UEStudio IDE and PICBASIC PRO Compiler Integration Tutorial

The purpose of this tutorial is to provide a high-level overview on how to setup UEStudio from IDM Computer Solutions to support MELABS PICBASIC PRO source files and integrate the PBP Compiler and U2 Programmer into the UEStudio IDE.

By configuring the UEStudio IDE to work with PBP, the user will be provided with the following features and functions.

Provides the following features similar to the MCSX IDE:

- Customizable Syntax Highlighting
- Code Explorer Tree for: DEFINES, Constants, Variables and Labels
- Multi-file Tabs
- Comment/Uncomment selected lines
- Indent/Outdent selected lines
- Launch the MELABS U2 Programmer

Provides the following additional features not found in MCSX: (Just to name a few)

- Code Folding
- Auto Indentation
    - Code Folding and Auto Indentation for the following: If/Endif, For/Next, Do/Loop, While/Wend, Select Case, #Config/#End Config, #If/#Endif, etc.
- Debug/Release Build support
- Customizable Syntax Highlighting on Key Words (Supports up to 20 separate color groups)
- More Powerful Find/Replace including Regex (Regular Expressions)
- Bookmarks
- HEX Editing
- Line/Column Mode Editing
- Compare Source Files (Diff)
- Copy/Past with Windows Clipboard plus (9) User Clipboards
- Word Case Conversion of selected text (ToUpper, ToLower, Capitalize, Invert Case)
- Macros

Screen shot examples of the above features are listed in Appendix A of this document.

UEStudio's configurability makes the process of setting up external compiler tool chain and file-type specific syntax highlighting support fairly easy. The setup process involves adding two configuration files for UEStudio to use for PIC BASIC PRO source files, a compiler Configfile and a compiler Wordfile.

The compiler Configfile instructs UEStudio how to Compile and Build PBP source files by calling the PBP compiler and how to Program the PIC MCU with the compiled .hex file by calling the U2 Programmer application.

The compiler Wordfile instructs UEStudio how to perform color-coded syntax highlighting, code folding and auto-indentation for PBP source files.

The included .zip file includes these two files that the user will install on their system. These files may be modified further to customize the functionality for their particular setup.

- Configfile: PBP Application (no extension)
- Wordfile: pbp.uew

The information on how to create these files and their structure is covered in the UEStudio help file. The Configfile information can be found under the "Compiler Configuration" section of help. The Wordfile information can be found under the "Advanced->Configuration->Editor Display->Syntax Highlighting" section of help.

Setup

Configfile Install

The included "PBP Application" Configfile will need to be copied into the following directory on your system. C:\Program Files\IDM Computer Solutions\UEStudio\configs\PBP3 Compiler\

You will need to ensure that the "C" drive is changed to your specific installation setup. You will need to create the "PBP3 Compiler" subdirectory under the "configs" subdirectory above.

Wordfile Install

The included "pbp.uew" Wordfile will need to be copied into the following directory on your system.

C:\Users\xxx\AppData\Roaming\IDMComp\UEStudio\wordfiles

Xxx = User name on the target machine

You will need to ensure that the "C" drive is changed to your specific installation setup.

The exact location on the target machine can be located in the Editor Display - Syntax Highlighting in the Configuration dialog of UEStudio.

Using the UEStudio Compile/Build/Program features with PBP

To utilize the Compile, Build and Program features with the PICBASIC PRO compiler and the U2 Programmer with a given PBP source file, the user will need to use the UEStudio Project feature.

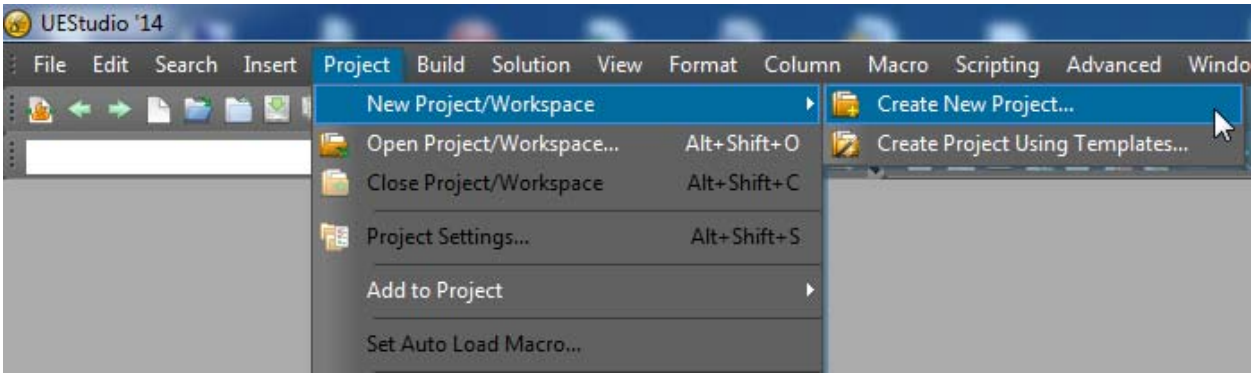
The high-level steps to begin source-code file creation, editing, compiling and ultimately programming the PIC MCU with the compiled .hex file are listed below.

1. Create a new UEStudio project
2. Set the compiler to use for the project
3. Add/Create the source file for the project

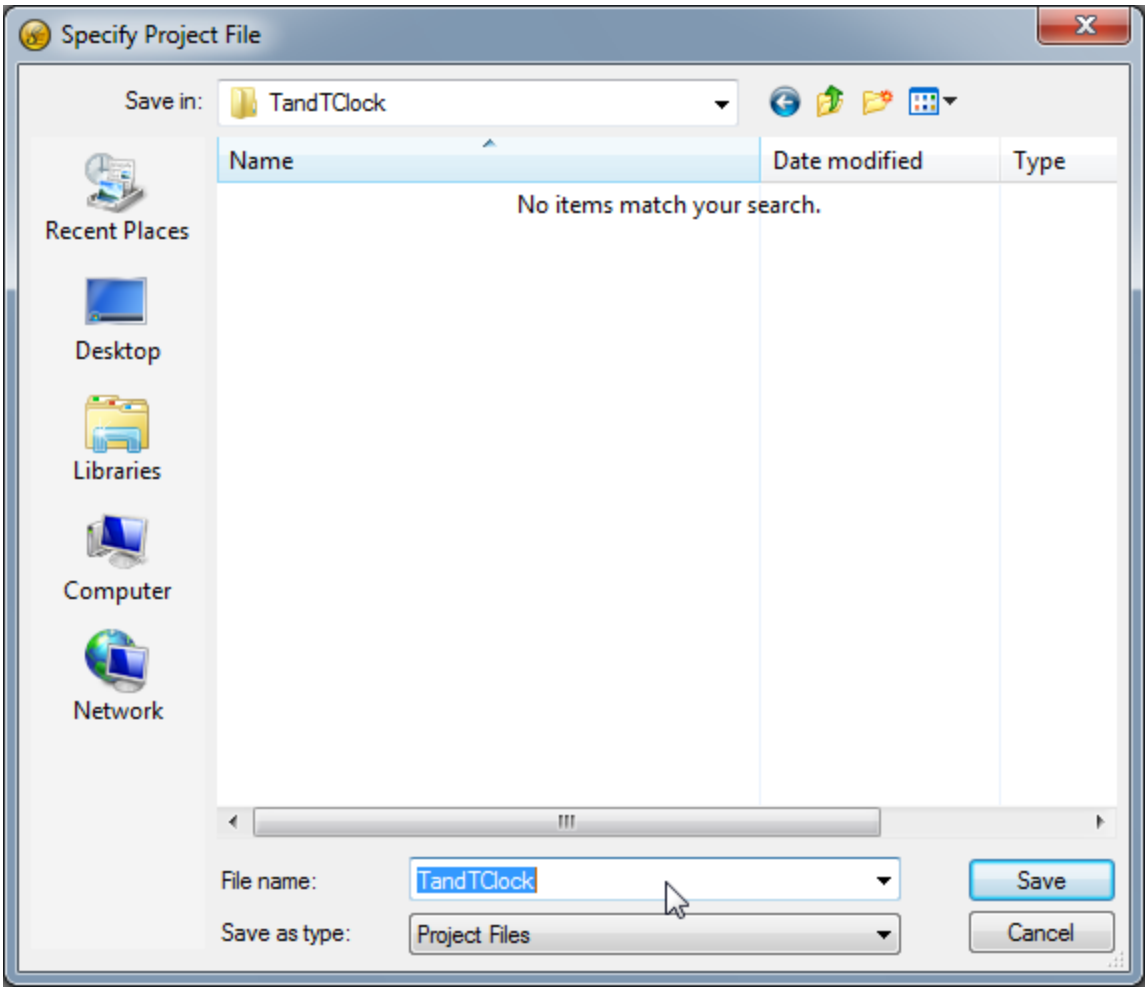
Create UEStudio PBP Project

Create a new Project:

1. From the main screen select Project->New Project/Workspace.

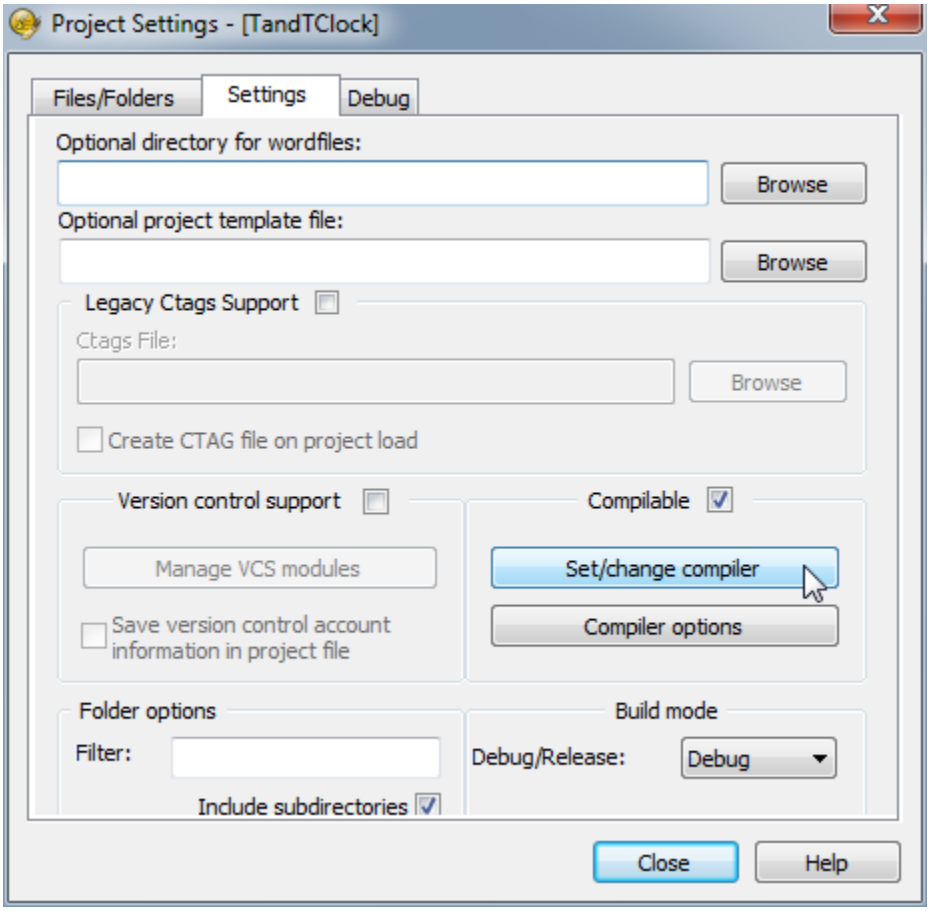


2. From the Project Wizard navigate to the directory you have setup for your new PBP Project and enter the File name for the new project. (NOTE: The project name must be the same as the PBP source file without the .pbp extension. E.g. TandTClock (the PBP source file is TandTClock.pbp)).

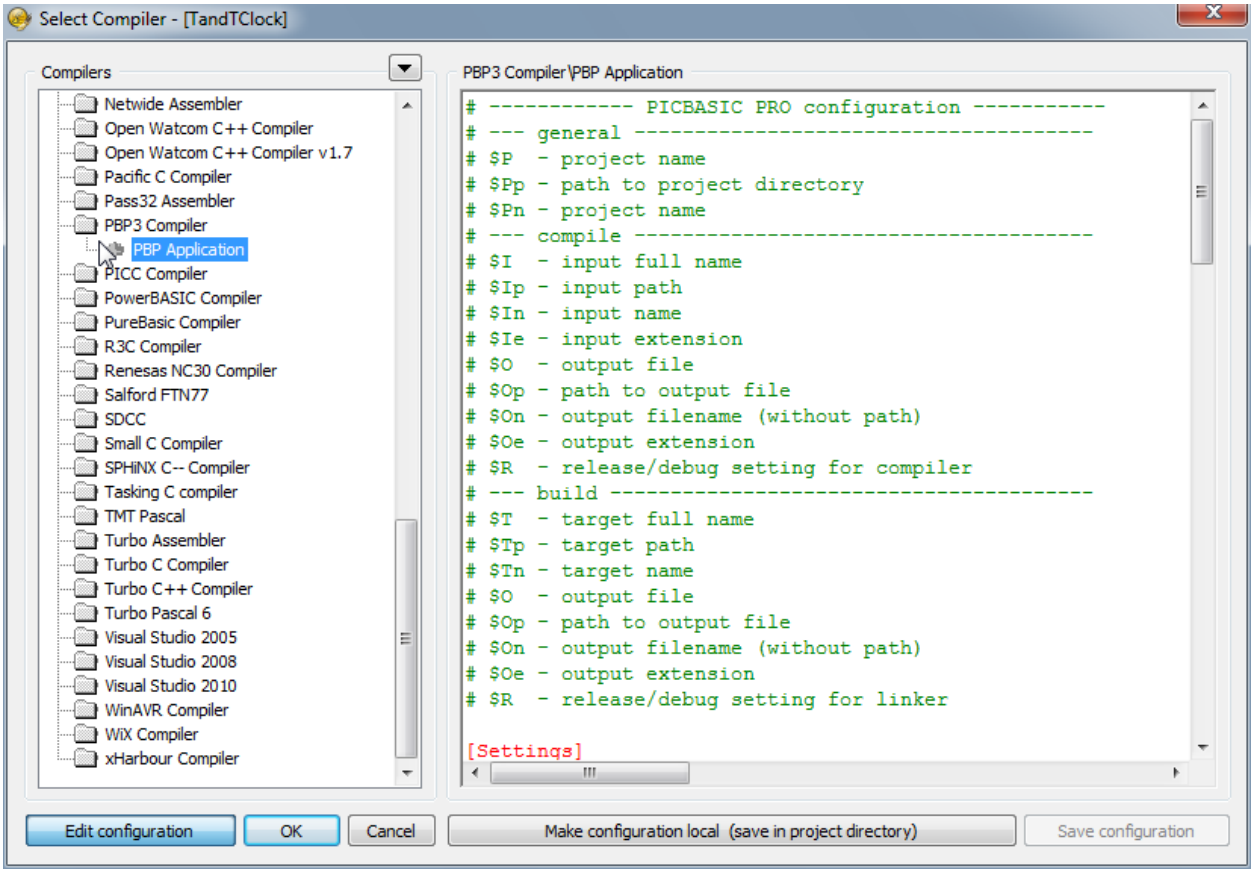


3. Click on the Save button.

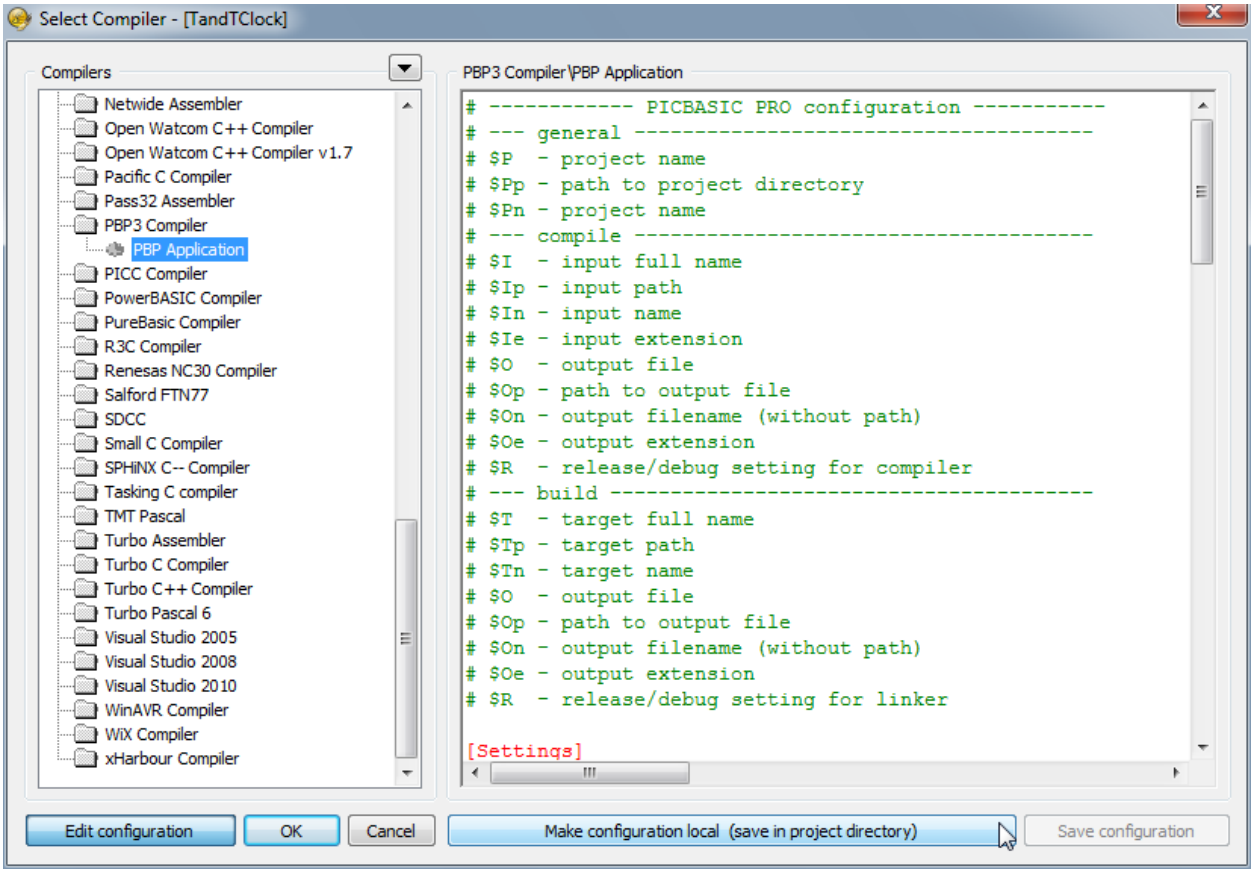
4. The Project Settings dialog is displayed. Select the Set/change compiler button.



5. The Select Compiler dialog is displayed.
6. Navigate the Compilers list tree on the left to the PBP3 Compiler leaf and select “PBP Application”. This is the Configfile that was installed earlier.

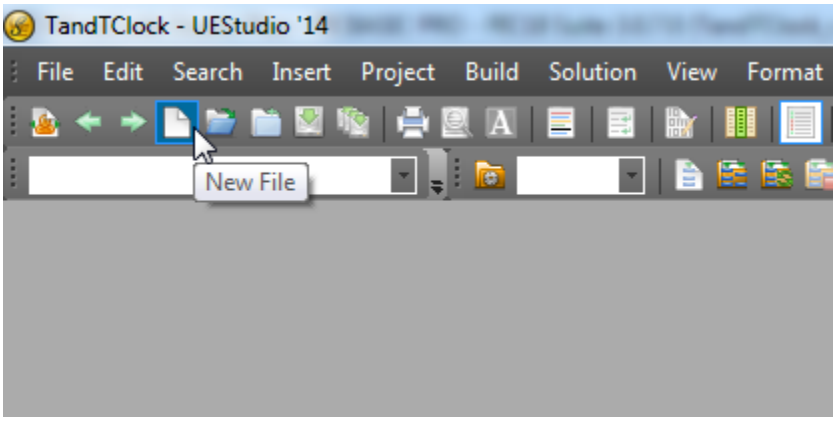


7. Select the “Make configuration local (save in project directory) button. This will create a copy of the PBP3 Application Configfile into your project directory. This is a measure that will allow you to edit the project copy of the Configfile without changing the master file.

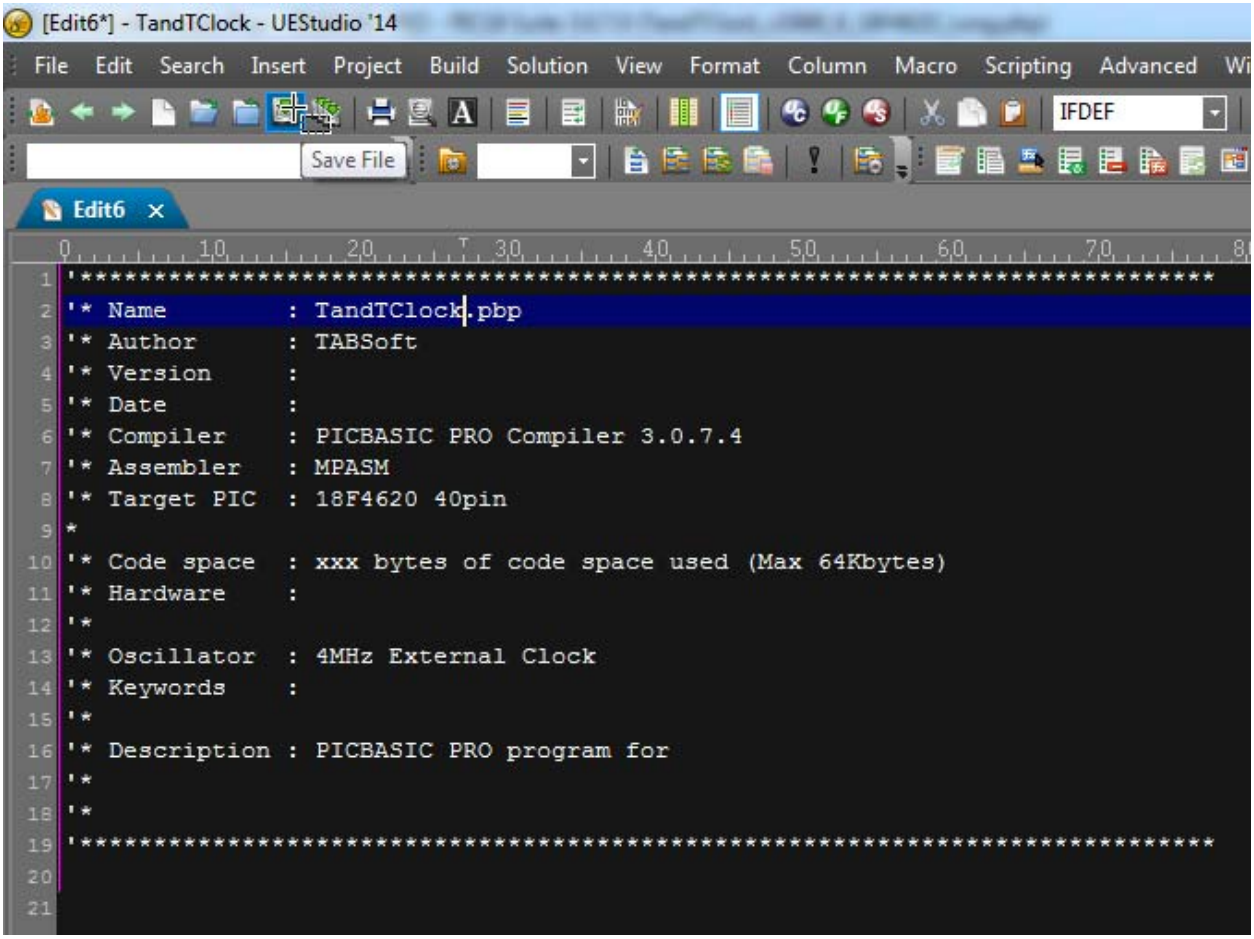


8. Select the OK button to close the dialog.

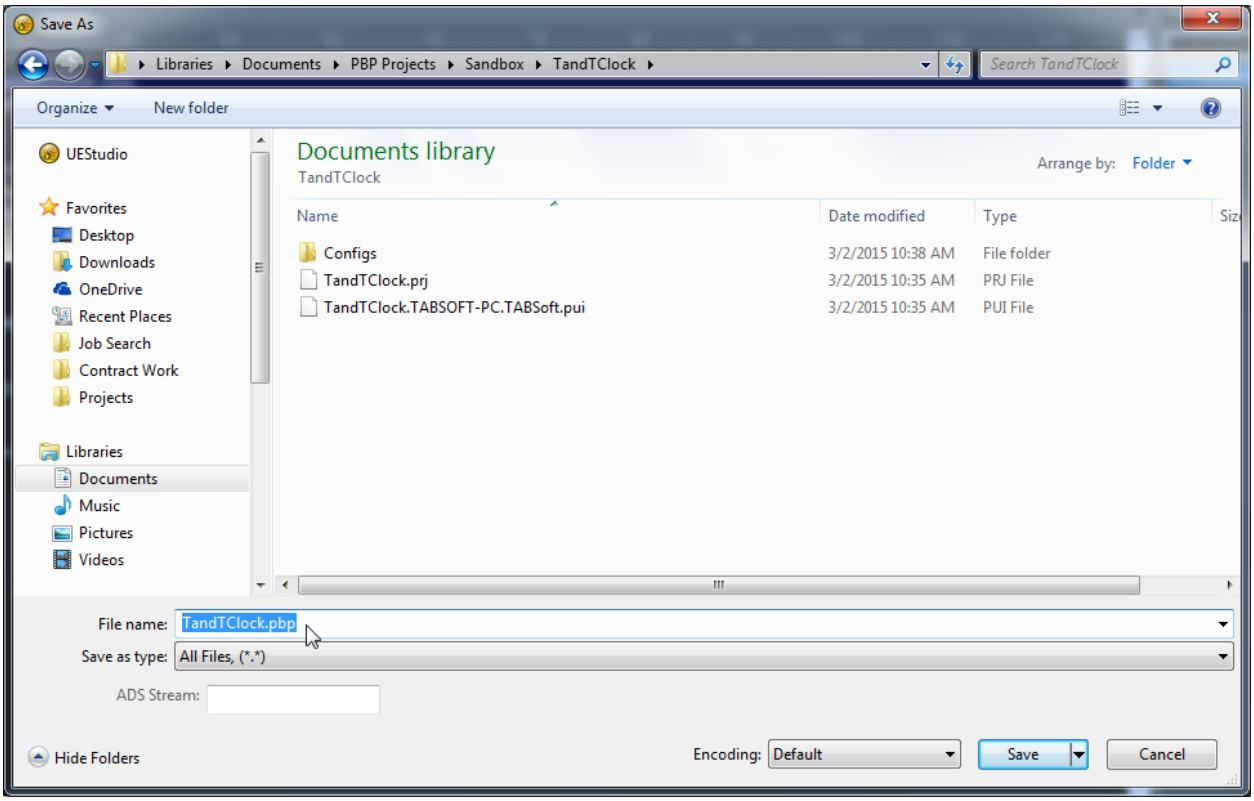
9. Now to create your new PBP Source file, from the UEStudio application select the “New File” button or “File->New” from the menu bar. A new empty file will be created in UEStudio and displayed as a file tab.



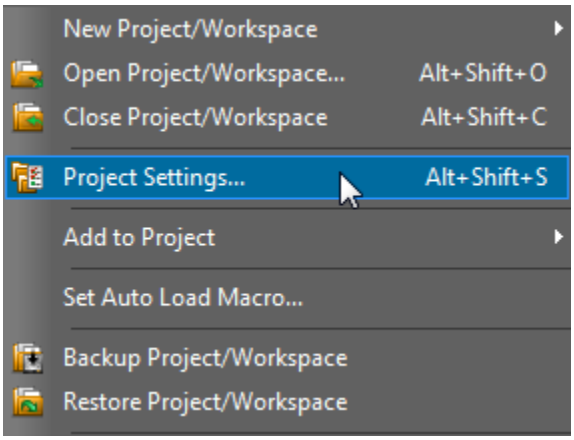
10. Create/edit the source file as you normally would. Depicted below is a simple header containing information for the program.
11. Select the “Save File” button or “File->Save” from the menu bar to save the newly created source file.



12. The “Save As” dialog will be displayed. Use the dialog navigation features to set the directory where your project was created earlier.
13. Enter the name of your source file in the “File Name” field making sure the PBP Source file matches the name of the UEStudio Project with the extension of “.pbp”. (E.g. TandTClock.pbp)
14. Select the “Save” button to complete the save of the file.

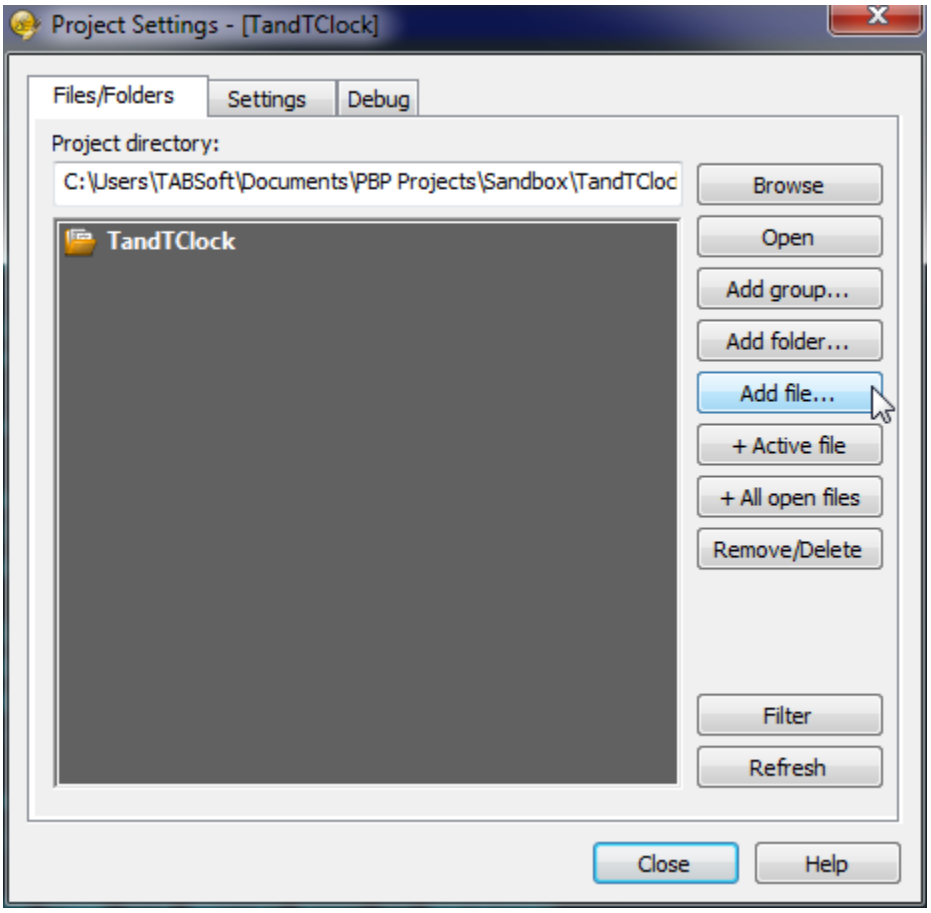

Save As
Libraries ▸ Documents ▸ PBP Projects ▸ Sandbox ▸ TandTClock ▸
Search TandTClock
Organize
New folder
UEStudio
Favorites
Desktop
Downloads
OneDrive
Recent Places
Job Search
Contract Work
Projects
Libraries
Documents
Music
Pictures
Videos
Documents library
TandTClock
Arrange by: Folder
Name
Date modified
Type
Configs
3/2/2015 10:38 AM
File folder
TandTClock.prj
3/2/2015 10:35 AM
PRJ File
TandTClock.TABSOFT-PC.TABSoft.pui
3/2/2015 10:35 AM
PUI File
File name: TandTClock.pbp
Save as type: All Files, (*.*)
ADS Stream:
Hide Folders
Encoding: Default
Save
Cancel

15. Next you will need to add the newly saved file above to your UEStudio Project. Select the "Project->Project Settings" option from the menu bar.

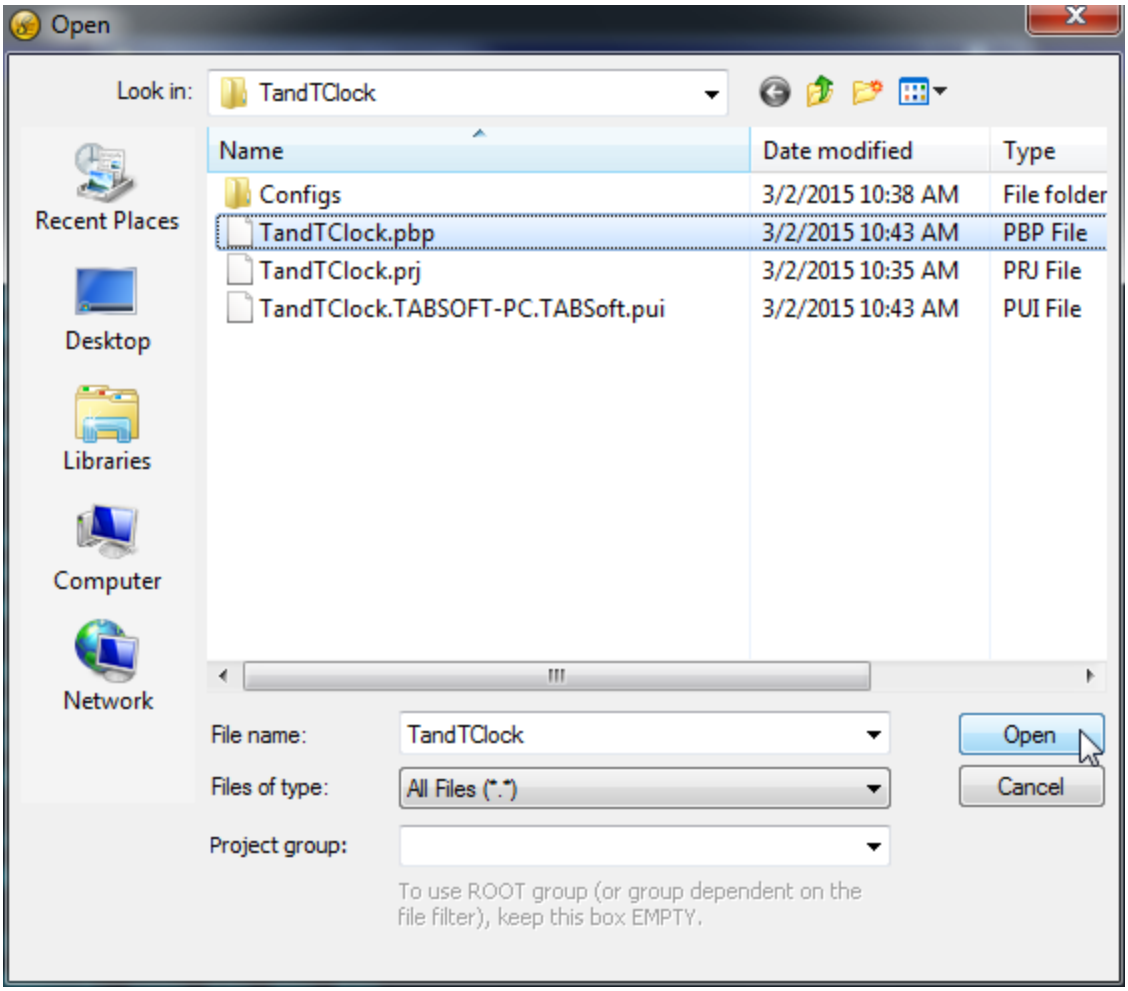


16. The "Project Settings" dialog will be displayed.

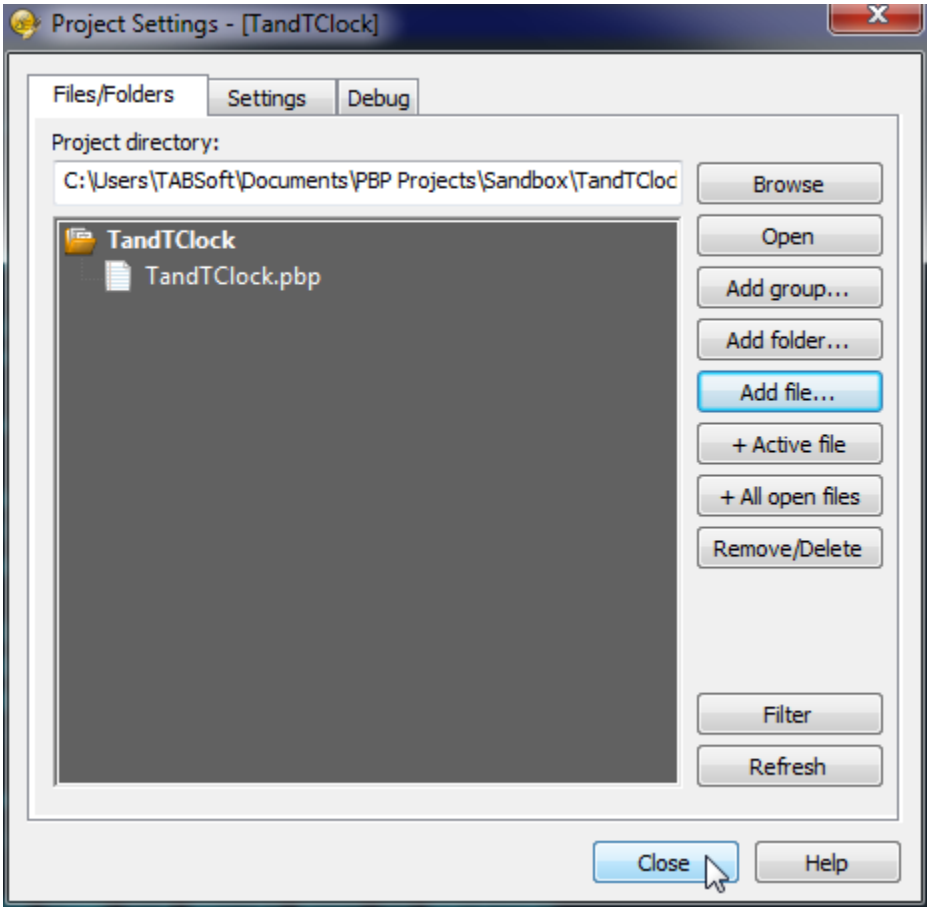
17. Select the “Files/Folders” tab.
18. Click the “Add file…” button.



19. The “Open” file dialog will be displayed.
20. Use the file/folder navigation features to select the PBP source file you created earlier and click the “Open” button.



21. You will now see in the “Files/Folders” tab of the “Project Settings” dialog that your source file has been added to your Project Tree.
22. Click the “Close” button to close the dialog.
23. You are now ready to finish coding your PBP program source file as you normally would for development.


Project Settings - [TandTClock]
Files/Folders
Settings
Debug
Project directory:
C:\Users\TABSoft\Documents\PBP Projects\Sandbox\TandTCloc
Browse
TandTClock
TandTClock.pbp
Open
Add group...
Add folder...
Add file...
+ Active file
+ All open files
Remove/Delete
Filter
Refresh
Close
Help

Compiling a PBP Project

Now that the earlier tasks of creating a PBP project, adding the source code file and completion of the code development has been completed, you can compile your new PBP project in two ways.

CompileOnly: Simply compiles the source code file using the PBPX Compiler/MPASMWIN tool chain.

Compile_Program: Compiles the source code file and launches the U2 Programmer application and sets the target device and loads the newly compiled .hex file.

Before compiling the PBP source code file, the user will want to set the appropriate Compiler Options for their specific PBP program. The PBP Compiler Configfile installed earlier defines several compiler options that the user can select for the PBP program.

Compiler Options:

- BuildMode : Select "CompileOnly" to compile the PBP source file; Select "Compile_Program" to compile the PBP source file AND launch the PIC Programmer.
- Processor : Select the type of PIC MCU target device.
- UseLongs : Select "no" to compile with PBPW; Select "yes" to compile with PBPL (PIC18 Only).
- SourceLevelDebug : Select "no" to disable source-level debug code in COFF file; Select "yes" to enable source-level debug code in COFF file.
- ShowProgress : Select "yes" to enable MPASMWIN Progress Window; Select "no" to disable MPASMWIN Progress Window.
- AutoClose : Select "yes" to automatically close the MPASMWIN Progress Window.
- Verbose : Select "no" for standard messages; Select "yes" for detailed messages.

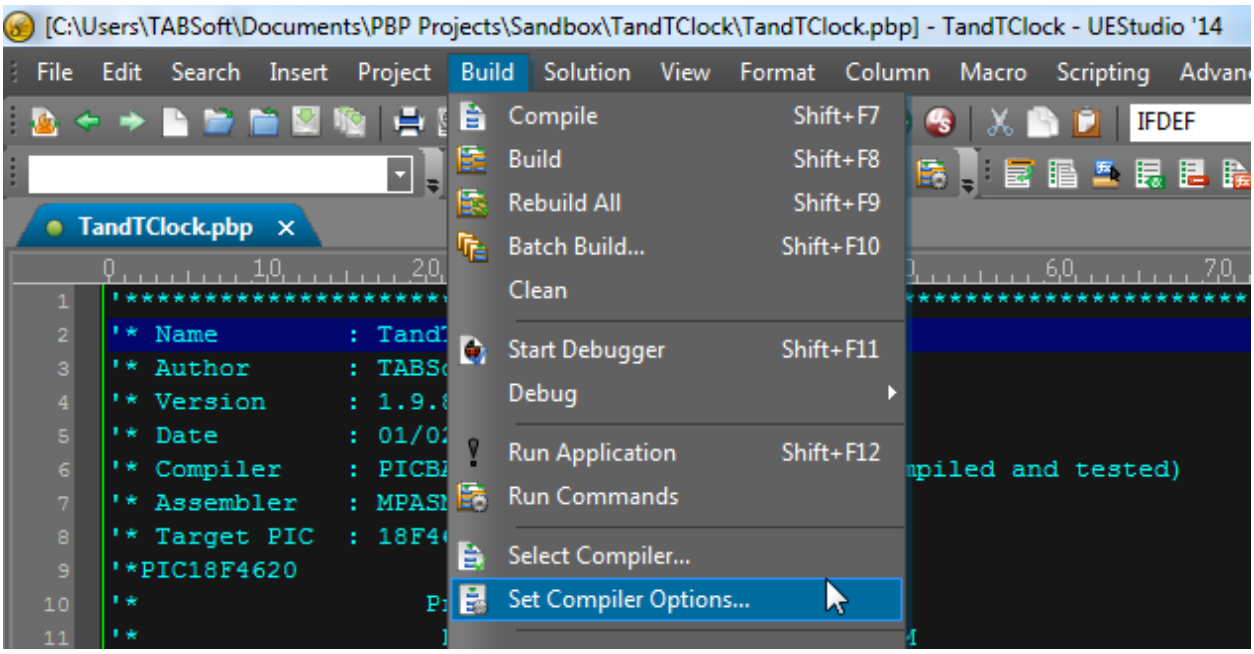
NOTE: The “Processor” option is a list of PIC MCU target devices. The included “PBP Application” Compiler Configfile adds just a few sample PIC MCUs. The user can edit the Configfile to delete or add the target MCUs of their choice.

CompileOnly Example

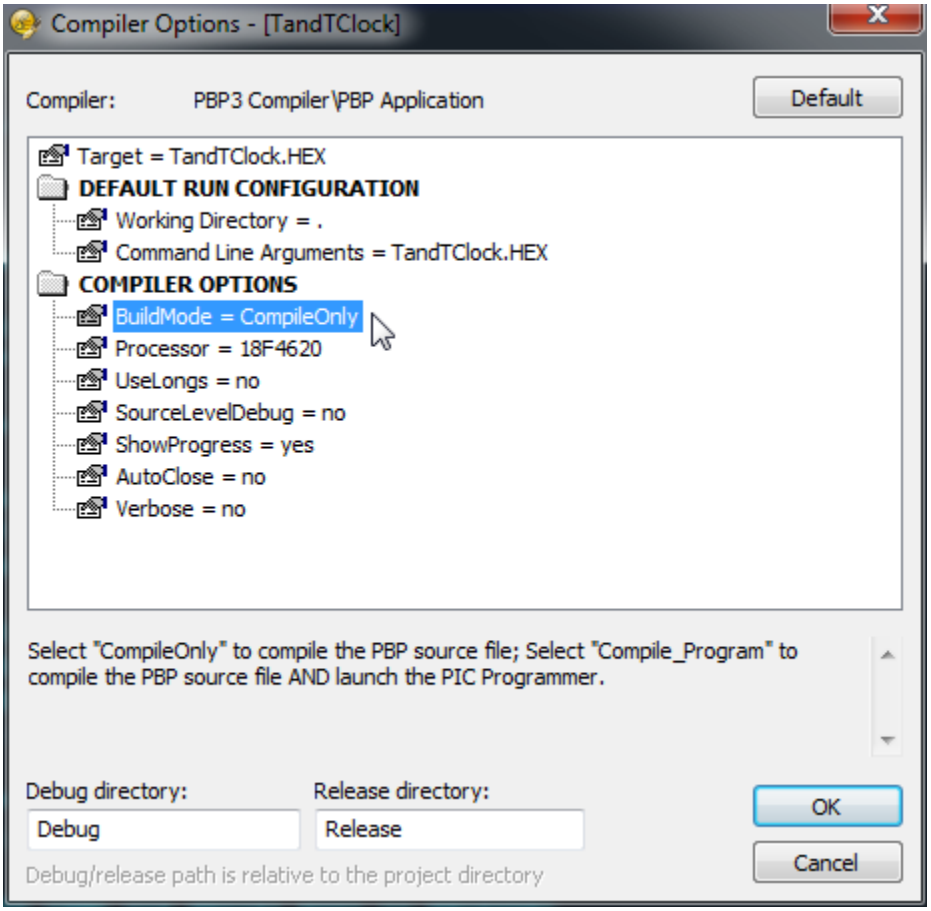
To compile a PBP program which will create a .hex PIC file, the UEStudio Compiler tool chain will use the “PBP Application” Compiler Configfile and the Compiler Options selected by the user to invoke the PBPX compiler and MPASMWIN assembler tool chain.

The steps to compile the PBP program are as follows:

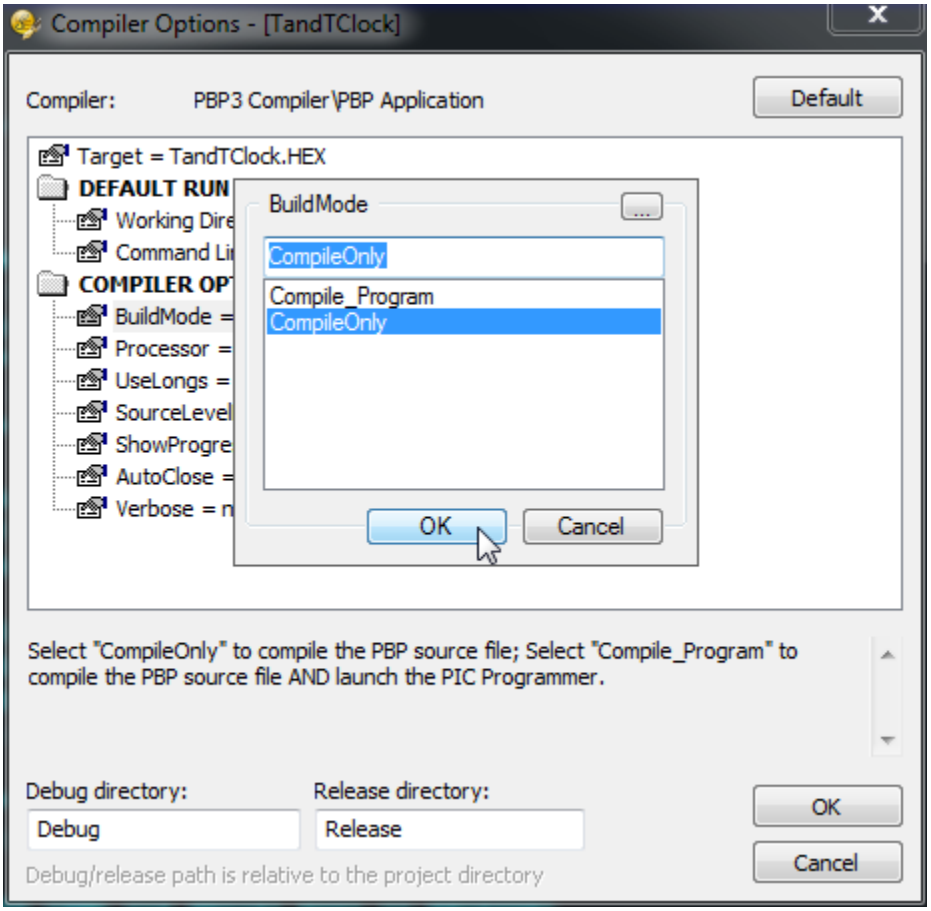
1. From UEStudio, select the “Build->Select Compiler Options” from the menu bar.



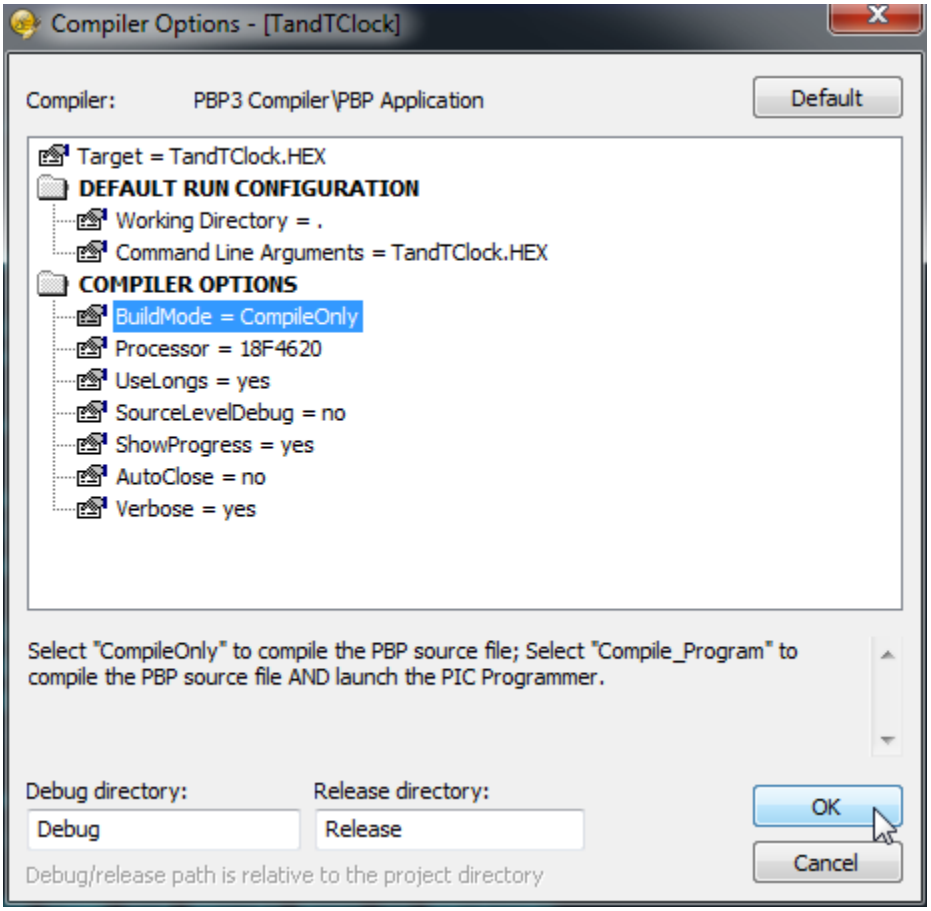
2. The “Compiler Options” dialog will be displayed.
3. Modify the Compiler Options as needed by double clicking the option from the dialog. A specific option list dialog will be displayed for that particular option and the user can select the appropriate option key.
   NOTE: If you single click on the option a description of the option is displayed.
4. To select the “BuildMode = CompileOnly” option, double click on “BuildMode” in the tree list. The “BuildMode” option list will be displayed.



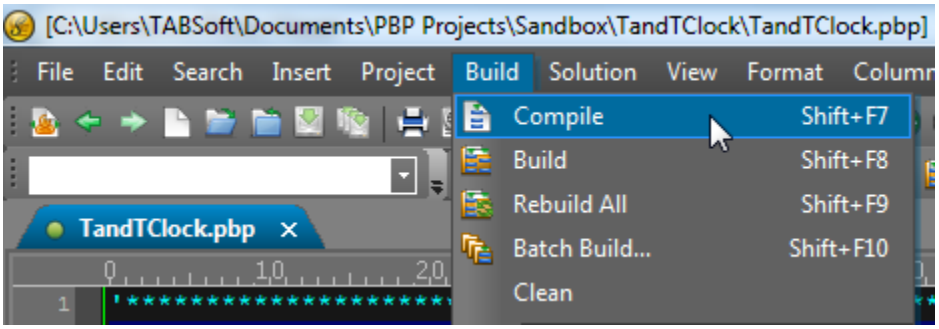
5. Click on “CompileOnly” and click the “OK” button.



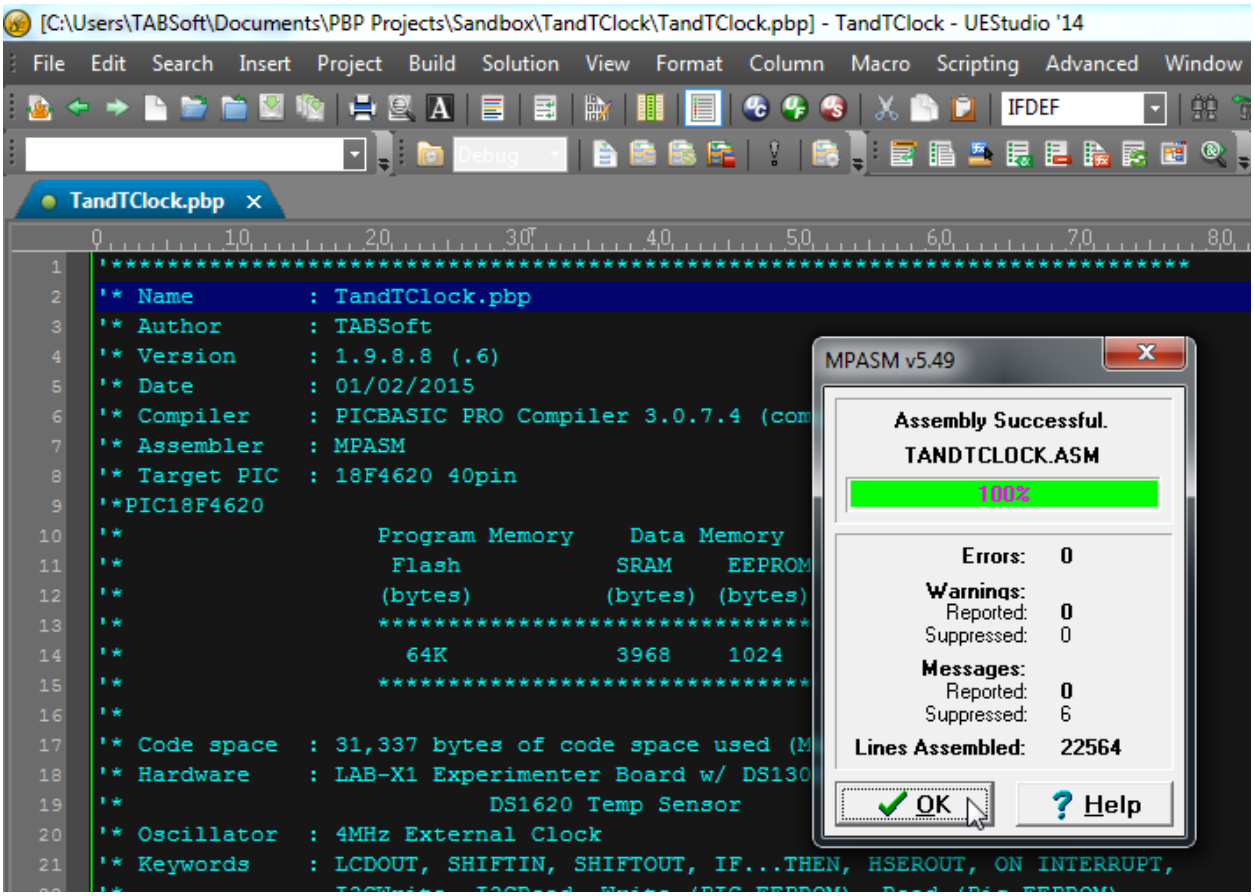
6. Modify any other Compiler Options as necessary.
7. Click the “OK” button from the “Compiler Options” dialog.



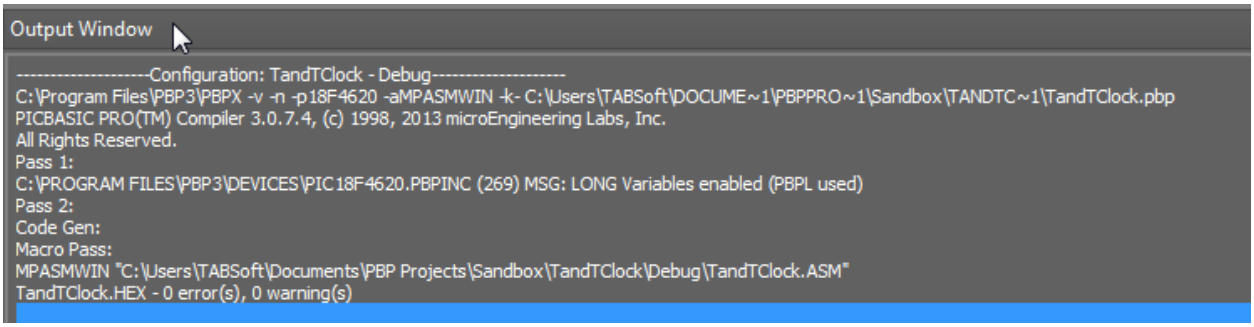
8. Now to compile the project, select the “Build->Compile” option from the menu bar.



9. The compile process will start by calling the PBPX compiler with the Compiler Options set earlier and then will invoke the MPASMWIN assembler/linker. The MPASMWIN progress dialog will be displayed. Once the process is complete, click the “OK” button on the progress dialog.



10. The UEStudio Output Window at the bottom of the screen will show the captured output including error messages during the compile process.
    NOTE: The output .hex" file will be located in the Release or Debug subdirectory in your project directory. This will be dictated by the "Debug/Release" Build Mode set in the Project Settings.



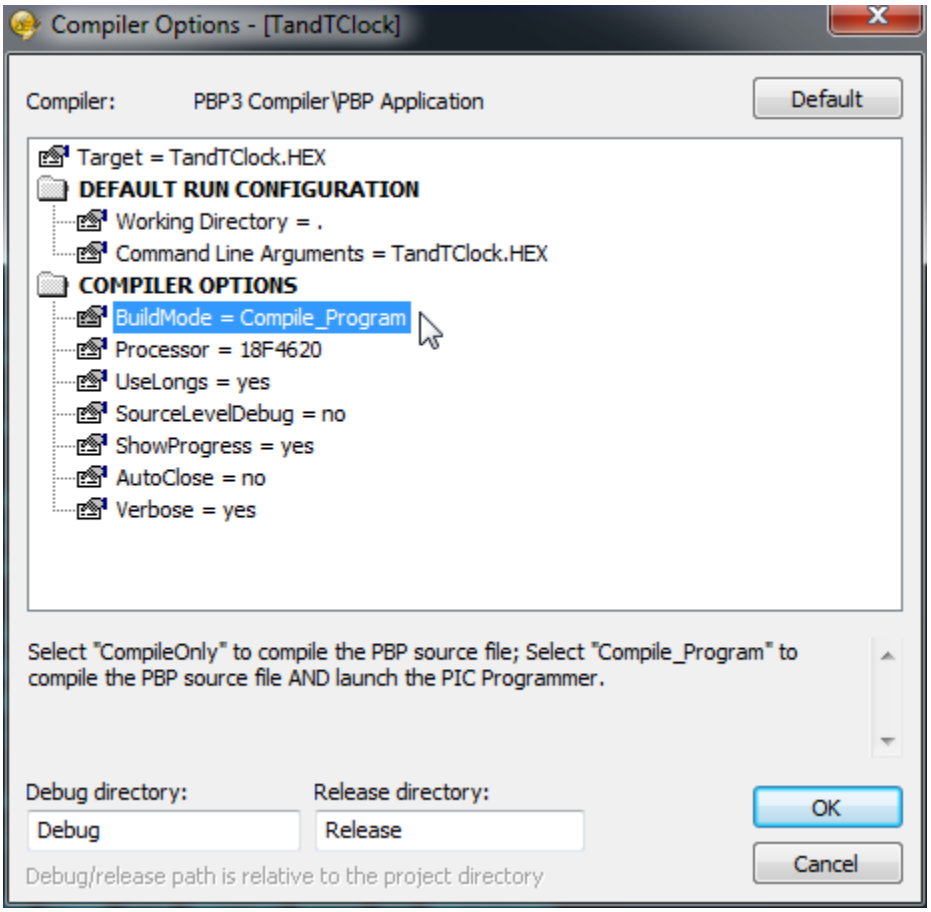
Compile_Program Example

To compile a PBP program which will create a .hex PIC file and launch the U2 Programmer application, the UEStudio Compiler tool chain will use the “PBP Application” Compiler Configfile and the Compiler Options selected by the user to invoke the PBPX compiler and MPASMWIN assembler tool chain and finally launch the U2 Programmer application.

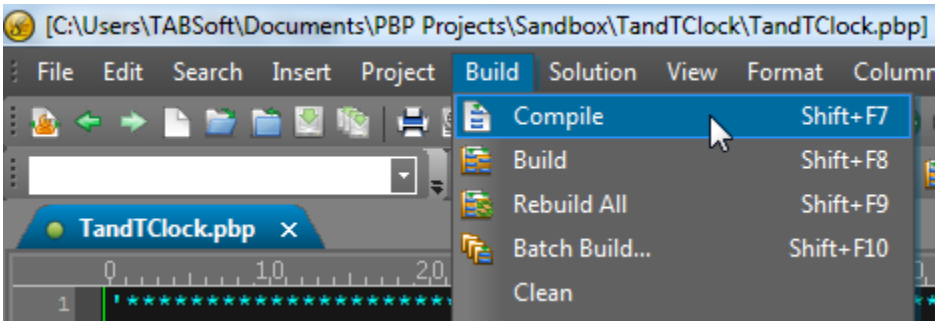
NOTE: The included “PBP Application” Configfile sets up the use of the MELABS U2 Programmer application to be called from the command line automatically. Other PIC Programmer applications may be used instead if they support launching the application from a command line.

The steps to Compile and Program the PBP program into the target PIC MCU are as follows:

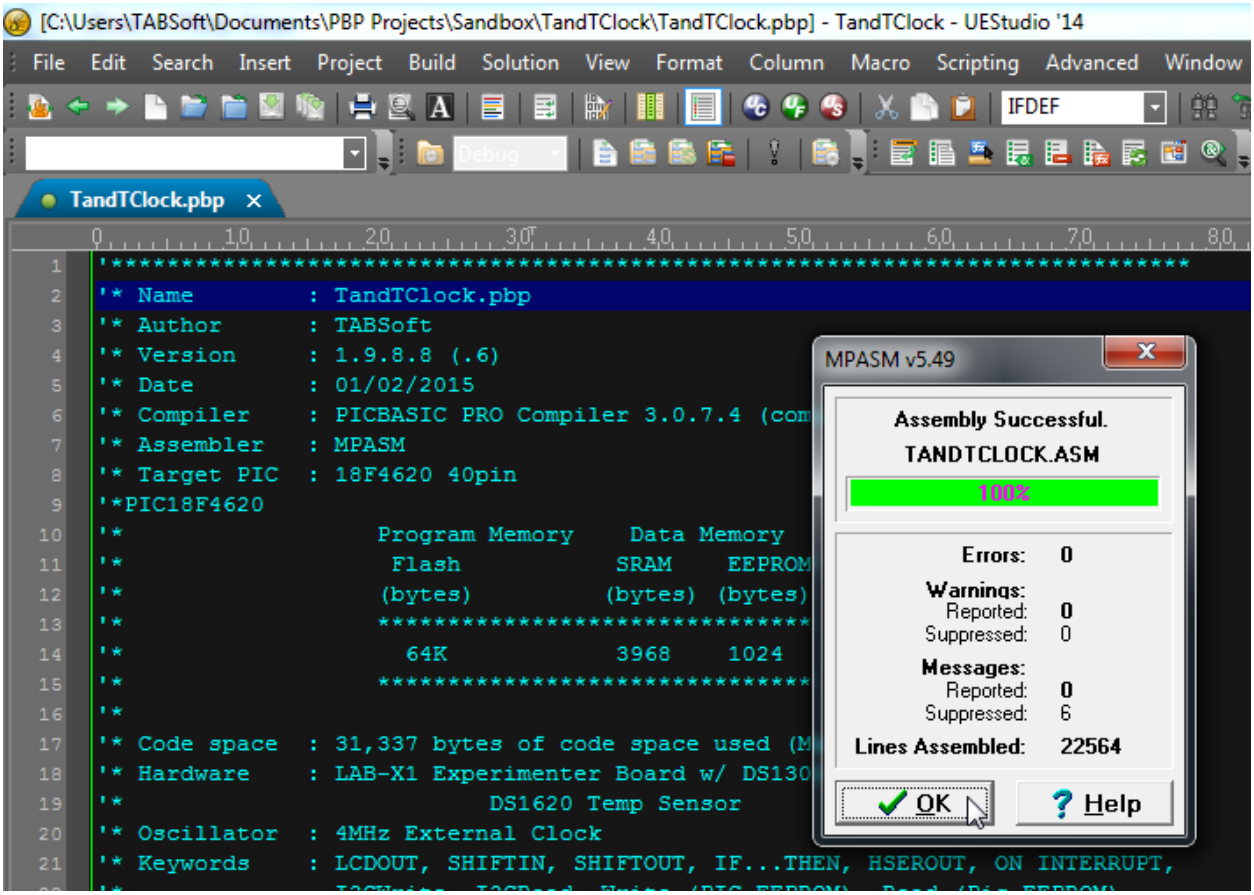
1. From UEStudio, select the “Build->Select Compiler Options” from the menu bar.
2. Change the BuildMode option to “Compile_Program” in the Compiler Options dialog.
3. Select the “OK” button from the Compiler Options dialog.



4. Select the “Build->Compile” option from the menu bar.
5. The compile process will begin just as in the “CompileOnly” example.



6. When the compile has completed, click the “OK” button from the MPASMWIN progress dialog.



7. The compile process will now launch the U2 Programmer application. It will select the Target PIC MCU (determined by the Compiler Options earlier) and load the .hex file created during the compile process previously (the appropriate .hex file will be loaded either from the Debug or Release folder depending upon the UEStudio Debug/Release Build Mode option in the Project Settings for the particular project).

8. The user can then conduct normal actions from the U2 Programmer application.
   NOTE: The UEStudio Output Window displays the command line used to invoke the U2 Programmer application.
9. Once the PIC Programming has been completed, the user can close the U2 Programmer application and return to UEStudio.
   NOTE: UEStudio simply launches the programmer application from a command line. The application is running independently from UEStudio once it is launched.