MPLAB X Appendix

MPLAB® X IDE Quick Reference Guide

Table of Contents

MPLAB X Appendix

This appendix is intended to be a supplement to lab manuals supplied with a Microchip Technical Training class. Although it may be useful on its own, it is not intended to provide complete instructions for using all aspects of the MPLAB X Integrated Development Environment. For more detailed information on the use of MPLAB X IDE, please consult one of the following Microchip Technical Training classes (for additional details, see **http://www.microchip.com/RTC**):
TLS0101—Getting Started with MPLAB X IDE
TLS0999—Transitioning to MPLAB X IDE for users of MPLAB IDE version 8
Or consult the Microchip Developer's Help Center at **http://microchip.wikidot.com/mplab:_start**

1. Managing Projects

Section 1.1
How to open a project

Unlike some other IDEs, there is no single icon you can double click on from your operating system's file manager. MPLAB X Projects must be opened from within the IDE.

1. There are several methods you can use to launch the **Open Project** dialog:
   Method 1:
   Click on the **Open Project** icon on the main toolbar:
   Method 2:
   From the menu, select **File ▸ Open Project**
   Method 3:
   Using the keyboard: Ctrl + Shift + O

2. Navigate to the project's directory and select the directory itself, which is represented by a chip icon instead of the usual folder. In MPLAB X IDE, there is no single project document, so the project's directory is used to represent the project in the **Open Project** dialog.

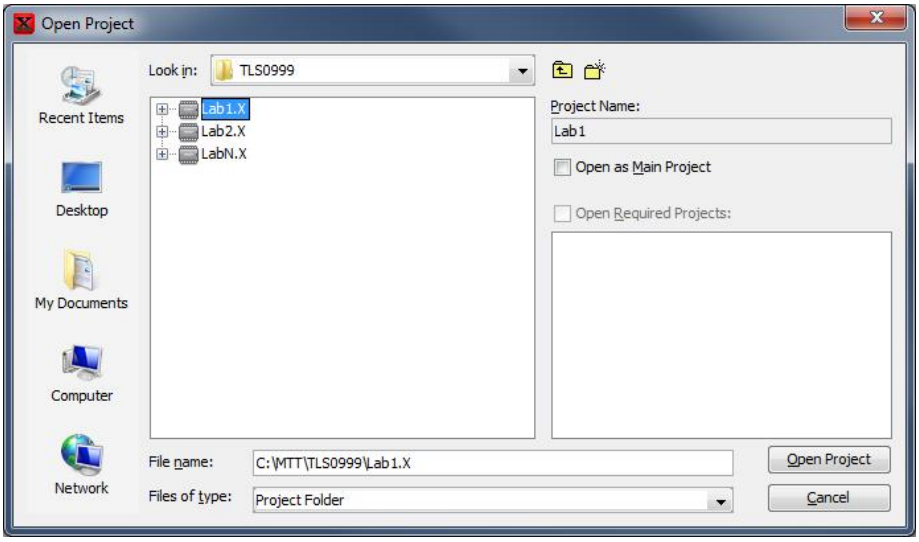


Figure 1.1.1
An Open Project dialog showing three projects in the TLS0999 directory

Information

In most Microchip Technical Training classes, projects are stored in the following directory:

C:\MTT*classcode*

where *classcode* is specified in the presentation or lab manual.

3. Click on the **Open Project** button. You should now see a populated project tree in the IDE (you may need to click on the '+' next to the chip icon to expand the subfolders.

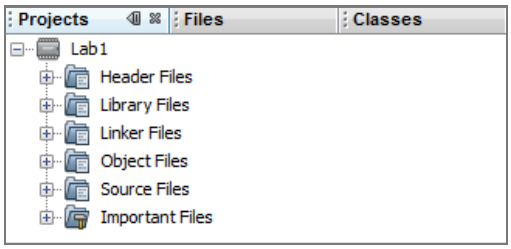


Figure 1.1.2
A populated project tree after opening a project

Section 1.2

How to add existing files to a project

Attention

Do not select **Add Existing Items from Folders...** as this will dump everything from the directory you choose into the selected logical folder.

1. Right click on the logical folder in the project tree (e.g. Source Files, Header Files, etc.) where you wish to add the file(s) and select **Add Existing Item...** from the popup menu.

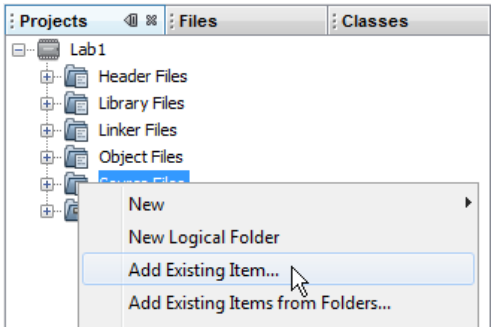


Figure 1.2.1
Popup menu displayed after right clicking on the Source Files logical folder in the project tree

2. The Select Item dialog box will open and display the contents of your project directory. Select one or more files (Ctrl + click to select additional files).

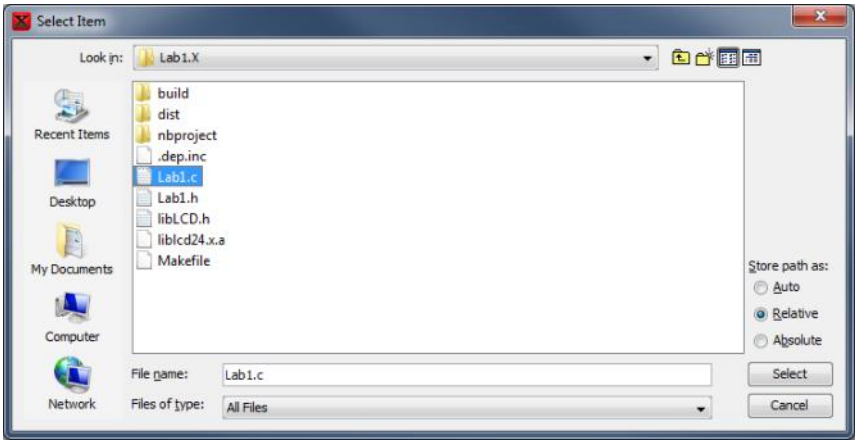
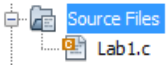


Figure 1.2.2
The Select Item dialog

In most cases you can leave the **Store path as:** radio buttons set to **Relative**, but you may select **Auto** or **Absolute** if required for your project.
Relative: Stores paths to files relative to the project directory. For example: **\Lab1.c**, or **..\OtherDirectory\Somefile.c**
Relative is usually the best choice for files inside your project directory.
Absolute: Stores paths to files with full path from root directory. For example: **C:\MTT\TLS0999\Lab1\Lab1.c**
Absolute is the best choice for files outside your project directory that won't be moved such as code shared among several projects or libraries.
Auto: Automatically uses Relative for files inside the project directory and Absolute for files outside of the project directory.

3. Click the **Select** button. You should now be able to expand the selected logical folder in the project tree and see that the files have been added to your project.



Section 1.3

How to create new files in a project

1. There are three methods you may use to launch the new file wizard:

Method 1:

Right click in the Projects window and select **New** ▸ ***file-type*** from the popup menu:

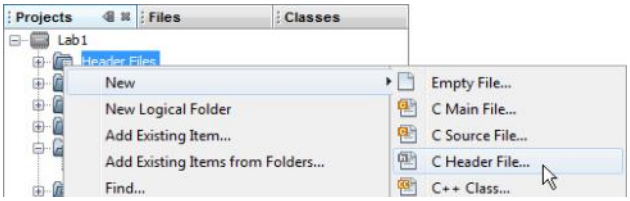


Figure 1.3.1
New file popup menu

Method 2:

Click on the New File icon on the main toolbar: 

Method 3:

From the main menu select **File** ▸ **New File...**

Method 4:

Using the keyboard: Ctrl N

2. If you chose method 1 above, skip to step 3. If you chose one of the other methods above, you will be presented with the screen shown in Figure A.6 below. In the **New File** dialog, select the type of file you wish to create. Any file created under the Microchip Embedded category or an Empty File from the Other category will be automatically added to the project tree. Other file types may need to be added manually.

Click the **Next >** button after you have made your selection.

Information

You can create a "main" file anytime you like and just delete the code that is automatically inserted to make it a regular C file.

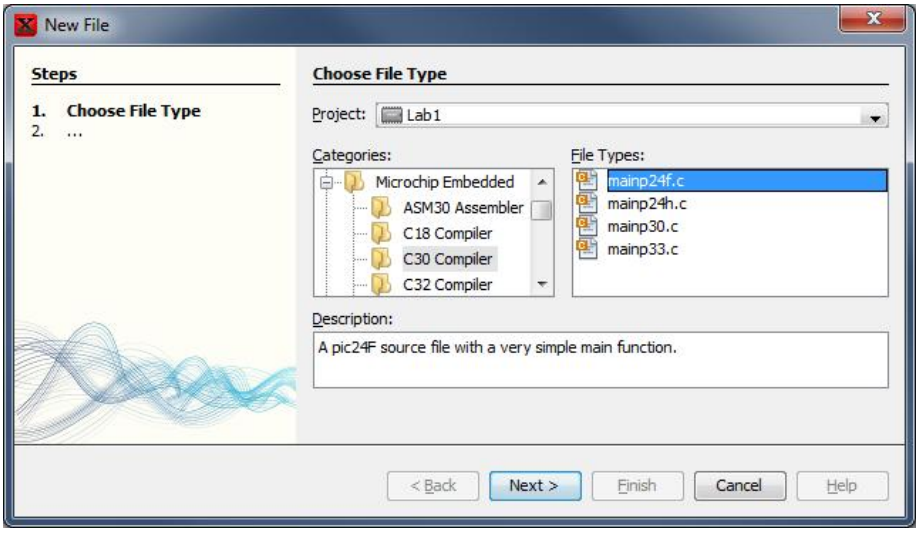


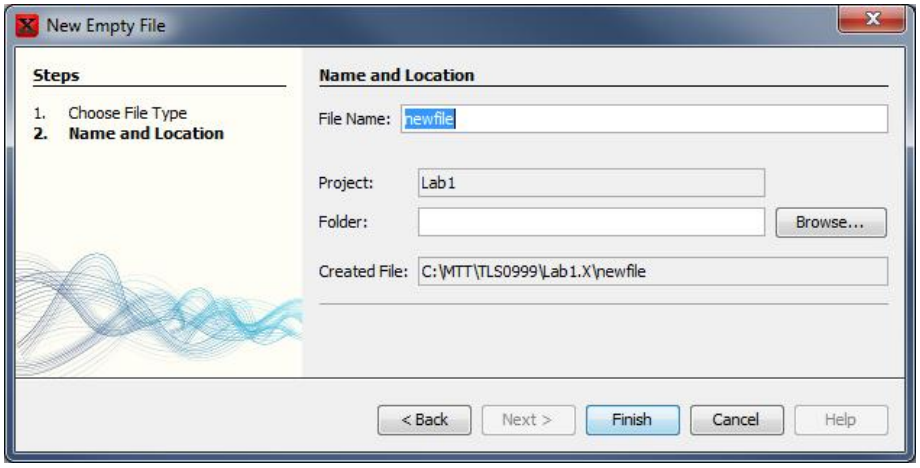
Figure 1.3.2
New File wizard—Choose File Type

3. The dialog now prompts you for a file name and potentially a file type depending on your initial selection. It also prompts you for a folder. You can leave this blank and the IDE will automatically create the file inside your project directory. If you wish to locate the file elsewhere, click on the **Browse...** button and choose a different location.

If you chose **Empty File** as the new file type, you will see the dialog in Figure A.7 on the next page.

If you chose one of the types under the **Microchip Embedded** category, you will see a dialog like the one in Figure A.8 on the next page.

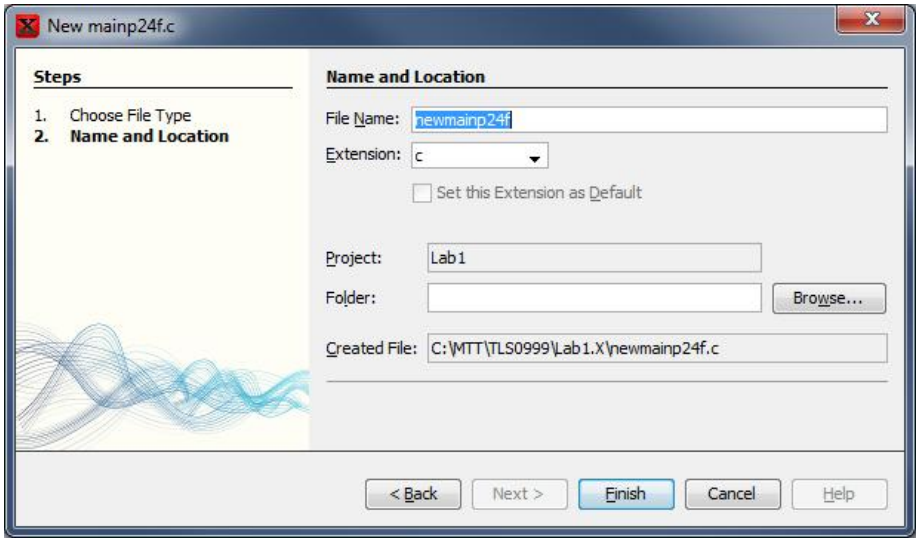
Figure 1.3.3
New Empty File Dialog



Attention

You must specify a file extension (e.g. myNewFile.c) as part of the File Name if you want the file to be added to the project tree automatically into the correct logical folder.

Figure 1.3.4
New Main File Dialog (Microchip Embedded category)

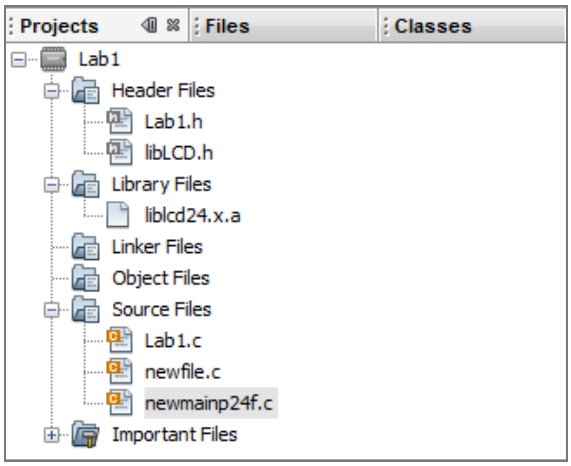


Information

It is not necessary to specify the file extension as part of the filename as long as the correct extension is selected below.

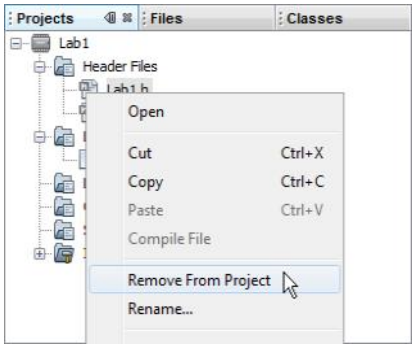
4. After providing all of the required information, click the **Finish** button. You should now see the new file in the project tree. If you don't see it, you may need to add the file to the project (see page A-4)

Figure 1.3.5
Project tree with newfile.c and newmainp24f.c added to the project.



Section 1.4
How to remove a file from a project

1. Right click on the file you wish to remove and select from the popup menu **Remove From Project**



DANGER!

DO NOT use the **Delete** key to simply remove a file from your project. This will also ***PERMANENTLY DELETE*** the file from your hard drive. It will not be recoverable from the Trash folder.

Figure 1.4.1
Project right click menu with Remove From Project selected

Section 1.5
How to *permanently* delete a file

1. Select a file in the project tree and press Delete

This will permanently delete the file from you system in addition to removing it from the project. The file will not be recoverable from the Trash.

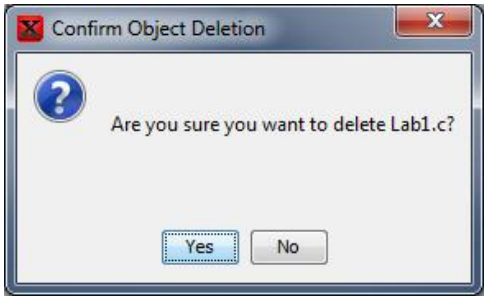


Figure 1.5.1
Confirm delete dialog. This will be your only warning.

Section 1.6
How to save a file or project

1. The entire project along with all of its files are saved automatically each time you build your code. However you may explicitly save the project and all its files using one of two methods.

Method 1:
Click the "double floppy disk" icon in the toolbar: 

Method 2:
From the main menu select **File ▸ Save All**

Though it is not usually necessary to just save a single file, you can do so by selecting the file in the editor and then from the main menu **File ▸ Save**.

Section 1.7
How to close a project

1. There are two methods you can use to close a project:

Method 1:

Right click on the top node of the project in the project tree (the chip icon) and select **Close** from the popup menu (about 2/3 of the way down).

Method 2:

From the main menu, select **File ▸ Close Project (*project name*)**

where ***project name*** is the name of the project you wish to close—there may be multiple similar menu items if you have more than one project open in the IDE.

Section 1.8
How to modify project settings

1. There are three ways to access a project's settings:

Method 1:

Right click on the top node (chip icon) of a project in the project tree and select **Properties** at the very bottom of the long popup menu.

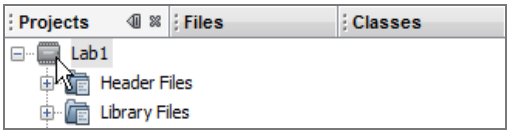


Figure 1.8.1
The top node of a project in the project tree

Method 2:

From the main menu select **File ▸ Project Properties (*project name*)**

Method 3:

If the **Project Environment** window is open (bottom left corner by default), you can click on the "wrench and bolt" icon in its left margin.

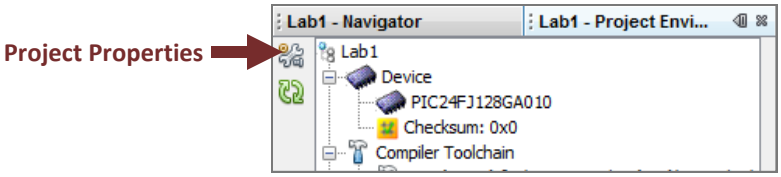
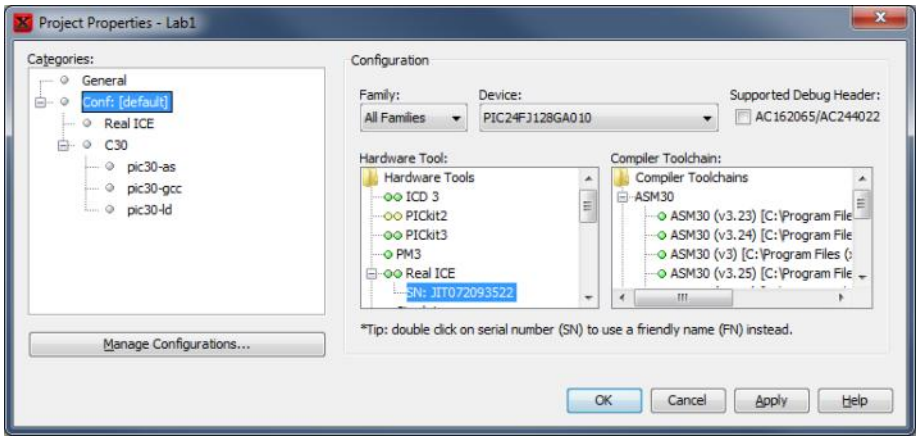


Figure 1.8.2
Project Properties button in the Project Environment window.

2. From here you can select a different device, debug tool or build tool and you can modify any of their settings. When choosing a new tool, click the Apply button to make it show up in the tree on the left side.



2. Building Projects

Section 2.1
How to build a project

There are several ways to build a project in MPLAB X depending on what you intend to do with the results. This method is only used to see the results of a build or to produce a release mode hex file.

There are two different types of build you can do:
Build:
This will build only the files in your project that have changed since the last build or it will build everything if nothing has been built previously. It will generally be faster to use this type of build, especially for larger projects.
Clean and Build:
This will remove any intermediate files generated by the previous build and will build every file in your project regardless of whether or not it has changed since the last build to ensure a full, clean build.

Information

The **Build** and **Clean and Build** functions are not intended for use before **Run Project**, **Debug Project** or **Make and Program Target**. All three of those functions automatically do a build before performing further steps. No harm will be done by using **Build** or **Clean and Build** first, but it will be a duplication of effort and will waste time.

1. There are two ways to access these two build types:
From the Main Toolbar:


Build — ***Make*** in MPLAB IDE 8


Clean and Build — ***Build All*** in MPLAB IDE 8

Alternatively, you can right click on the top node of a project (chip icon) in the project tree and select either **Build** or **Clean and Build** from the popup menu.

Section 2.2
How to build and run a project with a debugger

When you want to build a project for the purpose of programming a target to run with a debugger like the MPLAB® ICD 3 or REAL ICE, this is the method to use.

Information

It is not necessary to do a **Build** or **Clean and Build** before doing a **Debug Project** because a build will be done automatically.

1. There are three ways to build and run your code through a debugger:
Method 1:
Click on the **Debug Project** button on the main toolbar 
Method 2:
Right click on the top node of the project (chip icon) in the project tree and select **Debug** from the popup menu.
Method 3:
From the main menu, select **Debug ▸ Debug Project (*project name*)**

This will perform the following tasks automatically:
a. Build (make) project in debug mode
b. Program target (unless using simulator)
c. Run code on target

Section 2.3

How to build and run a project without a debugger

Information

It is not necessary to do a **Build** or **Clean and Build** before doing a **Run Project** because a build will be done automatically.

When you want to build a project for the purpose of programming a target to run without a debugger, this is the method to use.

1. There are three ways to build and run your code on a target:

Method 1:

Click on the **Run Project** button on the main toolbar 

Method 2:

Right click on the top node of the project (chip icon) in the project tree and select **Run** from the popup menu.

Method 3:

From the main menu, select **Run ▸ Run Project (*project name*)**

This will perform the following tasks automatically:

a. Build (make) project in release mode
b. Program target
c. Run code on target

3. Debugging Projects

Section 3.1
How to set or change the debugger

1. Open the project properties window (see page A-8 “How to modify a project’s settings” for details)

2. In the center column under “Hardware Tool”, click on the serial number under the name of the tool you wish to use. If choosing the simulator, just click on “Simulator” since no serial number is associated with it. Some tools do not provide a serial number. In that case, click on the text right below the name of the tool (see PICkit 2 in the figure below for an example).

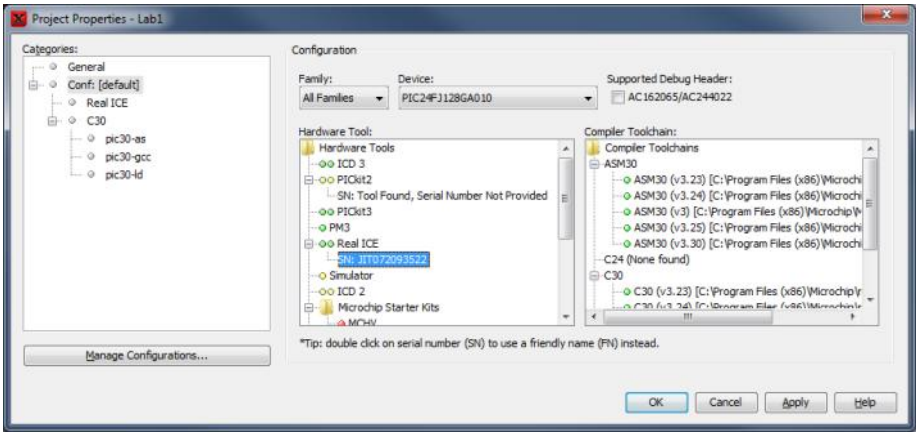


Figure 3.1.1
Selecting a hardware tool in the Project Properties window

3. Click on the **Apply** button after you make your selection and you should see the selected tool in the tree of the left column.

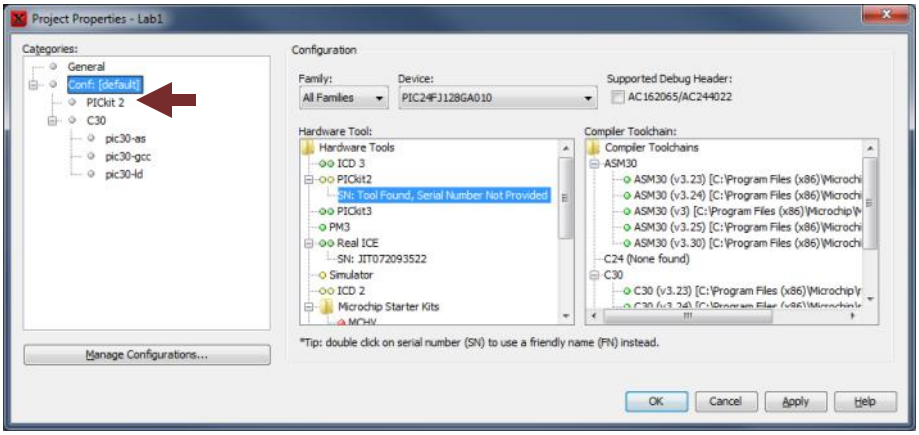


Figure 3.1.2
Changing a hardware tool in the Project Properties window

4. Clicking on the tool in the tree of the left column will display the tool’s properties in the right side of the window, where you can modify them to suit your project’s needs. Click **OK** when finished.

Section 3.2

How to control program execution when debugging

> **Information**
>
> When you hover the mouse pointer over a toolbar button a tool tip will be displayed explaining the function of the button.

Debug Toolbar Buttons

Finish Debug Session (Shift + F5)
This is required before you make any changes to your project settings or source code.

Pause

Reset

Continue (F5)

Step Over (F8) - Execute each line without stepping into functions (functions are executed without stepping through each line)

Step Into (F7) - Execute each line and step into functions

Run to Cursor (F4)

Set PC at Cursor

Focus Cursor at PC

Additional functions may be found in the **Debug** menu.

Section 3.3

How to set and clear breakpoints

1. Standard line breakpoints may be set or cleared by clicking on the line number in the glyph margin.

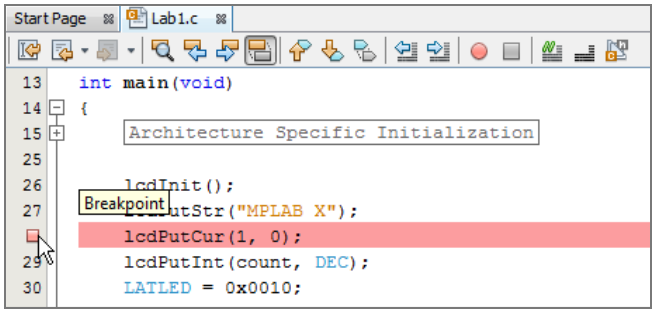


Figure 3.3.1
Setting / clearing a breakpoint

More advanced breakpoint features may be accessed by opening the breakpoints window. From the main menu select **Window ▸ Debugging ▸ Breakpoints**.
In the breakpoint window, right click on a breakpoint in the list and select **Customize** or **Complex Breakpoint** from the popup menu for advanced options.

Section 3.4

How to use the stopwatch

1. From the main menu select **Window** ▸ **Debugging** ▸ **Stopwatch**

2. In the stopwatch window, click on the Properties button in its margin.

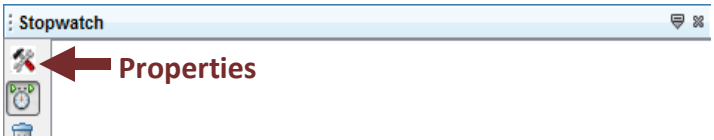


Figure 3.4.1
Stopwatch properties button

3. Select existing breakpoints for the start and stop conditions

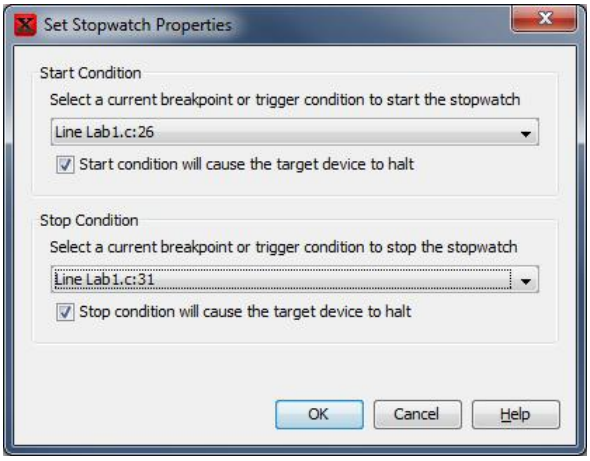


Figure 3.4.2
Stopwatch properties window

4. Run your code. The cycle count will be displayed each time you hit one of the selected breakpoints. The Trash icon in the margin will reset the cycle count.

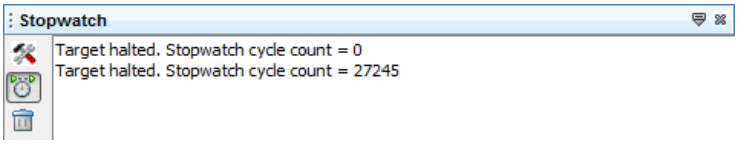


Figure 3.4.3
Stopwatch displaying results from the starting and ending breakpoints

Section 3.5

How to display and use Watches

1. From the main menu select **Window** ▸ **Debugging** ▸ **Watches** or from the keyboard press Alt ⇧ Shift 2

2. The **Watches** window should appear as a new tab near the **Output** window in the bottom center of the IDE. There are three ways to add a watch value:

Method 1:

Double click on the first empty line in the **Name** column and type in the name of the variable/register

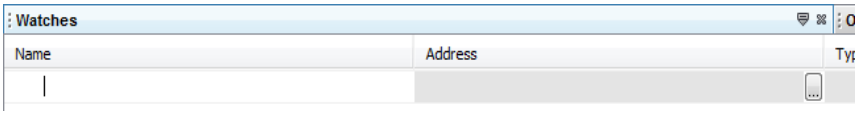


Figure 3.5.1
Adding a watch variable directly to the watches window

Method 2:

Double click on a variable or register name in the editor to select it. Then left click and drag it to the Watches window.

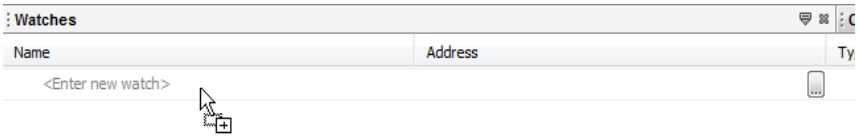


Figure 3.5.2
Dragging a variable from the editor to the watches window

Method 3:

Right click on a variable/register name in the editor or on a new line in the **Watches** window and from the popup menu select **New Watch…** , then in the window that opens, enter the desired variable/register name in the text box or select it from the list by choosing **Global Symbols** or **SFRs**

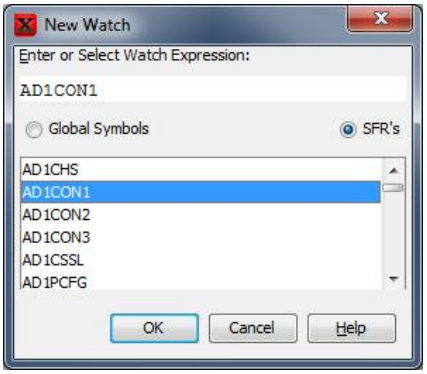


Figure 3.5.3
New Watch dialog box

Information

Watch variables may be sorted by clicking on the **Name** column.

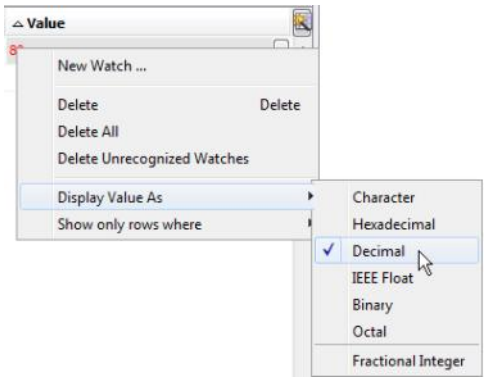


Figure 3.5.4
Changing the Value display format

Information

To change the display value, right click on an entry in the Value column and select **Display Value As** from the popup menu.

Section 3.6

How to view Embedded Memory

(RAM, SFRs, Flash, EEPROM or Configuration Bits)

1. From the main menu select **Window ▸ PIC Memory Views ▸ Memory View *n*** where ***n*** is a value from 1 through 4. It doesn't matter which one you choose initially as they are all identical and configurable.

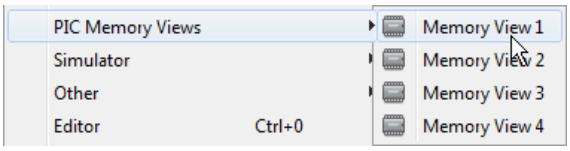


Figure 3.6.1
PIC Memory Views menu item

2. A new tab will open by default in the bottom center part of the IDE. At the bottom left of this window is a combo box labeled "Memory" that is used to configure this window to display any valid memory type for the currently selected device.

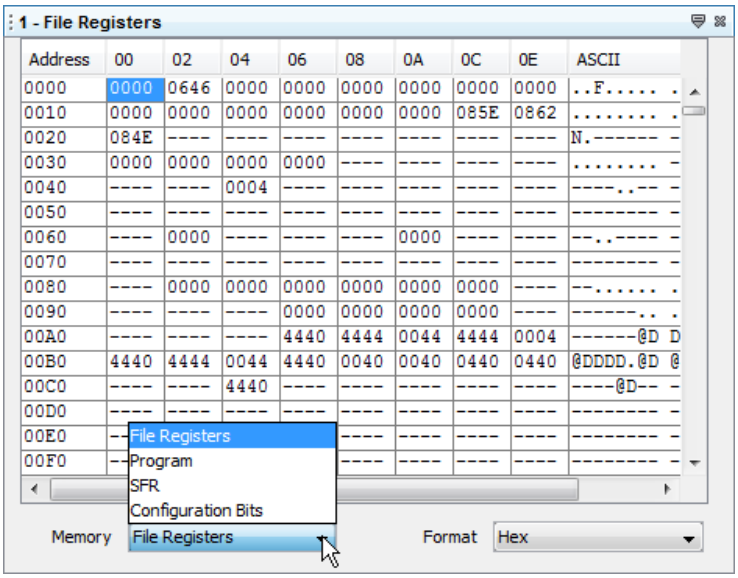


Figure 3.6.2
PIC Memory View showing File Registers with other options in the Memory combo box

Information

The **Format** combo box at the bottom right of the memory view window is different for each memory type, but will configure the display in a variety of numeric and symbolic formats.

MPLAB X Appendix