ModusToolbox™ BSP Assistant user guide

ModusToolbox™ tools package version 3.9.0
BSP Assistant version 1.80

About this document

A newer version of this document may be available on the web here

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
Italics	Denotes file names and paths
`Courier New`	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:

- ModusToolbox™ tools package user guide
- VS Code for ModusToolbox™ user guide
- Eclipse for ModusToolbox™ user guide
- Device Configurator user guide

Table of contents

About this document . . . 1

Table of contents . . . 2

1 Overview . . . 3

2 Launch the BSP Assistant . . . 4

2.1 Stand-alone GUI . . . 4

2.2 GUI from application . . . 5

2.3 CLI executable . . . 5

3 Quick start . . . 6

3.1 BSP Assistant use cases . . . 6

3.2 Basic workflow . . . 6

3.3 Opening existing BSP . . . 7

3.4 Creating new BSP based on template . . . 7

3.5 Creating a new BSP by specifying devices . . . 7

4 Editing the BSP . . . 9

4.1 Making changes . . . 9

4.2 BSP Sections . . . 9

4.3 Viewing changed files . . . 15

4.4 Show Diff . . . 15

4.5 Updating launch settings . . . 15

5 GUI Description . . . 16

5.1 Menus . . . 16

5.2 BSP location . . . 17

5.3 BSP Sections . . . 17

5.4 Messages . . . 17

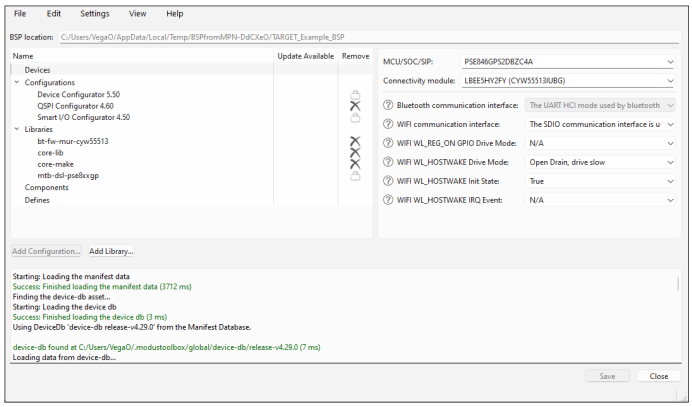
5.5 Save and Close buttons . . . 17

6 CLI Description . . . 18

Disclaimer . . . 20

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, or you can create a BSP by specifying the device(s) it contains. The BSP Assistant helps to simplify the process of creating or modifying a BSP to suit your needs.



***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 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.)*

2 Launch the BSP Assistant

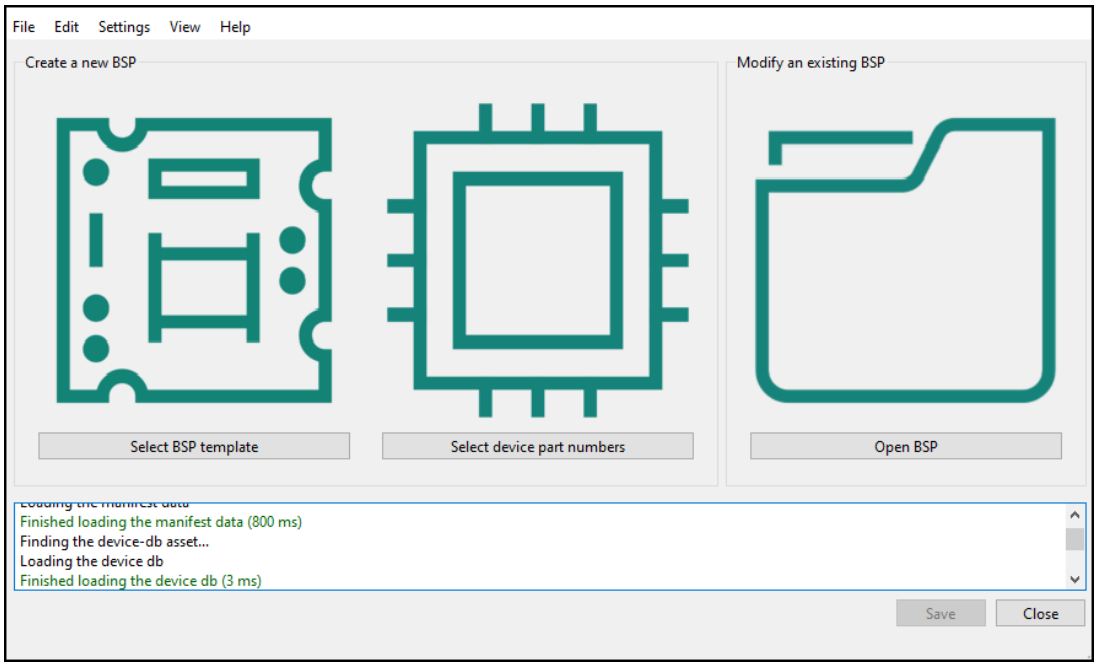
There are numerous ways to launch the BSP Assistant, depending on how you choose to use the various tools in ModusToolbox™ software.

2.1 Stand-alone GUI

You can launch the BSP Assistant as a stand-alone GUI by running its executable as applicable for your operating system (for example, select it using the Windows **Start** menu or macOS Finder). 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.



2.1.1 From Dashboard

You can open the BSP Assistant GUI tool in stand-alone mode from the ModusToolbox™ Dashboard. Refer to the Dashboard user guide for more details.

2.1.2 GUI command-line options

You can open the BSP Assistant GUI tool in stand-alone mode from the command line using:

```
$ bsp-assistant
```

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

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

In this example, the BSP Assistant GUI tool opens with a specified BSP. The `--bsploc` option is used to specify a directory containing an existing BSP to load. For more information about the command-line options, run the executable using the `-h` option.

2.2 GUI from application

If you already have a ModusToolbox™ application, you can open its active BSP from a make command or from VS Code and Eclipse.

When you open a BSP from an application, the device-db is locked to the version that was available when the application was created. If you want newer devices not available in the device-db, you will need to update it using the Library Manager.

2.2.1 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. Navigate to the application directory (for example, Hello_World) 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.

2.2.2 VS Code and Eclipse

VS Code and Eclipse have tools to launch the BSP Assistant from within an open application. Refer to the applicable user guide for more details:

- VS Code for ModusToolbox™ user guide
- Eclipse for ModusToolbox™ user guide

2.3 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.

3 Quick start

3.1 BSP Assistant use cases

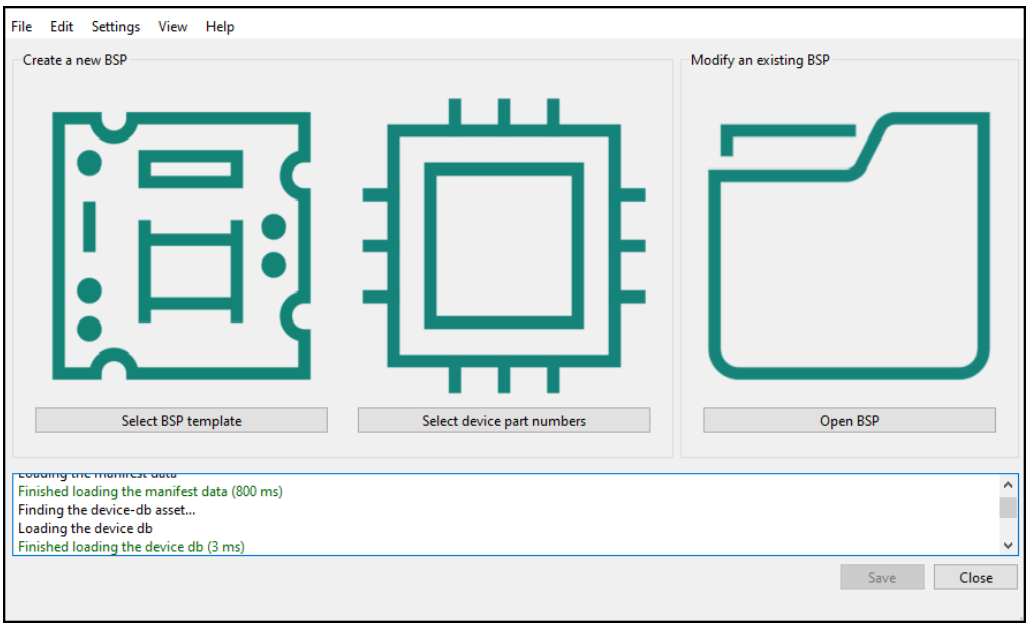
The following provide different scenarios for using the BSP Assistant:

- **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 existing 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 library 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:

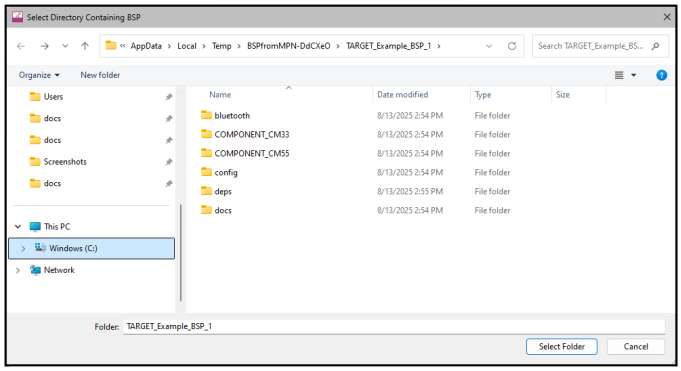
1. Launch the BSP Assistant, and either open an existing BSP or create a new BSP.



   - An existing BSP may be in an application or you may browse to a location independent of an application.
   - Create a new BSP based on a sample board, or by specifying the devices it contains.
2. Edit the various areas of the BSP.
3. Save changes, and optionally export the BSP.

3.3 Opening existing BSP

On the main BSP Assistant window, click the **Open BSP** 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**.

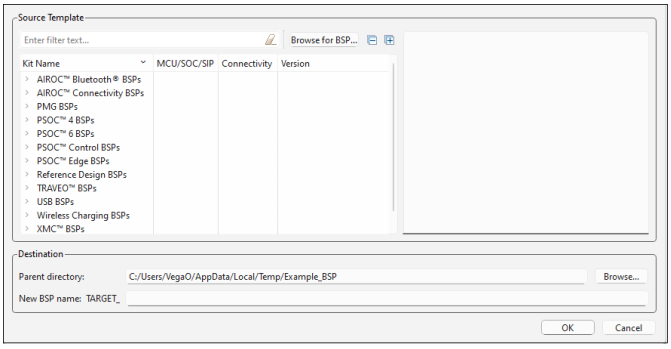


***Note**: If the specified BSP has a .git directory that points to an Infineon repository, a warning dialog will indicate that you might want to use the New... dialog to import this BSP into a new location that is not connected to Infineon's Git repositories before customizing it.*

3.4 Creating new BSP based on template

1. On the main BSP Assistant window, click the **Select BSP Template** link or select **File > New > From BSP Template** to open the Create New BSP from Template 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.
3. Alternatively, click **Browse for BSP** 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 and click **OK**.

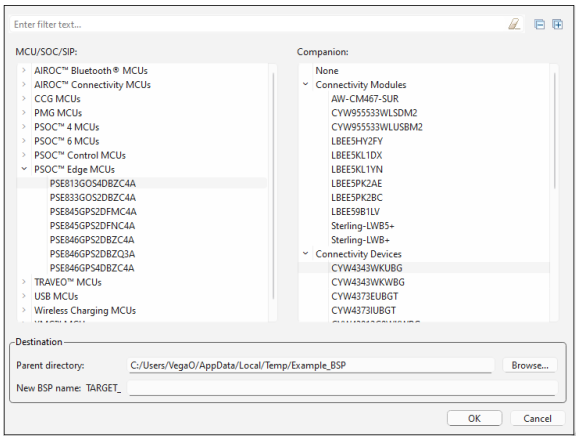
   ***Note**: To prevent copy-and-paste errors, an error message displays if you enter "TARGET_" in the field so that you do not mistakenly enter "TARGET_TARGET_".*

3.5 Creating a new BSP by specifying devices

1. On the main BSP Assistant window, click the **Select device part numbers** link or select **File > New > From Device Part Numbers** to open the Create New BSP from Contained MPNs dialog.

   This dialog provides a list of all devices by family category, as well as various companion devices.

3 Quick start



2. On the left, select the MCU/SOC/SIP for your BSP. You can use the filter option to limit the number of devices displayed.
3. On the right, after selecting an MCU/SOC/SIP, select the appropriate companion device (or none).

   ***Note**: We recommend when adding a companion device that you add a connectivity module instead of a connectivity device; see Connectivity module for more details.*

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 and click **OK**.

   ***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_".*

4 Editing the BSP

After opening or creating a BSP, the GUI shows various settings. The following sections provide details about editing these settings.

4.1 Making changes

When you make a change within the BSP Assistant, such as device information, libraries, defines, etc., the **Save** button becomes active to submit the change. You must save these types of changes before making changes to configuration data, because configuration changes happen in separate tools.

When you make a change that involves configuration data (CAPSENSE™, QSPI, etc.), you save those changes in that particular tool. The BSP Assistant automatically incorporates those changes without needing to use **Save**.

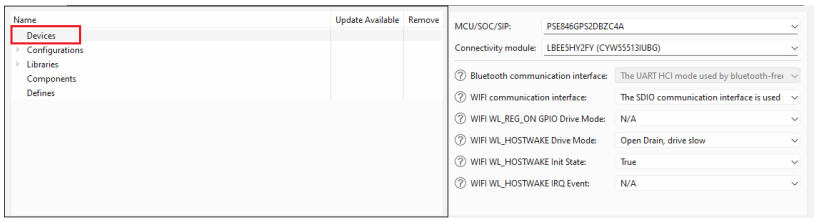
You can use the **Undo <action>** menu to revert any change. If that change was saved (either manually or automatically), you must click **Save** to submit the undo. The same is true for **Redo <action>**.

4.2 BSP Sections

There are various BSP sections to edit and view different parts of the current BSP's information.

4.2.1 Devices

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



***Note**: If you are changing the device in a BSP for an existing application, newer devices may not be available. If not, you will need to close the BSP Assistant and update the device-db version in the Library Manager. You may also need to update the PDL version. Then, reopen the BSP Assistant to change the device.*

4.2.1.1 MCU/SOC/SIP

The **MCU/SOC/SIP** pull-down menu contains all of the central devices that are compatible with the chip architecture of the current BSP. 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 chip architecture. If you want a device in a different chip architecture, you must create a new BSP.

4.2.1.2 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.

4.2.1.3 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.

4.2.1.4 Other settings

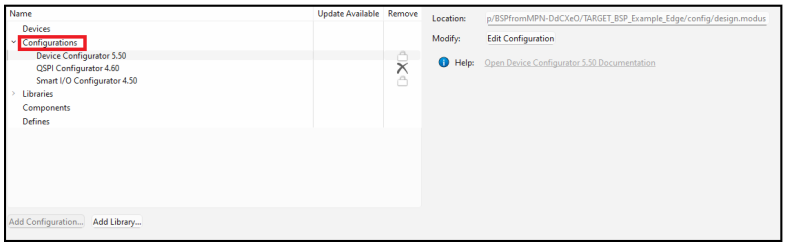
The remaining rows in the Devices section are options that are specific to the MCU/SOC/SIP and connectivity selections that you have made. These are used to select the settings required to get the desired behavior of the devices.

***Note**: For PSOC™ 6 only, there is a "CM0+ pre-build firmware image" setting used for single-core applications. If the current BSP is for a PSOC™ 6 multi-core application, the Makefile uses a `DISABLE_COMPONENTS` option to disable the default CM0+ image, and therefore overrides this setting.*

4.2.2 Configurations

The Configurations section allows you to edit, remove, or add BSP configuration settings for the selected MCU/SOC/SIP and connectivity device. The list on the left shows all configurators that currently have files in the BSP. The **Location** field on the right shows the path to the selected configuration file.

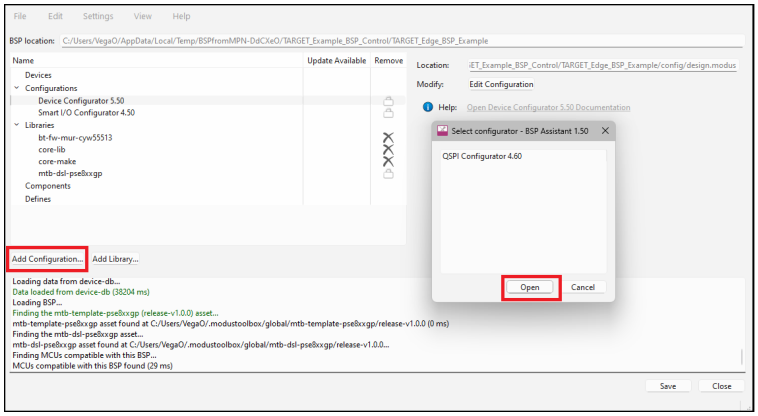
***Note**: As mentioned in the Making changes section, the BSP Assistant automatically incorporates configuration changes from these separate tools without needing to use the **Save** button.*



***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.*

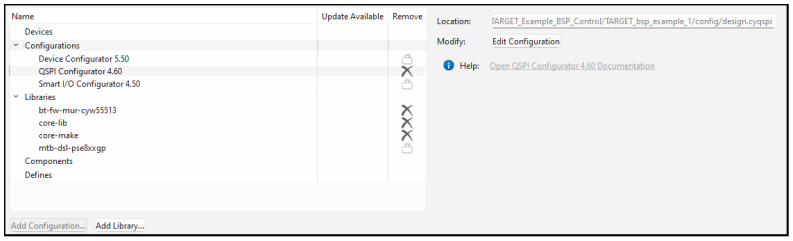
4.2.2.1 Add configuration

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



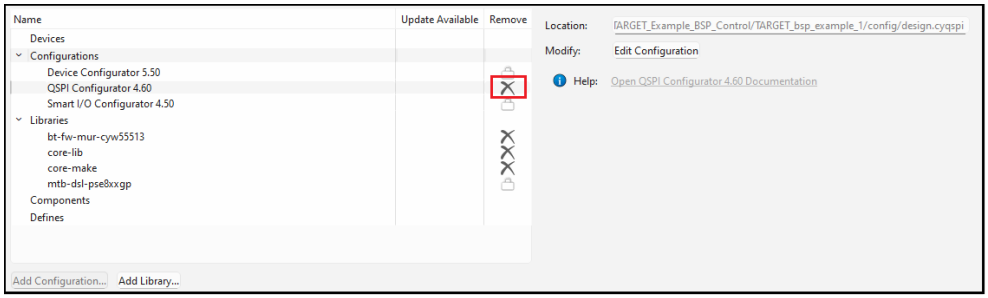
Select the desired configurator from the list and click **Open**. This will open the applicable configurator.

Once you save your settings from that configurator, it will appear in the list in the BSP assistant.



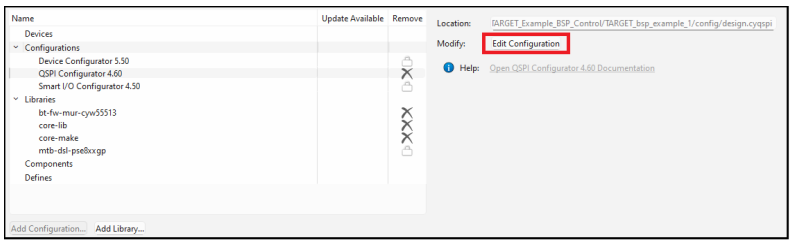
4.2.2.2 Remove configuration

To remove a selected configuration file from the BSP, click the **X** icon, if applicable. Some have a lock symbol and cannot be removed.



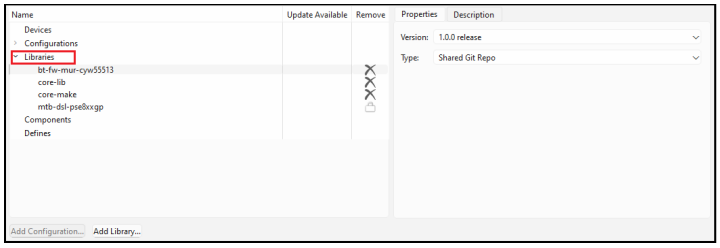
4.2.2.3 Edit configuration

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



4.2.3 Libraries

The Libraries section shows all of the libraries that the BSP needs to operate along with any other libraries that have been 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** or **Type** of any library. The type is either shared (that is, in *mtb_shared*) or local (that is, in the application's *libs* directory). To change these values, use the pull-down menus on the right side of the dialog.



***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.*

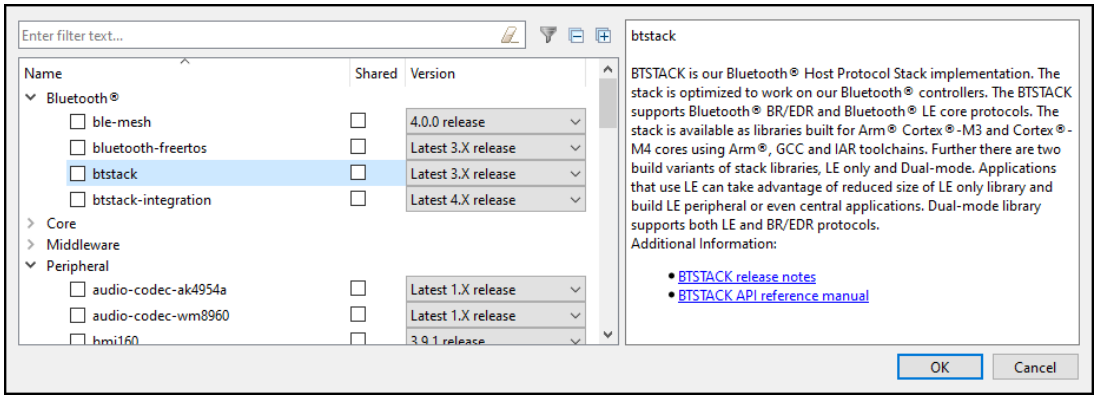
When you add additional Libraries, the default is "Latest V.X release" where "V" is the major version number. This means that those libraries will be set to the latest versions when the BSP is used to create an application. If you want fixed versions in the BSP, you can use the drop-down to select the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

*If you modify the libraries in a BSP, any application(s) that use the BSP must be updated to include those libraries. 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.*

You can remove a library, change a library from shared to local, and select a different version.

- To remove a library, click the **X** icon, if applicable. Some have a lock symbol and cannot be removed.
- To add libraries, click the **Add Library...** button to open the Add Libraries dialog.

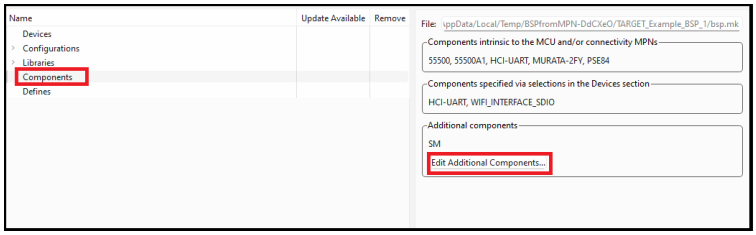
This will open a dialog similar to the Add Library dialog from the Library Manager. Refer to the Library Manager user guide for more details about that tool. Use the plus and minus buttons to expand or collapse all categories.



- Enable the **Filter** button to show only the libraries that are already included in the BSP.
- Use the plus and minus buttons to expand or collapse all categories.

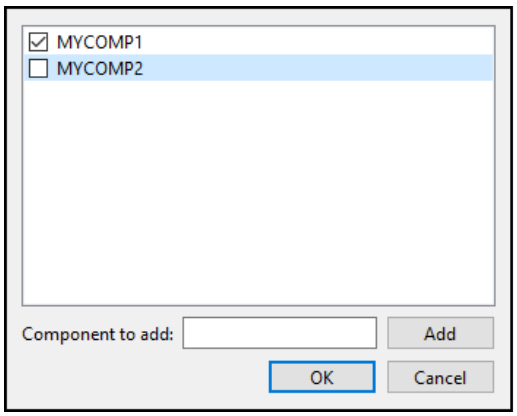
4.2.4 Components

The Components section lists values that are included in the `COMPONENTS` variable in the *bsp.mk* file.



There are three sections: components that are required by the devices that your BSP contains, components that are required based on option selections from the Devices section, and any additional component values. The first two sets are set based on the devices section, so they cannot be modified in the Components section.

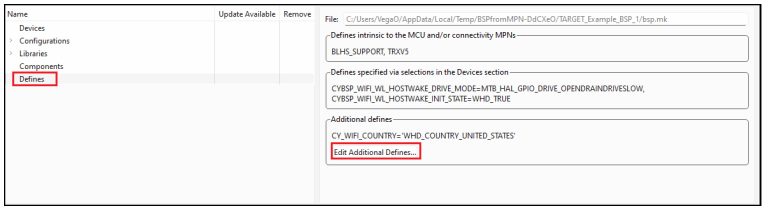
You can add or remove additional components by clicking the **Edit Additional Components** button.



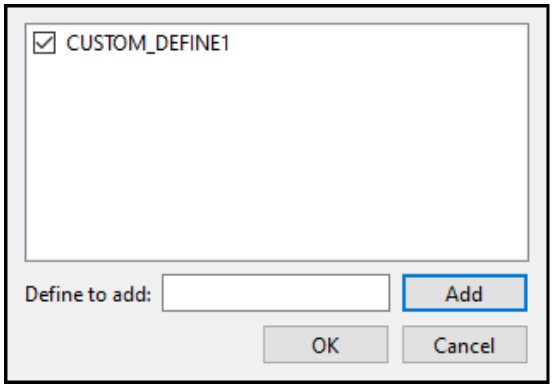
Type the name of a component to add in the box and click **Add** to add it to the list. Deselect the box next to any existing components to remove them. Click **OK** when you are done making changes.

4.2.5 Defines

The Defines section is similar to the Components section, except that it lists values for the `DEFINES` variable in the *bsp.mk* file.

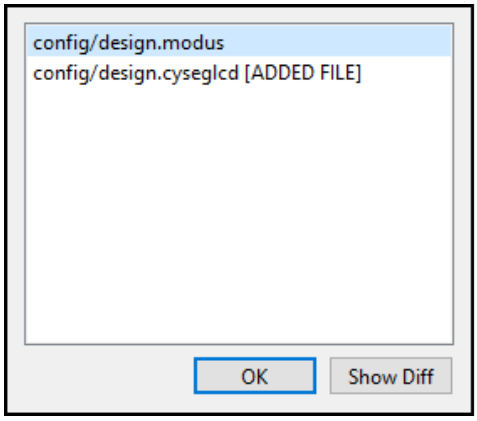


As with the Components section, there are three categories of defines. The first two are read-only values based on the selections in the Devices section. The third includes additional defines. Click **Edit Additional Defines** to add or remove additional define values.



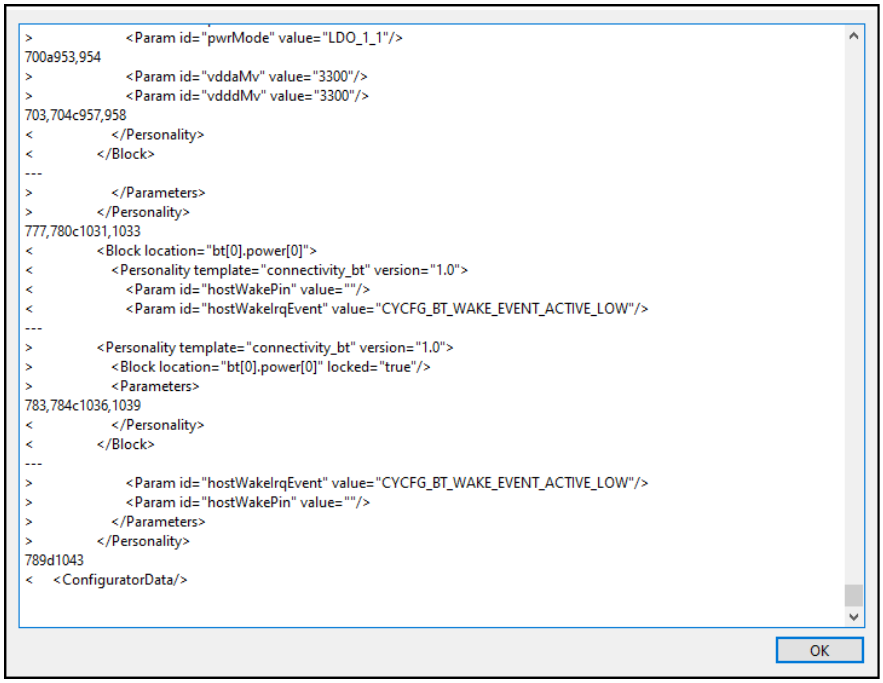
4.3 Viewing changed files

To view a list of all the files that have been modified since you opened the tool for this session or created in the current BSP, select **View > List Changed Files...**



4.4 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**.



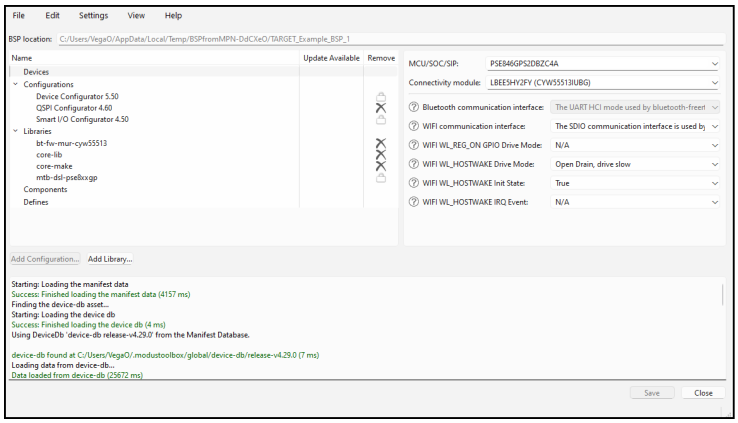
This dialog shows the output of the "diff" command, comparing the file as it existed when the BSP Assistant opened the BSP to the file as it exists now. Deleted lines include the "<" character while added lines include the ">" character.

4.5 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.

5 GUI Description

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



5.1 Menus

5.1.1 File

- **New** – Provides options to create a new BSP:
  - From Sample Board
  - From Contained MPNs
- **Open** – Opens an existing BSP; see Opening 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.

5.1.2 Edit

- **Undo <action>/Redo <action** – Reverts/redoes the previous action made in the BSP Assistant tool, such as change devices, run configurators, or add libraries.

5.1.3 Settings

- **ModusToolbox™ Settings**:

  This opens the Settings tool, an editor that allows you to configure a wide range of settings for your environment, such as proxy settings, content modes, and manifest DB settings. See the Settings tool user guide for more details on specific features.

5.1.4 View

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

5.1.5 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.

5.2 BSP location

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

5.3 BSP Sections

These are various sections to edit and view the current BSP's information: see Editing the BSP for more details.

5.4 Messages

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

5.5 Save and Close buttons

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

- **Save** – This button becomes active when you make a change, such as device information, libraries, defines, etc. See Making changes section.
- **Close** – This button closes the BSP Assistant. If you have any unsaved changes, a dialog displays asking if you want to save the changes.

6 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 "PSC":

```
bsp-assistant-cli list_mcus --filter PSC
```

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 (for example, release-v1.2.0), or
- The special string "latest", which indicates that you want the latest version that 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.

create_from _mpn <mpn-list> <parent-dir> <bsp-name>

This argument creates a BSP that supports the specified MPN or MPNs. The mpn-list is a comma-separated list of MPNs, where the first listed entry is an MCU. It displays an error in the following cases:

- If `parent-dir` does not exist
- If `bsp-name` begins with "TARGET_"

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 kits 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

Trademarks

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

Edition 2026-08-15

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-vdl1711574043530

Important notice

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability. Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

If this document is marked as 'Preliminary', 'Target', 'Draft' or 'Proposal', Infineon reserves the right to change all information given in this document at any time without notice. Subject to the development and release of a Product for series supply by Infineon, the technical specifications of the Product are set forth in the relevant datasheet provided by Infineon without such marking.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury.

However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer. Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG). If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited.

For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit www.infineon.com.