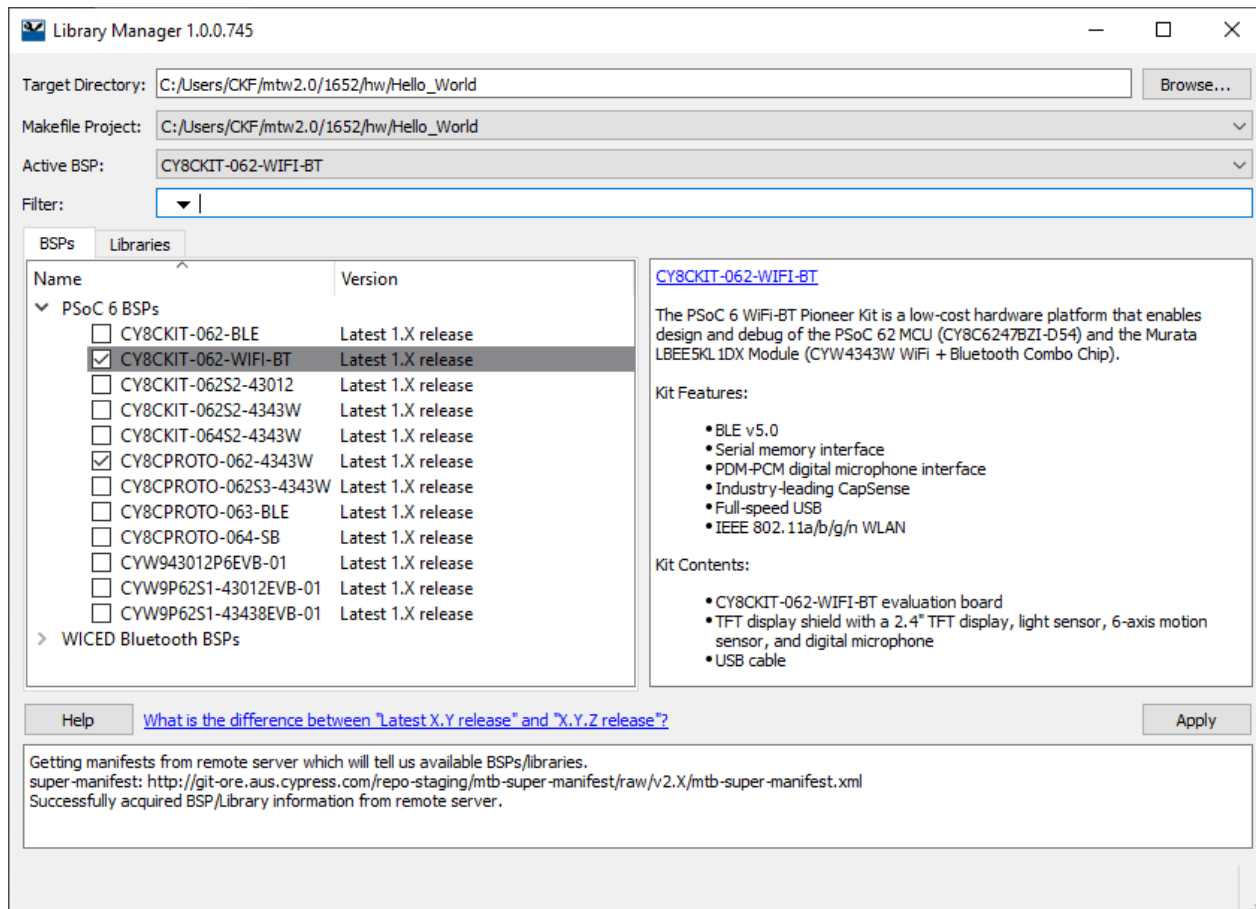


Overview

The Library Manager provides a GUI to select which Board Support Package (BSP) and version should be used by default when building a ModusToolbox application. The tool also allows you to add and remove libraries, as well as change their versions.



Definitions

The following are the terms used in this guide that you may not be familiar with:

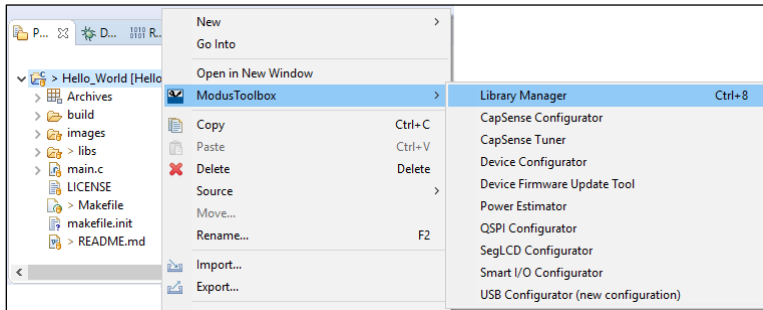
- Local lib file – a .lib file present directly in the project whose library is being edited.
- Shared lib file – a .lib file that is used indirectly via the shared library mechanism.
- Target project – the project directory whose library is being directly edited.
- Fully editable mode – user is able to add, remove, and change the version of library.
- Restricted edit mode – user can manage version of selected library but cannot add or remove library.

Launch the Library Manager

You can launch the Library Manager from, and use it with, a ModusToolbox IDE application. You can also run it independently of the ModusToolbox IDE.

From a ModusToolbox IDE Application

To run the Library Manager from an application within the ModusToolbox IDE, right-click on the project and select **ModusToolbox > Library Manager**.



You can also open the Library Manager by clicking the link in the ModusToolbox IDE Quick Panel.

When the Library Manager opens, it collects a list of available BSPs and libraries for the selected project, and all the necessary metadata from a webservice. The tool also checks the current project directory for all *.lib files to find the currently selected libraries.

Note The ModusToolbox project creation process runs the `make getlibs` command in addition to cloning the repo itself. So, your project comes prepared for the Library Manager to parse it if you operate within the ModusToolbox environment.

Independent of the ModusToolbox IDE

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

```
<install_dir>\tools_<version>\library-manager
```

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

When run independently, the Library Manager opens with the target directory set as <user-home> or the directory used from the previous session. If you are using the Library Manager in stand-alone mode, you must have first created and deployed a project. That is, the `make getlibs` command must have been run to download all the default project dependencies and .lib files.

From the Command Line

You can run the tool 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 tool using the `-h` option.

GUI Description

Fields

The Library Manager contains the following fields at the top of the GUI:

- **Target Directory:** Location of the application's top-level directory, which contains the ModusToolbox makefile project. Use the **Browse...** button to select a different directory.
- **Makefile Directory:** Location of the application's selected makefile. Use the pull-down menu to select another file, if applicable.
- **Active BSP:** The name of the currently active target board. Use the pull-down menu to choose any of the boards with a selected check box in the **Boards** tab. You can have multiple sets of BSP code in a project; however, the make project is built only for the selected **Active Board**.
- **Filter:** Text field used to limit the number of boards or libraries shown on the selected tab.

Tabs

The Library Manager may display one or more of the following tabs, depending several variables:

- **BSP tab:** Shows available BSPs and versions. This tab is hidden if the project uses shared BSPs.
- **Libraries tab:** Shows libraries supported by the Active BSP. This tab is hidden if the project type is a library project.
- **Shared BSPs tab:** Shows a read-only view of BSPs that are available via the shared library mechanism. This tab is hidden if the project is not using any shared BSPs.
- **Shared Libraries tab:** Shows a read-only view of code libraries that will be used by the project. This tab is hidden if the project is not using any shared libraries.

BSP and Library Versions

The Library Manager contains a **Version** column for each BSP and library shown in the tool. You can select "Latest X.Y release" or "X.Y.Z release." These represent tags for versions of BSPs/libraries in Cypress GitHub repos:

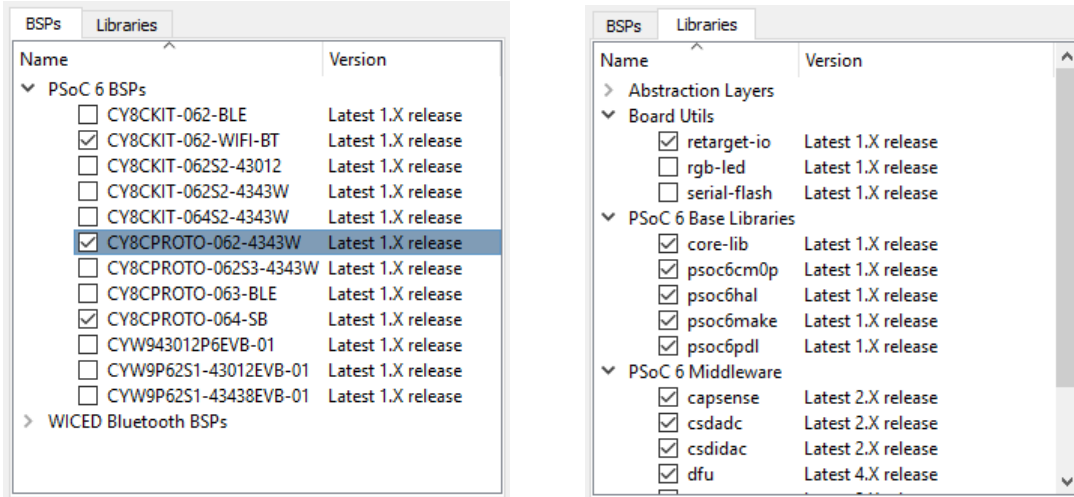
- The "latest-vX.Y" tag is updated on an ongoing basis to point to a newer, backward-compatible version of the library when Cypress releases it.
- The "release-vX.Y.Z" tag always points to a specific, official release, and it does not change.

When you create a new project, Cypress instantiates it with .lib files that point to the "latest-vX.Y" tag on the GitHub repo. When Cypress releases a new version of a BSP/library, the "latest-vX.Y" tag on GitHub moves to point to that newer version. The next time you open the Library Manager for that project and click **Apply**, the tool will display a dialog to allow you to update the BSPs/libraries with "Latest X.Y release" selected as the **Version**.

Cypress also applies the "release-vX.Y.Z" tag to all releases on GitHub repos, and these tags do not move. So, when you select "X.Y.Z release" for the **Version**, it ensures that the selected BSPs/libraries will never be updated. If you decide you do need an updated version, you can change your selection back to "Latest X.Y release" at any time.

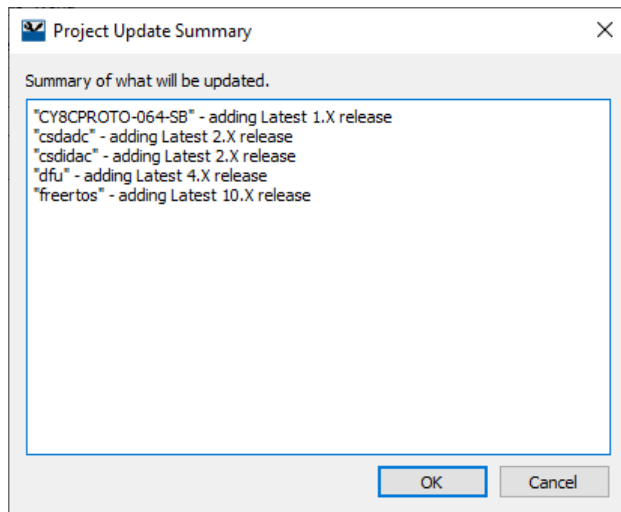
Adding/Removing BSPs/Libraries

The **BSP** and **Libraries** tabs (not Shared tabs) show selection check boxes next to each of the listed BSPs and libraries supported in the application. Selected check boxes represent BSPs/libraries already included in the application, while those without selected check boxes are not included. The highlighted BSP in the list is the Active BSP for the application.

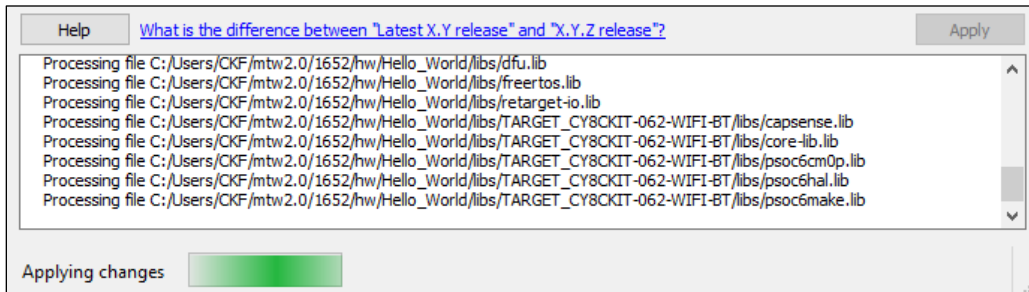


Adding BSPs/Libraries

To add a BSP/library to your application, enable (select) the check box for one or more item on each tab, and click **Apply**. The Project Update Summary displays a list of the changes that will be made to the project.



Click **OK** and the tool starts updating the project and displaying progress in the status box.



When complete, the additional items will display with selected check boxes.

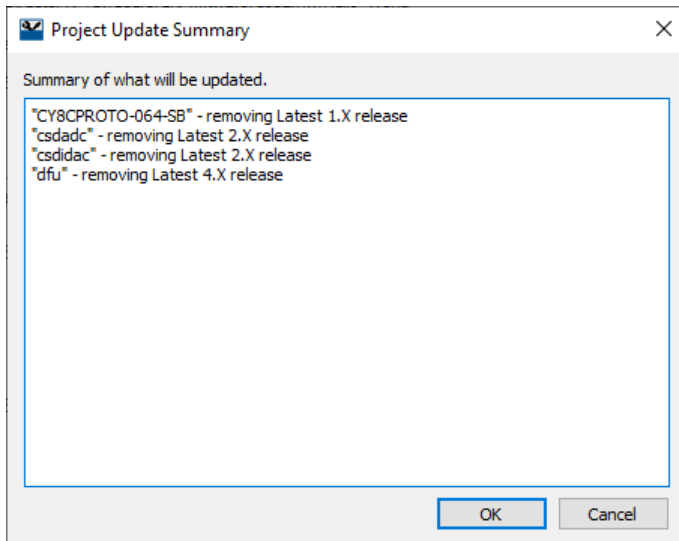
Removing BSPs/Libraries

To remove a BSP/library to your application, disable (unselect) the check box for one or more item on each tab, and click **Apply**.

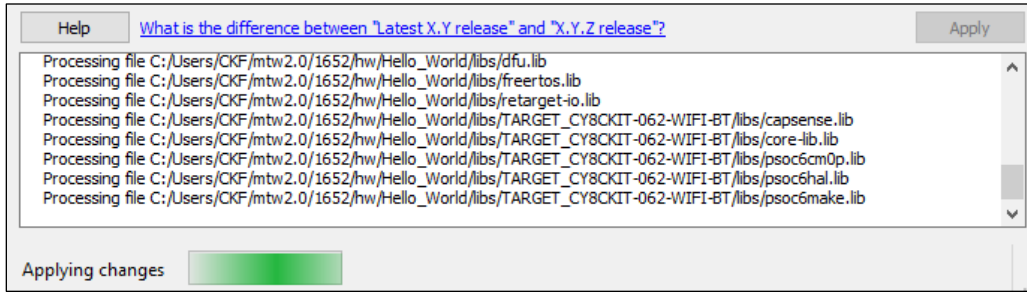
IMPORTANT NOTE: Use caution when removing all Cypress BSPs from your application. BSPs are responsible for ensuring key libraries are present, including "psoc6make" for PSoC 6 projects and "baselib" for Bluetooth projects. Before you remove all Cypress BSPs, make sure a custom BSP is present in your application. For information on creating and customizing BSPs, refer to the following:

- **PSoC 6:** In the *BSP User Guide*, which you can find in a link in the Quick Panel, under Documentation, for the BSP you are using.
- **Bluetooth:** In the *readme.md* file for the selected application.

The Project Update Summary displays a list of the changes that will be made to the project.



Click **OK** and the tool starts updating the project and displaying progress in the status box.



When complete, the removed items will display in the Library Manager without selected check boxes. On disk, the libraries are removed by creating a special .lib file with a commit set to a sentinel value that indicates the library is removed.

Tool Change Description

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

Version	Change Descriptions
1.0	New tool.

© Cypress Semiconductor Corporation, 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.