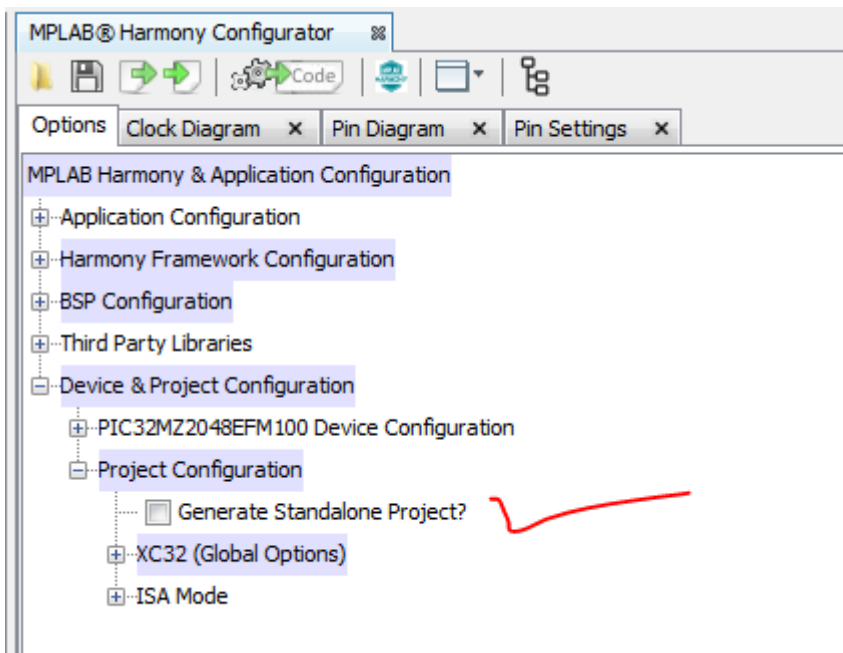


Migrating **Standalone** MPLAB Harmony Project to **Standard** MPLAB Harmony Project

Standalone MPLAB Harmony Project: An MPLAB Harmony project which is generated and build under the MPLAB Harmony project with the option of standalone **checked** under MHC.



- **Once generated and build a standalone project need not be kept inside the Harmony framework (directory structure).** It can be placed anywhere in your system and you should still be able to build and debug it using the appropriate MPLAB X IDE and XC32 compiler.
- A standalone project copies all the necessary files (peripheral libraries, drivers, system services, middleware libraries, etc.) needed by the project into a folder under the project folder.
- A standalone project does not need to have MPLAB Harmony installed.

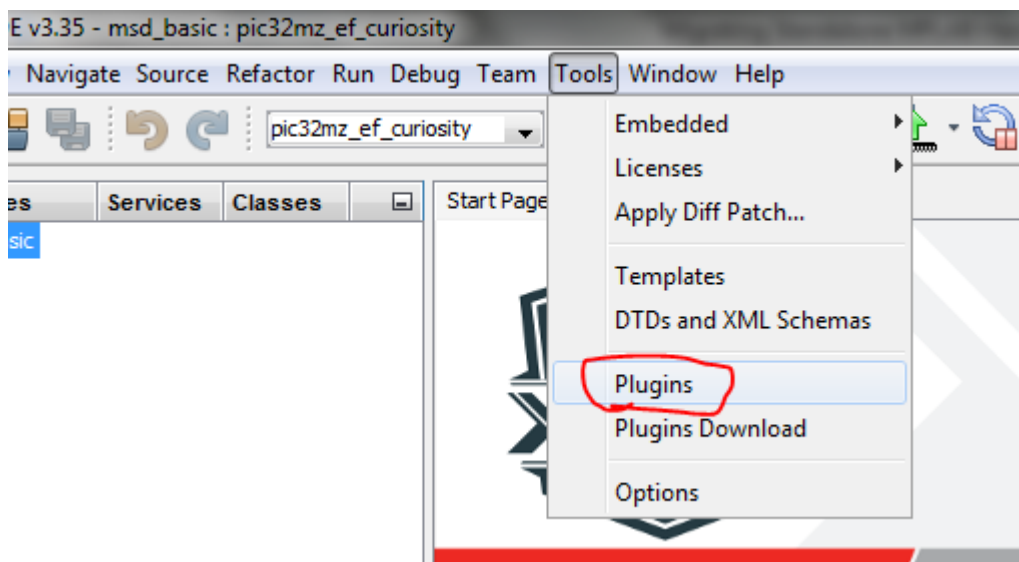
Standard MPLAB Harmony Project: An MPLAB Harmony project which is generated and build under the MPLAB Harmony project with the option of standalone **unchecked** under MHC.

- Once generated and build, a standard project needs to be kept inside the Harmony framework (directory structure). It cannot be placed anywhere in your system.
- A standard project refers to the necessary files (peripheral libraries, drivers, system services, middleware libraries, etc.) relative to its location in the Harmony Framework directory structure (Typically placed under **apps** folder)
- A standard project always needs to be under MPLAB Harmony framework therefore you need to have MPLAB Harmony installed.

The Migration Scenario (Standalone to Standard): Suppose that you have received a Standalone Harmony project to evaluate a PIC32 based development board. After you are done with your evaluation you want extend the standalone project and add more harmony software modules so that you can evaluate other hardware components or further develop/enhance your application. **Is it possible to extend the standalone Harmony project to add other needed harmony modules which were originally not present in the standalone project?**

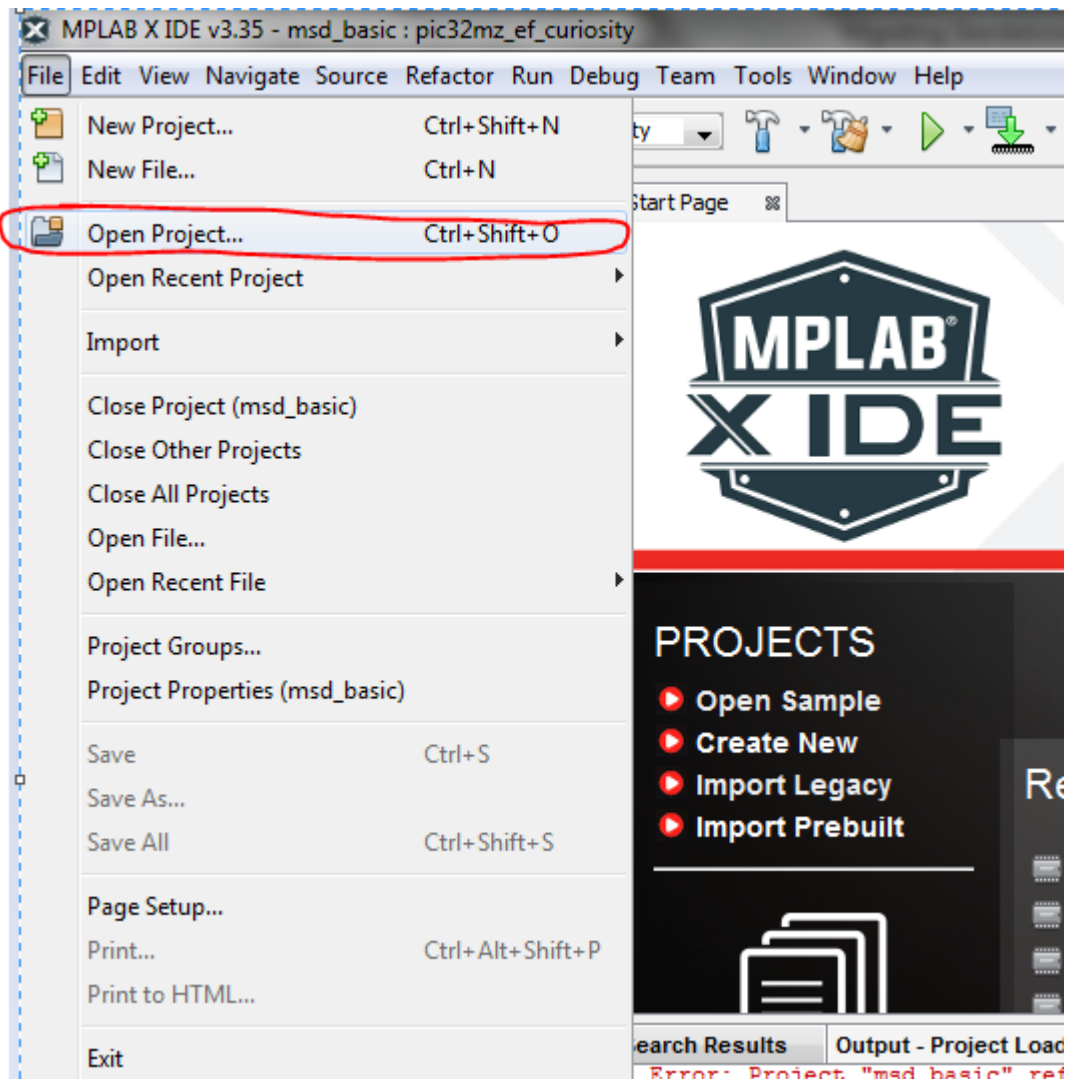
The answer is **YES!** This is possible. Follow the instructions below.

1. If not done already, Download and Install the [MPLAB® Harmony Integrated Software Framework](#).
[Refer to the documentation you received with the standalone project to know the version of MPLAB Harmony the standalone project was generated and build. It is recommended you use the same version]
2. Place your standalone project under the **apps** folder of your MPLAB Harmony installed path. I.e. <Harmony install path>/apps/. (Example Harmony install path = c:/microchip/harmony/v2_00b)
3. Open [MPLAB X IDE](#).
4. Install [MHC](#) plugin. (Ignore this step if you have already installed MHC plugin)
 - a. To Install [MHC](#), In [MPLAB X IDE](#), under [Tools](#), Click [Plugins](#).

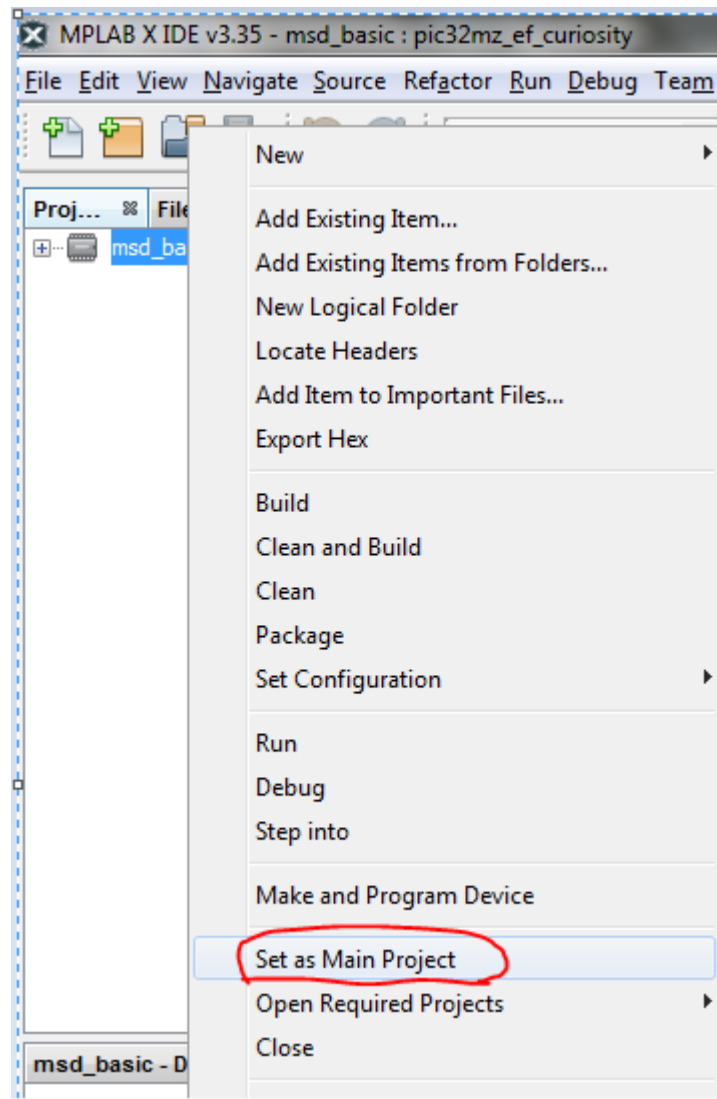


Install the appropriate plugin from the plugin path. Typically from c:\microchip\harmony\v2_00b\utilities\mhc

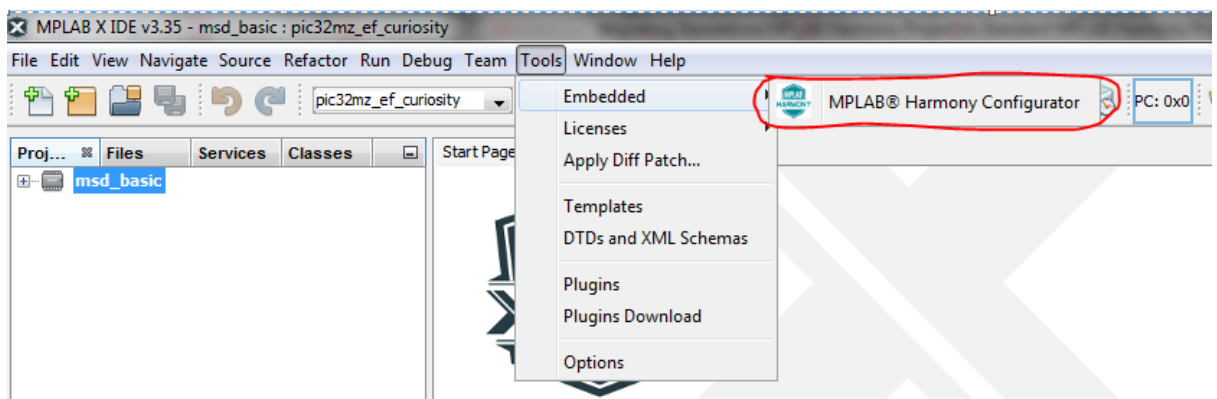
5. Open your standalone project (Which you had just placed under <Harmony install path>/apps/)



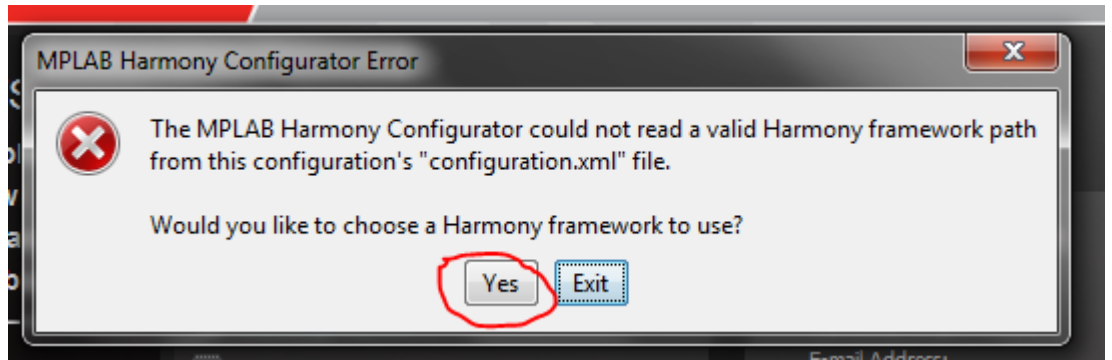
6. Set this project as Main project.
 - a. In MPLAB X IDE, under project explorer window, Right click on the project name, and click on Set as Main Project.



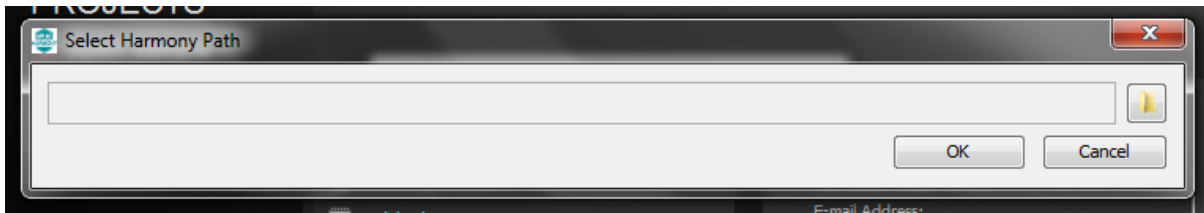
7. Open MHC by clicking **Tools->Embedded->MPLAB Harmony Configurator**.



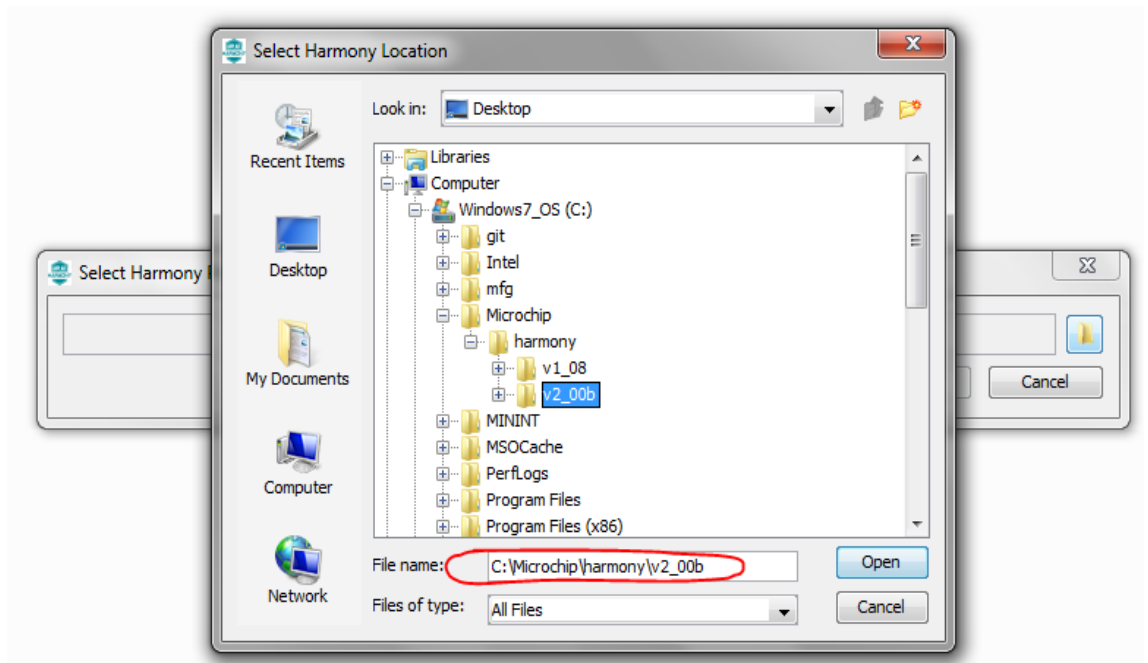
- a. After opening MHC, it could prompt a MHC configuration error as below



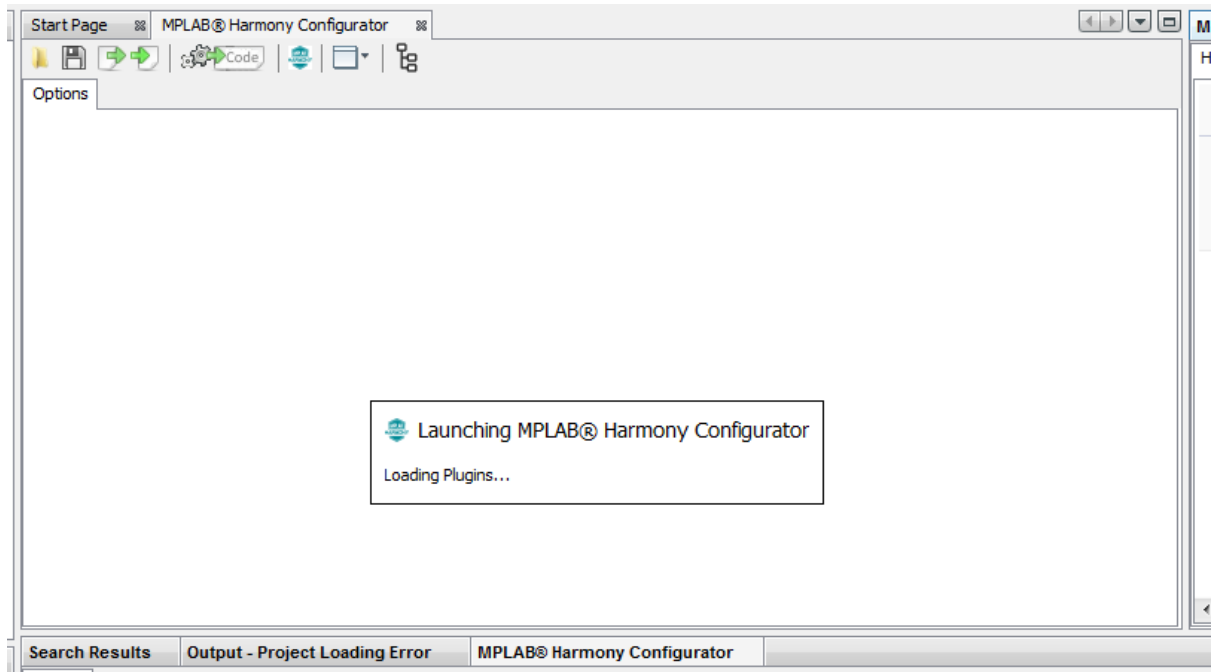
- b. Select [Yes](#) on the MPLAB Harmony Configurator Error.
- c. A dialog box asking the MPLAB Harmony path would appear



- d. Click on the browse folder and select the MPLAB Harmony Path



- 8. [MHC](#) would be launched as shown below.

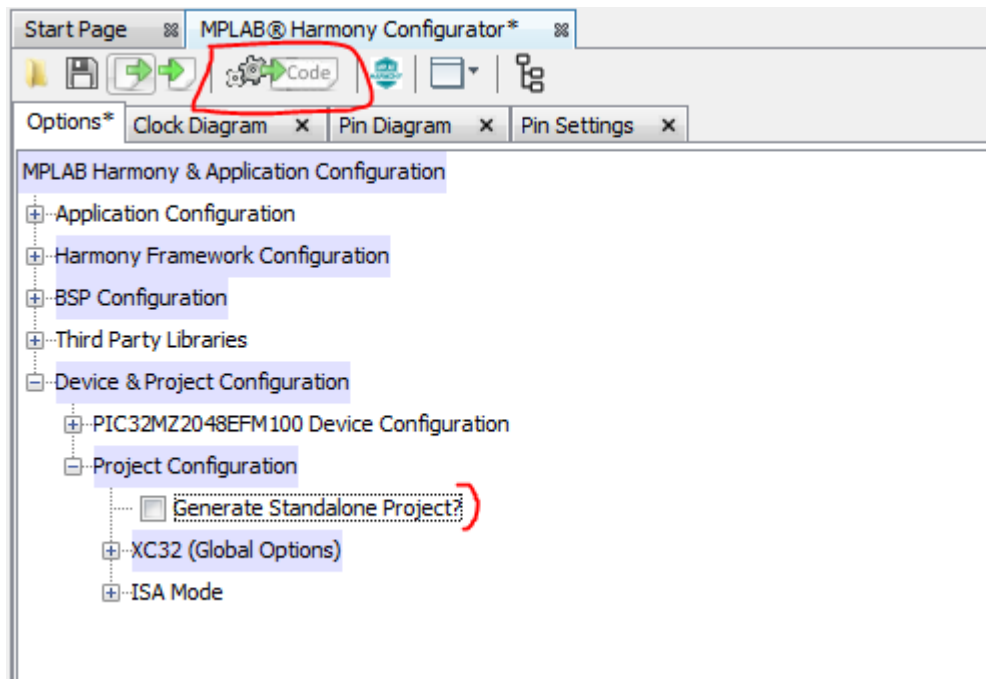


9. In MHC, Navigate to [Project Configuration](#) under [Device & Project Configuration](#).
10. **Uncheck** the option to [Generate Standalone Project?](#) And regenerate the project by hitting the [Generate Code](#) button.

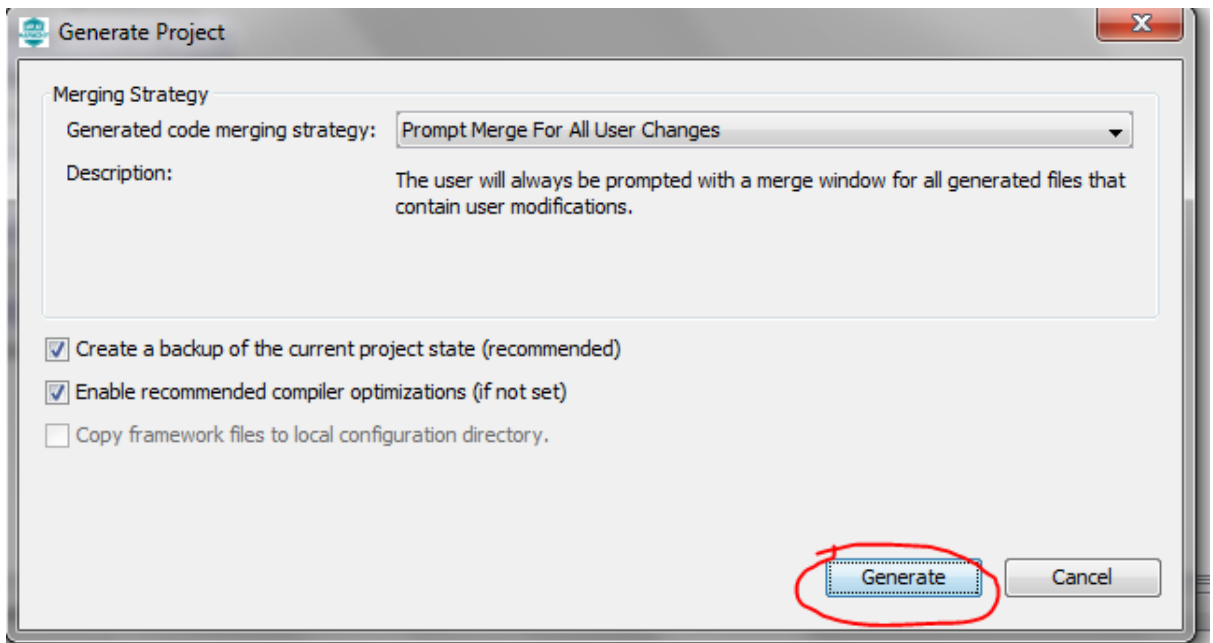
Note: Generate Standalone Project is the only option with which you can generate a MPLAB Harmony project (selected configuration) as standard or standalone.

Checking this option would generate the project as standalone.

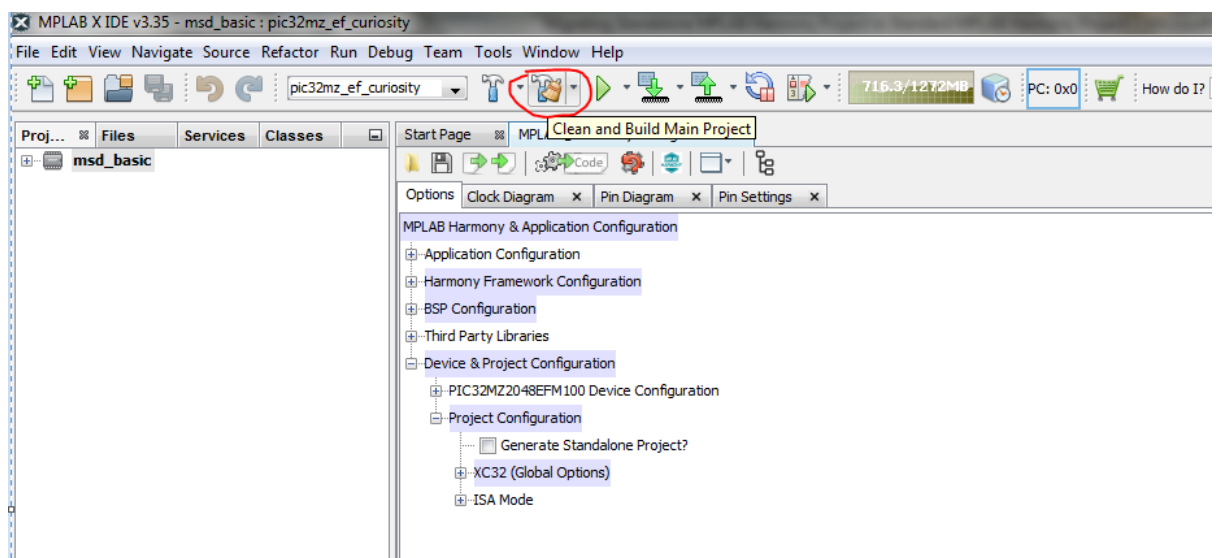
Unchecking this option would generate the project as standard.



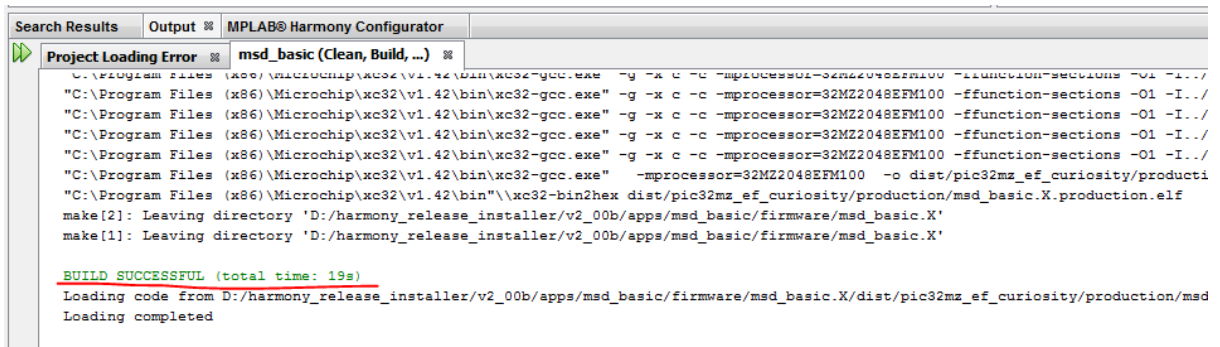
11. Select the desired merging strategy and hit the [Generate](#) Button.



12. Depending on the merging strategy selected the MHC might show options to merge code changes. Select appropriate margining suggestions.
13. After the project is regenerated, hit the [Clean and Build](#) button to clean and build the regenerated standard Harmony project.



14. A message of **BUILD SUCCESSFUL** indicates that the project is successfully migrated from standalone mode to standard mode.



```
Search Results Output MPLAB® Harmony Configurator
Project Loading Error msd_basic (Clean, Build, ...)
C:\Program Files (x86)\Microchip\xc32\v1.42\bin\gcc.exe -g -x c -c -mprocessor=32MZ2048EFM100 -ffunction-sections -O1 -I.../
"C:\Program Files (x86)\Microchip\xc32\v1.42\bin\gcc.exe" -g -x c -c -mprocessor=32MZ2048EFM100 -ffunction-sections -O1 -I.../
"C:\Program Files (x86)\Microchip\xc32\v1.42\bin\gcc.exe" -g -x c -c -mprocessor=32MZ2048EFM100 -ffunction-sections -O1 -I.../
"C:\Program Files (x86)\Microchip\xc32\v1.42\bin\gcc.exe" -g -x c -c -mprocessor=32MZ2048EFM100 -ffunction-sections -O1 -I.../
"C:\Program Files (x86)\Microchip\xc32\v1.42\bin\gcc.exe" -mprocessor=32MZ2048EFM100 -o dist/pic32mz_ef_curiosity/producti
"C:\Program Files (x86)\Microchip\xc32\v1.42\bin"\xc32-bin2hex dist/pic32mz_ef_curiosity/production/msd_basic.X.production.elf
make[2]: Leaving directory 'D:/harmony_release_installer/v2_00b/apps/msd_basic/firmware/msd_basic.X'
make[1]: Leaving directory 'D:/harmony_release_installer/v2_00b/apps/msd_basic/firmware/msd_basic.X'

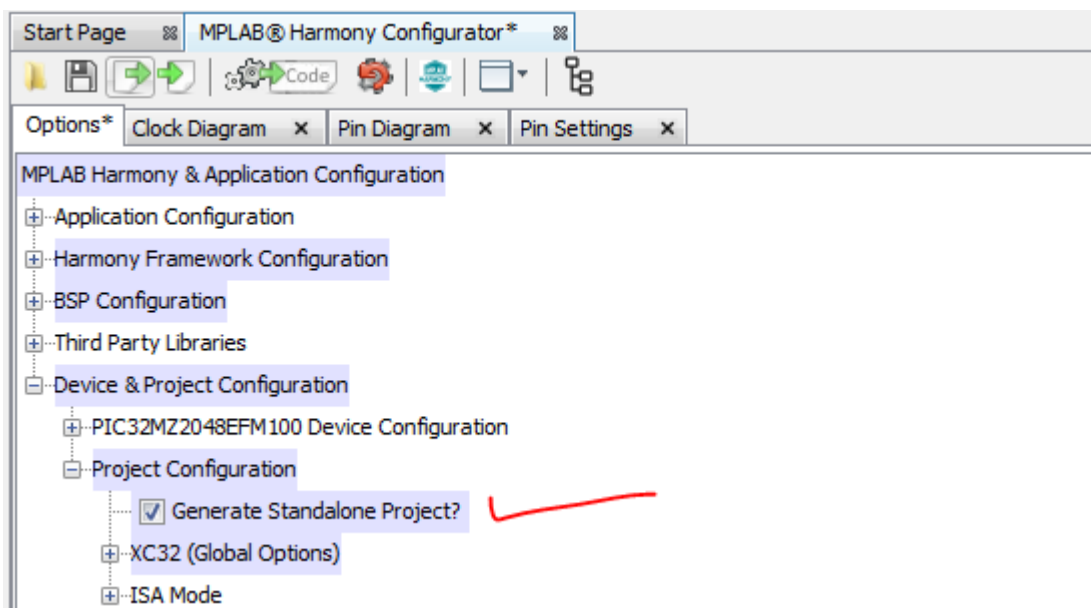
BUILD SUCCESSFUL (total time: 19s)
Loading code from D:/harmony_release_installer/v2_00b/apps/msd_basic/firmware/msd_basic.X/dist/pic32mz_ef_curiosity/production/msd
Loading completed
```

15. Now since the project is a standard Harmony project you can add any of the Harmony modules (drivers, system service, libraries, etc) by using the MHC options. Regenerate and build the project to realize the newly added modules as part of your project.

Creating a Standalone Harmony Project: Now that you have a standard Harmony project and you want to make a standalone version of it so that you share it with others (probably your customers) and that they could try it out at their end.

This is as simple as a mouse click.

1. In MHC, Navigate to **Project Configuration** under **Device & Project Configuration**.
2. **Check** the option to **Generate Standalone Project?** And regenerate the project by hitting the **Generate Code** button.



3. After the project is regenerated, hit the [Clean and Build](#) button to clean and build the regenerated standalone Harmony project.
4. The project is now available to be shared as a standalone project.

Note: When you create a standalone/standard project or migrate to/from standalone from/to standard, only the active configuration will have the changes applied automatically. The other configurations need to be manually handled.