

ModusToolbox™ BSP Assistant user guide

ModusToolbox™ tools package version 3.0.0

BSP Assistant version 1.0

About this document

Scope and purpose

The Board Support Package (BSP) Assistant helps you to create and manage custom BSPs for ModusToolbox™ applications using either a GUI or a command line interface (CLI).

Intended audience

This document helps application developers understand how to use the BSP Assistant tool.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
<i>Italics</i>	Denotes file names and paths
<code>Courier New</code>	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- BSP – board support package
- CLI – command line interface

Reference documents

Refer to the following documents for more information as needed:

- [Eclipse IDE for ModusToolbox™ user guide](#)
- [Device Configurator user guide](#)
- [ModusToolbox™ tools package user guide](#)

Table of contents

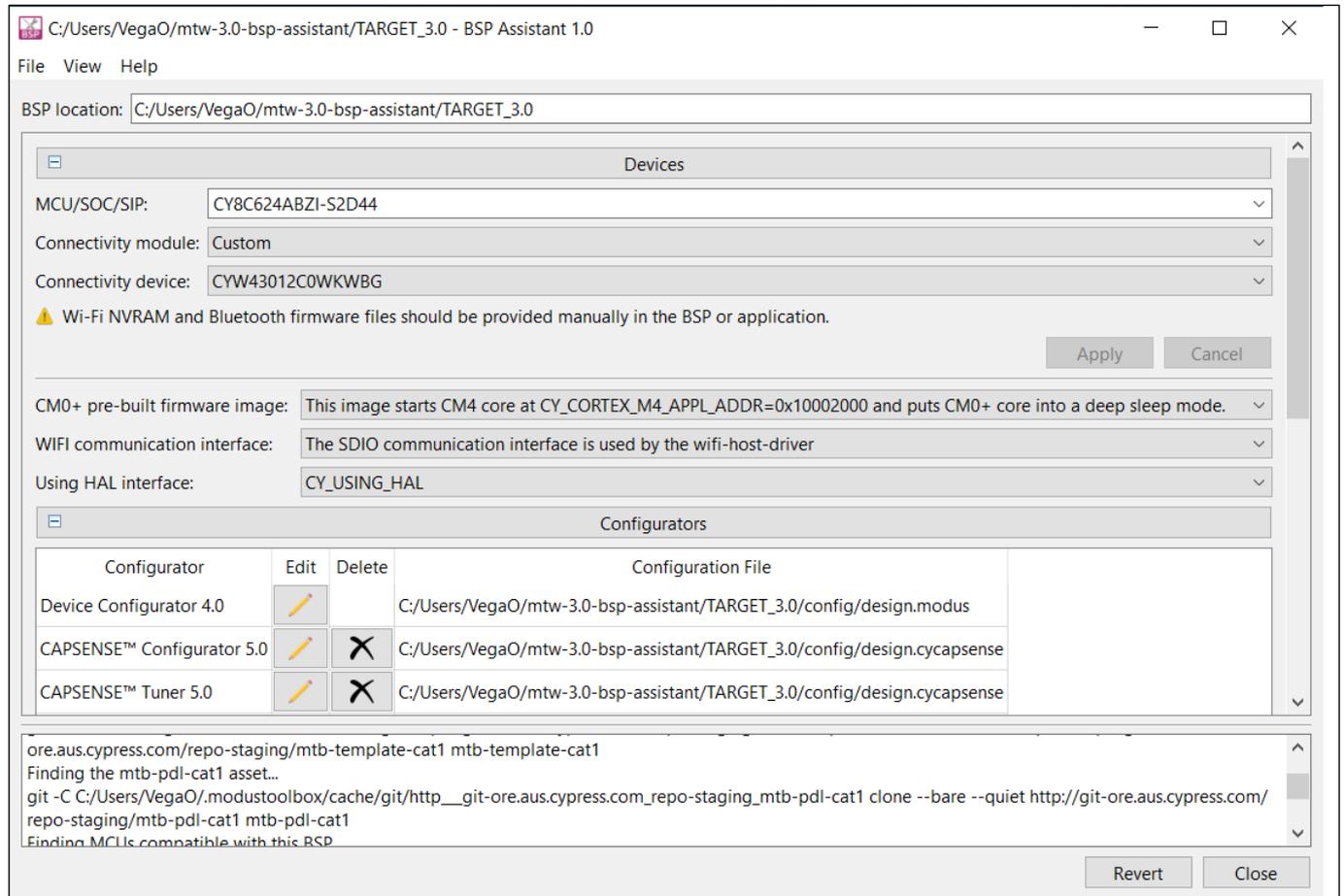
Table of contents

- 1 Overview..... 3**
- 2 Launch the BSP Assistant 4**
 - 2.1 make command4
 - 2.2 Eclipse IDE4
 - 2.3 Stand-alone GUI4
 - 2.4 CLI executable5
- 3 Quick start..... 6**
 - 3.1 BSP Assistant use cases6
 - 3.2 Basic workflow6
 - 3.3 Opening existing BSP7
 - 3.4 Creating new BSP7
 - 3.5 Editing the BSP8
 - 3.6 Confirm Cancelling Edits.....9
 - 3.7 Viewing changed files.....10
 - 3.8 Show Diff.....10
 - 3.9 Updating launch settings10
- 4 GUI Description11**
 - 4.1 Menus.....11
 - 4.2 BSP location11
 - 4.3 BSP Sections.....11
 - 4.4 Messages.....15
 - 4.5 Close and Revert buttons.....15
- 5 CLI Description.....16**
- 6 Version changes18**

Overview

1 Overview

All ModusToolbox™ applications require a target BSP. Infineon provides BSPs for all of our kits, as well as generic BSPs for each chip architecture, to use as a starting point. When working with your own hardware, you can modify an Infineon BSP to match that hardware. The BSP Assistant helps to simplify that process.



When you change device information, click **Apply** to submit the changes. When you make edits to a BSP that requires other changes (including linker scripts, startup code, and so on), the BSP Assistant automatically makes those changes. You can click **Revert** to undo all changes before closing the tool.

Note: WICED_BTSDK BSPs and ModusToolbox™ version 2.x BSPs are not supported by the BSP Assistant at all. In addition, the following features are not supported in the first release of the BSP Assistant:

- Modular BSPs
- Multi-MCU BSPs (The MCU explicitly named in the bsp.mk file is supported, but the other MCU is not changeable using the BSP Assistant.)

Launch the BSP Assistant

2 Launch the BSP Assistant

There are numerous ways to launch the BSP Assistant, depending on how you choose to use the various tools in ModusToolbox™ software.

2.1 make command

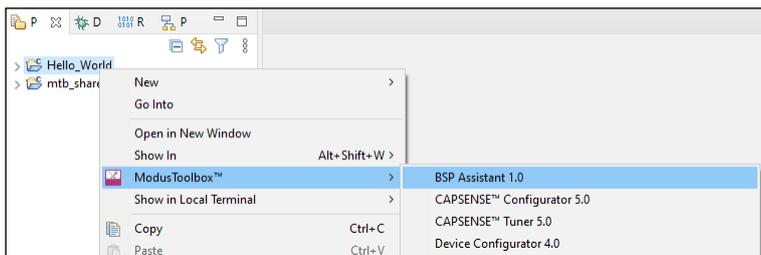
As described in the [ModusToolbox™ tools package user guide](#) Build System chapter, you can run numerous make commands in the application directory, including launching the BSP Assistant. After creating a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

```
make bsp-assistant
```

When launched this way, the bsp-assistant opens with the active BSP for the application.

2.2 Eclipse IDE

If you use the Eclipse IDE for ModusToolbox™, you can launch the BSP Assistant for the selected application. In the Project Explorer, right-click on the project and select **ModusToolbox™ > BSP Assistant <version>**. You can also click the **BSP Assistant <version>** link in the IDE Quick Panel.



Similar to the make command method, launching the BSP Assistant using the Eclipse IDE opens the tool for the selected application. Refer to the [Eclipse IDE for ModusToolbox™ user guide](#) for details about the IDE.

2.3 Stand-alone GUI

You can launch the BSP Assistant GUI as a stand-alone GUI by running its executable as applicable for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, it is installed here:

```
<install_dir>/ModusToolbox/tools_<version>/bsp-assistant-<version>
```

When you launch the GUI this way, it first loads the technology packs, tools information, and manifest data. Once that's done, the BSP Assistant opens without a BSP. You can then either open an existing BSP or create a new one.

You can launch the BSP Assistant GUI tool without a BSP from the command line using:

```
bsp-assistant
```

Launch the BSP Assistant

You also have the choice to add optional arguments, for example:

```
bsp-assistant --bsploc <bsp-location> --offline
```

In this example, the BSP Assistant GUI tool opens with the specified BSP. The `--bsploc` option is used to specify a directory containing an existing BSP to load.

The `--offline` option specifies that offline content should be used. Refer to the [ModusToolbox™ tools package user guide](#) for more details about offline content.

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

2.4 CLI executable

Additionally, there is a "cli" version of the executable that does not open the GUI. Running the cli executable from the command line can be useful as part of batch files or shell scripts. The exit code for the executable is zero if the operation is successful, and non-zero if the operation encounters an error. See [CLI Description](#) for more details.

Quick start

3 Quick start

3.1 BSP Assistant use cases

The following provide different scenarios for using the BSP Assistant:

- **Modify BSP for existing application:** You want to experiment with changes to a BSP for an application created from a code example. Open the BSP from the application, make and save changes, and then re-build the application. Changes will only apply to the current application and not to any others.
- **Modify BSP without application:** You want to modify a BSP independent of an application. Open the existing BSP, make and save changes, and then you can optionally export the BSP to a zip file. Once edited and/or exported, import the BSP using the Project Creator or Library Manager. Any applications that reference the BSP from the saved location will see the updated BSP. These applications need to be updated through Library Manager or by running `make getlibs` to see any dependency changes.
- **Create new BSP:** You have a new piece of hardware and want to create a BSP for it. Create a new BSP, make and save changes, and then you can optionally export the BSP to a zip file. Once edited and/or exported, import the BSP using the Project Creator or Library Manager. You may also use this as a starting point to create additional BSPs using the BSP Assistant.

3.2 Basic workflow

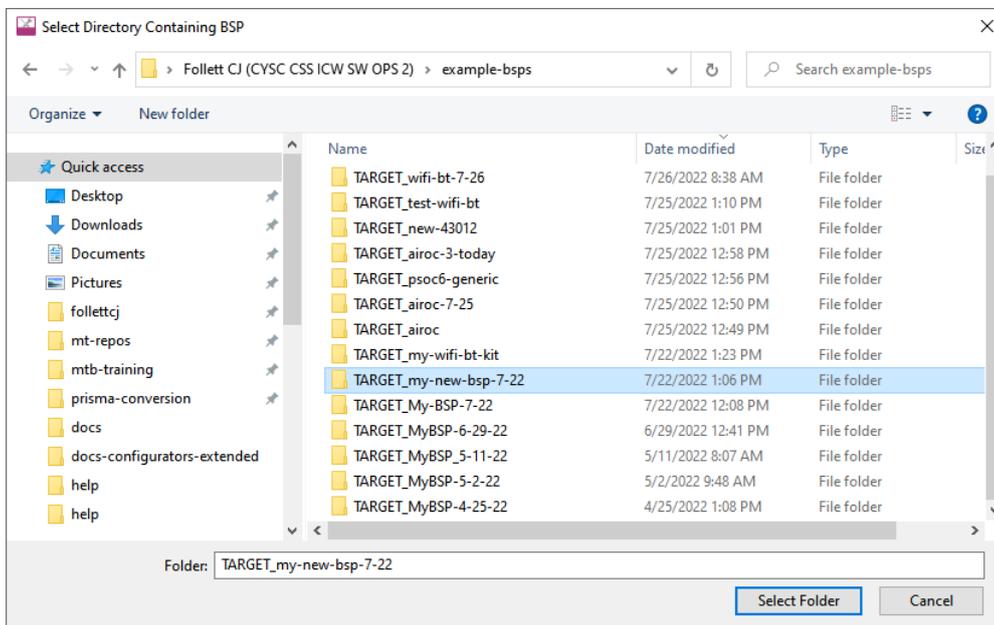
The following is the basic workflow for using the BSP Assistant:

1. [Launch the BSP Assistant](#), and either open an existing BSP or create a new BSP.
 - An existing BSP may be in an application or you may browse to a location independent of an application.
 - Create a new BSP by selecting one from the standard or custom manifest, or by browsing to the location of a starting BSP.
2. Edit the various areas of the BSP, including:
 - Devices
 - Configurators
 - Dependencies
 - Capabilities
3. Save changes, and optionally export the BSP.

Quick start

3.3 Opening existing BSP

On the main BSP Assistant window, click the **Open** link or select **File > Open** to access the standard Open dialog. Then, navigate to the directory that contains the desired BSP, and click **Select Folder**.

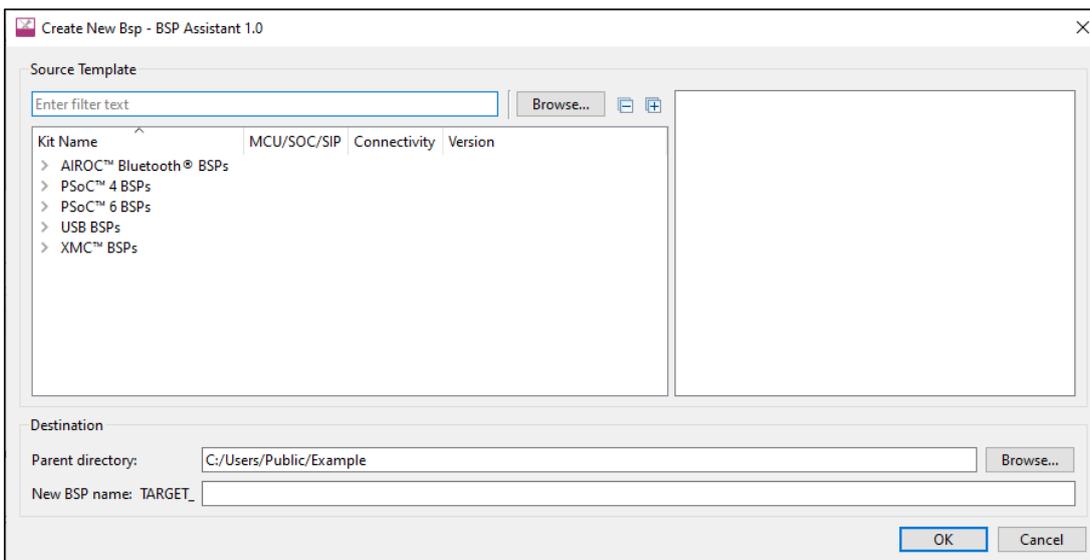


Note: If the specified BSP has a .git directory that points to an Infineon repository, a warning dialog will indicate that you might want to use the New... dialog to import this BSP into a new location that is not connected to Infineon's Git repositories before customizing it.

3.4 Creating new BSP

1. On the main BSP Assistant window, click the **create** link or select **File > New** to open the New BSP dialog.

This dialog is similar to the ModusToolbox™ Project Creator tool in that it provides a list of BSPs for Infineon kits and generic devices. Refer to the [Project Creator user guide](#) for more details about that tool.



2. Select a BSP that closely matches what you intend to use for your design.

Quick start

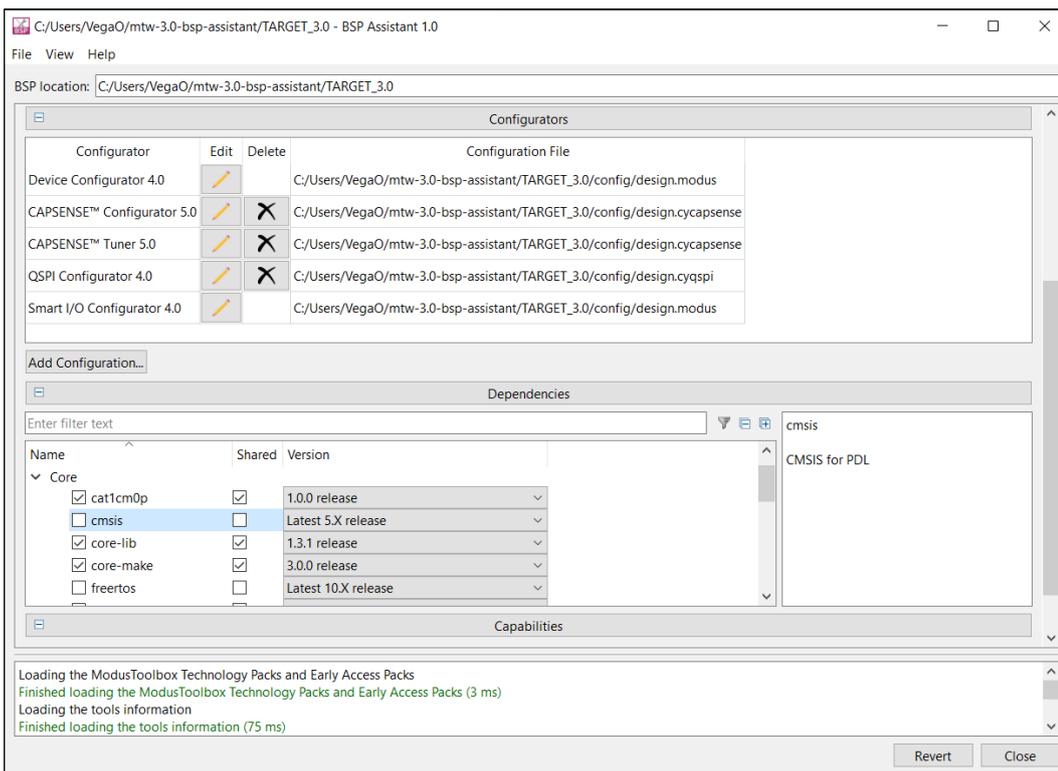
3. Alternatively, click **Browse** in the Source Template section and then navigate to and select the BSP you wish to import.
4. Enter the **Parent directory** location, or click **Browse** in the Destination section and navigate to the location where the BSP will be saved.
5. Type a name for the BSP in the **New BSP name** field.

Note: To prevent copy-and-paste errors, an error message displays if you enter "TARGET_" in the field so that you do not mistakenly enter "TARGET_TARGET_".

3.5 Editing the BSP

After opening or creating a BSP, the GUI shows various sections with settings, including:

- Devices
- Configurators
- Dependencies
- Capabilities



The following are brief descriptions of these sections. See [BSP Sections](#) for more details.

3.5.1 Devices

The Devices section includes various details about the BSP hardware. These settings vary for different types of devices and BSPs.

Select an option for changing the desired files. Click **Apply** to save the changes or **Cancel** to return to the previous selections. Once you click **Apply**, the tool makes all necessary changes to the BSP makefile, linker scripts, startup code files, etc.

Quick start

3.5.2 Configurators

The Configurators section shows the current configurations available for the BSP. Any changes you make to this section are saved automatically.

- Click the **Edit** icon to open the applicable configurator. Refer to that configurator's documentation for details about using it.
- Click the **Delete** icon to remove the selected configuration file from the BSP.
- To add a new configuration to the BSP, click the **Add Configuration...** button. This will open the applicable configurator. Once you save your settings from that configurator, it will appear in the list in the BSP assistant.

3.5.3 Dependencies

The Dependencies section shows all of the libraries currently used by the BSP. Any changes you make to this section are automatically saved. You can remove a library, change a library from shared to local, and select a different version.

Enable the **Filter**  button to show only the dependencies that are already included in the BSP.

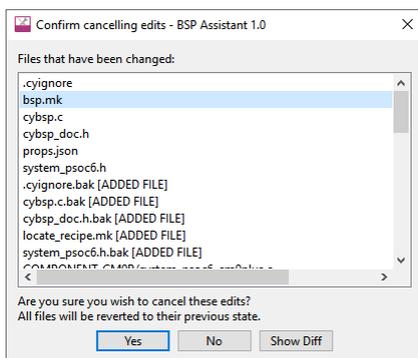
3.5.4 Capabilities

This section contains a read-only list of capabilities for the BSP. The list indicates the capabilities that the BSP can provide. Applications and libraries specify which capabilities they require, so this list is used to determine which applications and libraries are compatible with the BSP.

To update this list, you must edit the BSP's *props.json* file.

3.6 Confirm Cancelling Edits

When you click **Revert**, a dialog displays a list of all the files that have been modified since you opened the tool for this session or created the current BSP. You can view the differences in any of the files before accepting or cancelling the revert operation.

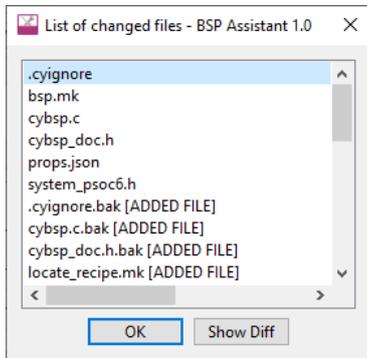


If you click **Yes**, the BSP will be restored to the state when you opened the tool for this session or created the current BSP. You cannot undo a revert operation.

Quick start

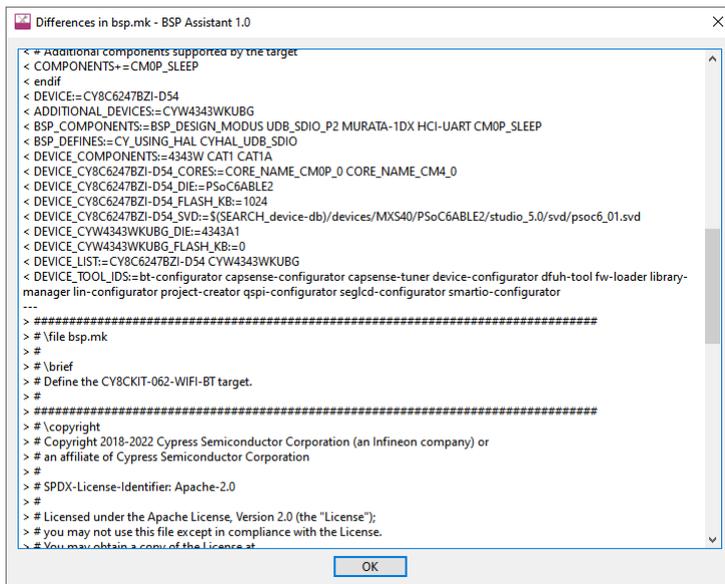
3.7 Viewing changed files

Another way to view a list of files with various changes us by selecting **View > List Changed Files...**



3.8 Show Diff

To open a diff tool that displays the changes made to a specific file, double-click a file or select a file and click **Show Diff**.



This dialog shows the output of the "diff" command, comparing the file as it existed when the BSP Assistant opened the BSP to the file as it exists now. Deleted lines include the "<" character while added lines include the ">" character.

3.9 Updating launch settings

If you are using an IDE, you may need to regenerate the launch settings to reflect the new BSP. Use the appropriate command(s) for the IDE(s) that are being used. For example: `make vscode` for VS Code or click the **Generate Launches...** link in the Quick Panel for Eclipse. This step must be done whenever the target device is changed since the launch settings may change from device to device.

GUI Description

4 GUI Description

The BSP Assistant GUI contains various menus and fields used to create and modify BSPs.

4.1 Menus

4.1.1 File

- **New** – Opens a New BSP dialog to create a new BSP; see [Creating a new BSP](#).
- **Open** – Opens an existing BSP; see [Opening an existing BSP](#).
- **Export as ZIP file...** – This opens a standard save file selection dialog for the host operating system, and if a file name is selected, a ZIP file containing the BSP as it is at that moment will be created at that location.
- **Open in System Explorer** – This opens your computer’s file explorer tool to the folder that contains the BSP.
- **Exit** – Closes the BSP Assistant.

4.1.2 View

- **List Changed Files** – This opens a dialog to display a list of changed files; see "Viewing changed files."

4.1.3 Help

- **View Help** – Opens this document.
- **About** – Opens the **About** box for version information, with links to open Infineon.com and the current session log file.

4.2 BSP location

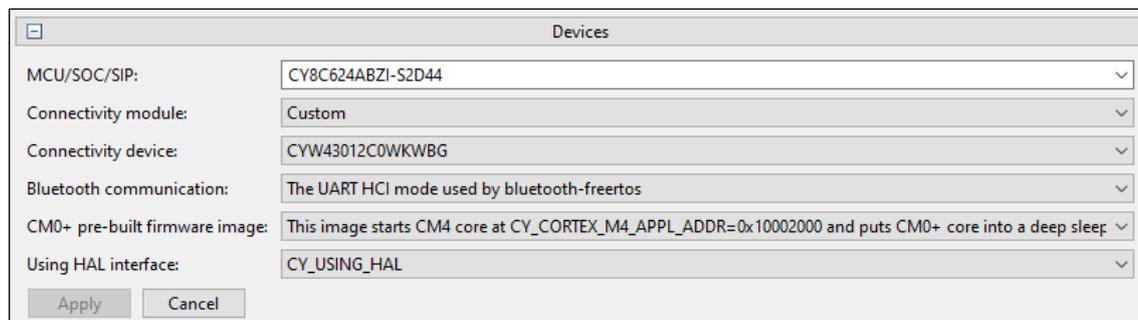
The "BSP location" field is not editable, and contains the location of the currently-loaded BSP.

4.3 BSP Sections

These are various sections to edit and view the current BSP's information.

4.3.1 Devices

The Devices section has selections for the MCU/SOC/SIP and connectivity module or device on the board, as well as any other settings specific to the MCU/SOC/SIP.



Each available setting has a pull-down menu to select a different option. Click the **Apply** button to submit the change. Click **Cancel** to undo the changes.

GUI Description

4.3.1.1 MCU/SOC/SIP

The **MCU/SOC/SIP** pull-down menu contains all of the central devices that are compatible with the category of the current BSP (this is expected to be bounded by chip architecture: CAT1A vs. CAT1B vs. CAT2 etc.). You can also search for such devices by typing into the field. If you change MCU/SOC/SIP significantly, a warning will indicate that you should run the Device Configurator and fix any errors displayed there. Refer to the [Device Configurator user guide](#) for more details about that tool.

The MCU/SOC/SIP can be changed to any device within the same family. If you want a device in a different family, you must first create a BSP for a device in that family.

4.3.1.2 Connectivity module

The values displayed in the **Connectivity module** field depend on the selected MCU/SOC/SIP.

- If the MCU/SOC/SIP has built-in connectivity, this field will be greyed out. Information from the database specifying the built-in connectivity hardware will be displayed.
- Otherwise, this pull-down menu will contain the value "None", a list of all connectivity modules compatible with the central device, and the value "Custom..."

The connectivity module can be set to "None," any module that is supported for the selected MCU/SOC/SIP, or "Custom." In the case of "Custom," you must then specify the exact connectivity device that you are using and you must manually supply the required Bluetooth® and/or Wi-Fi firmware files for your device and hardware in your BSP or application. If you choose a supported module, the appropriate files will be automatically provided.

4.3.1.3 Connectivity device

The **Connectivity device** pull-down menu will display as follows:

- If the module is "None," then this menu will be empty and greyed out.
- If the module is set to a specific module, then this menu will be greyed out but will have the name of the connectivity device in the module.
- If the module is "Custom...," then this menu will contain a list of all connectivity devices compatible with the central device.

4.3.1.4 Other settings

The remaining rows in the Devices section are options that are specific to the MCU/SOC/SIP and connectivity selections that you have made. These are used to select the settings required to get the desired behavior of the devices. For example, "CM0+ pre-built firmware image" is used to select the firmware that is loaded onto a PSoC™ 6 MCU for the CM0+ core.

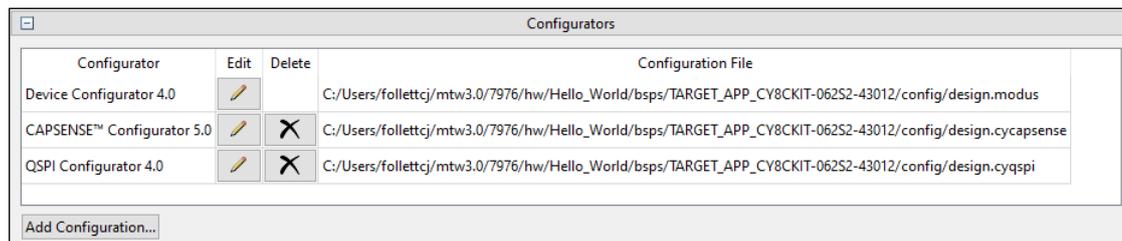
Note: The "CM0+ pre-build firmware image" setting is used for single-core applications. If the current BSP is for a multi-core application, the Makefile uses a `DISABLE_COMPONENTS` option to disable the default CM0+ image, and therefore overrides this setting.

GUI Description

4.3.2 Configurators

The Configurators section allows you to edit, remove, or add BSP configuration settings for the selected MCU/SOC/SIP and connectivity device. The list shows all configurators that currently have files in the BSP. You can click the **Edit** button to open the associated configurator to make changes or click the **Delete** button to remove a configuration file (some configurators are required and files cannot be removed).

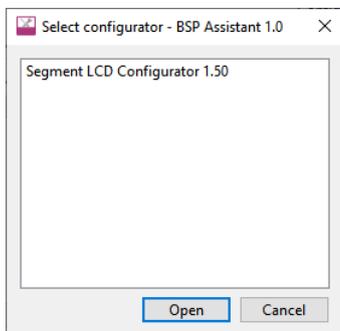
Note: The BSP assistant only operates on BSP configurator files. Library configurators such as the Bluetooth® configurator store their files in the application rather than the BSP, so they are not relevant to the BSP Assistant. For descriptions of the various configurators, refer to the [ModusToolbox™ tools package user guide](#).



Click the **Edit** icon to open the applicable configurator. Refer to that configurator's documentation for details about using it.

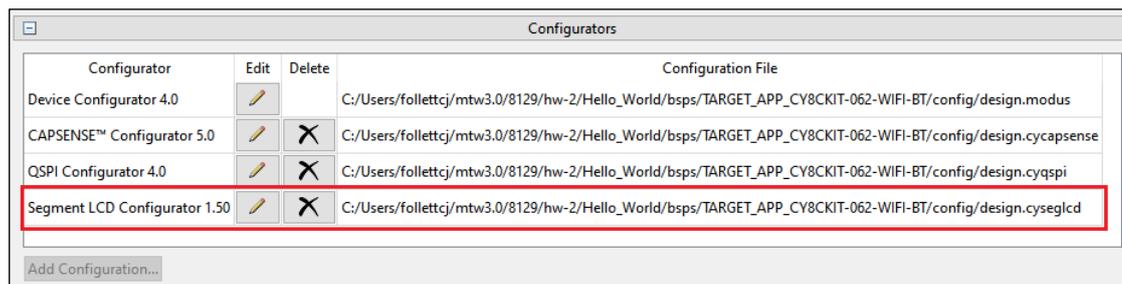
Click the **Delete** icon to remove the selected configuration file from the BSP.

To add a new configuration to the BSP, click the **Add Configuration...** button. This opens the Select configurator dialog.



Select the desired configurator from the list and click **Open**.

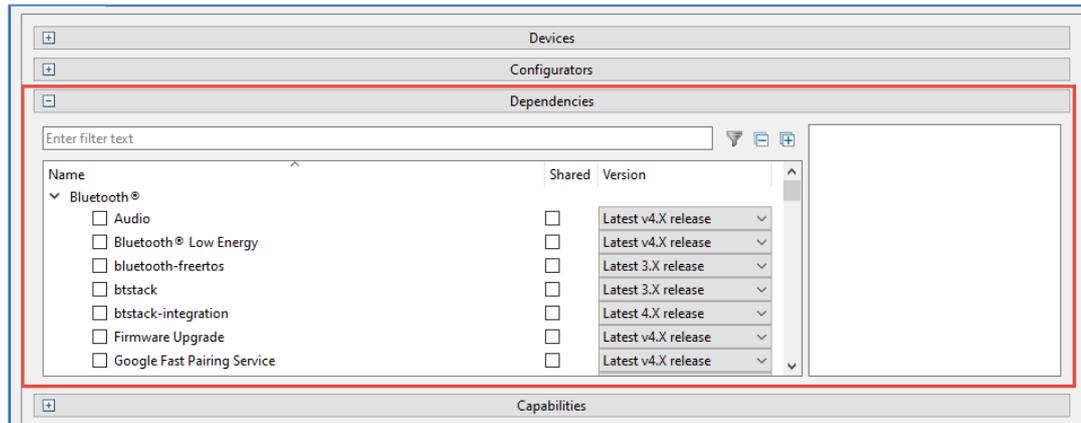
When you click **Open**, the associated configurator opens. Once you save changes from that configurator, it will appear in in the BSP Assistant list.



GUI Description

4.3.3 Dependencies

The Dependencies section shows all of the libraries that the BSP needs to operate along with any other libraries that can be added. When you first create a BSP, it will list all of the libraries that are required for your BSP, such as the core libraries, recipe libraries, PDL, and HAL. You can change the version of any dependency and the location where it will be placed for any applications that use the BSP. The location is either shared (i.e. in `mtb_shared`) or local (i.e. in the application's `libs` directory).



Select the **Filter** button to show only those libraries that are currently listed as dependencies. You can also use the filter box to search for libraries or the plus and minus buttons to expand or collapse all categories.

Note: The default versions for the required libraries are the latest release versions at the time the BSP was created. If you want the libraries to be set to the latest versions when the BSP is used to create an application, you can change the versions to "Latest V.X release" where "V" is the major version number.

Note: When you add additional dependencies, the default is "Latest V.X release" where "V" is the major version number. This means that those libraries will be set to the latest versions when the BSP is used to create an application. If you want fixed versions in the BSP, you can use the drop-down to select the release version that you require.

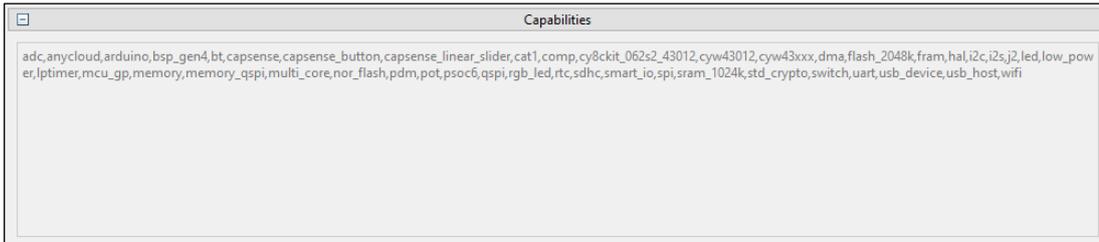
Note: If you modify the dependencies in a BSP, any application(s) that use the BSP must be updated to include those dependencies. This can be done by running the Library Manager and clicking **Update** or by running `make getlibs` from the application's root directory on the command line.

GUI Description

4.3.4 Capabilities

The Capabilities section is a read-only list of the capabilities that the BSP provides. This list is used by the Project Creator to show template applications (i.e. code examples) that are compatible with your BSP and by the Library Manager to show libraries that are compatible.

That is, the BSP contains "provided capabilities" while the code examples and libraries contain "required capabilities." The Project Creator and Library Manager match the contents of the provided and required capabilities lists to determine what to show.



If you want to modify the provided capabilities in a BSP, you can open the *props.json* file from the BSP. The capabilities are in the section **opt > props > capabilities**. For example:

```
"opt": {
  "props": {
    "capabilities": [
      "adc",
      "anycloud",
      "arduino",
      ...
    ]
  }
}
```

4.4 Messages

The text area at the bottom of the window displays progress and error messages from the system.

4.5 Close and Revert buttons

The BSP Assistant provides **Close** and **Revert** buttons, described as follows:

- **Close** – The BSP Assistant saves changes made to the Configurators and Dependencies sections automatically as you go. To save changes in the Devices section, you must click **Apply** in that section. If you click **Close** with changes already saved/applied, the BSP Assistant tool will quit. If there are unsaved changes, a dialog displays asking if you want to apply the changes.
- **Revert** – If you decide to back out all of your changes before closing the BSP Assistant (including applied changes in the Devices section), click the **Revert** button to review files that were changed and confirm the cancellation of the edits.

CLI Description

5 CLI Description

As noted previously, the BSP Assistant also includes a CLI executable to run various commands in a terminal window instead of a GUI. To see a list of the various arguments, run the following:

```
bsp-assistant-cli -h
```

As an example, the following command and argument lists all the MCUs that contain "XMC7":

```
bsp-assistant-cli list_mcus --filter XMC7 --offline
```

The `--offline` option specifies that offline content should be used. Refer to the [ModusToolbox™ tools package user guide](#) for more details about offline content.

The following sections describe each of the arguments available for the BSP Assistant CLI:

change_connectivity <bsp-location> <module-or-device-name>

This argument adds or changes the connectivity module or device specified by the BSP. It displays an error in the following cases:

- If `bsp-location` is not the location of a BSP.
- If `module-or-device-name` is not the name of a connectivity module or connectivity device present in the device database, or if it is incompatible with the MCU/SOC/SIP currently selected in the BSP.

change_mcu <bsp-location> <mcu-name>

This argument changes the MCU for the BSP exactly as if you were changing the **MCU/SOC/SIP** field in the GUI. It displays an error in the following cases:

- If `bsp-location` is not the location of a BSP.
- If `mcu-name` is not the name of an MCU, SOC or SIP present in the device database, or is incompatible with the currently-loaded BSP.

create <kit-name> <parent-dir> <bsp-name> [<version>]

This argument creates a new BSP at the specified location from the specified version of the specified board URI, exactly as using the **File > New...** menu option in the GUI with the appropriate information. It displays an error in the following cases:

- If `kit-name` does not match the name or ID of a BSP listed in the board manifests.
- If `parent-dir` does not exist.
- If `bsp-name` begins with "TARGET_".

If you provide the `version` option, it must be used as follows:

- A quoted user-facing version string (e.g., "1.2.0 release"), or
- A commit name (e.g., release-v1.2.0), or
- The special string "latest", which indicates that the user wants the latest version which specifies a specific version.

Strings such as "Latest 2.X release" and "latest-v2.X" result in an error. If you do not specify `version`, then the special string "latest" is assumed.

CLI Description

delete_configuration <bsp-location> <configuration-name>

This argument deletes a configuration file from the BSP. It displays an error if `bsp-location` is not the location of a BSP.

The `configuration-name` can be either a configurator name, such as `qspi-configurator`, or a base configuration file name, such as `design.cyqspi`.

import <source-bsp-location> <parent-dir> <bsp-name>

This argument imports an existing BSP and creates a new BSP based on it. It displays an error in the following cases:

- If `source-bsp-name` is not the location of a BSP.
- If `parent-dir` does not exist.
- If `bsp-name` begins with "TARGET_".

list_connectivity [--filter <filter>] [<bsp-location>]

This argument lists all of the connectivity modules and devices that are legal arguments, each on its own line.

- If you provide the `--filter` option, the output will only contain devices that match the filter.
- If you specify the `bsp-location` option, the output lists only connectivity modules and devices compatible with the specified BSP.

list_kits [--filter <filter>]

This argument lists all of the legal values for `kit-name`, each on its own line.

- If you provide the `--filter` option, the output only contains kits that match the filter.

list_mcus [--filter <filter>] [<bsp-location>]

This argument lists all of the MCUs, SOCs and SIPs that are valid arguments for the following command, each on its own line.

- If you provide the `--filter` option, the output only contains devices that match the filter.
- If you provide the `bsp-location` option, the output only contains MCUS, SOCs, and SIPs compatible with the specified BSP.

remove_connectivity <bsp-location>

This argument removes connectivity from the given BSP. It displays an error in the following cases:

- If `bsp-location` is not the location of a BSP.
- If the BSP currently specifies an SOC or SIP that has built-in connectivity.

Version changes

6 Version changes

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

Version	Change descriptions
1.0	New tool.

Revision history

Revision history

Date	Version	Description
2022-09-21	**	New document.
2022-10-04	*A	Edited images and text to reflect updated GUI.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-10-04

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

www.cypress.com/support

Document reference

002-36015 Rev. *A

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.