configurator from, and use it with, a ModusToolbox IDE application. You can also run it independently of the IDE. Then, you can either use the generated source with a ModusToolbox IDE application, or use it in any software environment you choose.

From a ModusToolbox IDE Application

1. Launch the Device Configurator from the ModusToolbox IDE.
2. Enable the Bluetooth resource.

   Note The Device Configurator contains various tabs for PSoC 6 devices.
3. On the **Parameters** pane, click the **Launch Bluetooth Configurator** button.



When you save changes, it generates/updates firmware in the ModusToolbox IDE application's *GeneratedSource* folder.

Independent of the ModusToolbox IDE

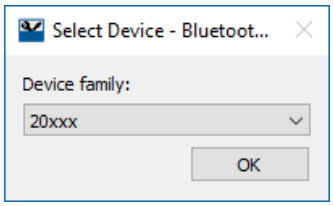
To run the Bluetooth Configurator independently, navigate to the install location and run the executable. On Windows, the default install location for the Bluetooth Configurator is:

[user_home]/ModusToolbox_<version>/tools/bt-configurator-<version>

For other operating systems, the installation directory will vary, based on how the software was installed.

Select Device Dialog

When running independently of the ModusToolbox IDE, the configurator asks to select the device for which the code will be generated. Two options are available: **20xxx** and **PSoC 6**.



You must choose a device since parameter configuration options depend on the selection.

When run independently, the configurator opens with the default configuration. You must open a header file for the application for which you want to configure Bluetooth. If a header file does not exist, then it will be created when clicking the **Save** button.

From the Command Line

You can run the configurator from the command line. However, there are only a few reasons to do this in practice. The primary use case would be to re-generate source code based on the latest configuration settings. This would often be part of an overall build script for the entire application.

For information about command line options, run the configurator using the `-h` option.

Quick Start

1. Launch the Bluetooth Configurator.
2. Use the Bluetooth Configurator to configure the application (for PSoC 6, GATT, GAP, L2CAP, LL, etc.; for 20xxx, GATT database). Refer to the Parameter Configuration section for more details.
3. Save the configuration file to generate source code.

The Bluetooth Configurator generates code into a GeneratedSource directory in your ModusToolbox IDE application, or in the arbitrary location for non-ModusToolbox 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 the hardware.

Note There is a link to the API documentation from the Device Configurator.

PSoC 6 Device

Include the *cycfg_ble.h* file in your application. Use the generated structures as input parameters for Cy_BLE_Init() function. Refer to section *Configuration Considerations* in theBluetooth Low Energy (BLE) Middleware API Reference Guide for more details about how initialize and enable Bluetooth Middleware.

20xxx Device

The following files are generated:

- *cycfg_bt.h*: Contains an XML representation of the configuration embedded into the comments.
- *cycfg_gatt_db.h* and *cycfg_gatt_db.c*: Contain the GATT database code.

Include *cycfg_gatt_db.h* in your application.

Menus

The Bluetooth Configurator contains the following menus.

File

- **Open:** Open and load an existing header file.
- **Save:** Save changes to the file. If the file does not exist, the Save file dialog will open.
- **Exit:** Close the configurator.

View

This menu contains a command to hide or show the **Notice List** pane. The pane is shown by default.

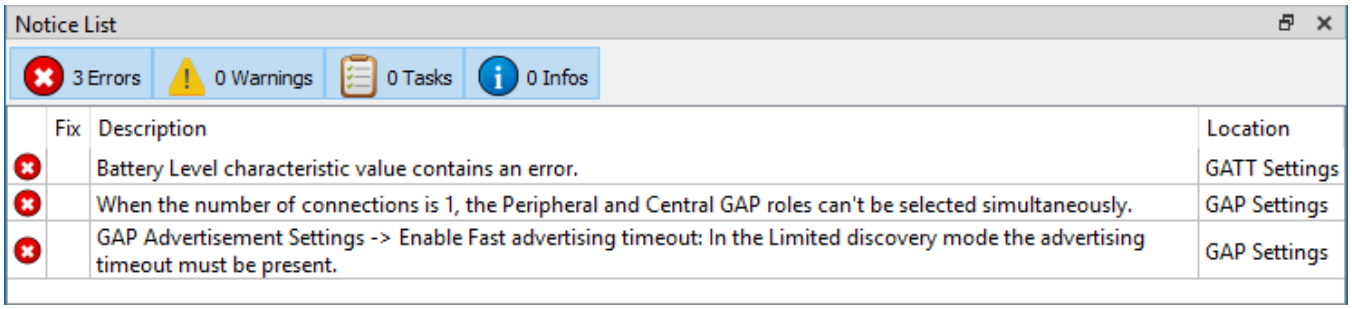
There is also a command to show or hide the toolbar.

Help

- **View Help:** Open this document.
- **About:** Open the About box for version information.

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, which 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.

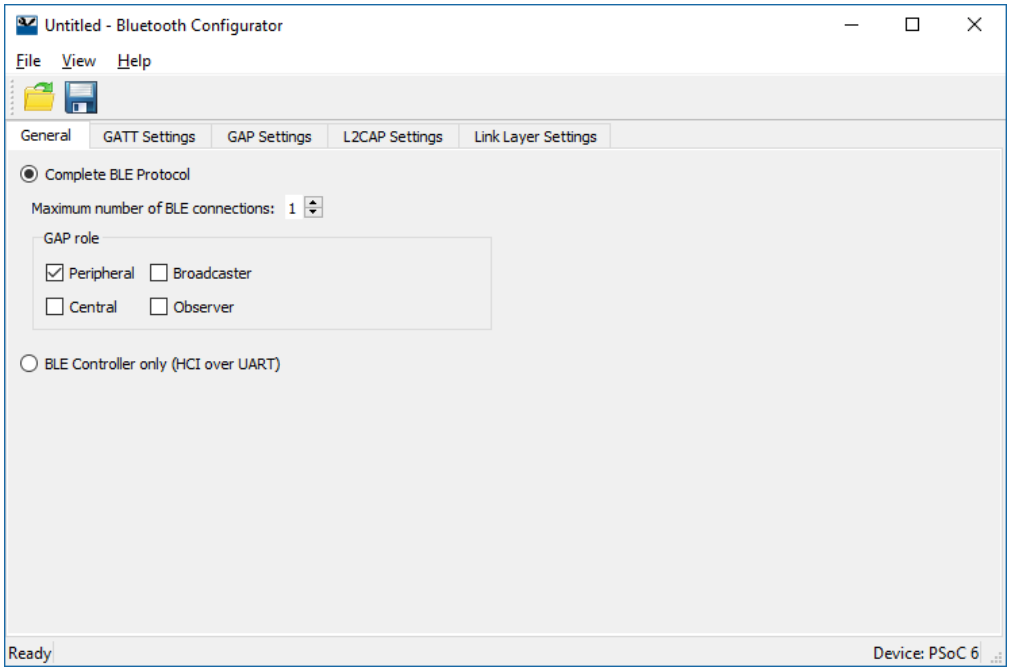
Parameter Configuration

The Bluetooth Configurator contains several tabs in which to configure parameters. The set of active tabs depends on the selected device and other parameters values. For **20xxx** devices, only the GATT Settings Tab is available.

General Tab

Note This tab is applicable for **PSoC 6** device only.

The **General** tab allows general configuration of the Bluetooth resource. This tab contains tools to load and save configurations. It also contains options for the type of configuration.



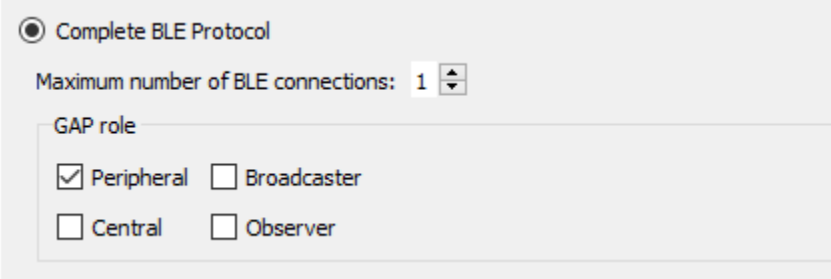
Mode Selection

On the main part of this tab, there are two options to select a mode:

- Complete BLE Protocol
- BLE Controller Only (HCI over UART)

General Tab – Complete BLE Protocol

The Complete BLE Protocol mode enables both BLE 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. Valid range is from 1 to 4. Refer to the Multi Connection Support section for more details.

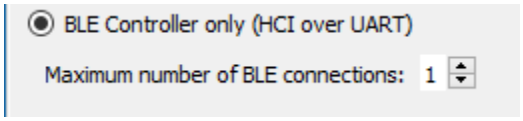
Gap Role

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 a 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** – Similar to the Peripheral role, the device sends advertising data. However, Broadcaster does not support connections and can only send data but not receive them.
- **Observer** – When in this role, the device scans for Broadcasters and reports the received information to an application. The Observer role does not allow transmissions.

General Tab – BLE Controller only (HCI over UART)

Choosing this configuration enables HCI mode, which enables use of the device as a BLE controller. It also allows communication with a host stack using an embedded UART. When choosing this mode all other tabs become unavailable.



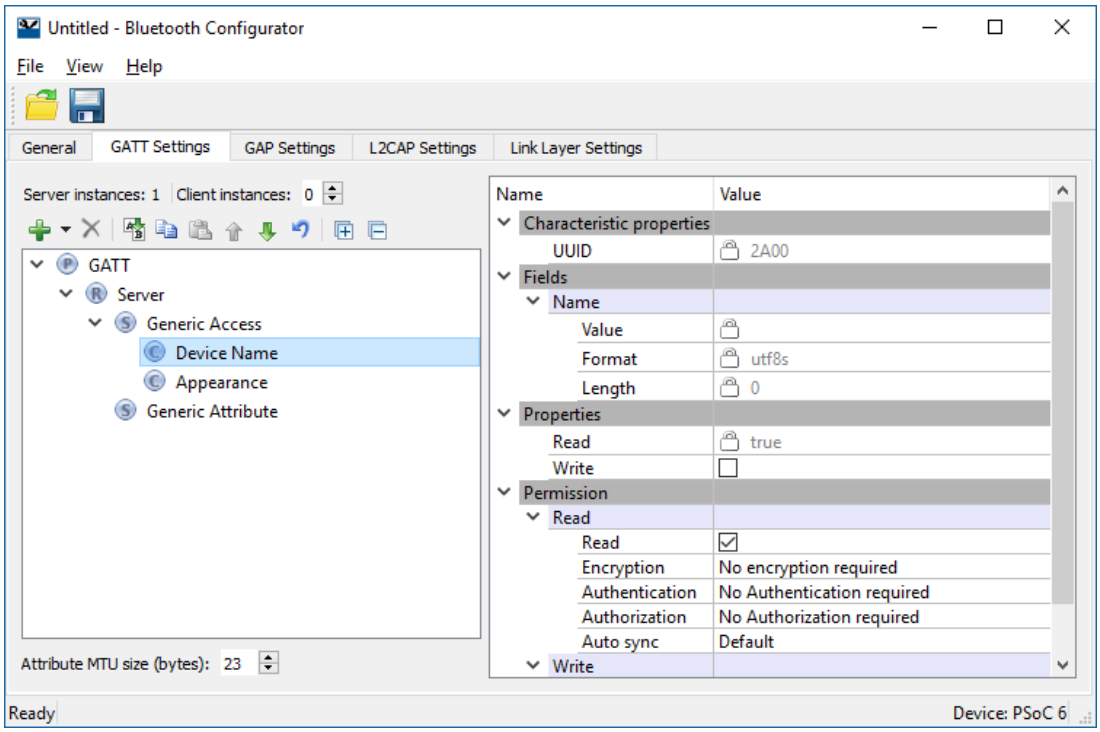
Maximum Number of BLE connections

This parameter displays how many BLE connections are allowed. Valid range is from 1 to 4. Refer to the Multi Connection Support section for more details.

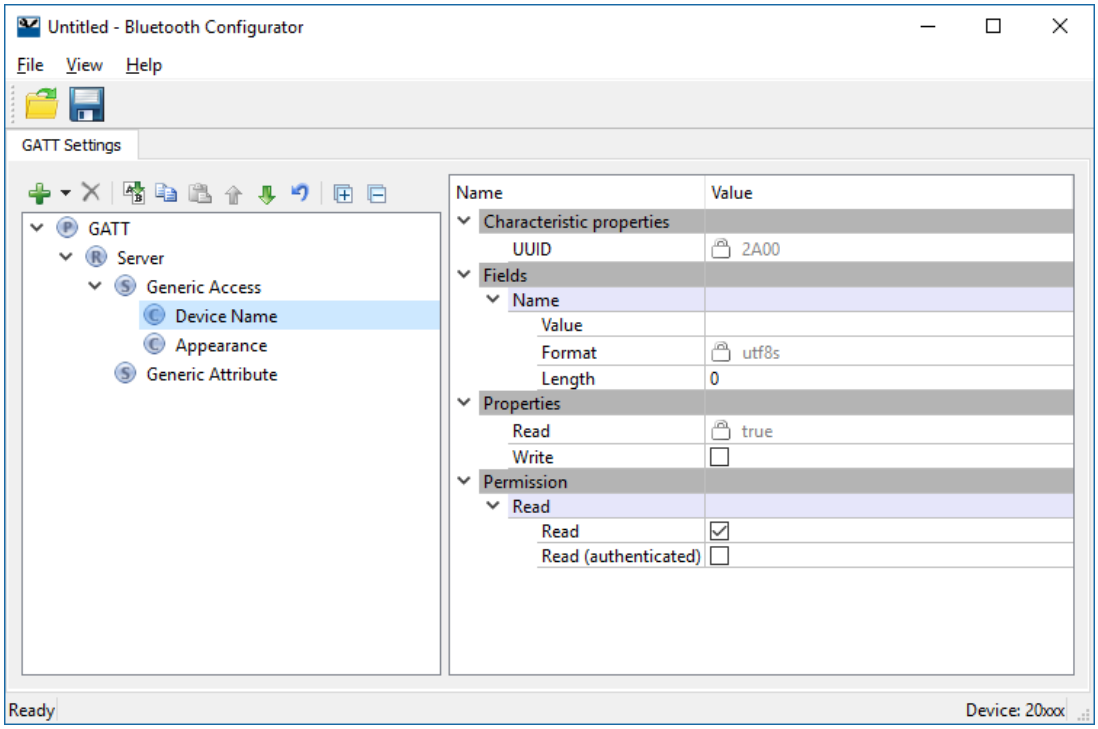
GATT Settings Tab

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 **PSoC 6 devices**, the tab appears as follows:



For **20xxx devices**, the tab looks as follows:



Toolbars

The toolbars contain navigation options and a means 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 reused across all BLE connections.

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

  Note Applicable for **PSoC 6** device 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 **PSoC 6** device only.

- **Add Profile –** This option is available when the GATT node is highlighted in the Profile tree. It allows adding a whole Profile to the Profiles tree. This option doesn't remove existing Services from the tree. Several Profiles can exist in the tree simultaneously.

- **Add Service –** This option is available when the **Profile Role** is highlighted in the Profile tree. It allows loading of Services in the selected **Profile Role**. In GATT server configuration, this option adds the selected service data to the server GATT database and enables service specific APIs. In GATT client configuration, the data structures for auto discovery of this service are created. If services that are not populated in the GUI are discovered during auto discovery, it ignores those service and the application is responsible for discovering the details of such services. Refer to the Profiles section for the available Services.

- **Add Characteristic –** This option is available when a Service is highlighted in the Profile tree. The Characteristic options are unique to each Service and are all loaded automatically when a Service is added to the design. The **Add Characteristic** button can be used to manually add new Characteristics to the Service. All Characteristics for the above mentioned Services plus Custom Characteristic are available for selection.

- **Add Descriptor –** This option is available when a Characteristic is highlighted in the Profile tree. Similar to the Characteristic options, 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 **PSoC 6** device 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.

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.

Attribute MTU Size

Maximum Transmission Unit size (bytes) of an attribute to be used in the design. 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 **PSoC 6** device only.

Profiles

You can add a whole Profile to the Profiles tree from a list of supported Profiles. Note that for **20xxx** device only **GATT Server** role is configurable. 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.
- **Cycling Power Observer** Profile role. Can only talk to a device with the **Cycling Power Broadcaster** role.

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

The Find Me 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 following Service: **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

The Indoor Positioning Service exposes location information to support mobile devices to position themselves in an environment where GNSS signals are not available. For example, in 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 is likely to be a sensor or actuator. A device supporting the Router role is likely to 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

The Proximity Profile enables 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

The Time 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

The Weight Scale 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, and may be also a GATT Client.

  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, and may be also a GATT Service.

  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 communication between Power Receiver Unit and Power Transmitter Unit in the 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 **PSoC 6** device only.

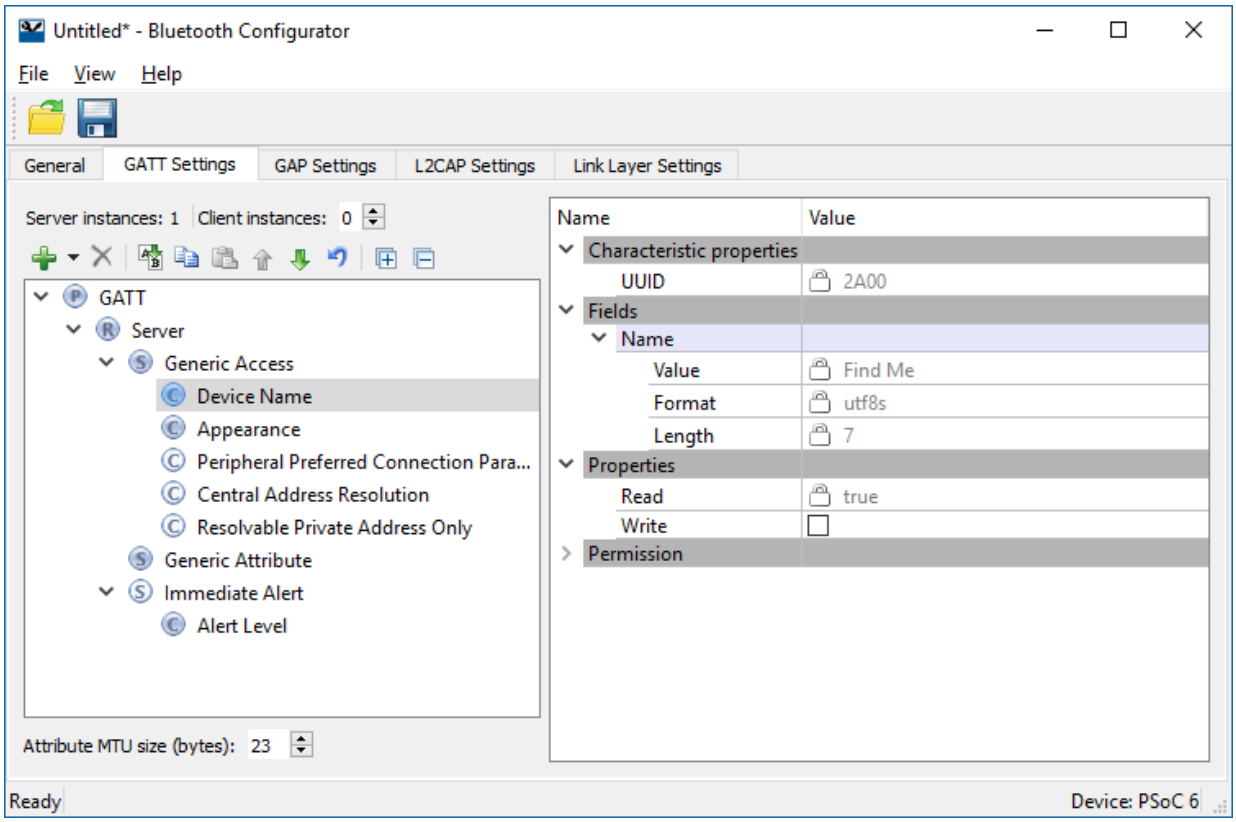
BLE supports the Bootloader Profile and Bootloader Service, which allow a Bootloader to update the existing firmware on the Cypress BLE device. The Bootloader Service uses the Bluetooth Low Energy interface as a communication interface. It can be added to any of the profiles if the design requires updating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

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

Notes

- All Profiles must have a **Generic Access Service** and a **Generic Attribute Service**.
- 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.
- Tree node icons may have two colors: blue and white. Blue color indicates that a node is mandatory and cannot be deleted. White color indicates that a node is optional.

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 that particular Characteristic settings. 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 **PSoC 6** devices, 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 **PSoC 6** devices, 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 Peripheral preferred connection parameters section for more information.

  - **Minimum Connection Interval** – This is a 2-byte parameter that denotes the minimum permissible connection time.
  - **Maximum Connection Interval** – This is a 2-byte parameter that denotes the maximum permissible connection time.
  - **Slave Latency** – This is a 2-byte value and defines the latency between consecutive connection events.
  - **Connection Supervision Timeout Multiplier** – This is 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.

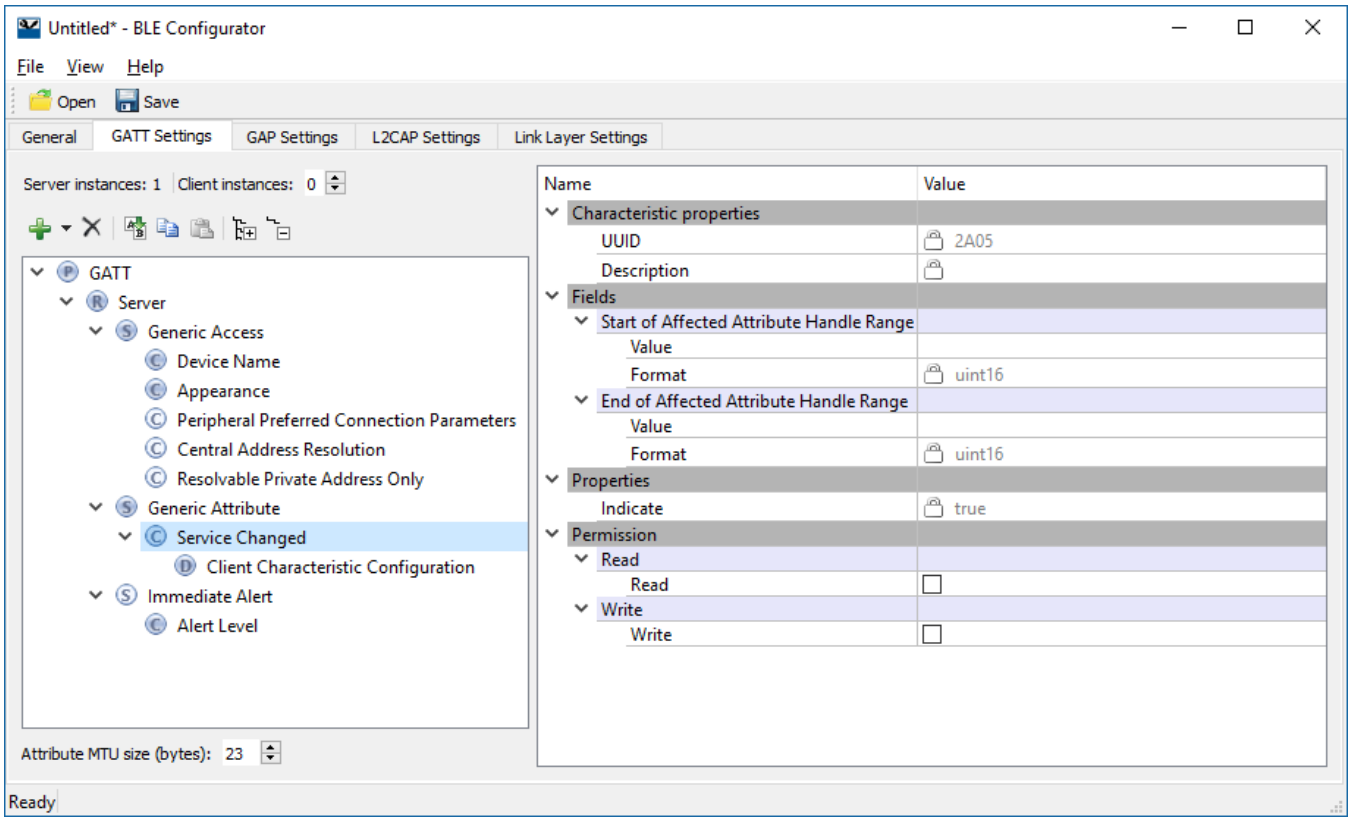
Note For proper operation, the Connection Supervision Timeout must be larger than **(1 + Slave latency) * Connection Interval * 2** (ms). Refer to Bluetooth Core Specification Volume 6, Part B, Chapter 4.5.2 for more information on Connection Supervision Timeout.

Note The above parameters are used for 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 should request a connection parameter update by sending an L2CAP connection parameter update request at an appropriate time.

A typical peripheral implementation should initiate 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

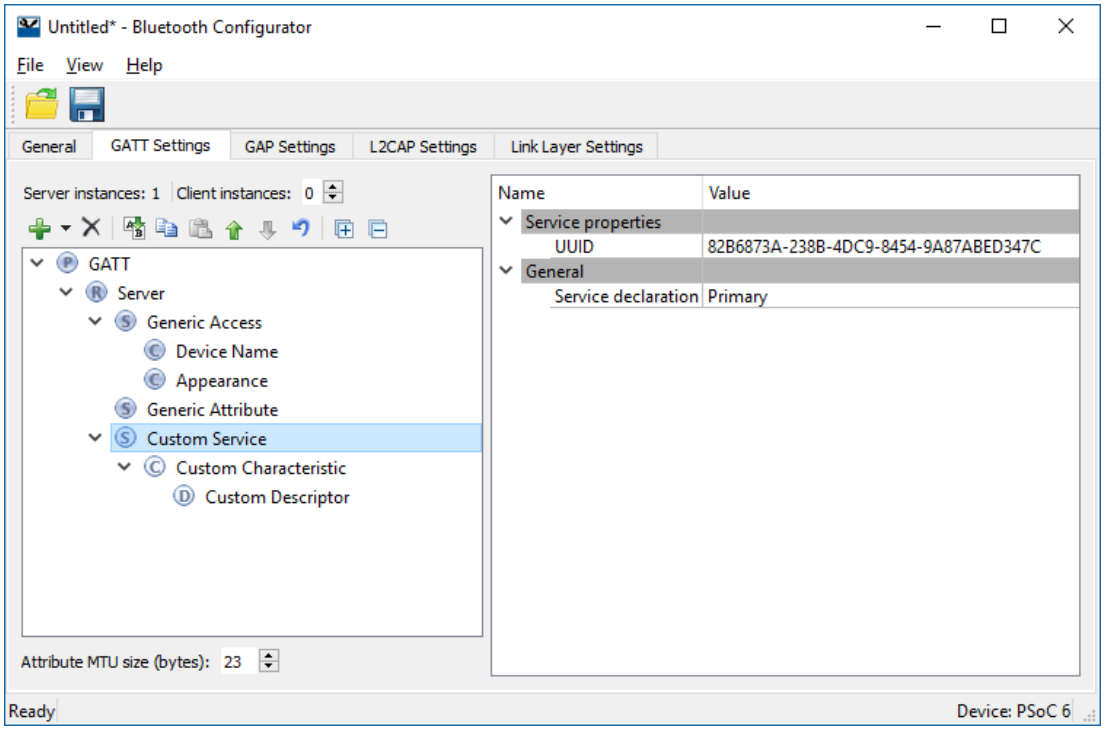


By default, this Service does not have any Characteristics. Add the optional Service Changed Characteristic under the Generic Attribute Service if needed.

- **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.

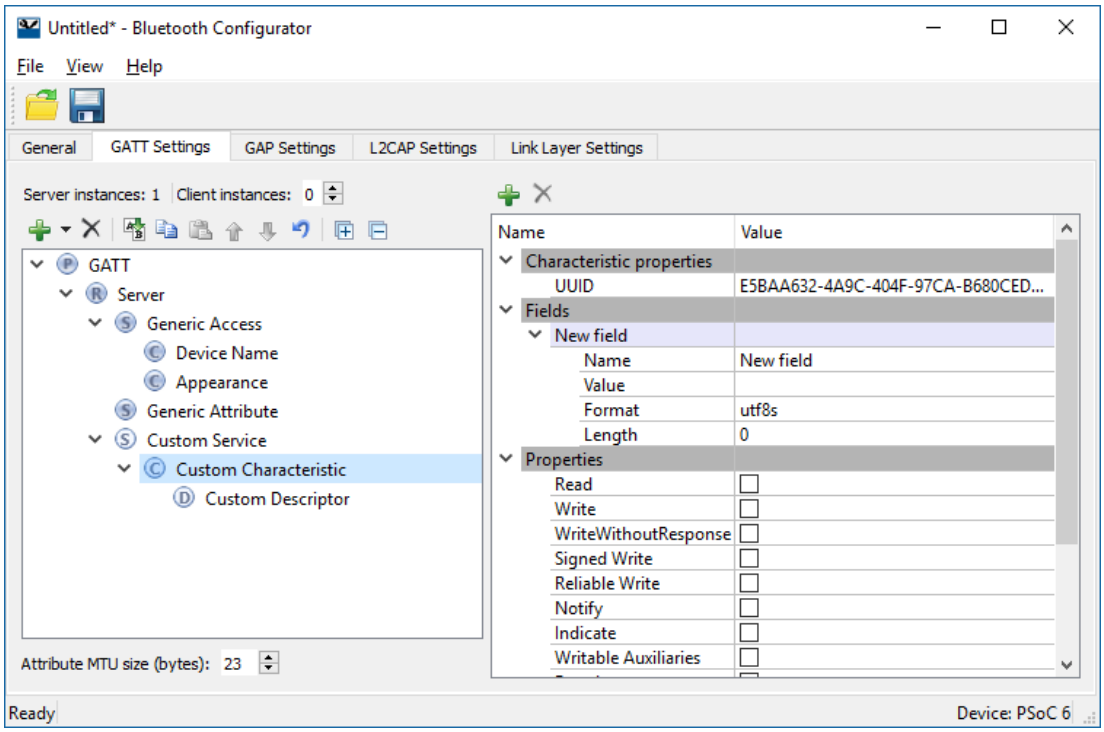
Service type

- **Primary** – Represents the primary functionality of the device.
- **Secondary** – Represents an 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 the 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. In case of the Custom Characteristic, the fields are customizable. 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, please see Bluetooth Core Specification Vol.3, part G (GATT), section 3.3.1.1 “Characteristic Properties”.

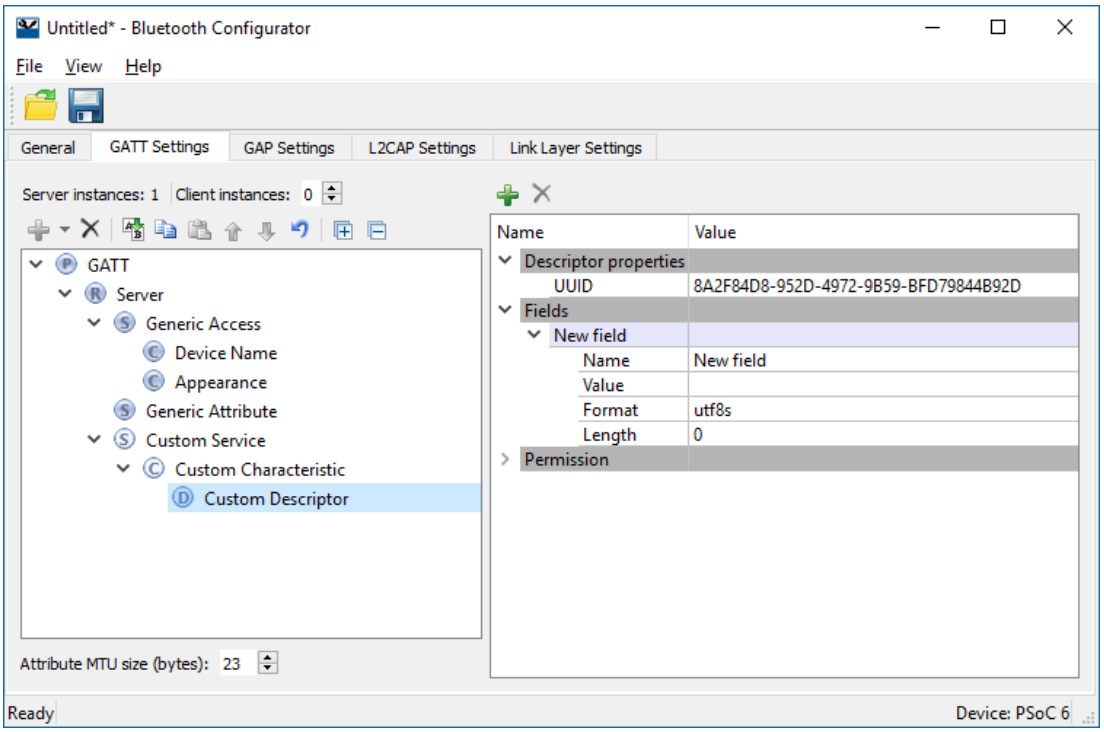
Permissions

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 **PSoC 6** devices, 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 the Security Configuration 0 on the **GAP Settings** tab. Additional Security configurations don’t 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

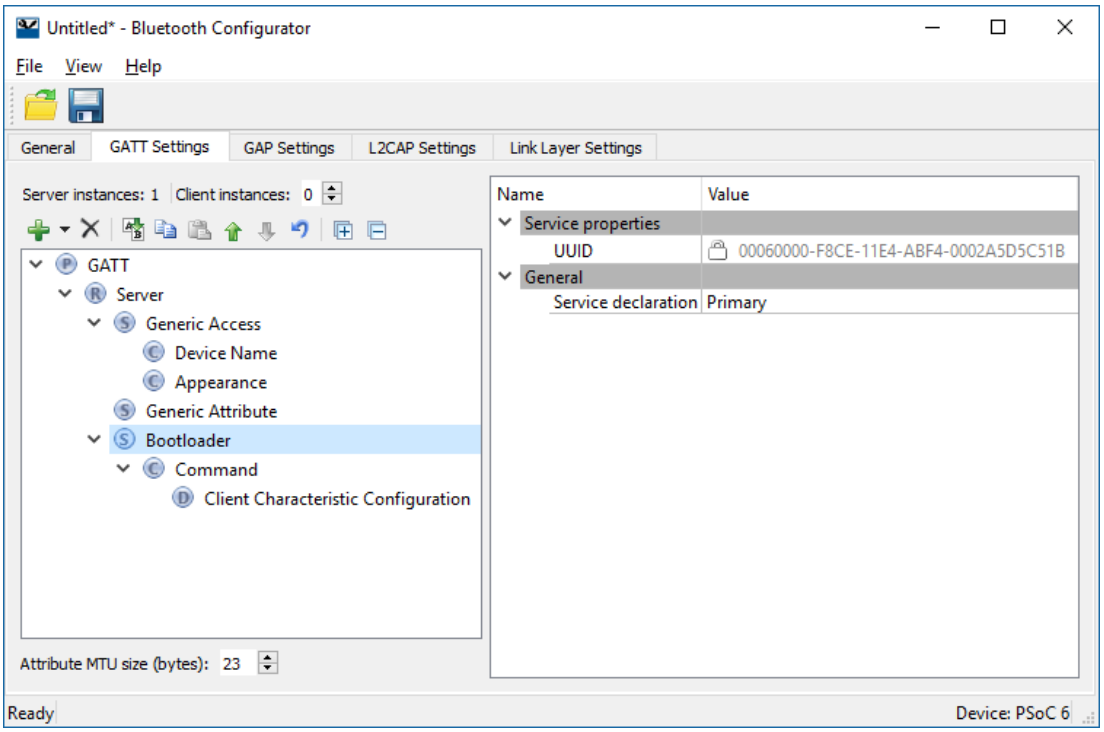
Fields represent a Descriptor value. The default value for each field can be set in the **Value** property. In case of the Custom Descriptor, the fields are customizable. You can add or delete fields using the tool buttons located above the Properties editor.

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

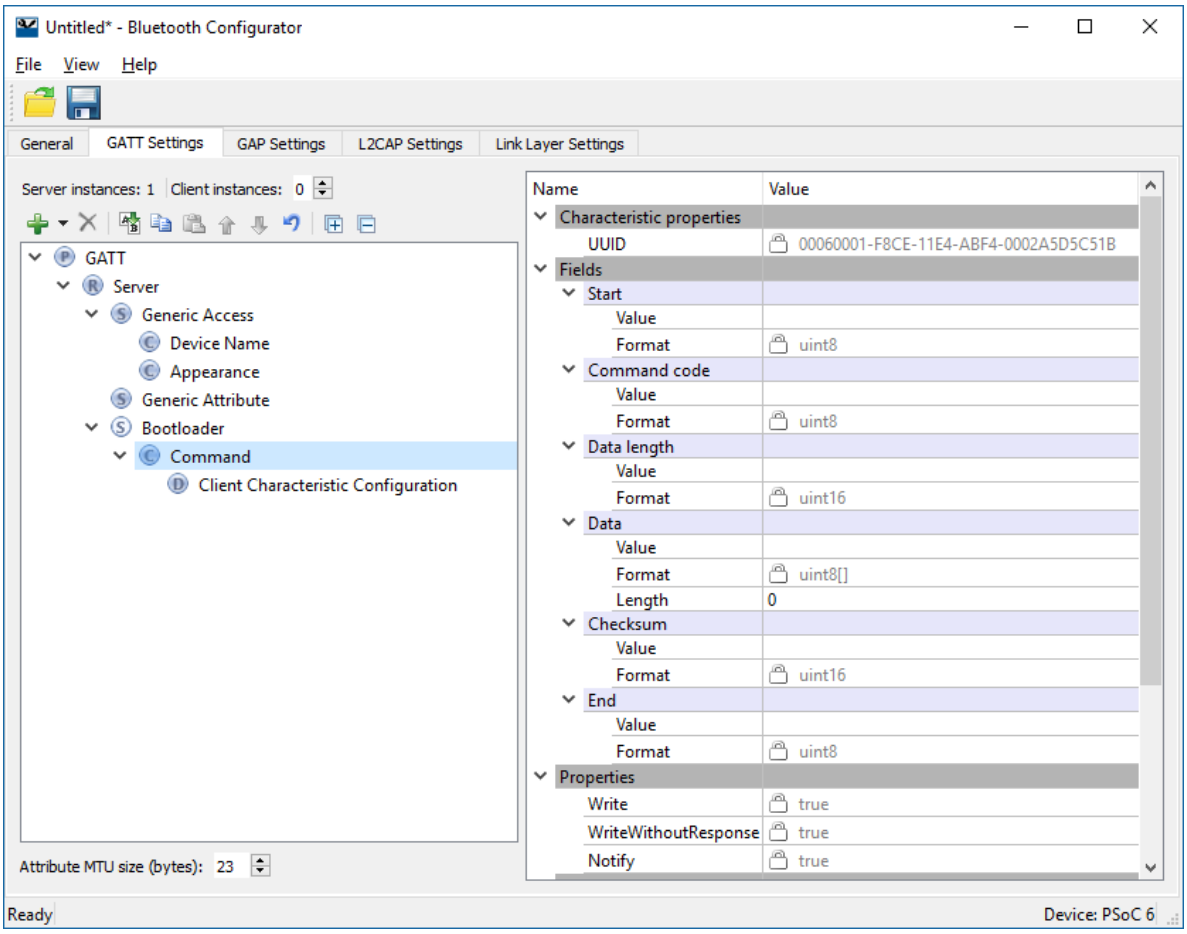
Note Available for **PSoC 6** device only.



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

Fields represent Command Characteristic values, such as the following.

- Start of packet – This constant defines the start of the bootloader packet.
- Command – This field defines the bootloader command.
- Status Code – This field defines the status code of the command.
- Data Length – This field defines the length of the bootloader command/response and should be set to the maximum command data length that can be used in the design.

  Per the specifics of the BLE protocol, if the command requires a response larger than 20 bytes, the attribute MTU size should be increased. To support the responses with data length set to 56 (response for **Get Metadata** command), the attribute MTU size should be set to 66. This can be seen from the following 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 this will result in the Bluetooth resource failing to send a response to the requested command.

- Data – This field defines the bootloader command data. The length of this field is specified by the Data Length field.
- Checksum – This field defines the checksum that is computed for the entire packet with the exception of the Checksum and End of Packet fields.
- End of Packet – This constant defines the end of the bootloader packet.

GAP Settings Tab

Note This tab is applicable for **PSoC 6** device only.

The GAP parameters define the general connection settings required when connecting Bluetooth devices. It 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.

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

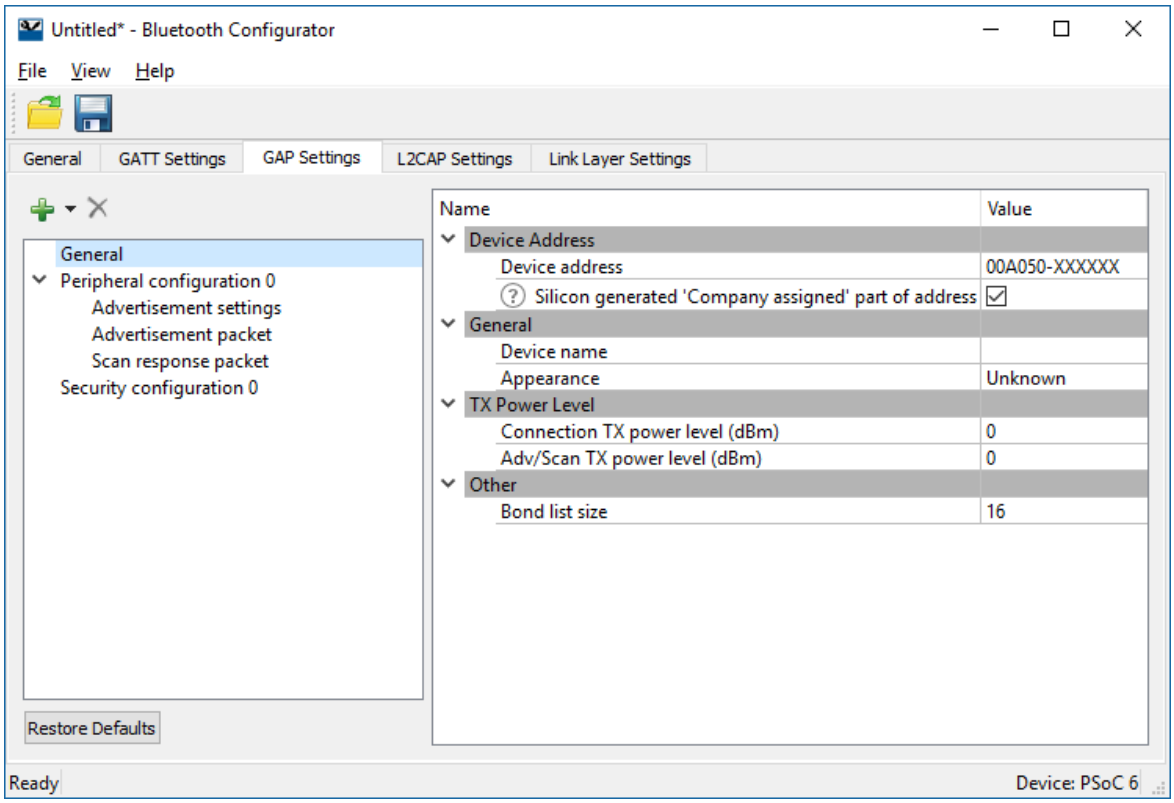
Toolbar

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

- **Add –** allows adding Peripheral, Central, Broadcaster, Observer, or Security configurations. 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 firmware.
- **Delete –** Deletes the selected Configuration.

GAP Settings Tab – General

This section contains general GAP parameters:



Public device address (Company ID – Company assigned)

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

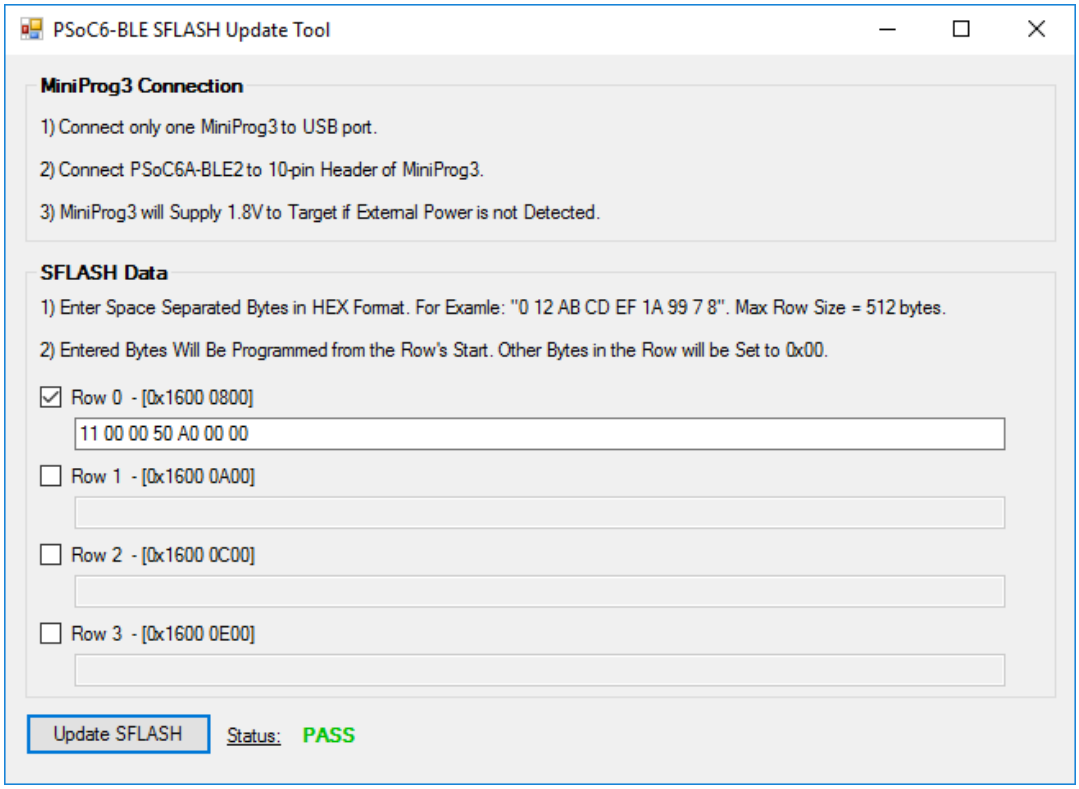
- The **"Company ID"** part is contained in the 24 most significant bits. It is a 24-bit Organization Unique Identifier (OUI) address assigned by IEEE.
- The **"Company assigned"** part is contained in the 24 least significant bits.

The address configured here is static and is designed to be used for development purposes only. During production, the device address should be programmed into the user's SFLASH location for the device address (row 0 of user SFLASH) via the SWD interface. Normally this address must be programmed only once during mass production, and then never changed in-field. However, user flash can be reprogrammed in-field many times.

During prototyping (FW design), the device address can be programmed into the user's SFLASH location using the MiniProg3 and the sample application installed in the following PSoC Programmer folder:

C:\Program Files (x86)\Cypress\Programmer\Examples\Misc\PSoC6-BLE2-SFLASH-Update\Executable

Enter the device address structure of type cy_stc_ble_gap_bd_addr_t in the Row 0 line to store it in the SFLASH.



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

This application is provided in the source code and can be used as a reference example for implementation in production programmers.

Silicon generated “Company assigned” part of device address

When checked, the “Company assigned” part of the device 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” option does not guarantee a unique device address. For mass production, Cypress strongly suggests that the device address be programmed into the user’s SFLASH location (row 0 of user SFLASH) via the SWD interface.

Device Name

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

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

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

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.

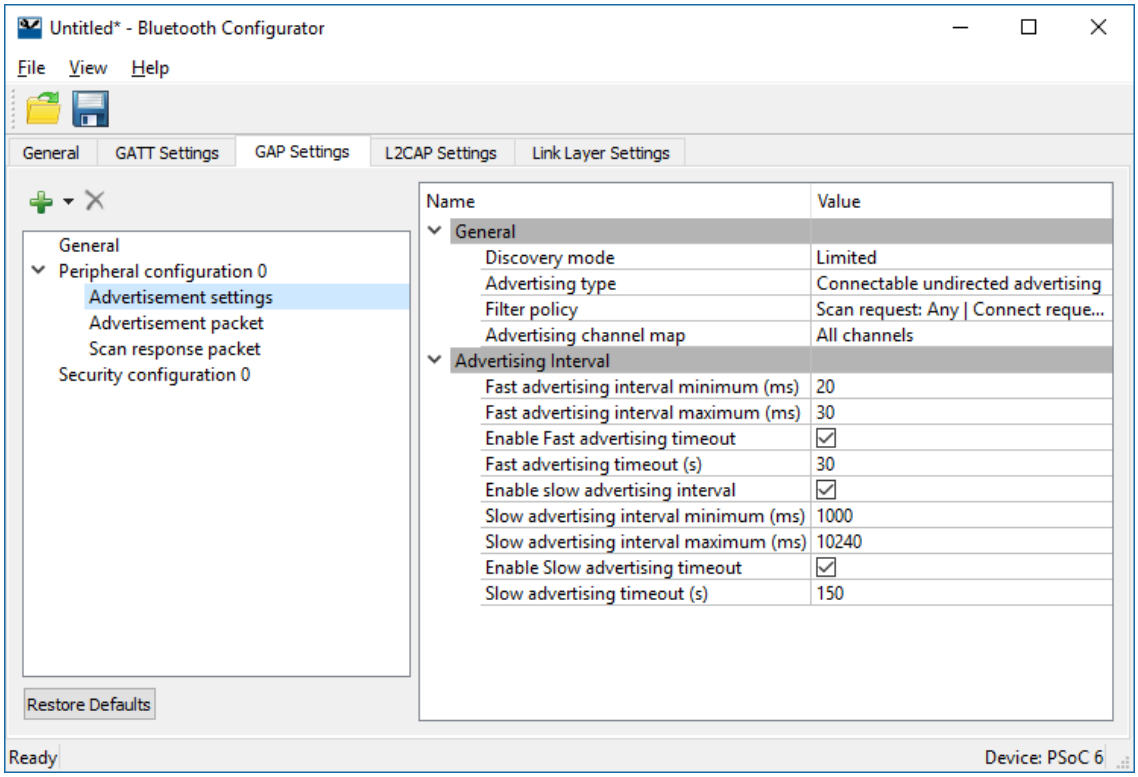
Bond list size

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

Note The maximum number of bonded devices is also limited by available Flash (emulated EEPROM area) size that will be consumed to store data. Consumed Flash size is calculated as multiple of number of supported services and multiple of number of supported bonded devices.

GAP Settings Tab – Advertisement Settings

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



Discovery mode

- **Non-discoverable** – In this mode, the device can't be discovered by a Central device.
- **Limited Discoverable** – This mode is used by devices that need to be discoverable only for a limited period of time, during temporary conditions, or for a specific event. The device which is advertising in Limited Discoverable mode are available for a connection to Central device which performs Limited Discovery procedure. The timeout duration is defined by the applicable advertising timeout parameter.
- **General Discoverable** – In this mode, the device should be used by devices that need to be discoverable continuously or for no specific condition. The device which is advertising in General

Discoverable mode are available for a connection to Central device which performs General Discovery procedure.

Advertising type

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

- **Connectable undirected advertising** – This option is used for general advertising of the advertising and scan response data. It allows any other device to connect to this device.
- **Scannable undirected advertising** – This option is used to broadcast advertising data and scan response data to active scanners.
- **Non-connectable undirected advertising** – This option is used to just broadcast advertising data.

Filter policy

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

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

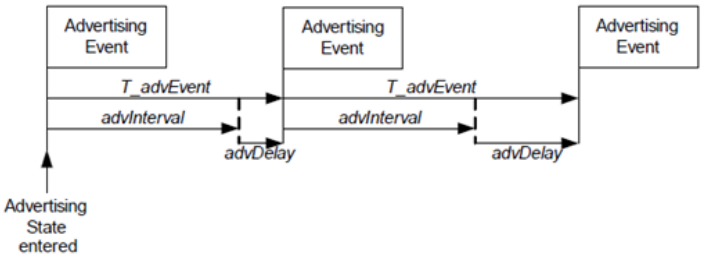
Advertising Interval

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.

Note In **General Discovery mode**, timeouts are not supported.

- **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 the data and establishing the LE Connection. The parameter is configured to increment in multiples of 0.625 ms. Valid range is from 20 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. 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. The timeout cannot occur before the advertising interval is expired, that is why if a timeout value is less than fast advertising interval minimum value, a warning is displayed. This parameter is not applicable in **General discovery mode**.
- **Slow advertising interval** – Defines the advertising interval for slow advertising. This is an optional parameter which, if enabled, allows to implement advertising with a lower duty cycle to save battery life. The Slow advertising interval parameters are applied to the device after the internal fast advertising interval timeout occurs. The minimum and maximum values defined using this parameter allow the BLE Stack to expect the 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. 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. Valid range is from 1000 ms to 10240 ms.
  - Timeout – The timeout value of advertising with slow advertising interval parameters. When equals 0, the device is advertising continuously. The timeout cannot occur before the advertising interval is expired, that is why if a timeout value is less than slow advertising interval minimum value, a warning is displayed.

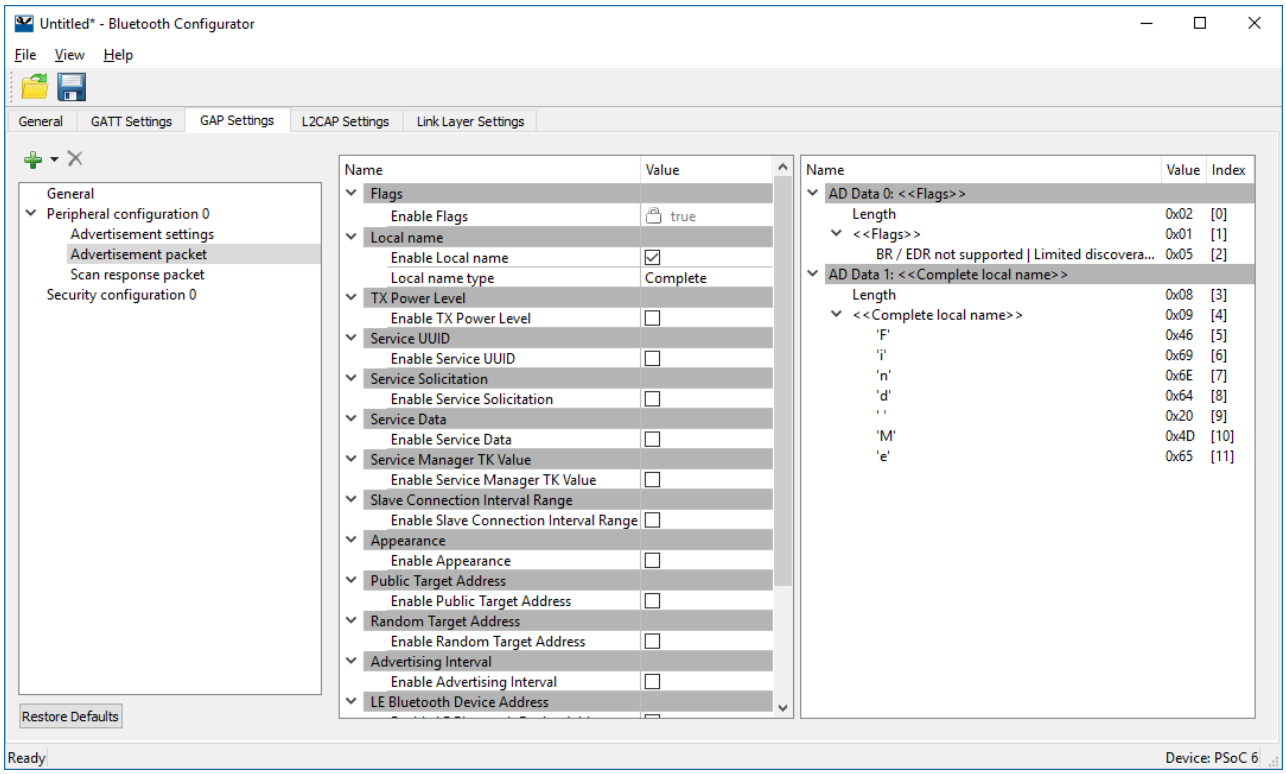


    - AdvDelay is a pseudo random delay 0-10 ms.

– The complete advertising Event consists of one advertising PDU sent on each of used advertising channels.

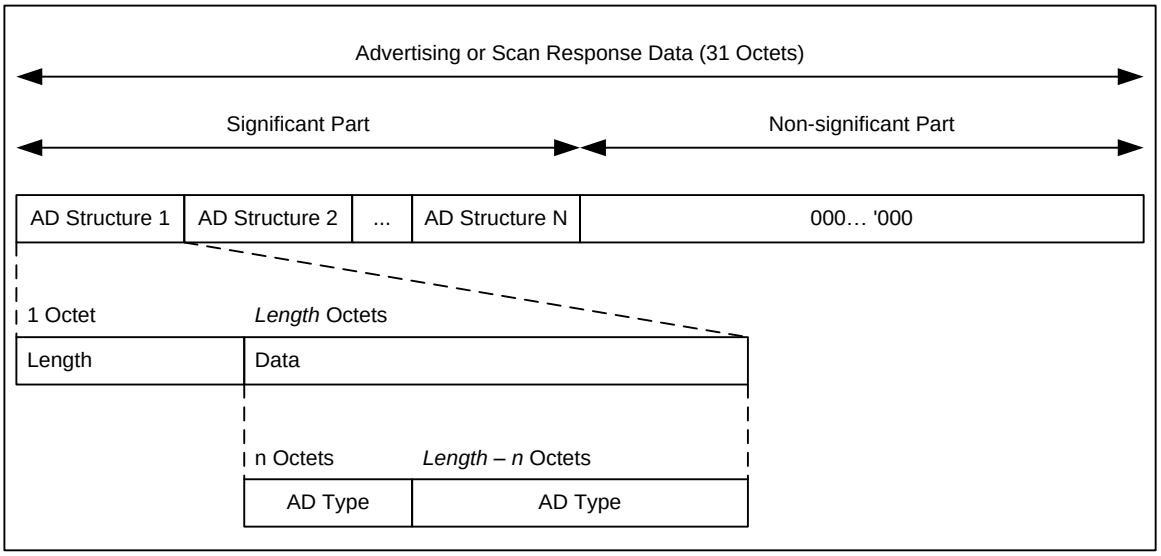
GAP Settings Tab – Advertisement packet

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



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** – The 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 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** field on the **GAP Settings** tab, under **General.**
Slave Connection Interval Range	Preferred connection interval range for the device. Not available in **Broadcaster** GAP role.
Service UUID	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.

AD Type	Description
Service Solicitation	List of Service UUIDs from the central device that the peripheral device would like to use. 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	Temporal key to be used at the time of pairing. Not available in **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	The Advertising interval value that is 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 **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	The data specified in the Indoor Positioning Service Specification. This is available when the Indoor Positioning Service is present in the Profile.

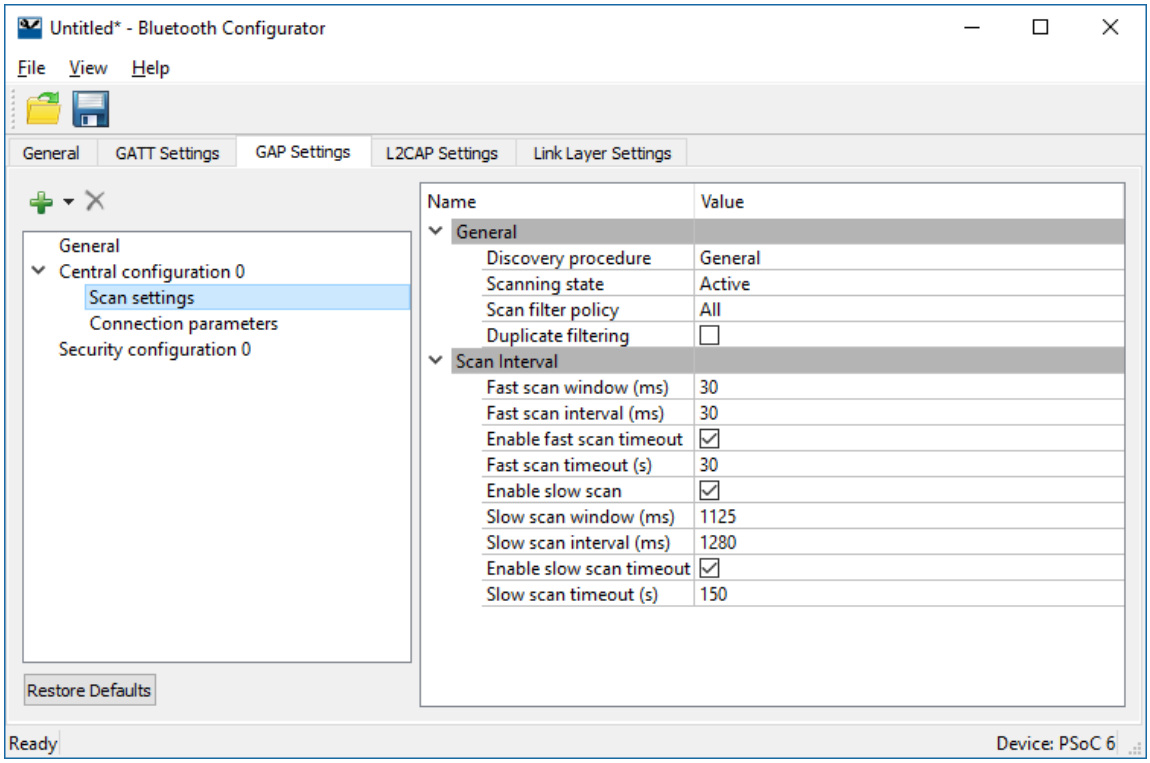
GAP Settings Tab – Scan Response Packet

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

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

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 performing this procedure will discover the device doing limited Discovery mode advertising only.
- **General** – A device performing this procedure will discover the device doing general and limited Discovery advertising.

Scanning state

- **Passive** – In this state, a device can only listen to advertisement packets.
- **Active –** In this state, a device may ask an 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.

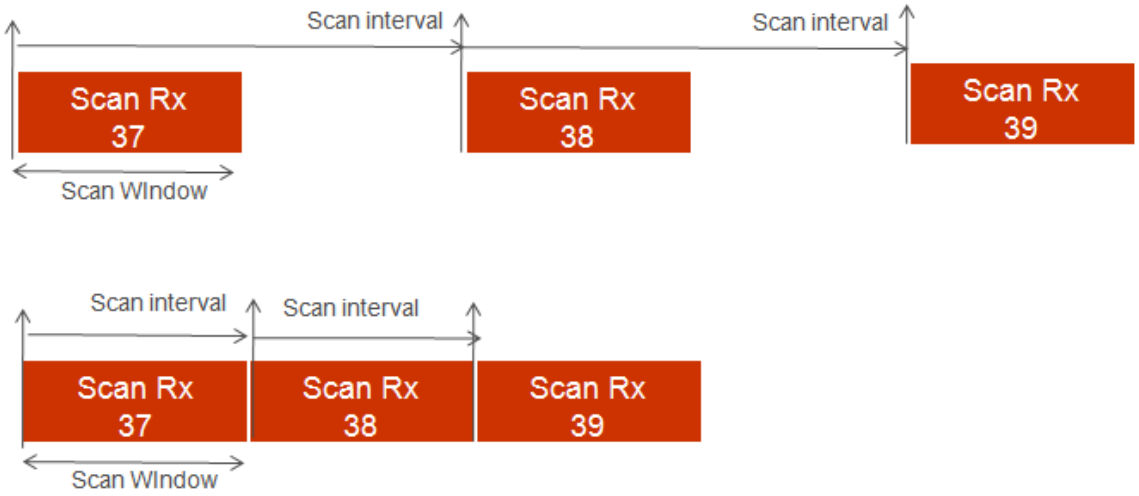
Duplicate filtering

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 8 address locations reserved to cache the previously seen advertiser devices and filter duplicate packets from them. If there are more than 8 advertising devices in the proximity of a scanner during the scan period, then the address storing buffer is exhausted. The firmware algorithm for overwriting the address cache buffer is implemented in FIFO fashion. When the scanner sees more than 8 advertisers, then 9th advertiser replaces the 1st one, 10th advertiser replaces the 2nd one, and so on in the address cache. After flushing the 1st advertiser from the address cache, if the scanner sees the first advertiser's ADV packet again, then it thinks that it is a new device (as 1st advertiser is no longer in the address cache) resulting in sending the 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 established, then the Profile switches to Slow scan parameters to save the battery life. After the **Slow scan timeout** value expires, 'CY_BLE_EVT_GAPC_SCAN_START_STOP ' event is generated. See API documentation.

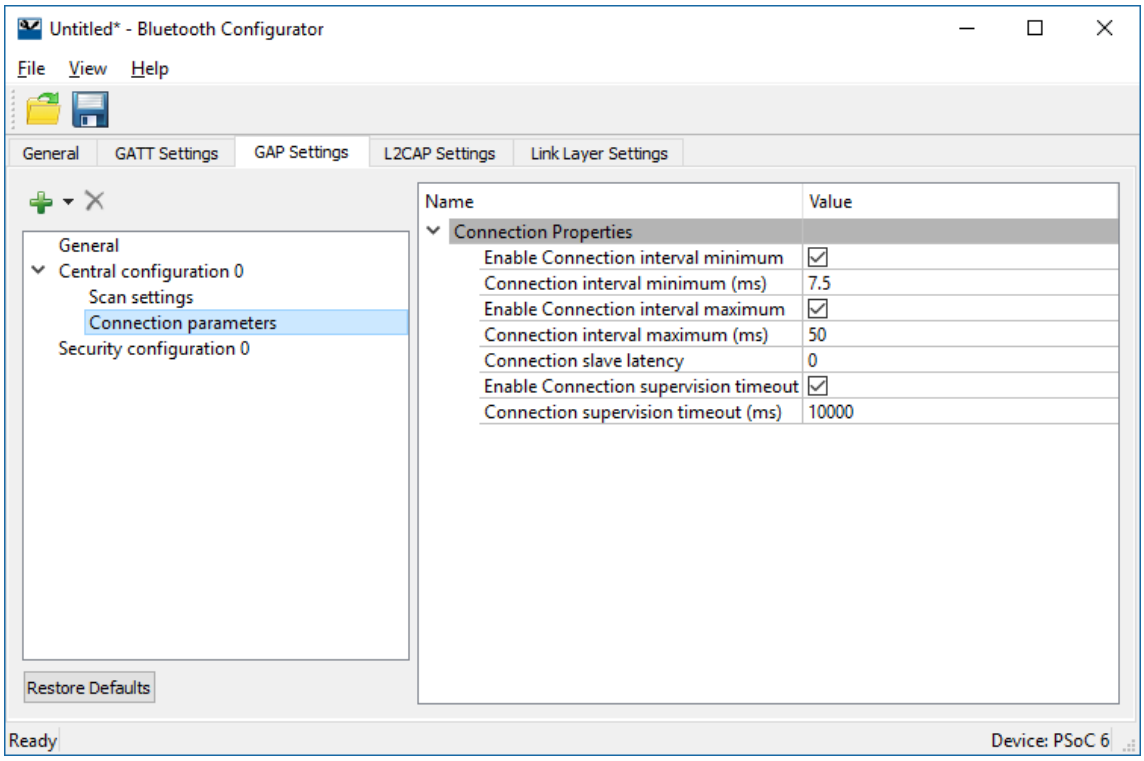


- **Fast scan parameters** – This connection type results in a faster connection between the GATT Client and Server devices than it is possible using a normal connection.
  - **Scan Window** – This parameter defines the scan window when operating in **Fast connection**. The parameter is configured to increment in multiples of 0.625 ms. 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. 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. The timeout cannot occur before the scanning interval is expired, that is why if a timeout value is less than slow scanning interval minimum value, a warning is displayed.

- **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. 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. 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. The timeout cannot occur before the scanning interval is expired, that is why if a timeout value is less than slow scanning interval minimum value, a warning is displayed.

GAP Settings Tab – Connection Parameters

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



Note The scaled values of these parameters 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 range is from 7.5 ms to 4000 ms. Not enabled means no specific minimum.
  - 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 range is from 7.5 ms to 4000 ms. Not enabled means no specific maximum.

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 = 24ms*

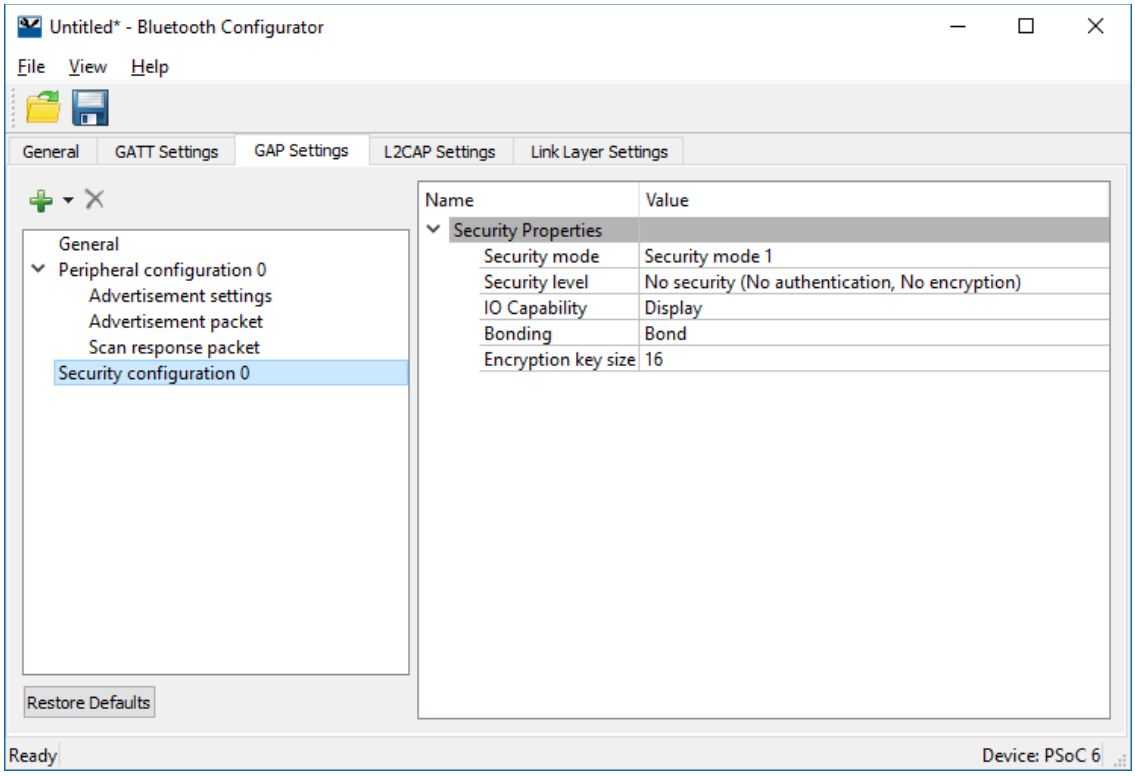
Note Connection intervals shorter than this value will still work, but under certain circumstances, the real time control procedures (connection update, channel map update etc.) with shorter update instance might result in a link disconnection.

- **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 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 in case of no response from peer device over the LE link. The time interval is configured in multiples of 10 ms. Not enabled means no specific value. The 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

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 GAP security modes. Both available modes may support authentication.

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

Security level

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

- If Mode1 is selected, then the following security levels are available.
  - No Security – With this level of security, the device will not use encryption or authentication.
  - Unauthenticated pairing with encryption – With this level of security, the device will send encrypted data after establishing a connection with the remote device.
  - Authenticated pairing with encryption – With this level of security, the device will send encrypted data after establishing a connection with the remote device. To establish a connection, devices should perform the authenticated paring procedure.
  - Authenticated LE Secure Connections pairing with encryption – With this level of security, 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.

- If Mode 2 is selected, then the following security levels are available.
  - Unauthenticated pairing with data signing – With this level of security, the device will perform data signing prior to sending it to the remote device after they establish a connection.
  - Authenticated pairing with data signing – With this level of security, the device will perform data signing prior to sending it to the remote device after they establish a connection. To establish a connection, the devices should perform 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 display capability and may display authentication data. GAP authentication is required.
- Display Yes/No – Used in devices with display and at least two input keys for Yes/No action. GAP authentication is required.
- Keyboard – Used in devices with numeric keypad. GAP authentication is required.
- No Input No Output – Used in devices that don't have 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 will store the link key of a connection after paring with the remote device in the flash memory and if a connection will be lost and re-established, the devices will use the previously stored key for the connection.

  Note Bonding information is stored in RAM and should be written to Flash if it needs to be retained during shutdown.
- **No Bonding** – The pairing process will be performed on each connection establishment.

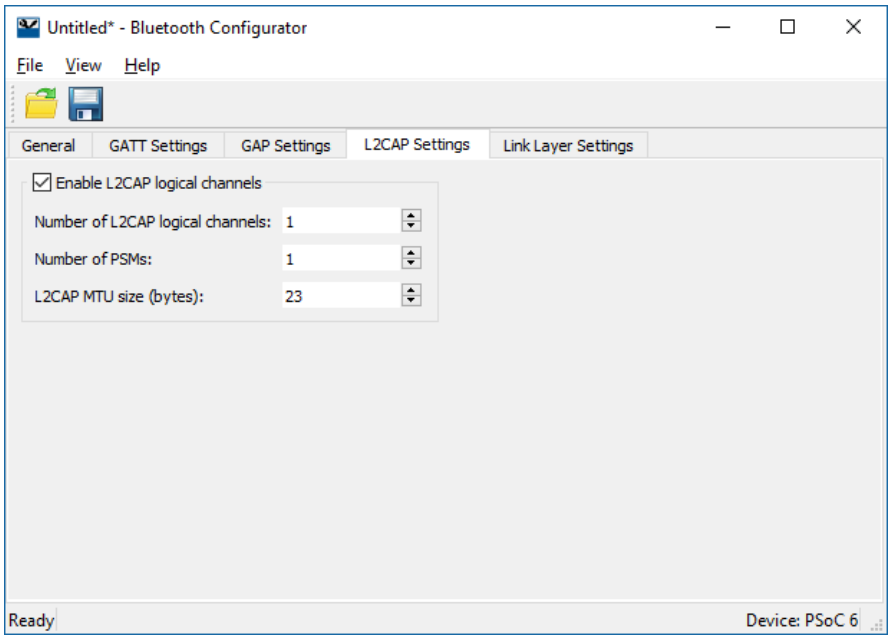
Encryption Key Size

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

Note This tab is applicable for **PSoC 6** device only.

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



Enable L2CAP Logical Channels

This parameter enables configuration of the L2CAP logical channels. Default: true.

Number of L2CAP Logical Channels

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

Number of PSMs

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

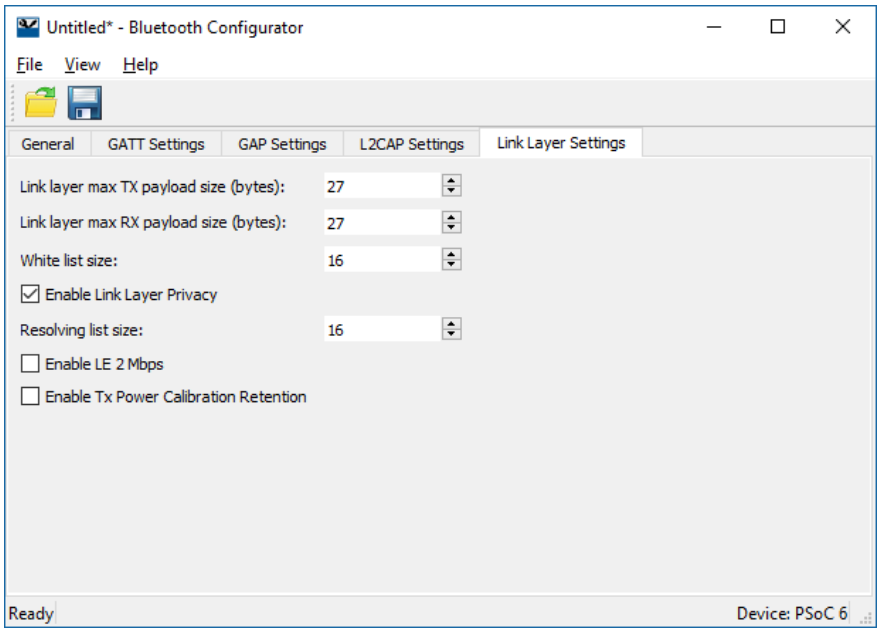
L2CAP MTU Size

This parameter defines the maximum SDU size of an L2CAP packet. 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

Note This tab is applicable for **PSoC 6** device only.

The Link Layer settings parameters for the Link Layer.



Link Layer Max Tx Payload Size

The maximum link layer transmits 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 CY_BLE_EVT_GAP_DATA_LENGTH_CHANGE event. Valid range is from 27 to 251 bytes.

Link Layer Max Rx Payload Size

The maximum link layer receives 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. Valid range is from 27 to 251 bytes.

Setting the 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. The 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. The default – 16.

Enable LE 2 Mbps

Enables LE 2 Mbps feature of Bluetooth 5.0.

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

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

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, BLE radio Tx power calibration is performed only once after programming and the calibration values are retained in 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, make sure 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 the user's row 0 (after BLE_DEVICE_ADDRESS) of the SFLASH location.

References

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

- Device Configurator Guide
- ModusToolbox IDE User Guide
- Cypress WICED Bluetooth API Reference Guide
- 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 WICED Bluetooth support.
	Fixed the issue with code generation for GAP Central role connection parameters. Their values were always generated as default.

© Cypress Semiconductor Corporation, 2018-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, ModusToolbox, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.