

How to Access Custom Libraries and Components

1.0

General description

This document is intended instruct the user of PsoC Creator on how to use custom libraries and/or components for their projects. These are libraries and components you can create or have received from others.

For the purposes of this document I will call libraries and components --- “libcomps”.

There are three ways to install these custom libraries and/or components for your version of PsoC Creator you are using.

I’ll start with the most global method (most flexible) and end with the most restrictive method.

Table of Contents

Global Install (Default Dependencies)	1
Project Install (Project Dependencies).....	4
Workspace Project Copy (No Libcomp Dependency Install)	6

Global Install (Default Dependencies)

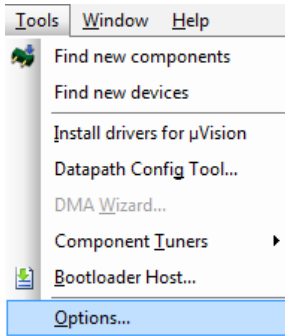
This method is the most flexible especially if you plan on using these custom libcomps for many of your projects.

This method installs the libcomps as Default Dependencies for the PsoC Creator as a whole.

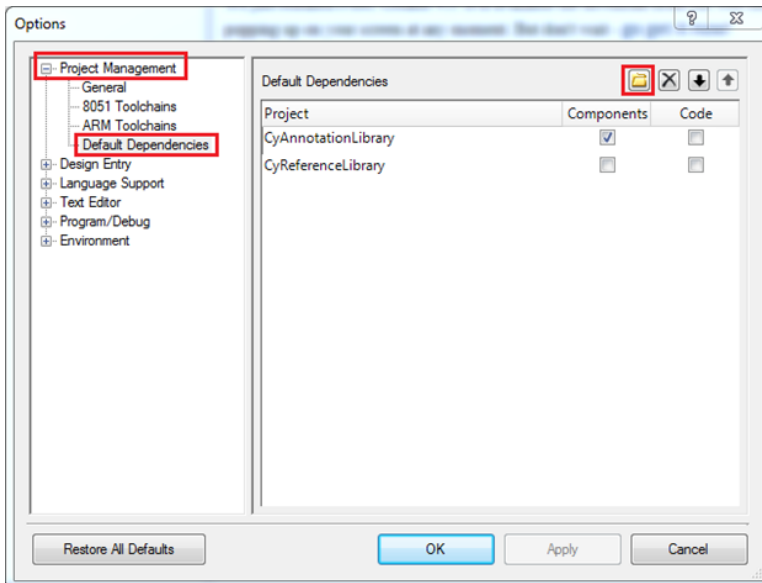
To install a custom libcomp as a Default Dependency:

1. Open PsoC Creator.

2. Select from the menu “Tools\Options...”

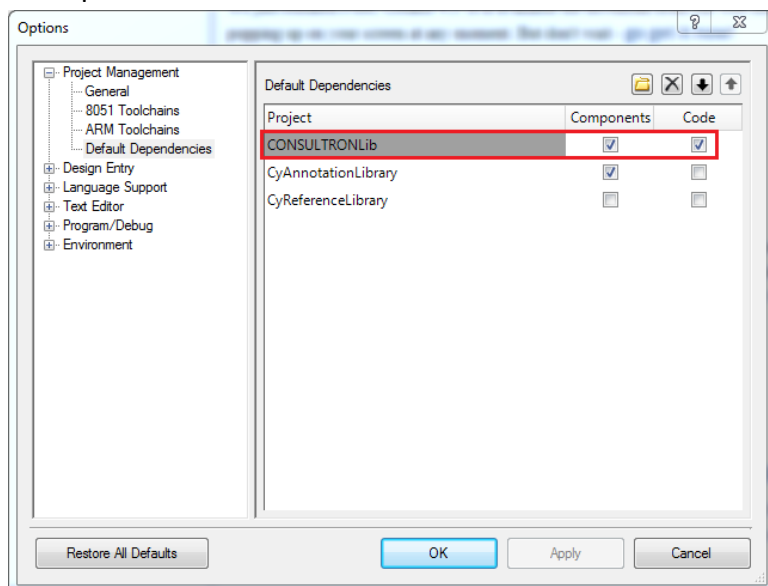


3. In the Options window select “Project Management\Default Dependencies”



4. Next select the folder icon.
5. The File Explorer will appear. Locate the custom libcomp and go into the libcomp's .cydsn directory to find the .cyprj file.
6. Select the libcomp's .cyprj and select “Open”.

7. You should now find the libcomp selected in the Default Dependencies List with the “Components” and “Code” columns enabled.



Advantages of this method

Note: This method adds this libcomp to PsoC Creator’s default dependency list. New projects will inherit this dependency and the libcomp will be available to all projects. See [Project Install \(Project Dependencies\) step 8](#) how to access the libcomp component in the TopDesign.

To use this new libcomp in a project already created, you need to open the project and start at [Project Install \(Project Dependencies\) step 3](#).

If you receive an updated libcomp (with the same libcomp name), install it in the same location. Then the feature “Update Components...” should allow for detecting if the libcomp has an updated component for your project.

Since the libcomp is installed and referred to in the Dependencies the libcomp code is not duplicated and is easily shareable among all projects that wants it.

Disadvantages of this method

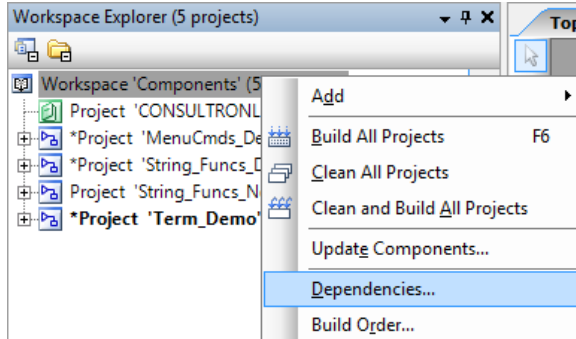
Since the libcomp is installed and referred to in the Dependencies the libcomp code is not duplicated, sharing your project/workspace with other people will require that the libcomp to be installed as a Dependency.

Project Install (Project Dependencies)

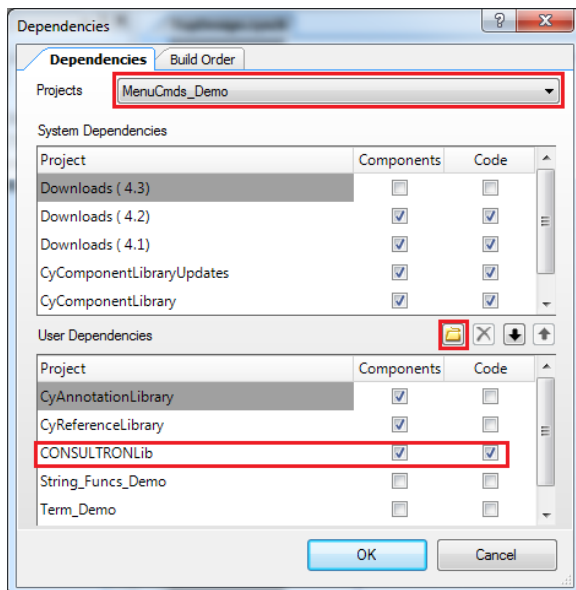
This method will install the custom libcomp on a specific project. If [Global Install \(Default Dependencies\)](#) had been performed you can skip to [step 7](#). Otherwise, this dependency will only apply to this project.

To install a custom libcomp as a Project Dependency:

1. Select the Workspace (or specific Project).
2. Right-mouse select “Dependencies”.

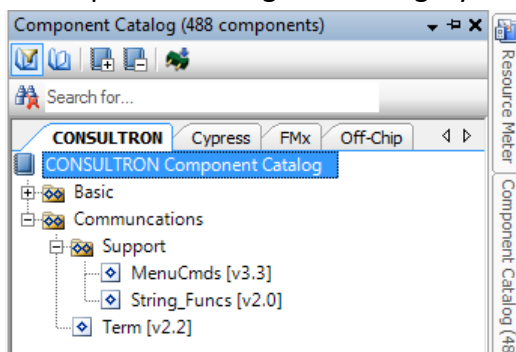


3. In the Dependencies window **make sure the target project is selected**. If not, select the project in the “Projects” field. Repeat steps 3 thru 6 for each project in the workspace you wish to use the custom libcomp.



4. If the custom libcomp is already listed in the “User Dependencies”, just make sure the “Components” and “Code” columns are enabled. Skip to [step 6](#).
5. If the custom libcomp is not listed in the “User Dependencies”, select the folder icon and locate the libcomp’s .cyprj file.
6. Select “OK”. The libcomp should now be available to the “User Dependencies”.

7. The project can now get access to the custom library code. You will have to access the library supplied header files (.h) to use the external functions, #defines, macros and library shared variables.
8. The project can now access the custom component(s). To do this open the project's TopDesign.
9. The Component Catalog window should have a tab that has the custom components. In the example here, you will find a tab "CONSULTRON" with subcategories. Note: If there is a tab "Default" check it out. This tab is used to contain components the author of the libcomp did not assign to a category.



10. You can now select the target custom component and drop it onto one of your schematic sheets of your TopDesign.
11. Consult the information provided with the custom component (such as a Datasheet) if there are custom configuration parameters to set.

Advantages of this method

Note: This method limits the libcomp install to just the specific project(s).

If you receive an updated libcomp (with the same libcomp name), install it in the same location. Then the feature "Update Components..." should allow for detecting if the libcomp has an updated component for your project.

Since the libcomp is installed and referred to in the Dependencies the libcomp code is not duplicated and is easily shareable among all projects that wants it.

Disadvantages of this method

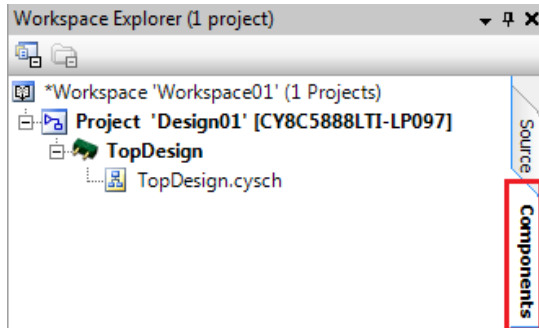
Since the libcomp is installed and referred to in the Dependencies the libcomp code is not duplicated, sharing your project/workspace with other people will require that the libcomp to be installed as a Dependency.

Workspace Project Copy (No Libcomp Dependency Install)

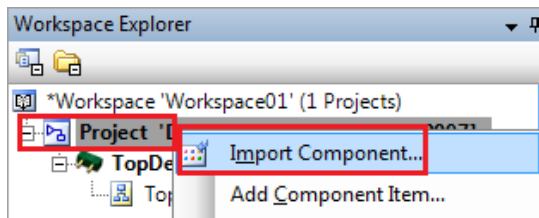
This method will copy the custom libcomp on a specific project. It does not require an install of Dependency information with PsoC Creator. This is the least flexible option of the three.

To copy a custom libcomp as a Workspace Project Copy:

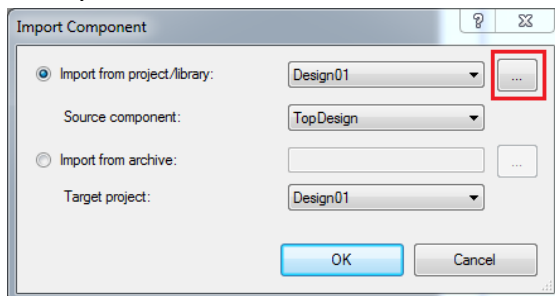
1. Open the Workspace where you want to use the custom libcomp. This Workspace **MUST** already include projects.
2. Select the “Components” tab of the Workspace Explorer.



3. Select the target project and Right-mouse select on “Import Component...”

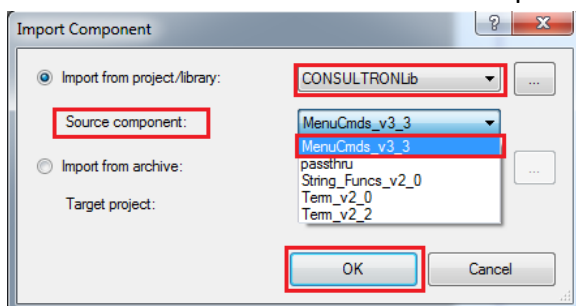


4. In the Import Component window select the “...” button of the “Import from project/library:”

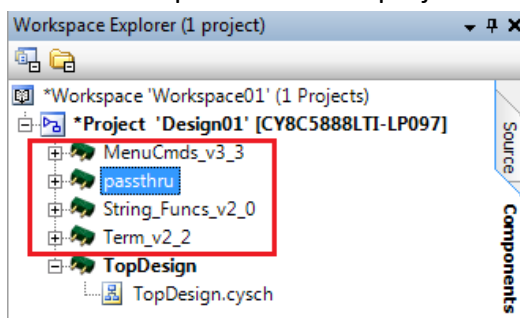


5. The File Explorer will appear. Locate the custom libcomp and go to it’s .cydsn directory and find the .cyprj file.
6. Select the libcomp’s .cyprj and select “Open”.

7. You should now see the custom libcomp in the “Import from project/library:” field.



8. Select the “Source Component:” you are interested in using. Example shown: “MenuCmds_v3_3”. Select “OK”.
9. Repeat step 4 thru 8 for each custom component you are looking to use from this custom libcomp. If the component you wish to copy uses embedded components from this libcomp, you may need to copy all the components of this libcomp.
10. The “Components” tab in Workspace Explorer should now list the component(s) you wish to incorporate into the project.



11. To use these components in the project proceed to [Project Install \(Project Dependencies\) step 8.](#)

Advantages of this method

Note: This method limits the libcomp library and component code to just the specific project(s). It prevents use of this libcomp to other projects and workspaces.

Since the libcomp files are copied into the project directory, a project/workspace archive used for code sharing includes these files.

Disadvantages of this method

For each project that this method is used, the libcomp is copied to the project’s directory. This can lead to project directory bloat.

Sharing this project (as an archive) needs to include these copied libcomp files.

If you receive an updated libcomp (with the same libcomp name) and you wish to inherit the library or component updates, you need to replace each of the files in the project directory. The feature “Update Components...” will not find the new component updates.

Changes

Version	Description of changes	Reason for changes/impact
1.0	first release of the information	

References

[PSoC® Creator™ Component Author Guide](#)

© CONSULTRON, 2020 All Rights Reserved. CONSULTRON allows PUBLIC use of the file.

CONSULTRON allows public or commercial use of this product.

DISCLAIMER: CONSULTRON provides NO WARRANTY expressed or implied.

This product is intended for non-critical or non-safety use and can be used for educational purposes.