



# 1601 MPL

## Transitioning to MPLAB® X IDE For Users of MPLAB IDE 8.X

*Author:* **Rob Ostapiuk**  
*Microchip Technology*

# Objectives

- **Show the differences from MPLAB<sup>®</sup> IDE 8**
- **Configure tools and compilers**
- **Create and convert projects**
- **Work with new project features**
- **Build, program, and debug projects**
- **Work with the new editor**
- **Work with the new navigation tools**
- **Customize the interface**
- **Explore some brand new features**

# Agenda

- **Rationale**
- **Major Differences**
- **IDE Overview**
- **Lab – Create a Standalone Project**
- **New Methods and Feature Tour**
  - Projects, Editor, Debugging and more
- **Multi-Project Notes**
- **Resources**



## Rationale

# Why make such a big change?

# Rationale

## Why make a completely new MPLAB® IDE?

- **MPLAB IDE 6-8 code base has run its course:**
  - IDE lacking several modern features
  - Difficult to add many requested features
  - Not easily extensible by 3<sup>rd</sup> parties
  - Lacking some infrastructure for advanced high-level language development
  - Limited to the Windows platform

# Rationale

## Why the NetBeans IDE?

- **The new platform provides:**
  - A mature IDE proven through years of use
  - A modern IDE with all the latest features
  - A well documented, extensible architecture
  - Many requested features already present
  - Architected for high-level language support
  - Runs on Windows, Mac and Linux

# Rationale

## Why the NetBeans IDE?

- **The new platform provides:**
  - A rich ecosystem of plug-ins
  - PC and embedded development in same IDE
  - File structure that doesn't require IDE
  - Many new features not previously available in MPLAB<sup>®</sup> IDE



# C Compilers for MPLAB<sup>®</sup> X IDE

## The new, simplified compiler offerings

# MPLAB<sup>®</sup> XC Compilers

## Coming in 2012

HI-TECH<sup>®</sup> PICC Based

**MPLAB XC8**

*8-bit Compiler*

PIC10, PIC12,  
 PIC16, PIC18

MPLAB C30 / GCC Based

**MPLAB XC16**

*16-bit Compiler*

PIC24,  
 dsPIC30, dsPIC33

MPLAB C32 / GCC Based

**MPLAB XC32**

*32-bit Compiler*

PIC32

***CCI – Common Compiler Interface***

**Free**

No cost, production worthy, optimizing compiler, community support

**Standard**

Entry level price, more optimization, access to priority support

**Professional**

Full price, whole program optimization, access to priority support



## Major Differences

**An overview of the most significant changes over MPLAB® IDE 8**

# Major Differences USB Drivers

## MPLAB® IDE 8

- ICDs and MPLAB REAL ICE™ in-circuit emulator use proprietary drivers
- PICkit™ programmer/debugger uses standard HID driver



## MPLAB X IDE

- ICDs and MPLAB REAL ICE™ in-circuit emulator use open source drivers
  - WinUSB on Windows
  - LibUSB on Mac and Linux
- PICkit programmer/debugger uses standard HID driver



# Major Differences

## USB Communications Link with Hardware Tools

### MPLAB® IDE 8

- Link established when tool selected or workspace opened with tool already selected
- Link severed when MPLAB IDE is closed
- Link is always on while tool enabled

### MPLAB X IDE

- Link established when debug session started 
- Link severed when debug session terminated 
- Link is always off except when actively debugging

*(By Default)*

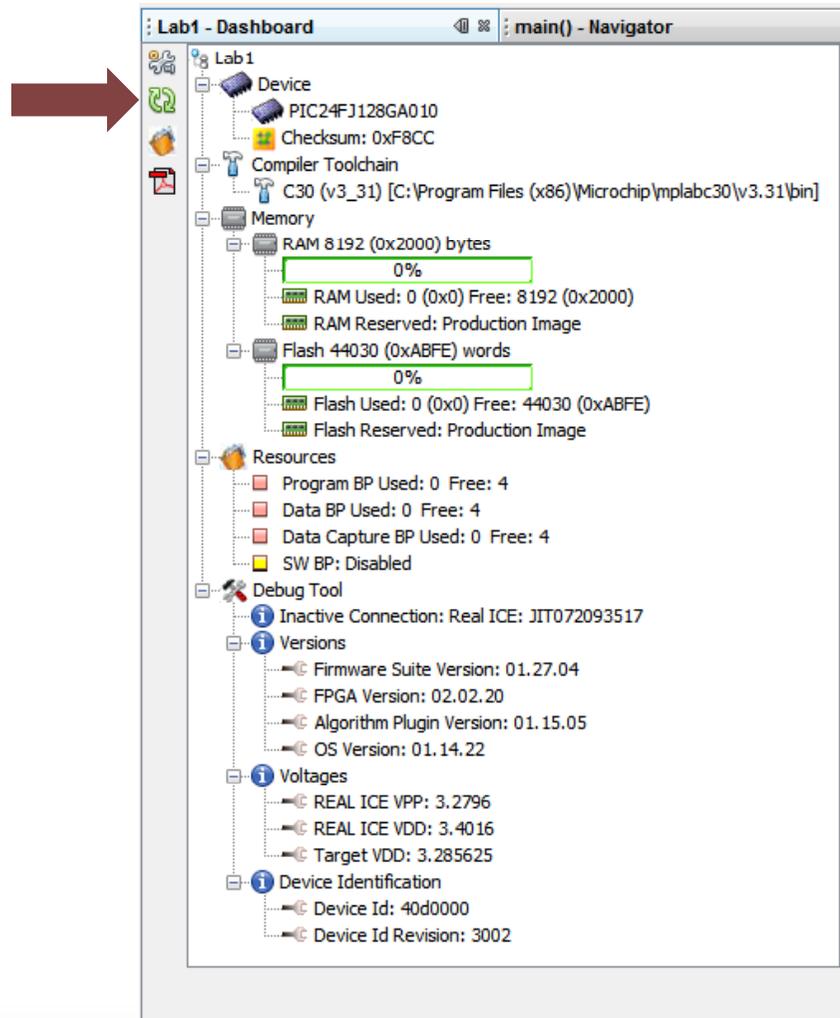
# Major Differences

## USB Communications Link with Hardware Tools

- Tool status can be refreshed without being in a debug session

**1** Open the project dashboard from the main menu:  
**Windows ► Dashboard**

**2** Click on the Refresh Debug Tool Status icon 



# Major Differences Projects and Workspaces

## MPLAB® IDE 8

- **Projects contain:**
  - Basic project settings, project files and locations, target device
- **Workspaces contain:**
  - Open projects, window placement, build and debug tools

## MPLAB X IDE

- **Projects contain:**
  - Everything – no workspaces needed
- **Project Groups:**
  - List of projects to simplify opening up multiple related projects that are frequently opened together

# Major Differences

## Project Files and Structure

### MPLAB® IDE 8

- **Projects and workspaces represented by individual files that may be double clicked to open the project or workspace:**
  - \*.mcp (project)
  - \*.mcw (workspace)

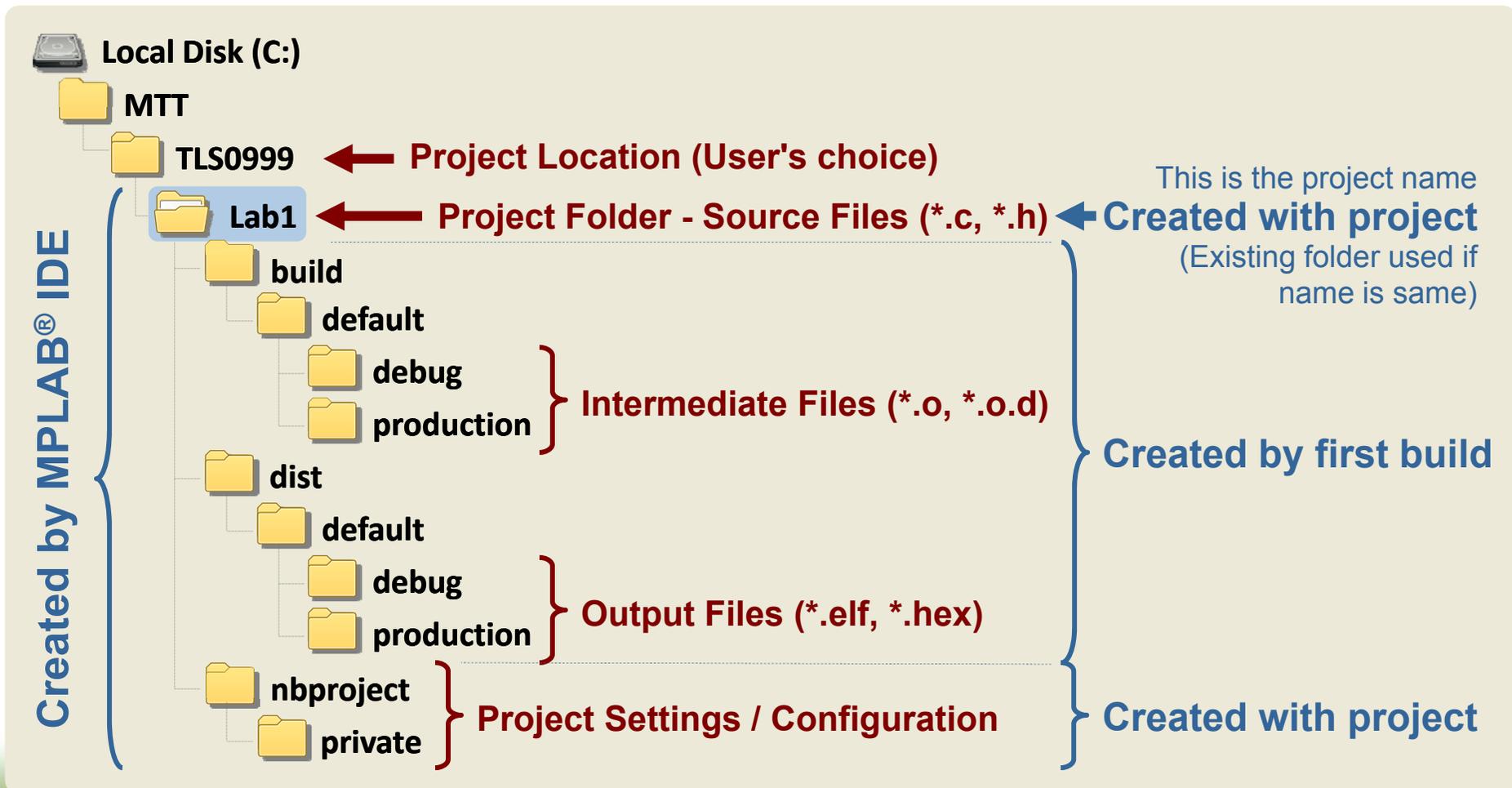
### MPLAB X IDE

- **Project represented by specific directory structure containing XML files**
- **There is no project file to double click**
- **Project must be opened from within IDE**

# Major Differences Project Files and Structure



## Automatically Generated Project Directory Structure



# Major Differences

## Project Based IDE

### MPLAB® IDE 8

- **Not all tasks required a project:**
  - Import Hex File
  - Quick Build

### MPLAB X IDE

- **All tasks require a project:**
  - Import Hex File ► Create “Prebuilt” project
  - Quick Build – not possible from within IDE

# Major Differences

## Debug/Release Build Settings

### MPLAB® IDE 8

- Debug/Release combo box must be explicitly chosen for the desired build type



Make

OR



Build All

### MPLAB X IDE

- Debug/Release implied by specific build command

**Release**



Build (Make)



Program Target



Clean and Build (Build All)

**Debug**



Debug Run Project

# Major Differences

## Legacy Tool Support

### Legacy tools NOT supported by MPLAB<sup>®</sup> X IDE

- **ICD 2 (2002)** No new device support since September 2010
- **ICE 2000 (1998), ICE 4000 (2003)**
- **PRO MATE II (1996)**
- **PICSTART<sup>®</sup> Plus (1997)**

### MPLAB X IDE – Current Tools Supported

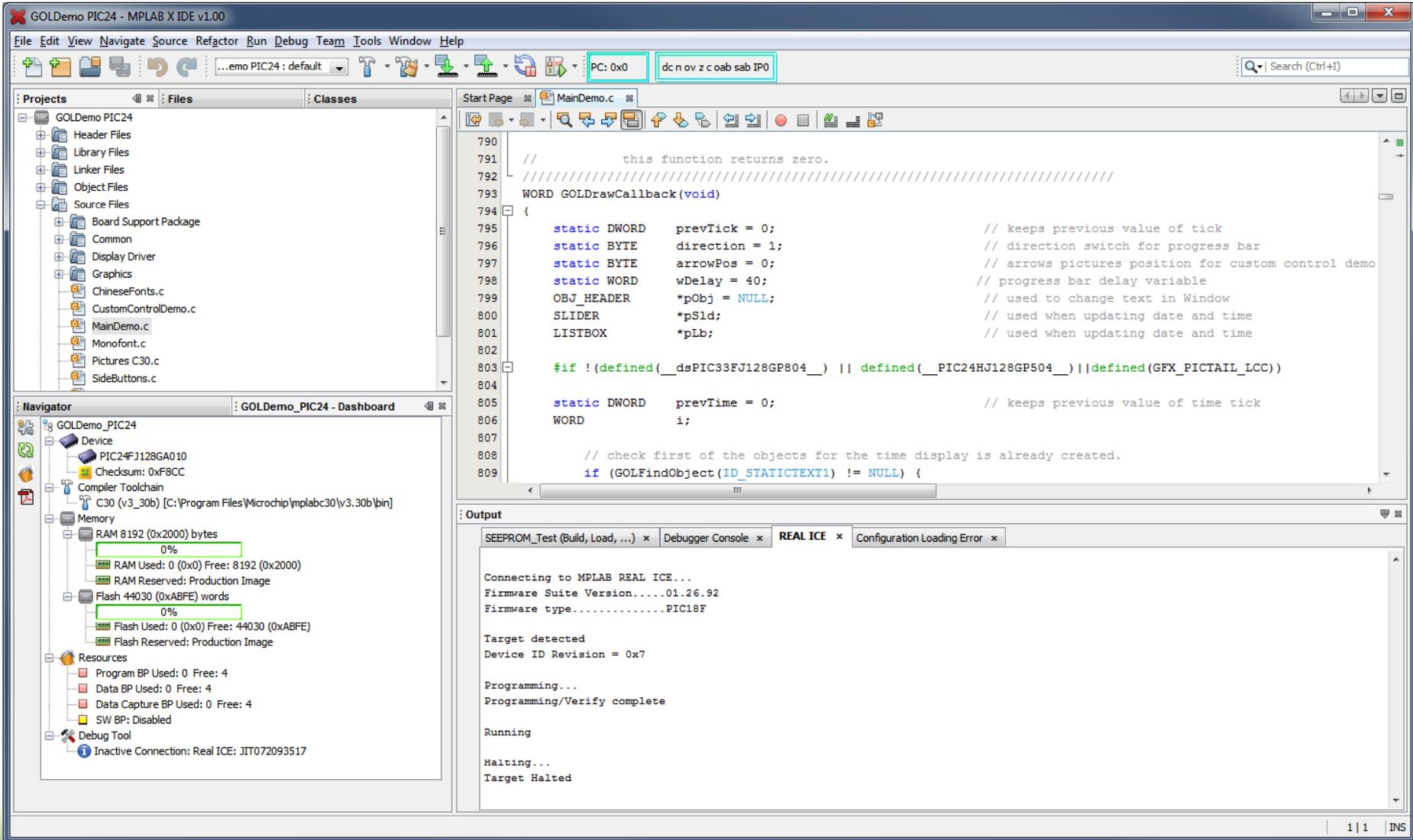
- **MPLAB ICD 3 (2008), PICKit<sup>™</sup> 3 (2008), PICKit 2 (2005)**
- **MPLAB REAL ICE In-Circuit Emulator (2006)**
- **MPLAB PM3 (2004)**
- **Starter Kits...**



# Overview

## A quick look at MPLAB<sup>®</sup> X IDE's components

# IDE Layout



The screenshot displays the MPLAB X IDE v1.00 interface. The main window is titled "GOLDemo PIC24 - MPLAB X IDE v1.00". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and development. The interface is divided into several panes:

- Projects:** Shows a tree view of the project structure, including Header Files, Library Files, Linker Files, Object Files, Source Files, Board Support Package, Common, Display Driver, Graphics, ChineseFonts.c, CustomControlDemo.c, MainDemo.c, Monofont.c, Pictures C30.c, and SideButtons.c.
- Navigator:** Shows the project hierarchy, including Device (PIC24FJ128GA010), Checksum (0xF8CC), Compiler Toolchain (C30 (v3\_30b) [C:\Program Files\Microchip\mplabc30\v3.30b\bin]), Memory (RAM 8192 (0x2000) bytes, Flash 44030 (0xABFE) words), and Resources (Program BP, Data BP, Data Capture BP, SW BP, Debug Tool).
- Source Code Editor:** Displays the code for "MainDemo.c". The code includes comments and defines static variables for a callback function:
 

```

790
791 // this function returns zero.
792 ///////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD   prevTick = 0;           // keeps previous value of tick
796     static BYTE    direction = 1;        // direction switch for progress bar
797     static BYTE    arrowPos = 0;        // arrows pictures position for custom control demo
798     static WORD    wDelay = 40;         // progress bar delay variable
799     OBJ_HEADER     *pObj = NULL;        // used to change text in Window
800     SLIDER         *pSlid;              // used when updating date and time
801     LISTBOX        *pLb;                // used when updating date and time
802
803     #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD   prevTime = 0;        // keeps previous value of time tick
806     WORD           i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {
      
```
- Output:** Shows the status of the target device and the progress of the programming process:
 

```

Connecting to MPLAB REAL ICE...
Firmware Suite Version.....01.26.92
Firmware type.....PIC18F

Target detected
Device ID Revision = 0x7

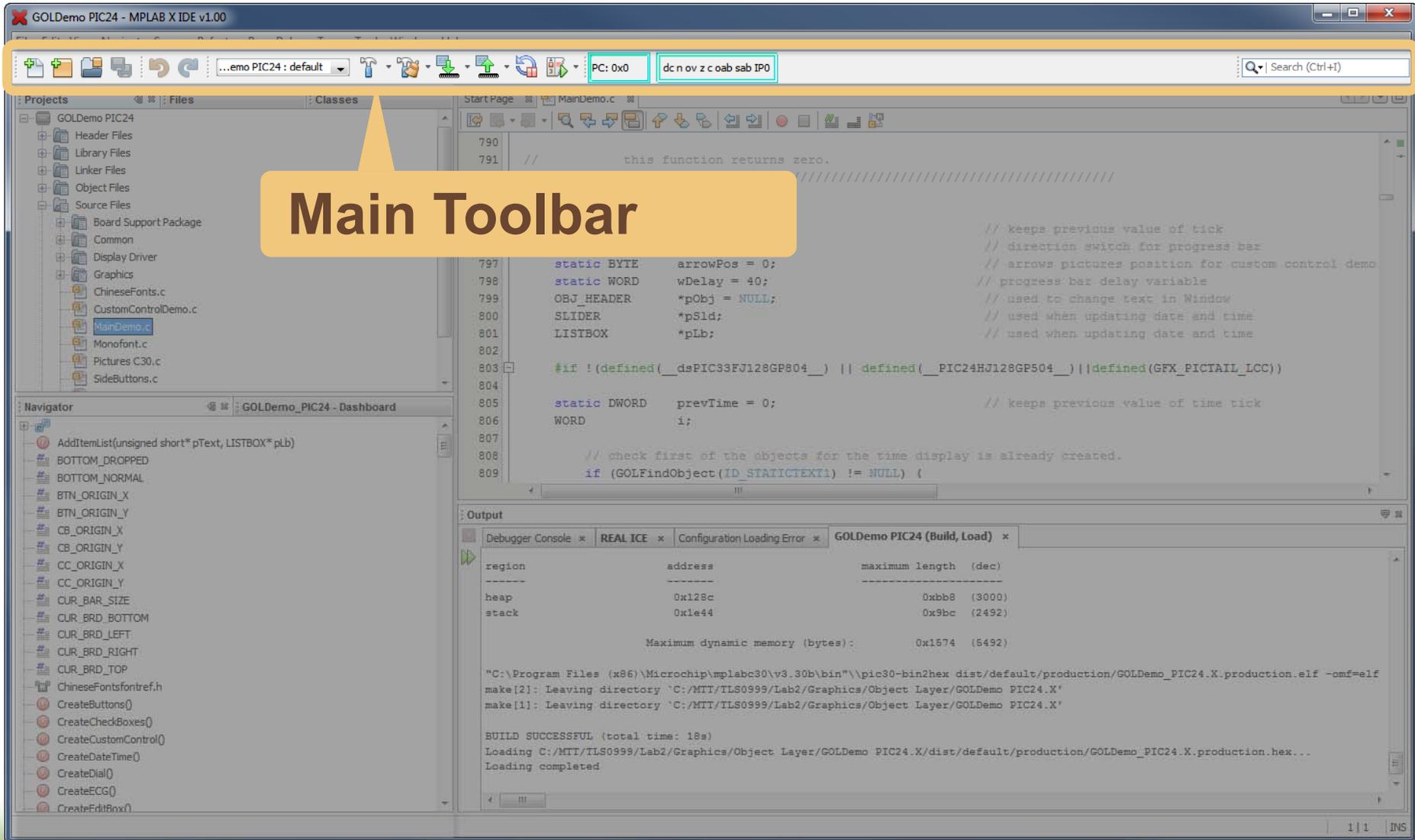
Programming...
Programming/Verify complete

Running

Halting...
Target Halted
      
```

# IDE Layout

## Main Toolbar

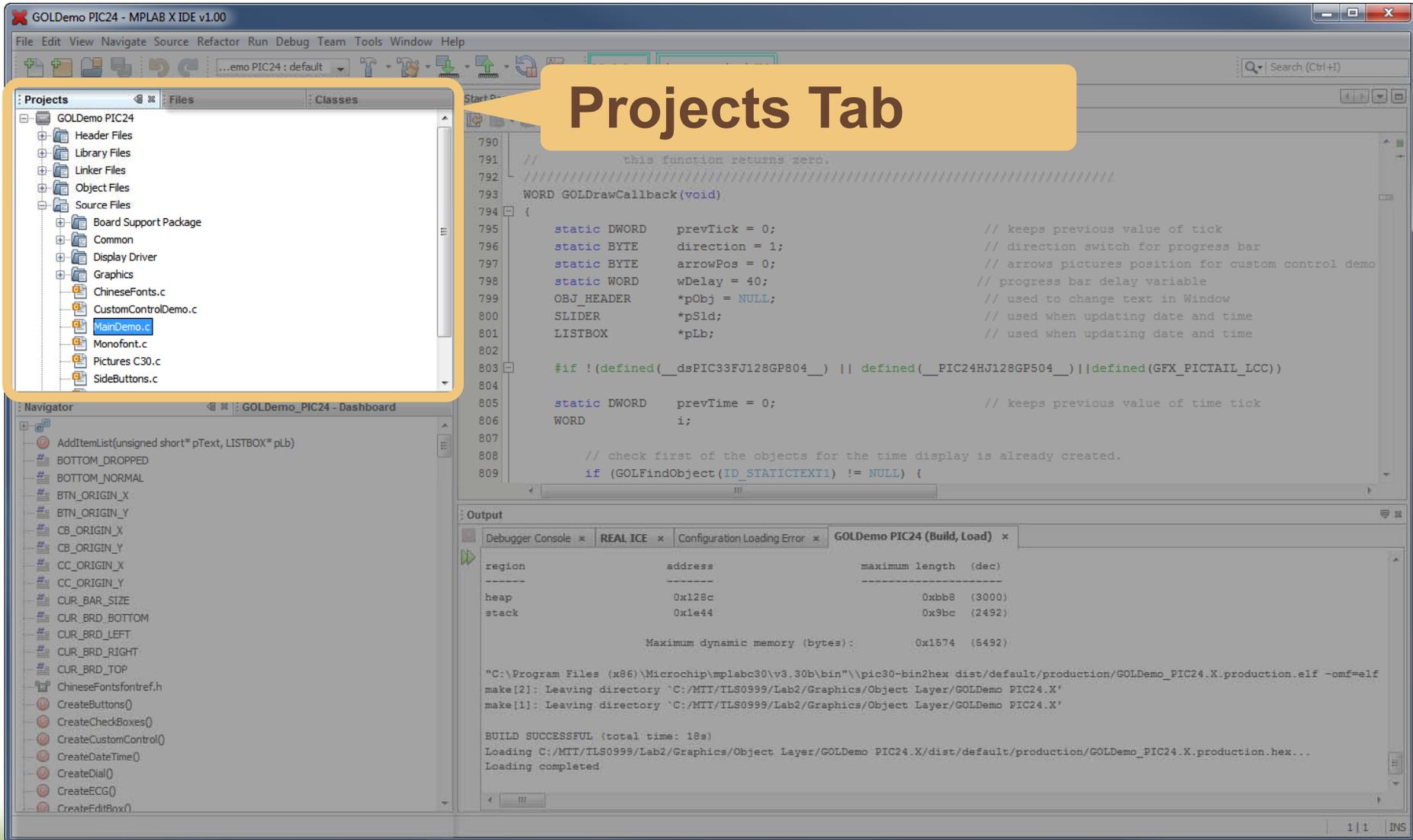


The screenshot shows the MPLAB X IDE v1.00 interface. The main toolbar is highlighted with a yellow box and a callout bubble labeled "Main Toolbar". The toolbar contains various icons for file operations, project management, and development tools. Below the toolbar, the IDE is divided into several panes:

- Projects:** A tree view showing the project structure, including Header Files, Library Files, Linker Files, Object Files, and Source Files.
- Navigator:** A list of symbols and functions, such as `AddItemList`, `BOTTOM_DROPPED`, `BOTTOM_NORMAL`, `BTN_ORIGIN_X`, `BTN_ORIGIN_Y`, `CB_ORIGIN_X`, `CB_ORIGIN_Y`, `CC_ORIGIN_X`, `CC_ORIGIN_Y`, `CUR_BAR_SIZE`, `CUR_BRD_BOTTOM`, `CUR_BRD_LEFT`, `CUR_BRD_RIGHT`, `CUR_BRD_TOP`, `ChineseFontsfontref.h`, `CreateButtons()`, `CreateCheckBoxes()`, `CreateCustomControl()`, `CreateDateTime()`, `CreateDial()`, `CreateECG()`, and `CreateEditBox()`.
- Code Editor:** Displays the source code for `MainDemo.c`. The code includes comments and declarations for variables like `arrowPos`, `wDelay`, `pObj`, `pSlid`, `pLb`, `prevTime`, and `i`. It also shows a conditional compilation block for different PIC models.
- Output:** Shows the build output for `GOLDemo PIC24 (Build, Load)`. It includes a table of memory regions and their sizes, followed by the build command and the successful completion message.

# IDE Layout

## Projects Tab (Project Tree)



**Projects Tab**

```

790
791 //      this function returns zero.
792 //      ////////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD    prevTick = 0;           // keeps previous value of tick
796     static BYTE     direction = 1;         // direction switch for progress bar
797     static BYTE     arrowPos = 0;         // arrows pictures position for custom control demo
798     static WORD     wDelay = 40;         // progress bar delay variable
799     OBJ_HEADER      *pObj = NULL;         // used to change text in Window
800     SLIDER          *pSlid;              // used when updating date and time
801     LISTBOX         *pLb;                // used when updating date and time
802
803     #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD    prevTime = 0;         // keeps previous value of time tick
806     WORD            i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {

```

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

```

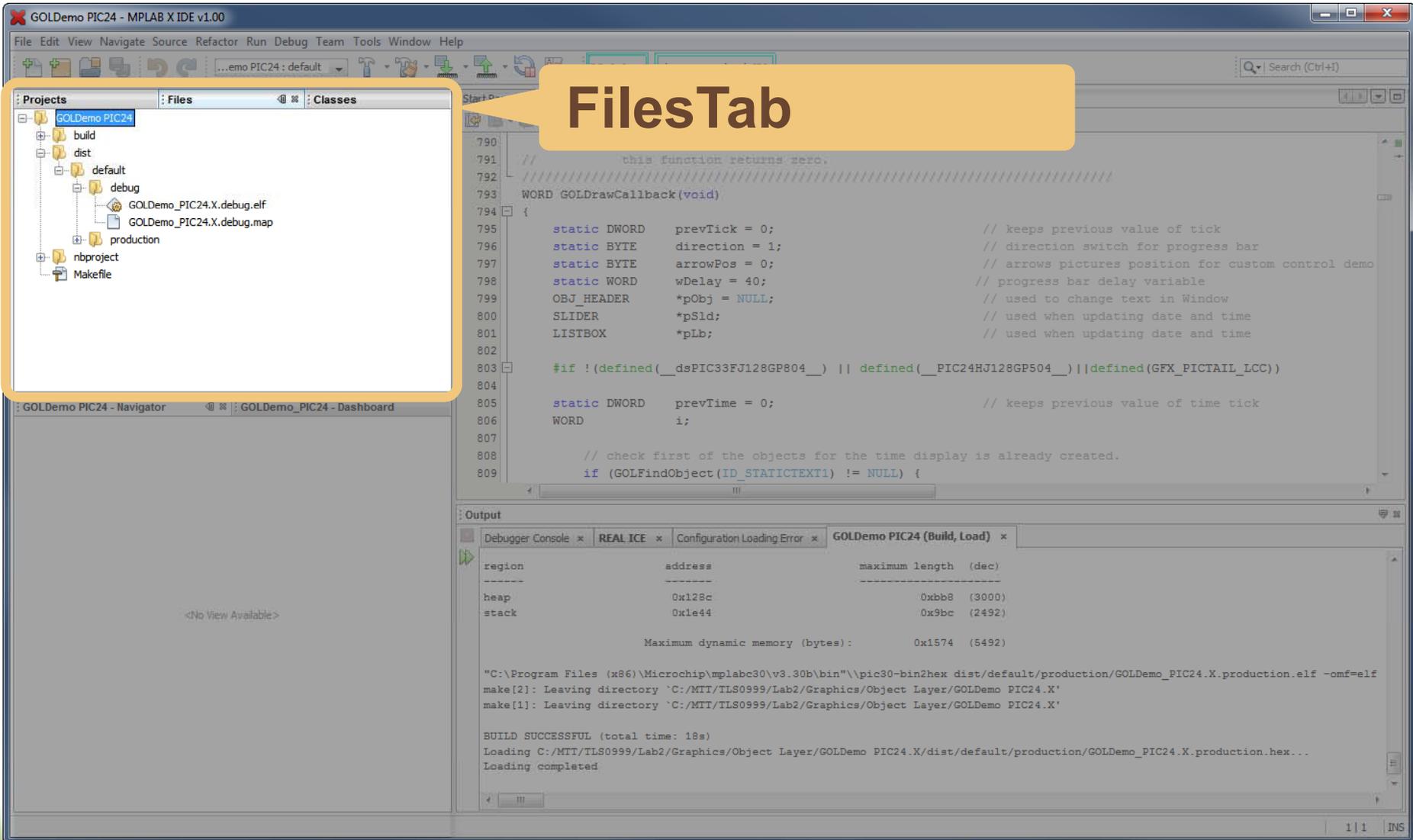
"C:\Program Files (x86)\Microchip\mplab30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

```

# IDE Layout

## Files Tab



The screenshot shows the MPLAB X IDE interface for a project named "GOLDemo PIC24". The Project Explorer on the left is divided into three tabs: "Projects", "Files", and "Classes". The "Files" tab is selected and highlighted with a yellow callout box containing the text "Files Tab". The "Files" tab displays a tree view of the project structure, including folders like "build", "dist", "default", "debug", "production", and "nbproject", along with files like "GOLDemo\_PIC24.X.debug.elf" and "GOLDemo\_PIC24.X.debug.map".

The main editor window displays C code for a function named "GOLDrawCallback". The code includes static variables for tick and time, and a loop that checks for the presence of certain macros. The code is as follows:

```

790
791 // this function returns zero.
792 ///////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD    prevTick = 0;           // keeps previous value of tick
796     static BYTE     direction = 1;         // direction switch for progress bar
797     static BYTE     arrowPos = 0;         // arrows pictures position for custom control demo
798     static WORD     wDelay = 40;         // progress bar delay variable
799     OBJ_HEADER      *pObj = NULL;        // used to change text in Window
800     SLIDER          *pSlid;             // used when updating date and time
801     LISTBOX         *pLb;              // used when updating date and time
802
803     #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD    prevTime = 0;         // keeps previous value of time tick
806     WORD            i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {

```

The Output window at the bottom shows the results of a build process, including memory addresses and lengths for heap and stack, and a successful build message.

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)

Maximum dynamic memory (bytes): 0x1574 (5492)

```

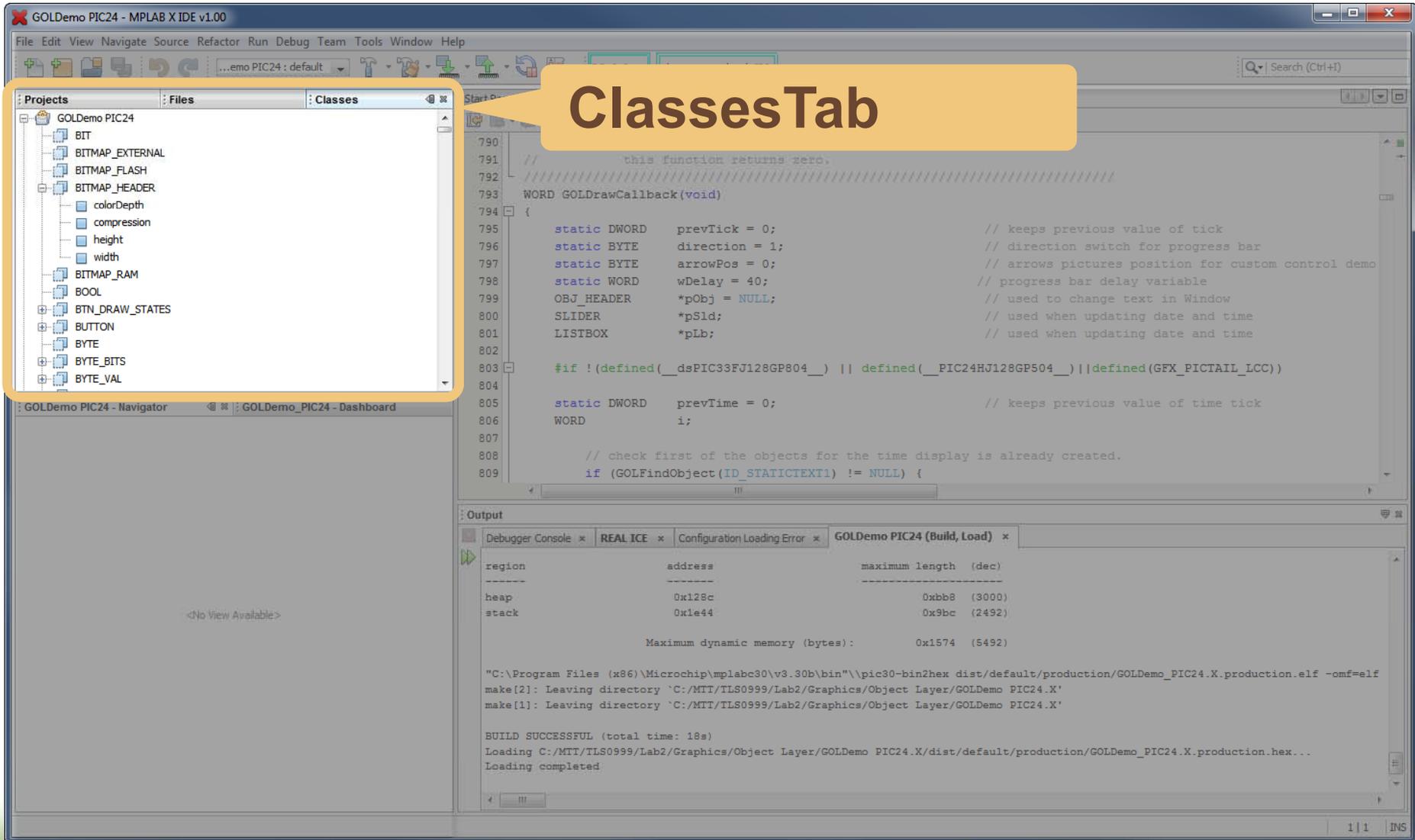
"C:\Program Files (x86)\Microchip\mplabc30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

```

# IDE Layout

## Classes Tab



The screenshot shows the MPLAB X IDE interface for a project named 'GOLDemo PIC24'. The 'Classes' tab is selected in the Project Explorer, showing a tree view of classes including BIT, BITMAP\_EXTERNAL, BITMAP\_FLASH, BITMAP\_HEADER, colorDepth, compression, height, width, BITMAP\_RAM, BOOL, BTN\_DRAW\_STATES, BUTTON, BYTE, BYTE\_BITS, and BYTE\_VAL. The main editor displays C code for a function named 'GOLDrawCallback'. The Output window at the bottom shows the build process, including memory layout information and the successful completion of the build.

**Classes Tab**

```

790
791 // this function returns zero.
792 ///////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD   prevTick = 0;           // keeps previous value of tick
796     static BYTE    direction = 1;         // direction switch for progress bar
797     static BYTE    arrowPos = 0;         // arrows pictures position for custom control demo
798     static WORD    wDelay = 40;          // progress bar delay variable
799     OBJ_HEADER     *pObj = NULL;         // used to change text in Window
800     SLIDER         *pSlid;               // used when updating date and time
801     LISTBOX        *pLb;                 // used when updating date and time
802
803     #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD   prevTime = 0;         // keeps previous value of time tick
806     WORD           i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {

```

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

```

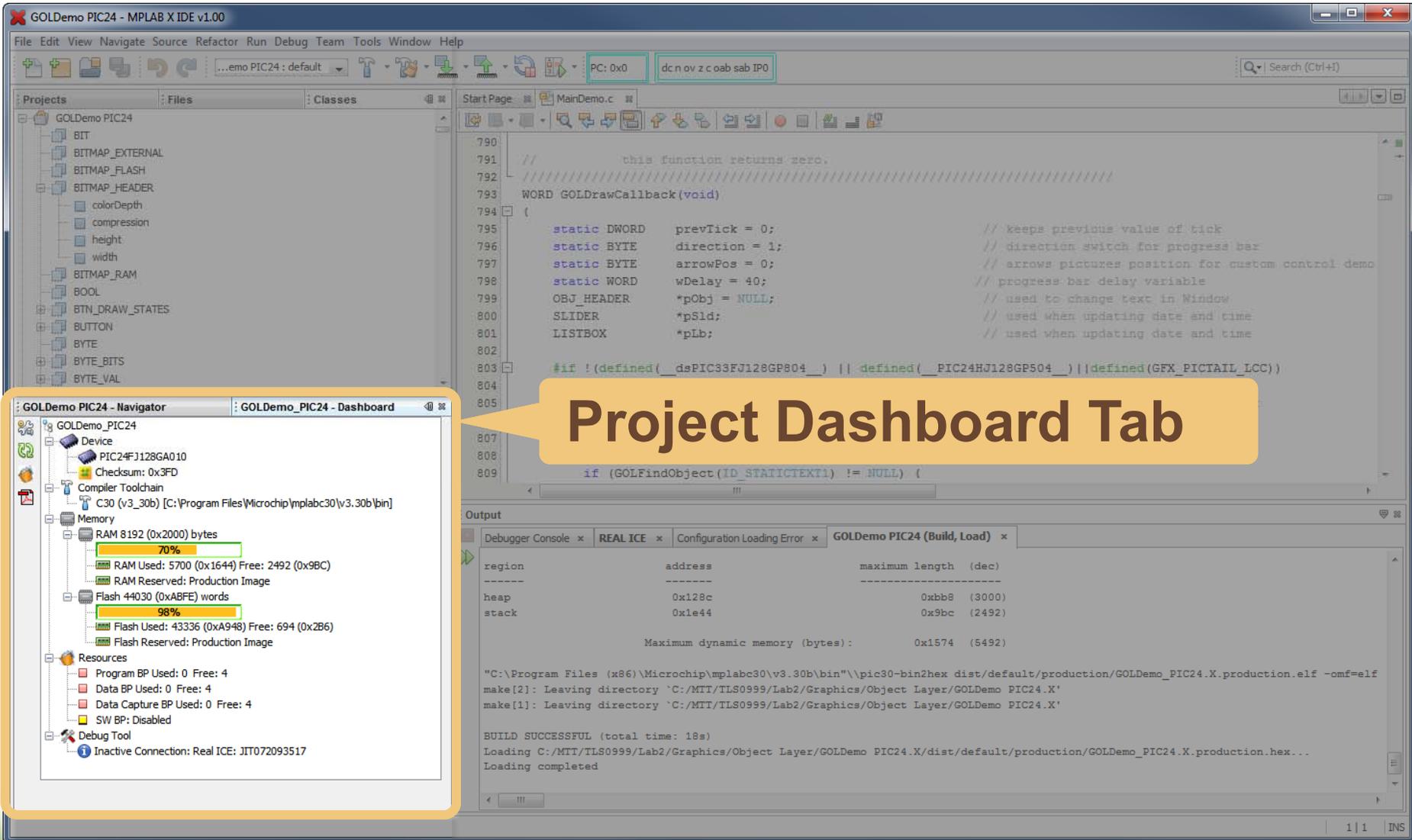
"C:\Program Files (x86)\Microchip\mplabc30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

```

# IDE Layout

## Project Dashboard Tab



The screenshot displays the MPLAB X IDE interface for a project named 'GOLDemo PIC24'. The main window shows a C source file 'MainDemo.c' with code for a drawing callback function. A yellow callout box points to the 'GOLDemo PIC24 - Dashboard' tab in the Navigator window, which is highlighted with a yellow border. The dashboard provides a hierarchical view of the project's components and their resource usage.

**Project Dashboard Data:**

- Device:** PIC24FJ128GA010
- Checksum:** 0x3FD
- Compiler Toolchain:** C30 (v3\_30b) [C:\Program Files\Microchip\mplabc30\v3.30b\bin]
- Memory:**
  - RAM:** 8192 (0x2000) bytes. 70% used. RAM Used: 5700 (0x1644) Free: 2492 (0x9BC). RAM Reserved: Production Image.
  - Flash:** 44030 (0xABFE) words. 98% used. Flash Used: 43336 (0xA948) Free: 694 (0x2B6). Flash Reserved: Production Image.
- Resources:**
  - Program BP Used: 0 Free: 4
  - Data BP Used: 0 Free: 4
  - Data Capture BP Used: 0 Free: 4
  - SW BP: Disabled
- Debug Tool:** Inactive Connection: Real ICE: J1072093517

The Output window at the bottom shows the build process for 'GOLDemo PIC24 (Build, Load)'. The build was successful, and the debugger (REAL ICE) is loaded.

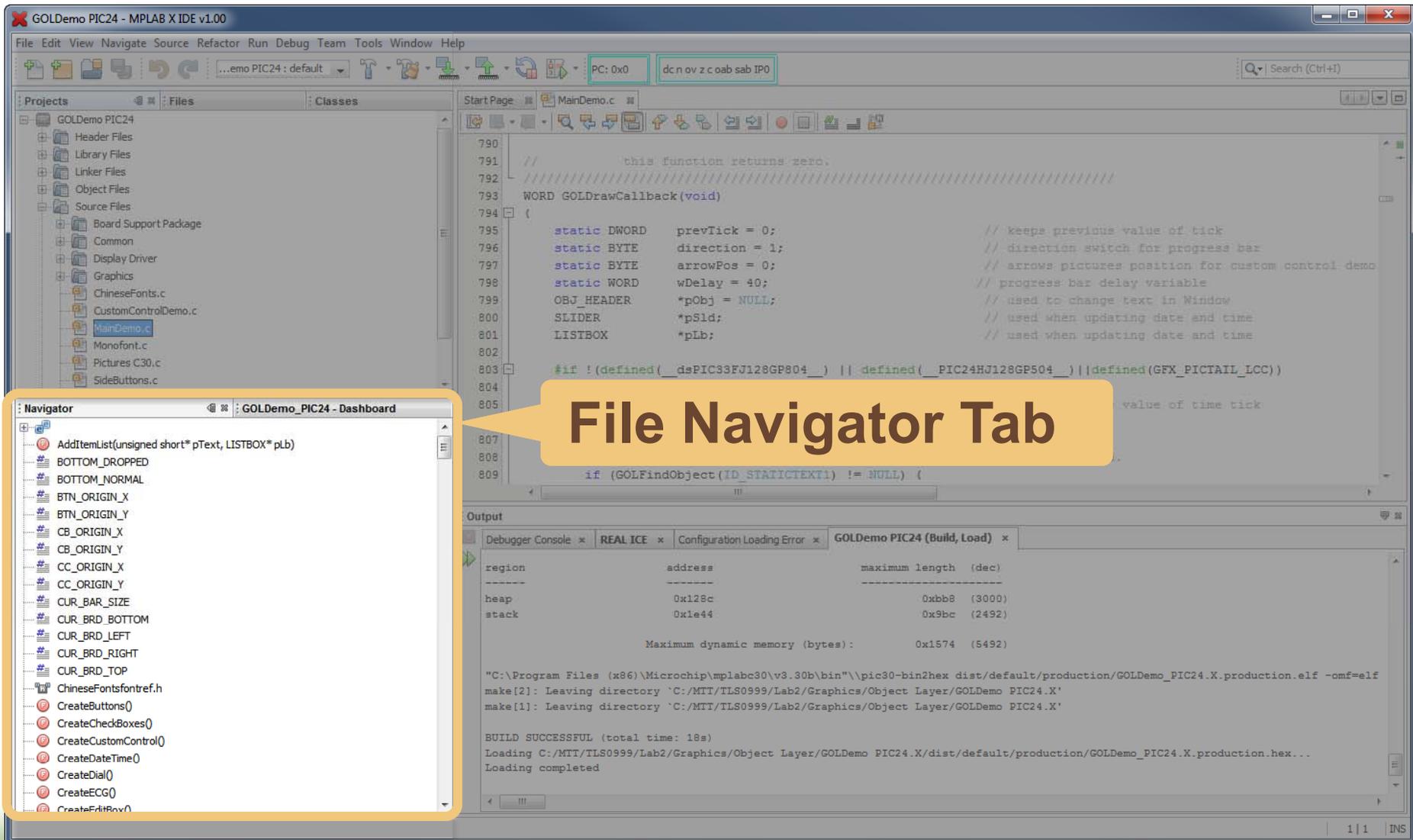
region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

```

"C:\Program Files (x86)\Microchip\mplabc30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed
  
```

# IDE Layout Navigator



The screenshot shows the MPLAB X IDE interface. The File Navigator tab is highlighted with a yellow callout box containing the text "File Navigator Tab". The File Navigator shows a list of files and folders for the project "GOLDemo PIC24 - Dashboard". The main editor window displays the source code for "MainDemo.c". The Output window shows the build process results, including the command "make" and the message "BUILD SUCCESSFUL (total time: 18s)".

**File Navigator Tab**

Debugger Console × REAL ICE × Configuration Loading Error × GOLDemo PIC24 (Build, Load) ×

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

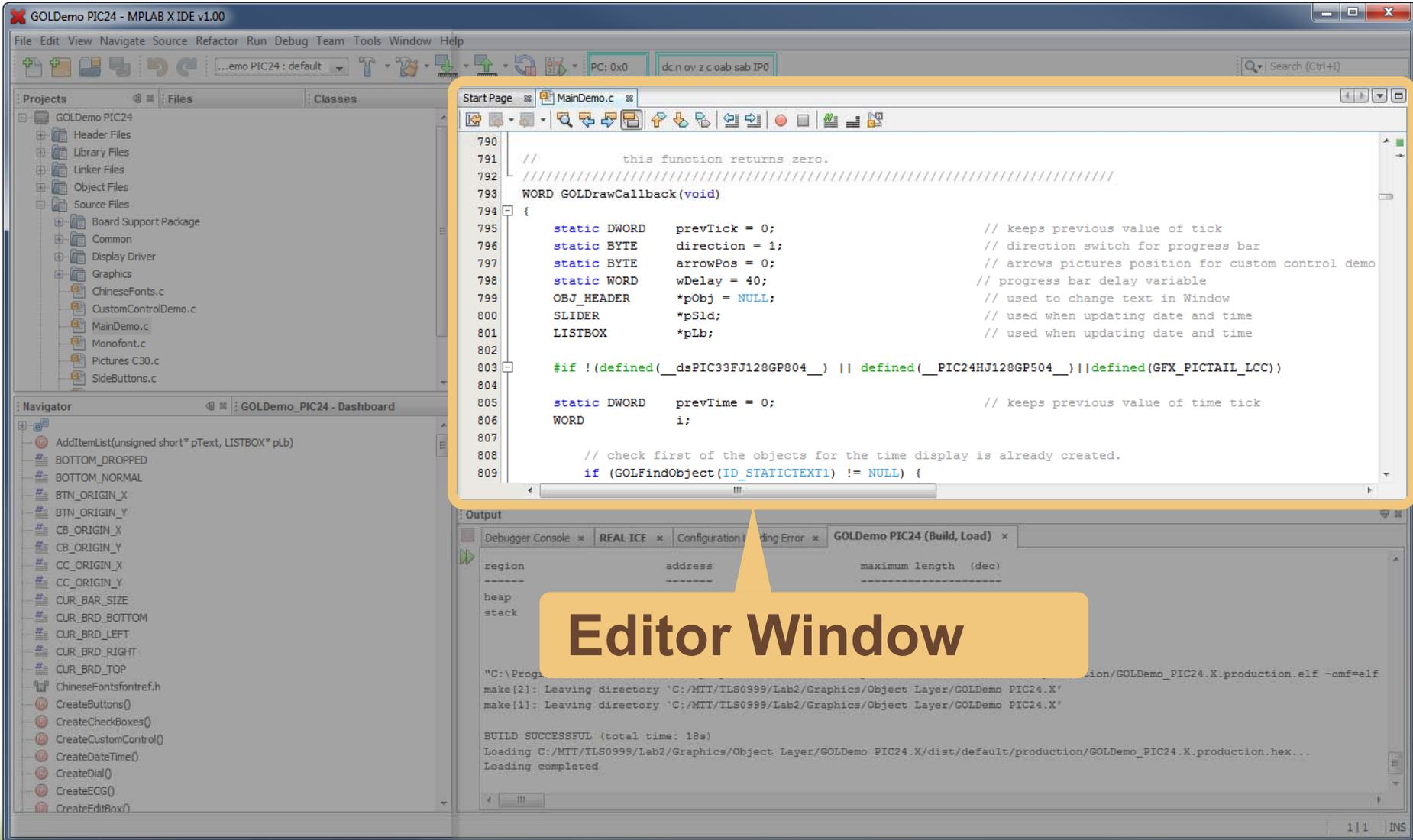
```

"C:\Program Files (x86)\Microchip\mplab30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MTT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

```

# IDE Layout Editor



The screenshot displays the MPLAB X IDE v1.00 interface. The main window is titled "GOLDemo PIC24 - MPLAB X IDE v1.00". The interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Team, Tools, Window, Help), a toolbar, and several panels:

- Projects:** Shows a tree view of the project structure, including "GOLDemo PIC24", "Header Files", "Library Files", "Linker Files", "Object Files", "Source Files", "Board Support Package", "Common", "Display Driver", "Graphics", "ChineseFonts.c", "CustomControlDemo.c", "MainDemo.c", "Monofont.c", "Pictures C30.c", and "SideButtons.c".
- Navigator:** Shows a list of symbols and functions, such as "AddItemList(unsigned short\*pText, LISTBOX\*pLb)", "BOTTOM\_DROPPED", "BOTTOM\_NORMAL", "BTN\_ORIGIN\_X", "BTN\_ORIGIN\_Y", "CB\_ORIGIN\_X", "CB\_ORIGIN\_Y", "CC\_ORIGIN\_X", "CC\_ORIGIN\_Y", "CUR\_BAR\_SIZE", "CUR\_BRD\_BOTTOM", "CUR\_BRD\_LEFT", "CUR\_BRD\_RIGHT", "CUR\_BRD\_TOP", "ChineseFontsfontref.h", "CreateButtons()", "CreateCheckBoxes()", "CreateCustomControl()", "CreateDateTime()", "CreateDial()", "CreateECG()", and "CreateEditBox()".
- Editor Window:** The central window, highlighted with a yellow border, displays the source code for "MainDemo.c". The code includes comments and function definitions for "GOLDrawCallback(void)".
- Output:** Shows the build process output, including "BUILD SUCCESSFUL (total time: 18s)" and "Loading completed".

A yellow callout box with the text "Editor Window" points to the central code editor.

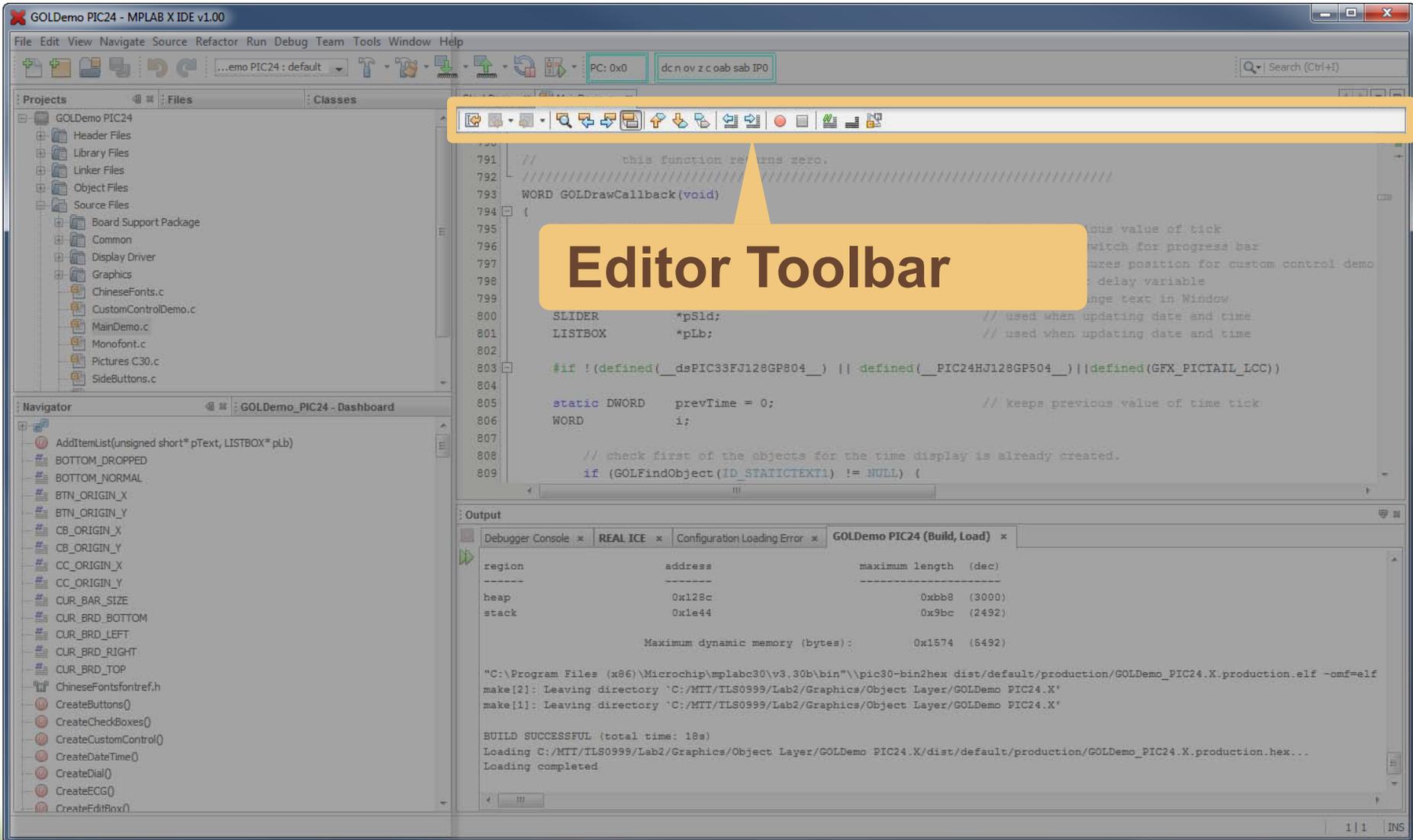
```

790
791 //      this function returns zero.
792 //      ////////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD    prevTick = 0;           // keeps previous value of tick
796     static BYTE     direction = 1;         // direction switch for progress bar
797     static BYTE     arrowPos = 0;         // arrows pictures position for custom control demo
798     static WORD     wDelay = 40;         // progress bar delay variable
799     OBJ_HEADER      *pObj = NULL;        // used to change text in Window
800     SLIDER          *pSlid;             // used when updating date and time
801     LISTBOX         *pLb;              // used when updating date and time
802
803     #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD    prevTime = 0;         // keeps previous value of time tick
806     WORD            i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {

```

# IDE Layout

## Editor Toolbar



**Editor Toolbar**

```

791 // this function returns zero.
792 ///////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     // returns value of tick
796     // switch for progress bar
797     // returns position for custom control demo
798     // delay variable
799     // change text in Window
800     SLIDER      *pSlid; // used when updating date and time
801     LISTBOX     *pLb; // used when updating date and time
802
803     #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD prevTime = 0; // keeps previous value of time tick
806     WORD i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {

```

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

```

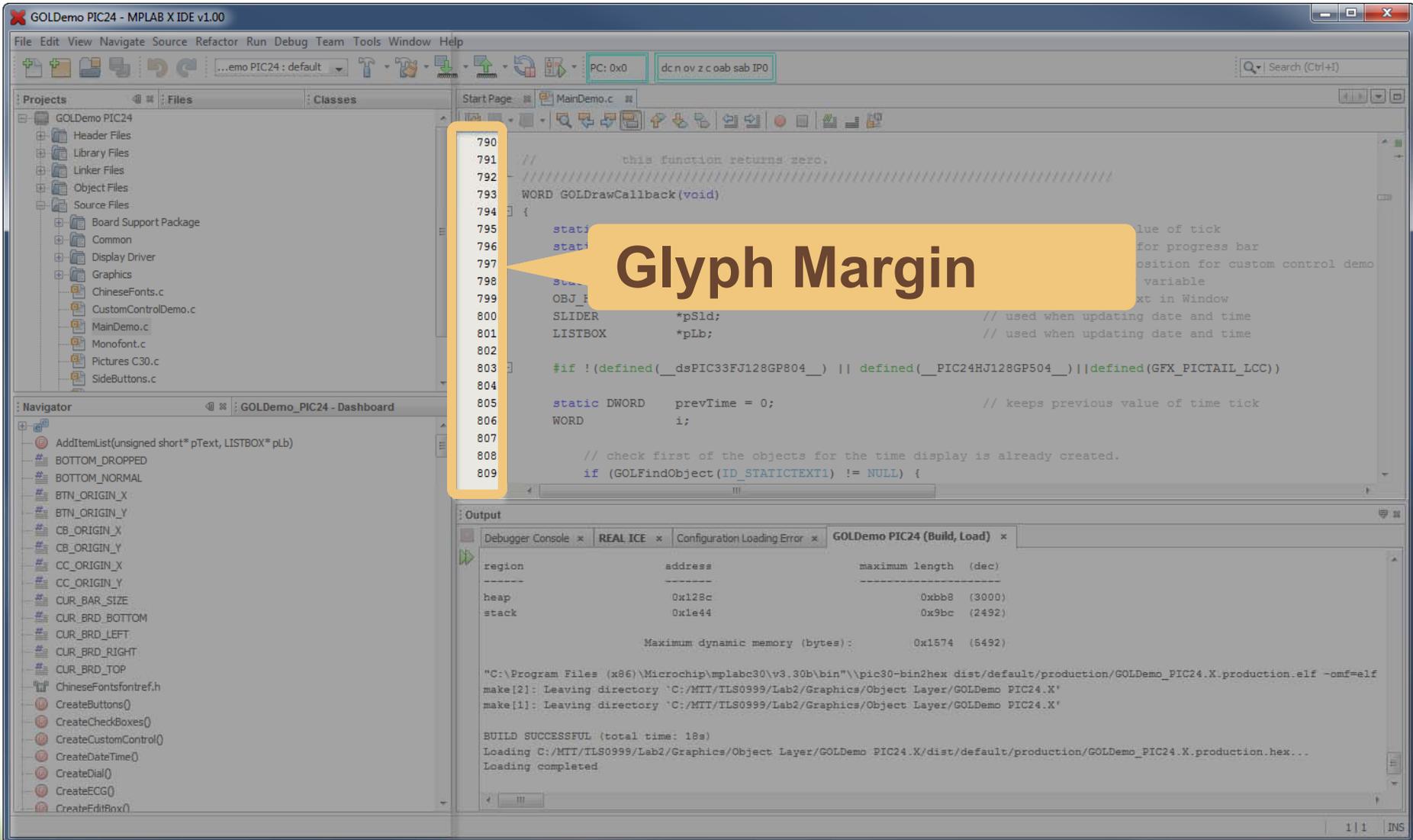
"C:\Program Files (x86)\Microchip\mplab30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

```

# IDE Layout

## Glyph Margin



The screenshot shows the MPLAB X IDE v1.00 interface. The main window displays a C source file named `MainDemo.c`. A yellow callout box with the text "Glyph Margin" points to the line numbers in the code editor, which are highlighted in a light yellow color. The code includes comments and function definitions. The Output window at the bottom shows the build process for `GOLDemo PIC24`, indicating a successful build.

```
790
791 //      this function returns zero.
792 //      ////////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD prevTime = 0; // keeps previous value of time tick
796     static WORD i;
797     // check first of the objects for the time display is already created.
798     if (GOLFindObject(ID_STATICTEXT1) != NULL) {
799         // used when updating date and time
800         SLIDER *pSlid;
801         LISTBOX *pLb;
802         #if !(defined(__dsPIC33FJ128GP804__) || defined(__PIC24HJ128GP504__) || defined(GFX_PICTAIL_LCC))
803             // used when updating date and time
804             static DWORD prevTime = 0; // keeps previous value of time tick
805             WORD i;
806             // check first of the objects for the time display is already created.
807             if (GOLFindObject(ID_STATICTEXT1) != NULL) {
```

**Output**

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)

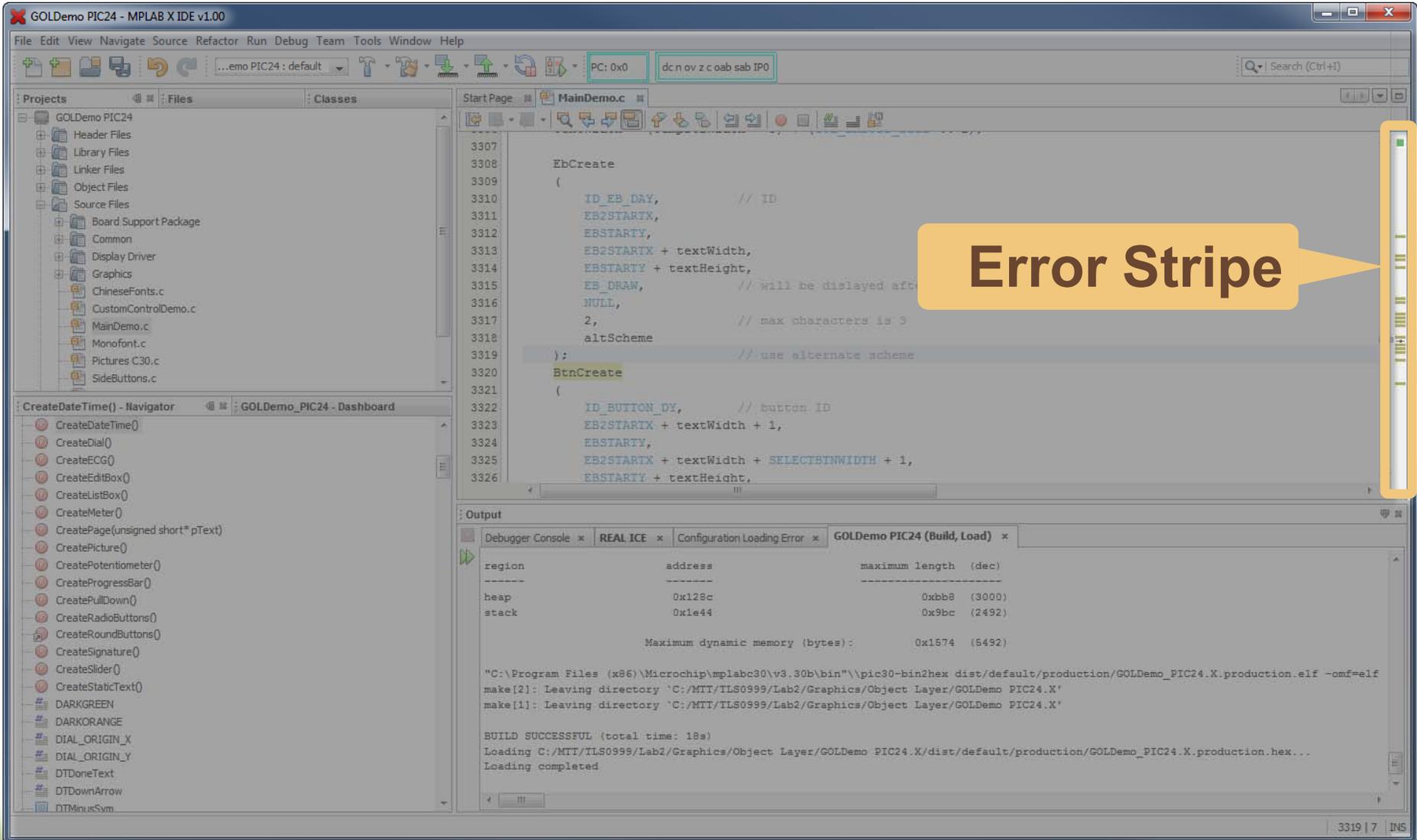
Maximum dynamic memory (bytes): 0x1574 (5492)

```
"C:\Program Files (x86)\Microchip\mplab30\v3.30\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed
```

# IDE Layout

## Error Stripe



The screenshot displays the MPLAB X IDE v1.00 interface. The main window shows the source code for 'MainDemo.c'. A vertical bar on the right side of the code editor, highlighted by a yellow callout box labeled 'Error Stripe', indicates the location of errors. The code editor shows the following code:

```

3307
3308
3309     EbCreate
3310     (
3311         ID_EB_DAY,           // ID
3312         EB2STARTX,
3313         EBSTARTY,
3314         EB2STARTX + textWidth,
3315         EBSTARTY + textHeight,
3316         EB_DRAW,           // will be displayed after
3317         NULL,              // max characters is 3
3318         2,
3319         altScheme          // use alternate scheme
3320     );
3321     BtnCreate
3322     (
3323         ID_BUTTON_DY,       // button ID
3324         EB2STARTX + textWidth + 1,
3325         EBSTARTY,
3326         EB2STARTX + textWidth + SELECTBINWIDTH + 1,
3327         EBSTARTY + textHeight,
3328         ""
3329     );

```

The Output window at the bottom shows the following information:

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

The Output window also shows the following text:

```

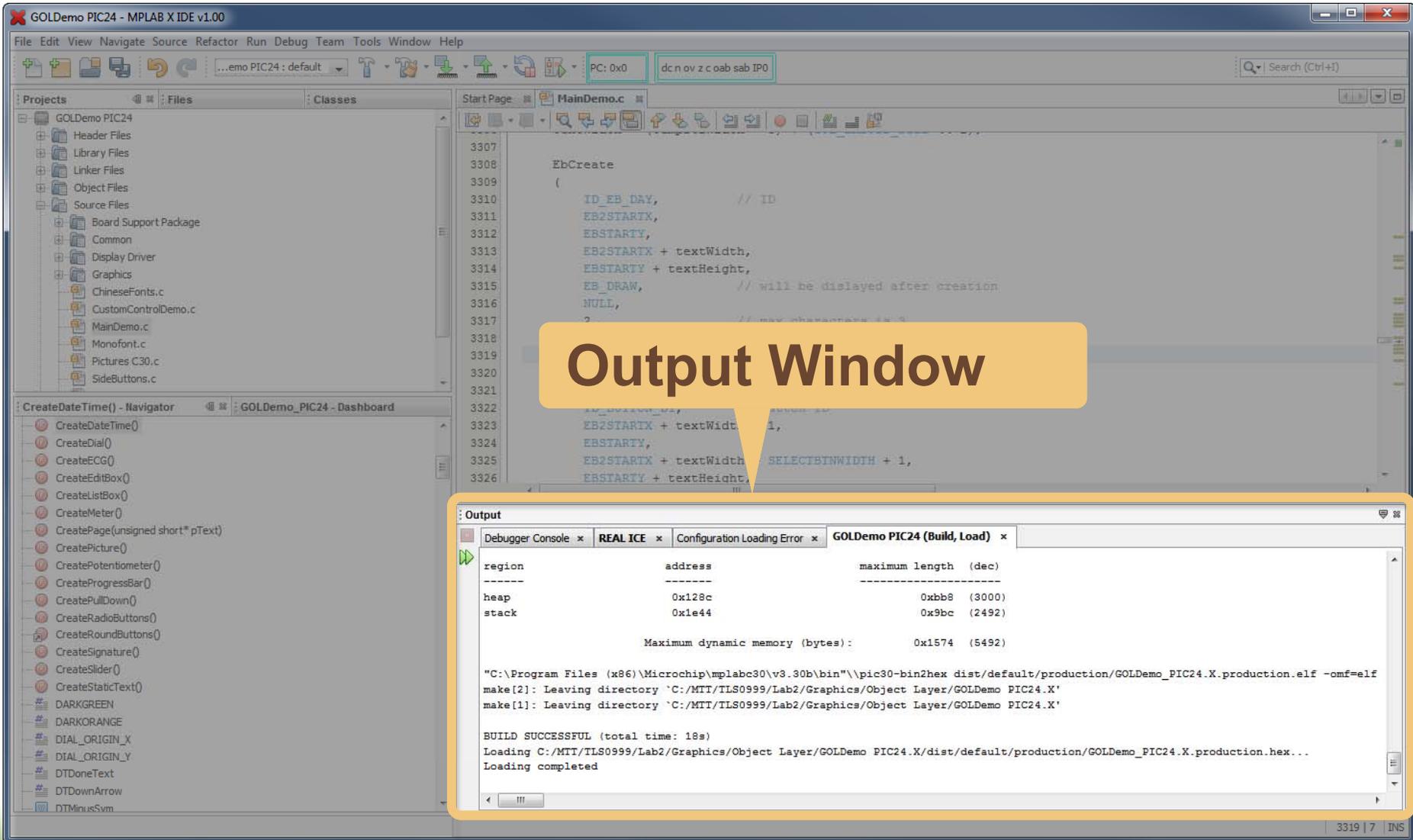
"C:\Program Files (x86)\Microchip\mplab30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MTT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MTT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MTT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

```

# IDE Layout

## Output Window



The screenshot shows the MPLAB X IDE interface. The main editor displays C code for a PIC24 microcontroller. The Output Window is open at the bottom, showing the results of a build process. A yellow callout box with the text "Output Window" points to the Output Window.

**Output Window Content:**

```

Debugger Console x REAL ICE x Configuration Loading Error x GOLDemo PIC24 (Build, Load) x
region                address                maximum length (dec)
-----
heap                   0x128c                    0xbb8 (3000)
stack                  0x1e44                    0x9bc (2492)

Maximum dynamic memory (bytes): 0x1574 (5492)

"C:\Program Files (x86)\Microchip\mplabc30\v3.30\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -omf=elf
make[2]: Leaving directory `C:/MIT/TL50999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'
make[1]: Leaving directory `C:/MIT/TL50999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/TL50999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed
  
```



# How to switch USB drivers

## Swapping drivers between MPLAB® IDE 8 and MPLAB X IDE

# How to switch USB drivers

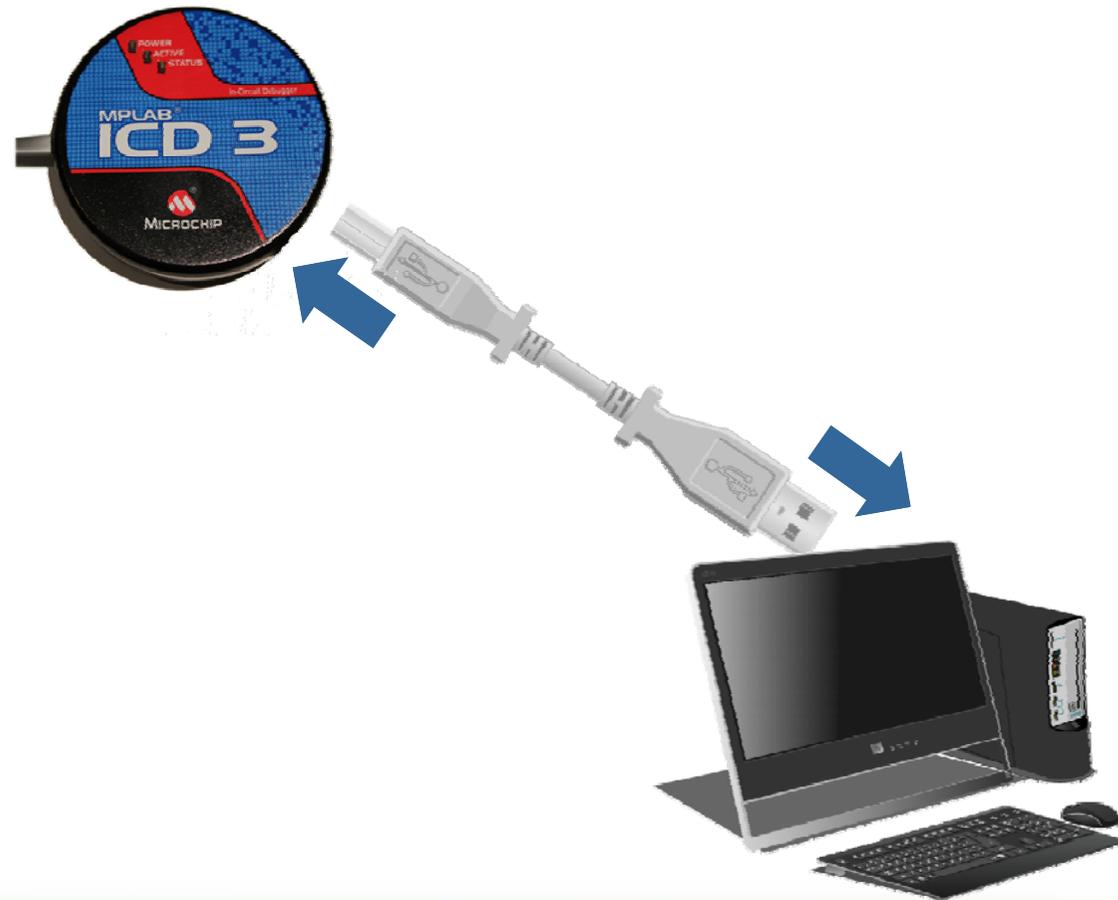
## 1 Plug-in the hardware tool



**Before you do anything else:**

**Plug in your MPLAB® REAL ICE™ in-circuit emulator or MPLAB ICD to a USB port on your PC.**

**The driver cannot be switched unless the tool is plugged in and the existing driver is loaded.**

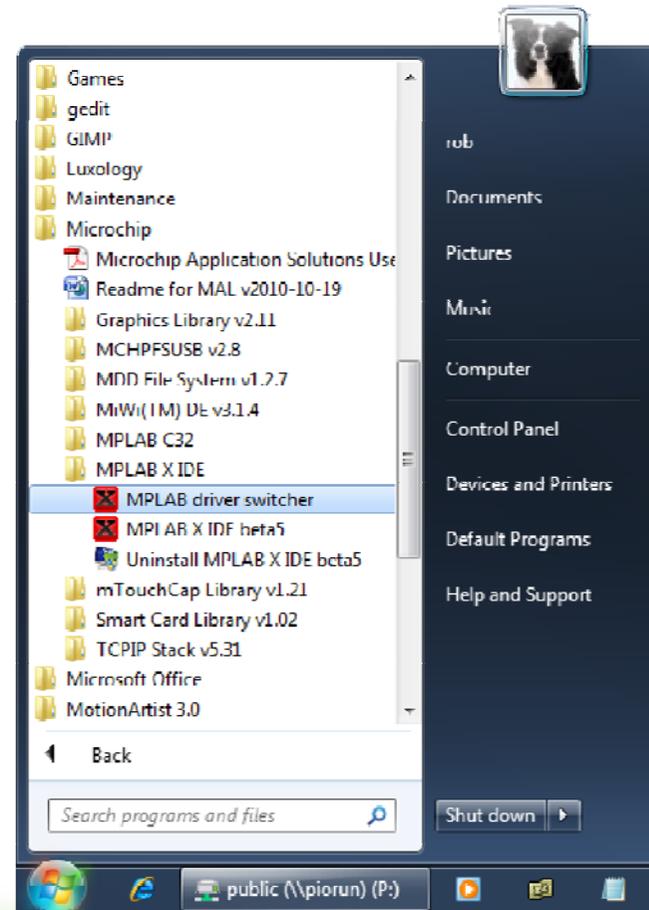


# How to switch USB drivers

## 2 Launch the Driver Switcher utility

Select from the Windows / Start menu:

- ▶ All Programs / Programs
- ▶ Microchip
- ▶ MPLAB® X IDE
- ▶ MPLAB Driver Switcher



# How to switch USB drivers

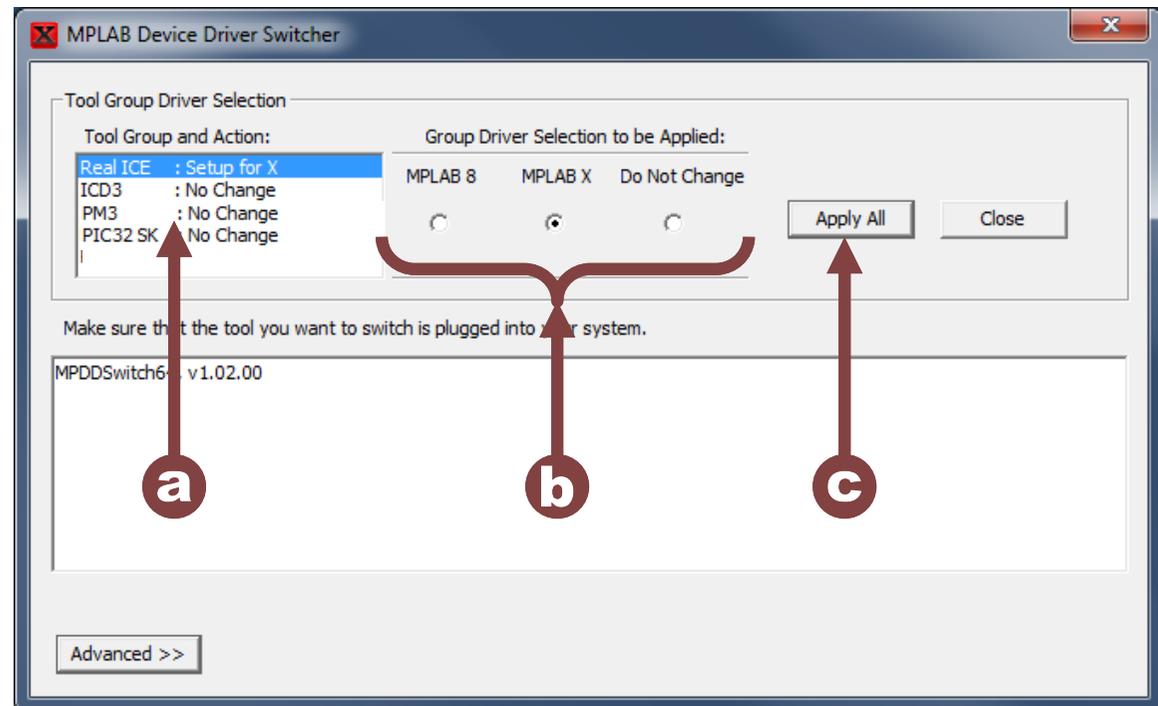
## 3 Select tool and driver

**a** Select tool whose driver you wish to switch

**b** Choose desired driver

**c** Click

Apply All

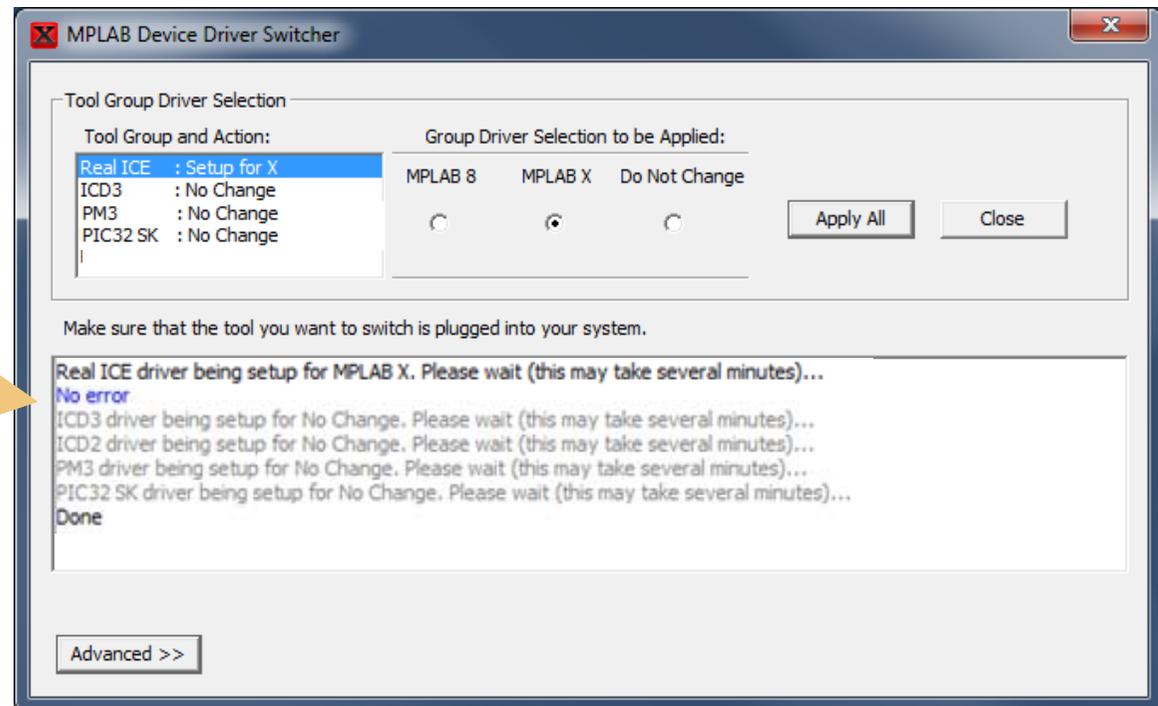


# How to switch USB drivers

## 4 Switch Complete

It may take a couple minutes for the change to be made.

When the driver has been switched successfully, you should see **No error** in the output window.





# How to configure a C compiler

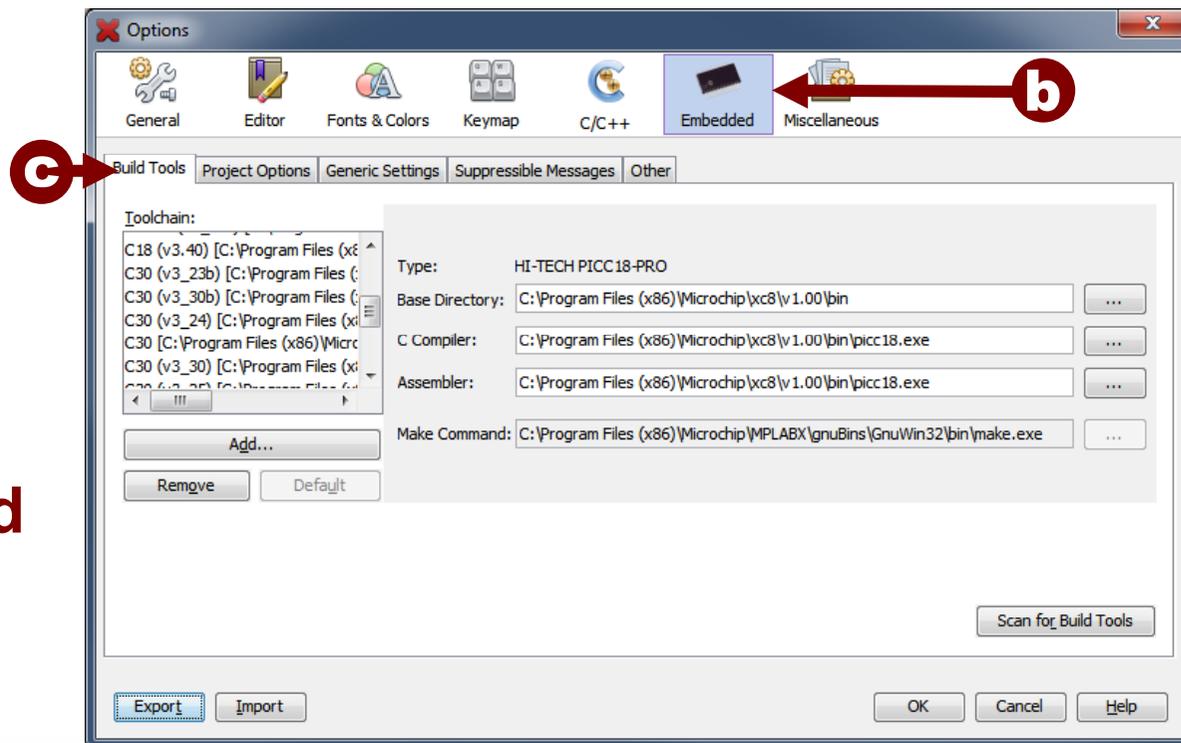
# How to configure a C compiler

## 1 Open the Embedded Options Window

a From the main menu, select **Tools ▶ Options**

b In the window that opens, select the **Embedded** category

c Select the **Build Tools** tab



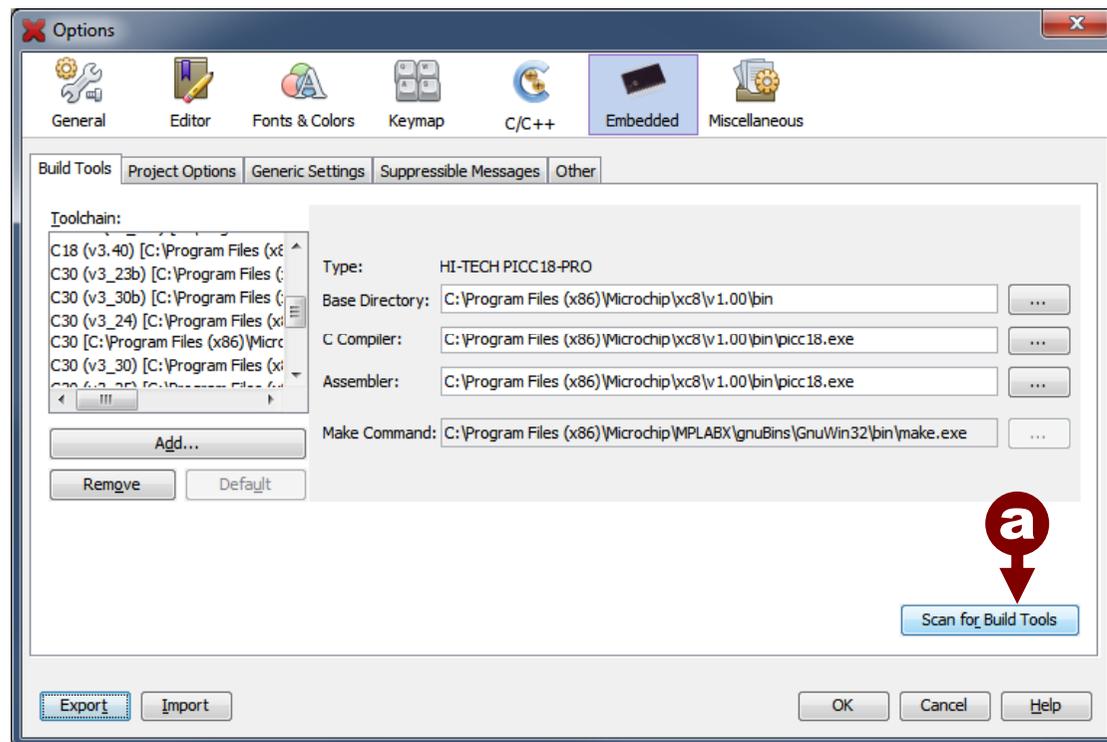
# How to configure a C compiler

## 2 Restore Defaults

### a Click on the **Scan for Build Tools** button

If the window gets populated with your installed compilers, then you are done.

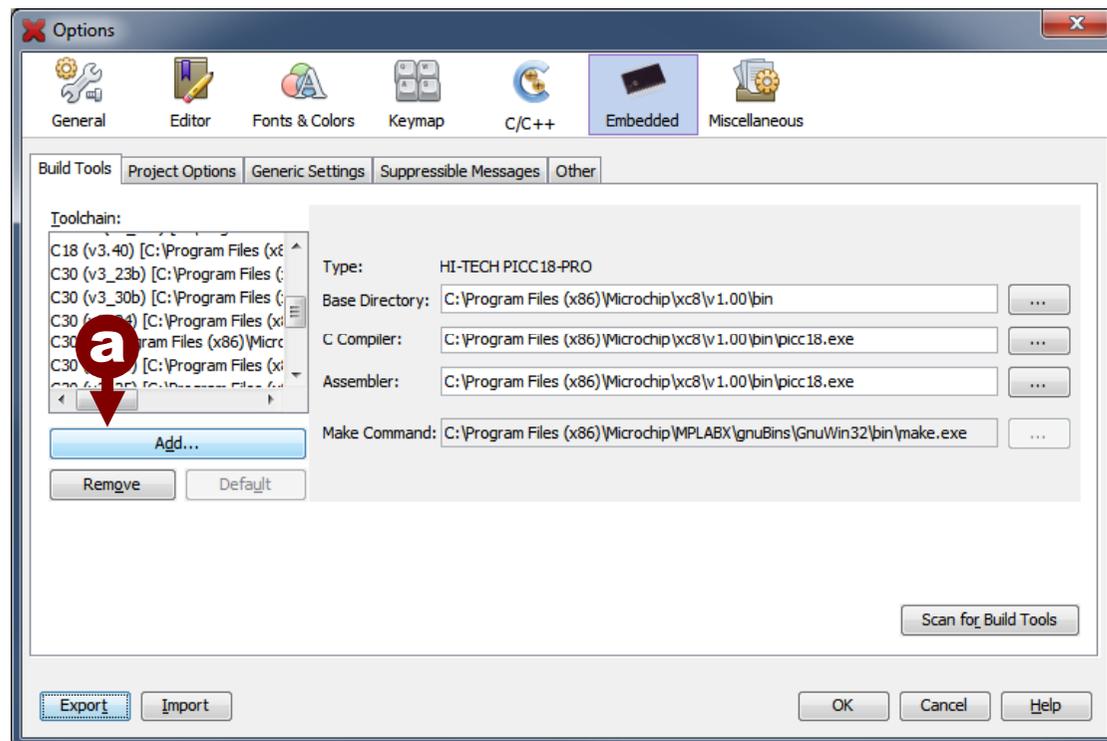
If a compiler is not found, continue to step 3.



# How to configure a C compiler

## 3 Add a new compiler

a Click on the **Add...** button

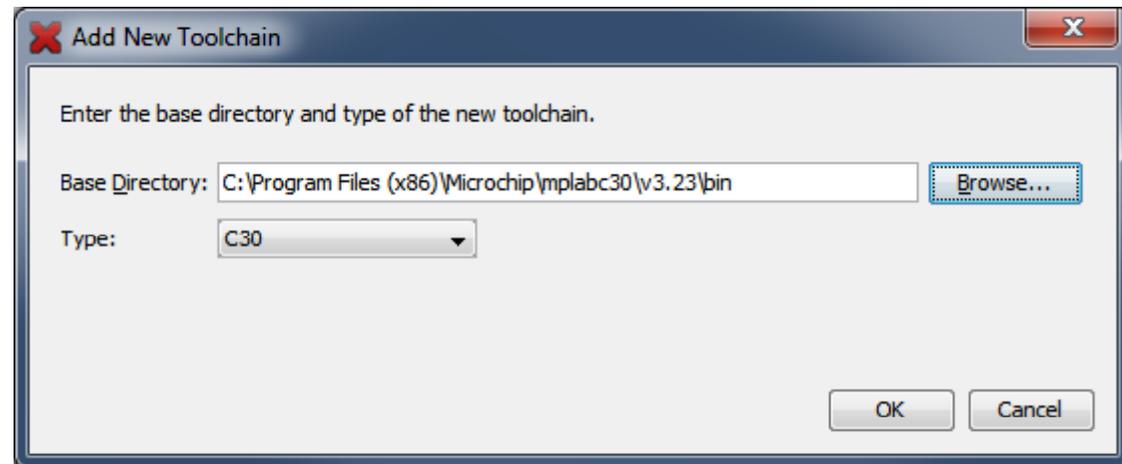


# How to configure a C compiler

## 4 Specify the base directory

- a** Click on the **Browse...** button and navigate to the compiler's base directory (usually called **bin**)

The **Type** box should be automatically populated with the correct type. It may be changed manually if the type wasn't properly detected.

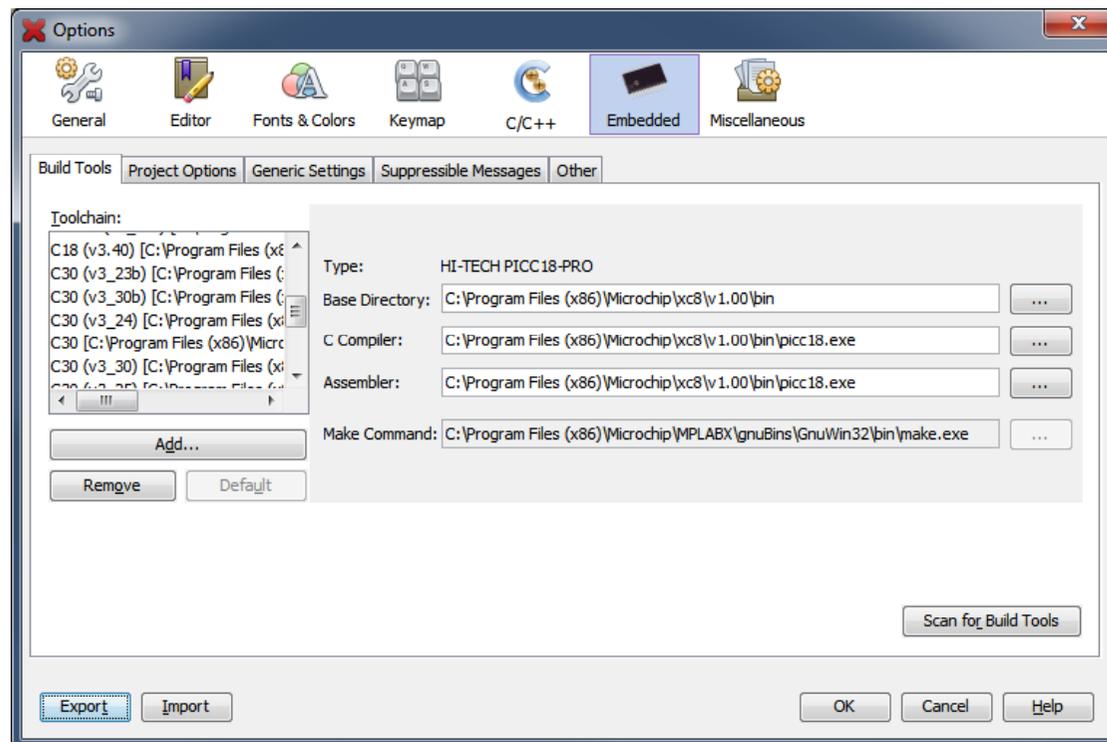


- b** Click on the **OK** button

# How to configure a C compiler

## 5 Done

All paths will be automatically populated





## Lab 1

# How to create a new standalone project

# Lab 1

## How to create a standalone project



### Purpose

- **Create a project from scratch using the project wizard**
- **Describe the various options chosen in the project wizard**
- **Provide a platform on which you can build your own projects**

# Lab 1

## How to create a standalone project

### Objective

- **Create a new MPLAB<sup>®</sup> X IDE project from scratch with the provided source files**
- **Blink two LEDs and display text on the LCD module of the Explorer 16 board**
  - Simulated by Proteus VSM Viewer for online classes
  - Executed on hardware for some on-site classes

# Lab 1

## How to create a standalone project



### Procedure

- **Choose project type**
- **Specify target device**
- **Specify debugger/programmer**
- **Specify compiler**
- **Specify project name and location**
- **Add files to the project**
- **Run project (in simulator or on hardware)**

# Lab 1

## How to create a standalone project

### 1 Launch the New Project Wizard

#### Toolbar



#### Menu

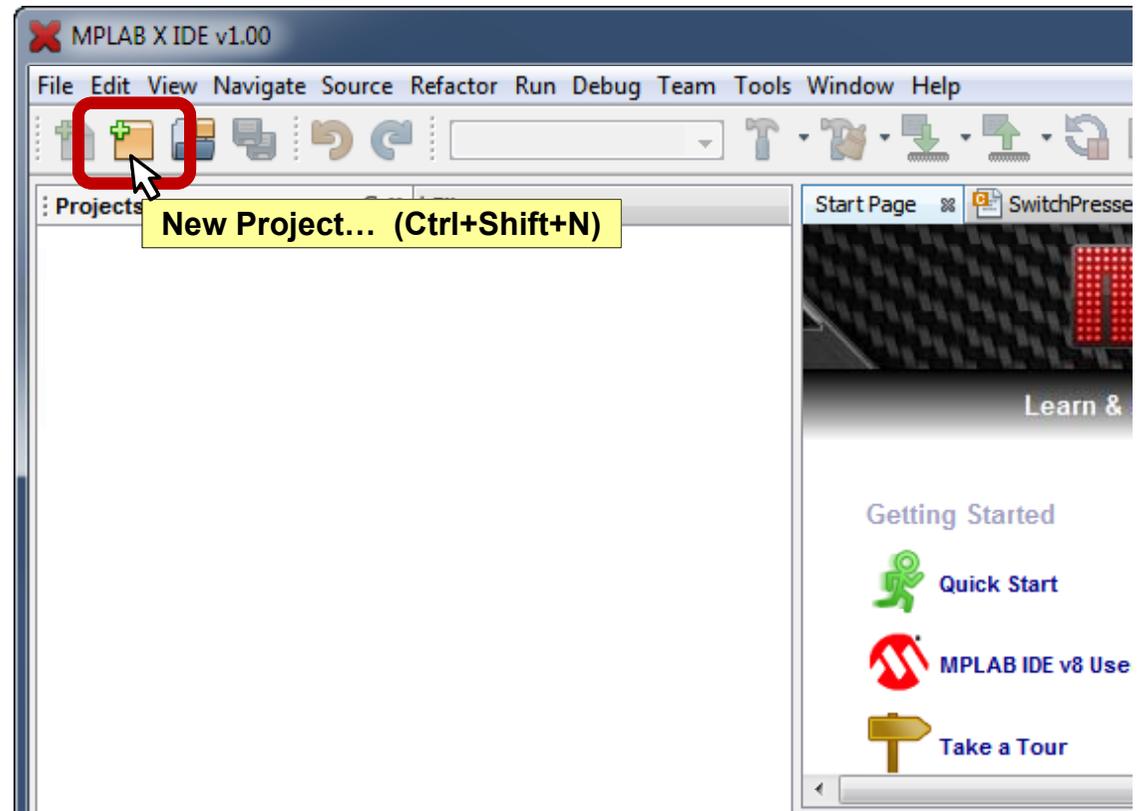
**File** ►  
**New Project...**

#### Keyboard

Ctrl

↑ Shift

N



# Lab 1

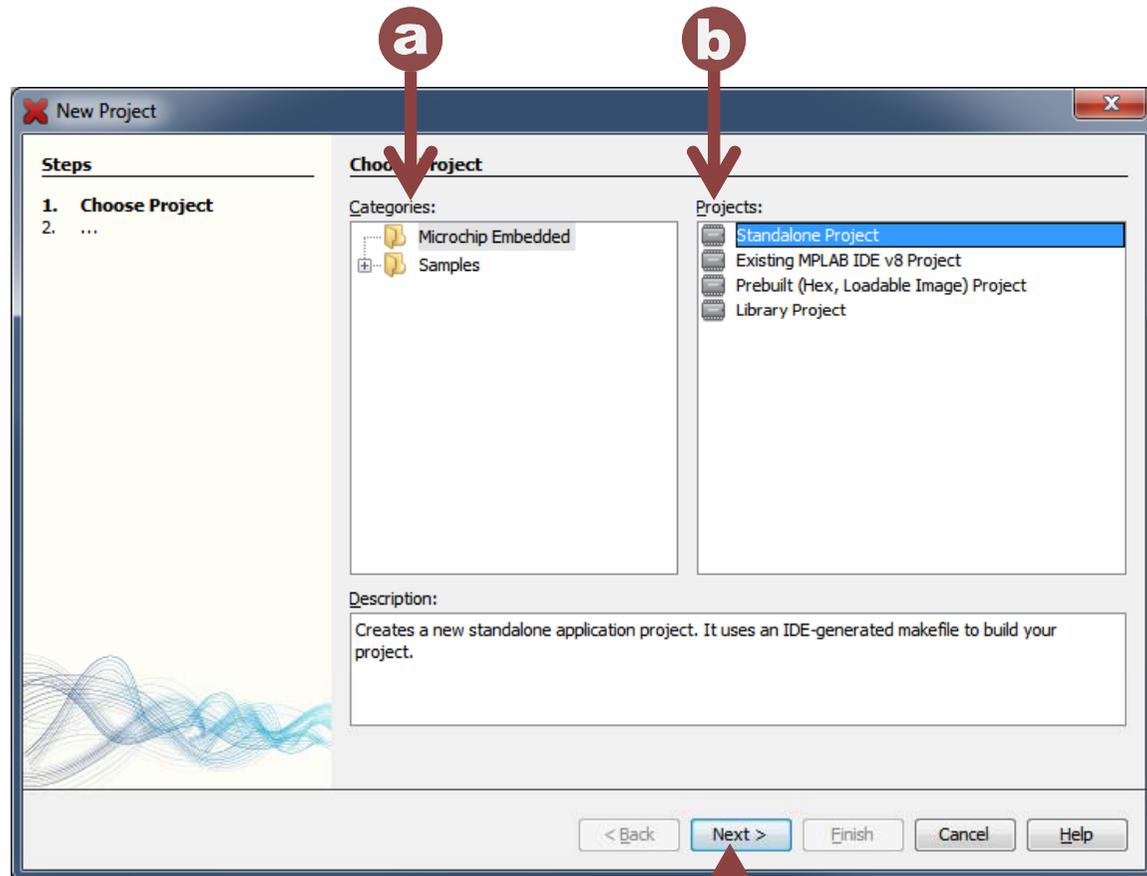
## How to create a standalone project

### 2 Choose Project

**a** Under “Categories” select:  
**Microchip Embedded**

**b** Under “Projects” select:  
**Standalone Project**

**c** Click **Next >**



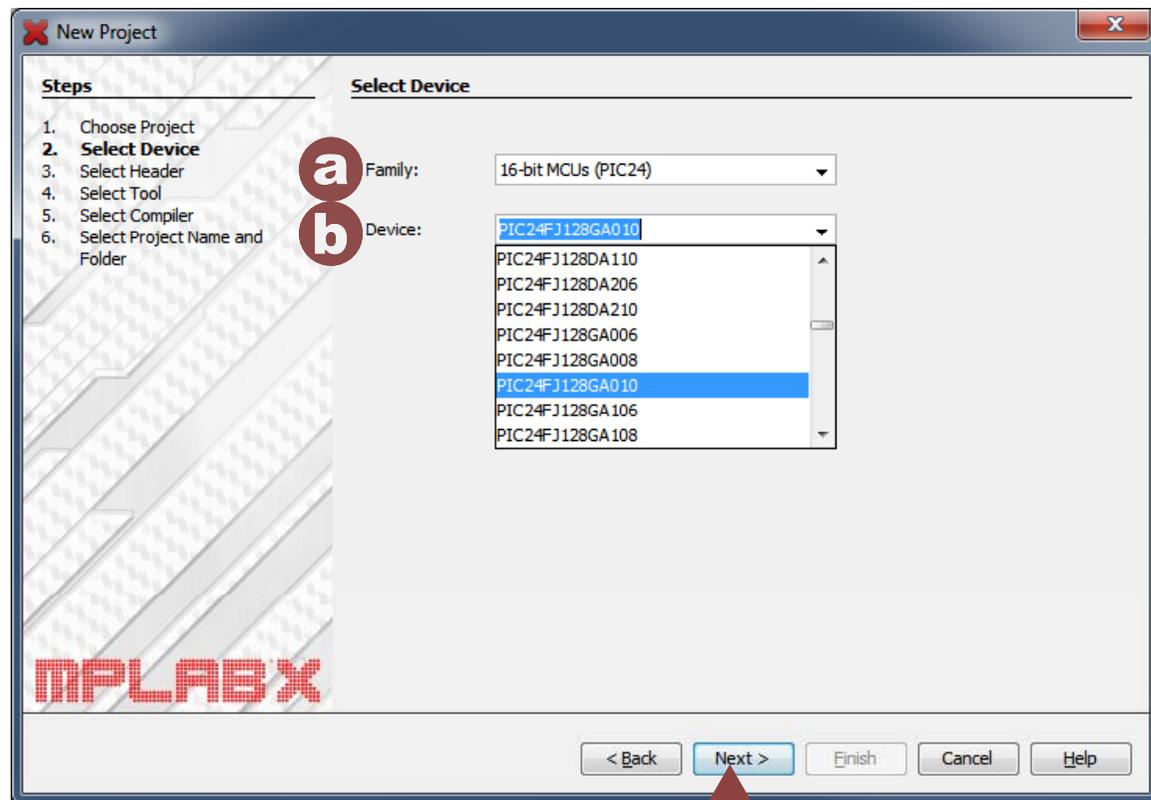
# Lab 1

## How to create a standalone project

### 3 Select Device

**a** For “Family” select:  
**16-bit MCUs (PIC24)**

**b** For “Device” select:  
**PIC24FJ128GA010**



**c** Click **Next >**

# Lab 1

## How to create a standalone project

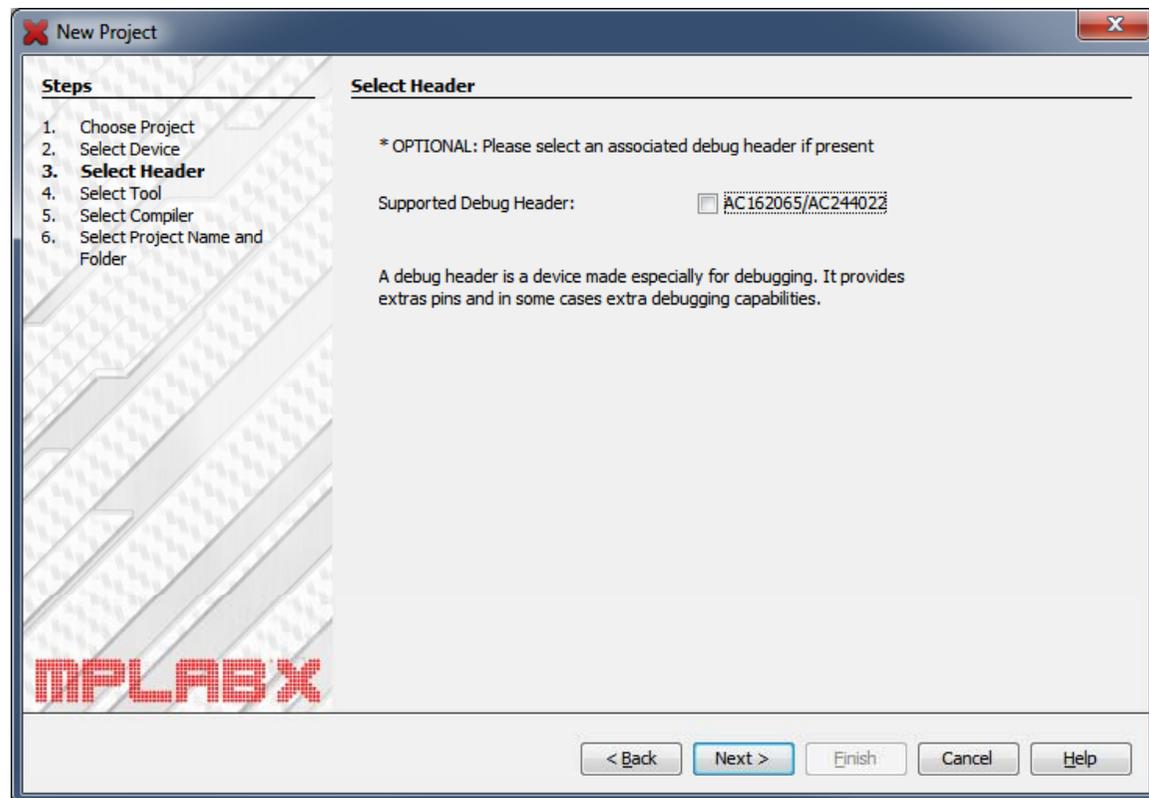
### 4 Select Header

No headers are required for the devices used in this class.

**Leave the box unchecked.**



Click **Next >**



# Lab 1

## How to create a standalone project

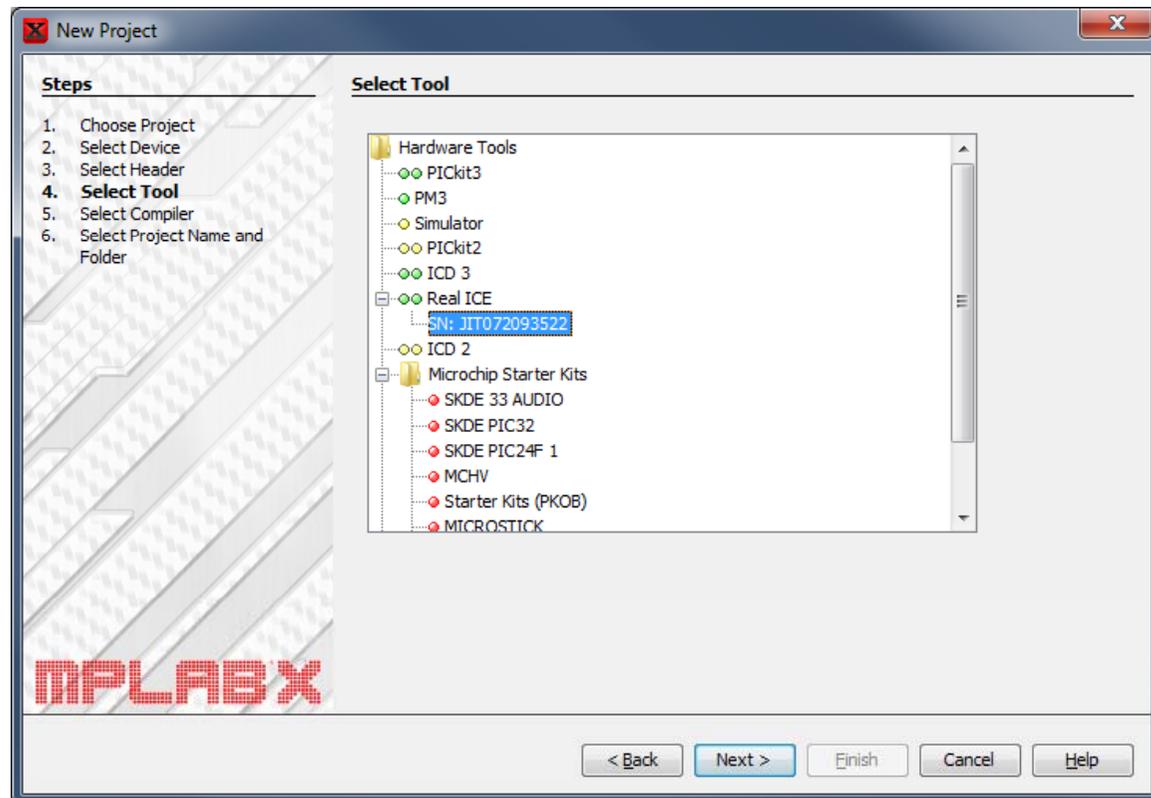
### 5 Select Tool

#### a Select Real ICE SN



If using a hardware debug tool, select its serial number as seen under REAL ICE on the right.

#### b Click **Next >**



# Lab 1

## How to create a standalone project

### 6 Select Compiler

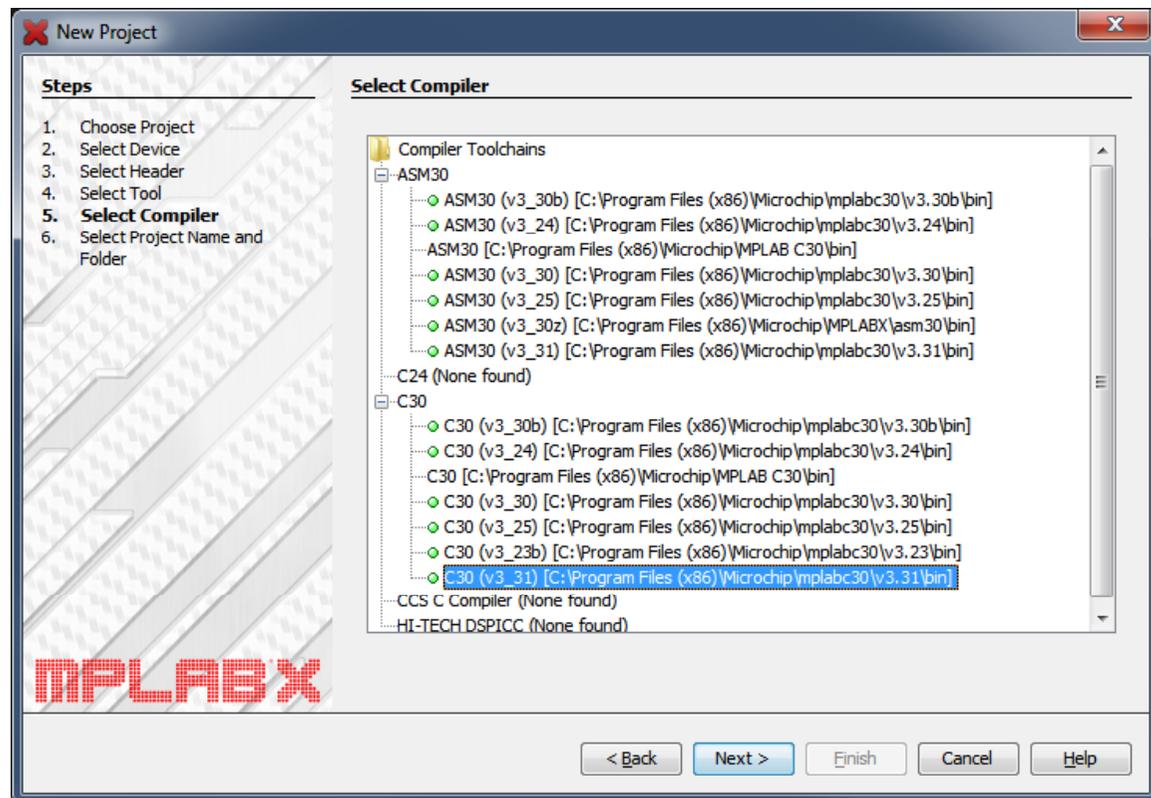
- a** Click on the compiler's version number below the name of the compiler you are using.



If you don't see a version number under a compiler name then it either isn't installed or the IDE cannot find it.

**b**

Click **Next >**



# Lab 1

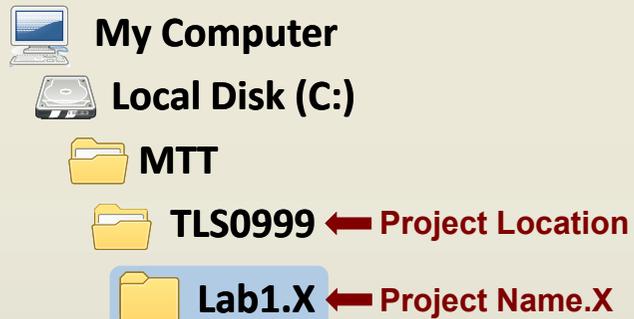
## How to create a standalone project

### 7 Select Project Name and Folder

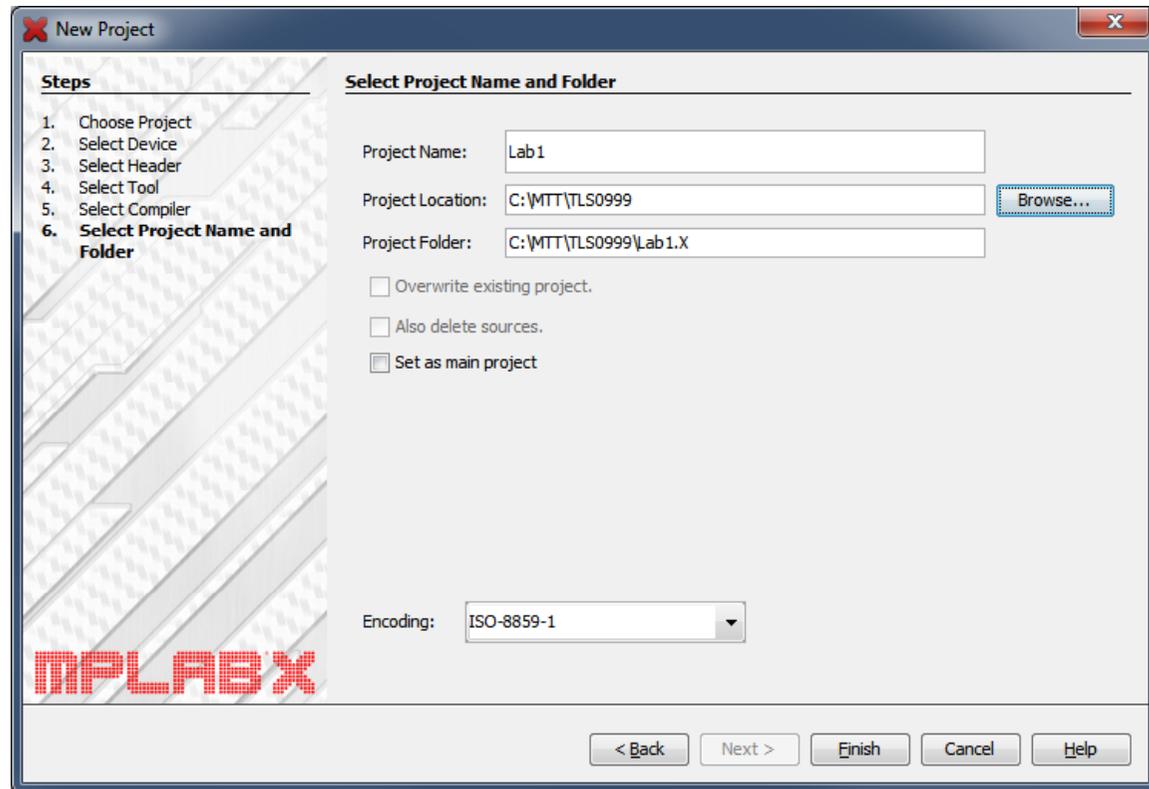
**a** Enter a *Project Name*:  
**Lab1**

**b** Enter a *Project Location*:  
**C:\MASTERS\1601**

A folder with the project name will be created in the project location.



**c** Click **Finish**



# Lab 1

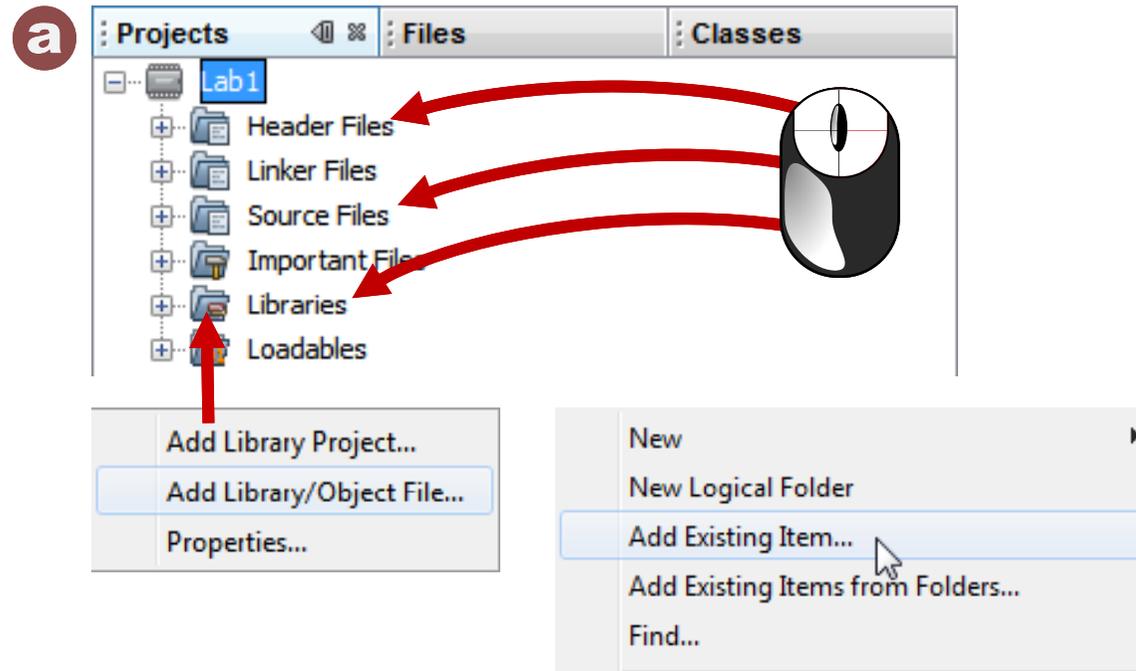
## How to create a standalone project

### 8 Add files to the new project

- a** In the project tree, right click on the appropriate logical folder and select **Add Existing Item...** from the popup menu for the Header and Source files and **Add Library/Object File...** for Libraries

We will add files to the following logical folders:

- b** Header Files
  - c** Source Files
  - d** Libraries
- on the next slide



# Lab 1

## How to create a standalone project

### 8 Add files to the new project (continued...)

Add the following files from the **Lab1.X** directory

**b** Header Files

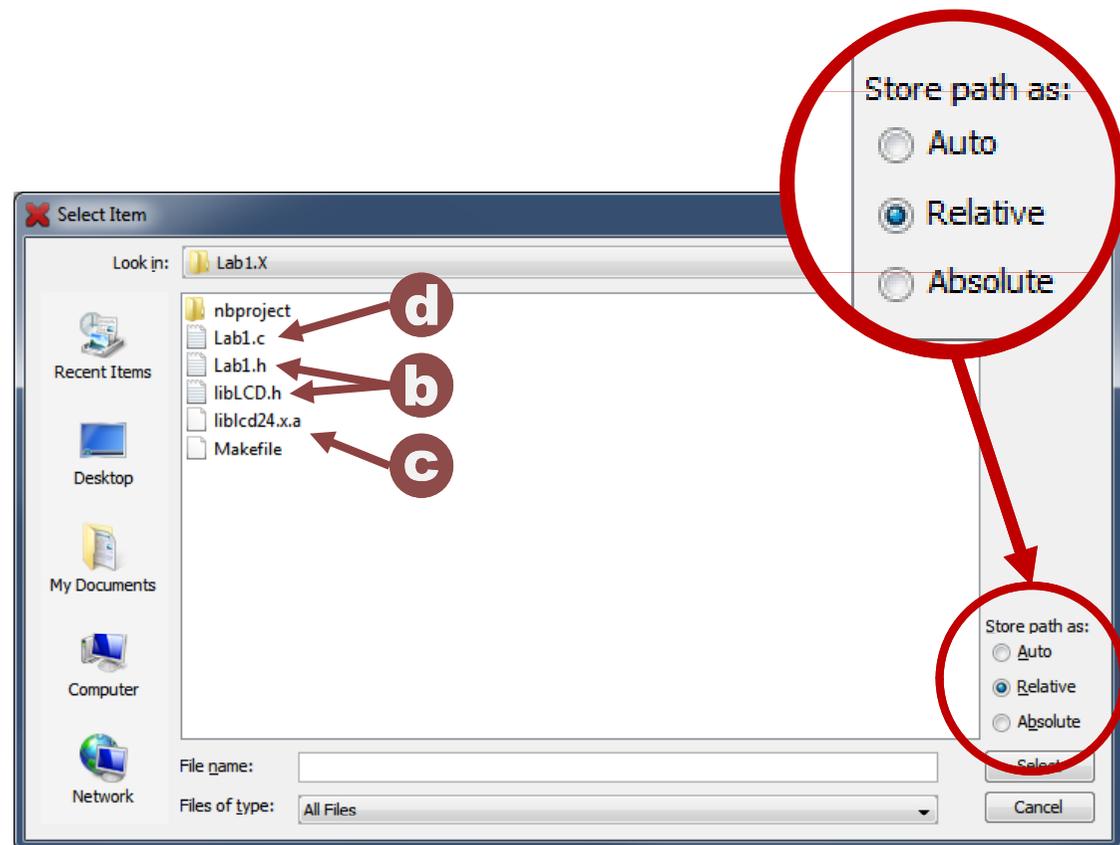
 **libLCD.h**  
 **Lab1.h**

**c** Source Files

 **Lab1.c**

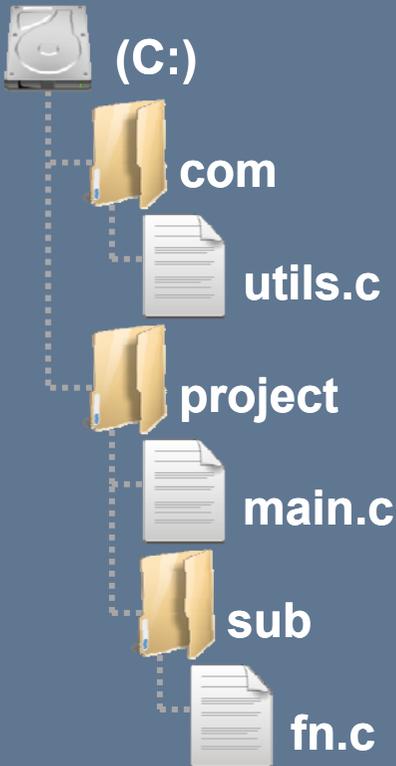
**d** Libraries

 **libLCD24.a**



# Lab 1

## How to create a standalone project

Path Storage Example	Absolute	Relative <small>.. = Go up one level</small> <small>. = Project directory</small>	Auto	
 <p>(C:)</p> <ul style="list-style-type: none"> <li>com           <ul style="list-style-type: none"> <li>utils.c</li> </ul> </li> <li>project           <ul style="list-style-type: none"> <li>main.c</li> <li>sub               <ul style="list-style-type: none"> <li>fn.c</li> </ul> </li> </ul> </li> </ul>	Outside Project Directory			
	<b>C:\com\utils.c</b>	<b>..\com\utils.c</b>	<b>C:\com\utils.c</b>	
				uses absolute
	Inside Project Directory			
	<b>C:\project\main.c</b>	<b>.\main.c</b>	<b>.\main.c</b>	
				uses relative
	Inside Project Subdirectory			
	<b>C:\project\sub\fn.c</b>	<b>.\sub\fn.c</b>	<b>.\sub\fn.c</b>	
				uses relative

# Lab 1

## How to create a standalone project

### 9 Build and Run Project

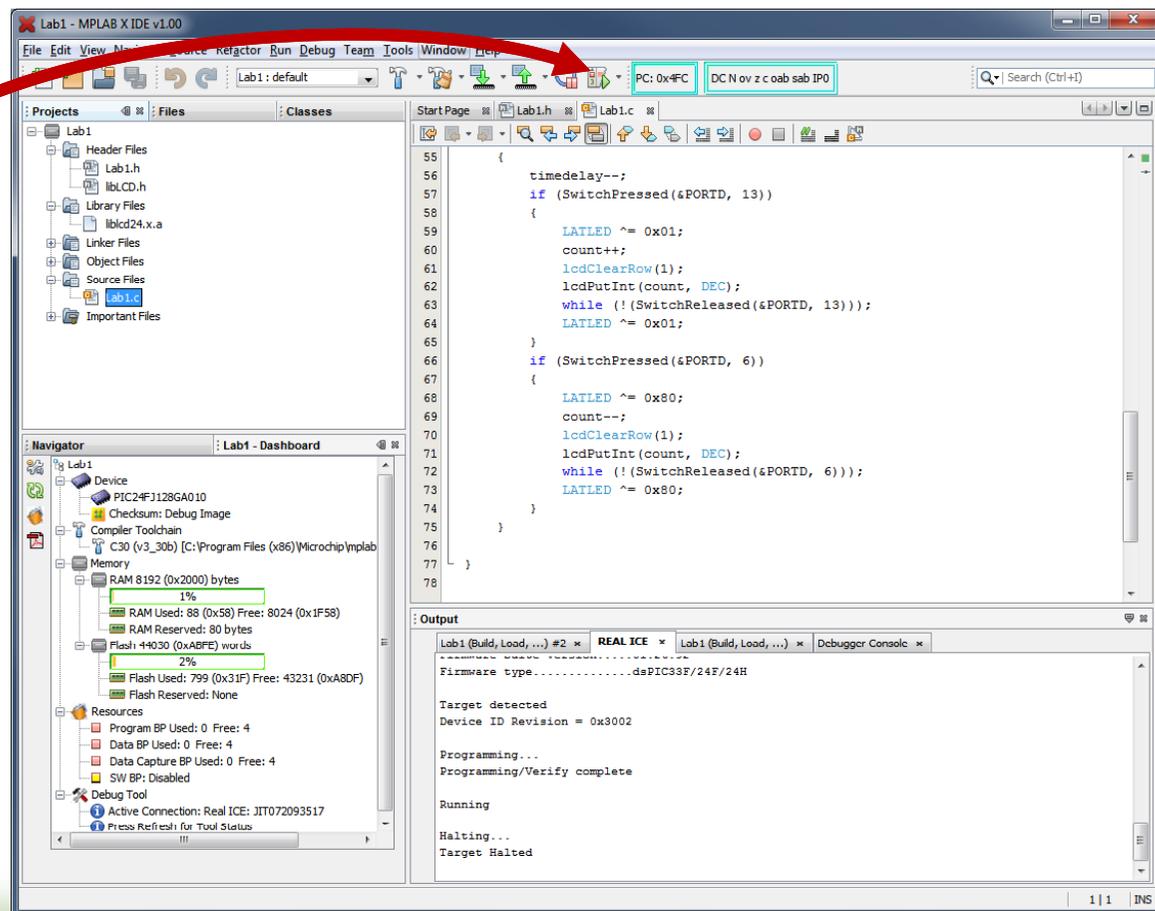
Build the project to ensure that everything was done properly.



Click on the **Debug Project** icon.

This button will:

1. Build (**make**) your project in **debug** mode
2. Program the target device on the board
3. Run your code



# Lab 1

## How to create a standalone project

### Results

If everything was done correctly, you should see "80" on the LCD and two of the LEDs should be blinking.

When done:

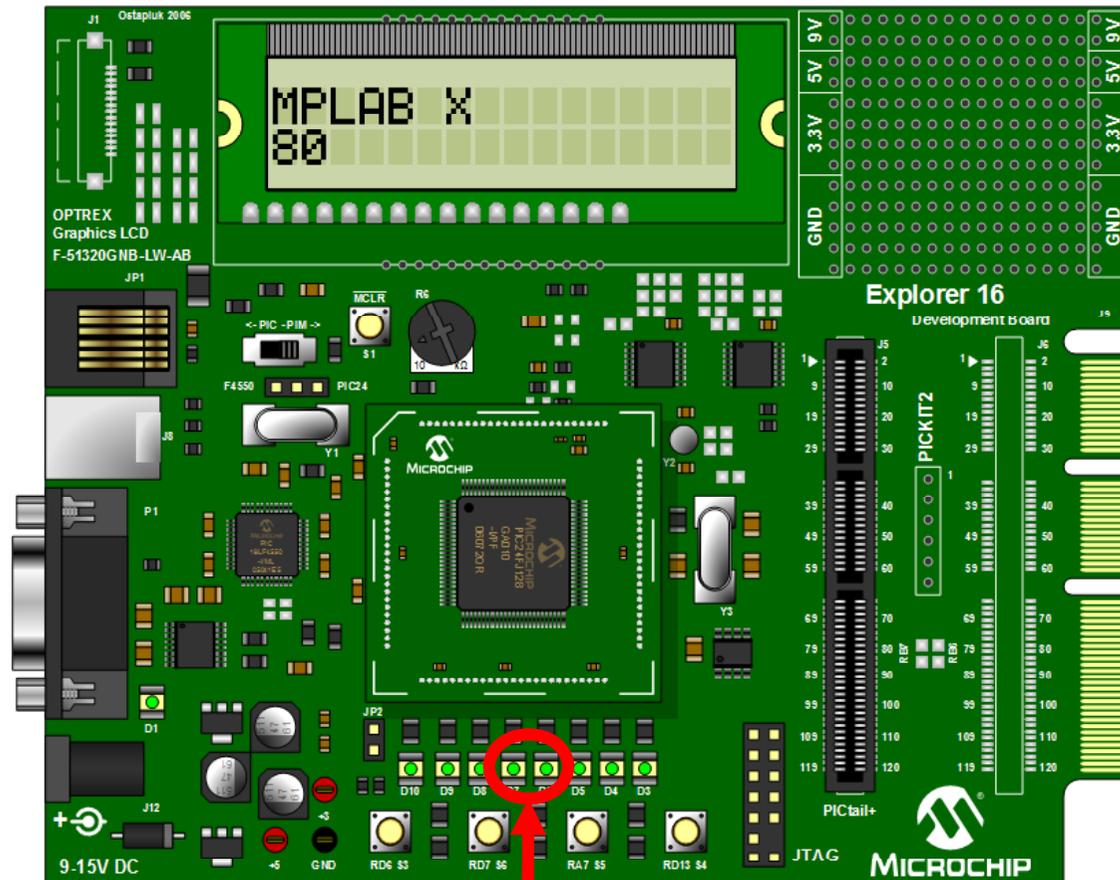
Click on the **Pause** button:



Click on the **End Debug Session** button:



(more on these later...)



**Blinking LEDs**

# Lab 1

## How to create a standalone project



### Conclusions

- **You now know how to create a standalone project including:**
  - Selecting a target device
  - Selecting a build and debug tool
  - Choosing a project directory
  - Building and running code on a target board or in the simulator



# How to build projects

# How to build applications

## MPLAB® IDE 8



### Make

(Menu: **Project** ▶ **Make**)

Build only files that have changed since the last build



### Build All

(Menu: **Project** ▶ **Build All**)

Builds all files regardless of whether or not they have changed since the last build

## MPLAB X IDE



### Build Project

Build only files that have changed since the last build



### Clean and Build Project

Builds all files regardless of whether or not they have changed since the last build

# How to build applications

## Working with a Debugger

### MPLAB® IDE 8

Debug  **Debug Mode**



### MPLAB X IDE



#### **Debug Project**

- Debug Mode
- Make
- Program Target Device
- Reset
- Run (Optional)



In most situations, it is not necessary to hit the "Build" or "Clean and Build" button before hitting this button.

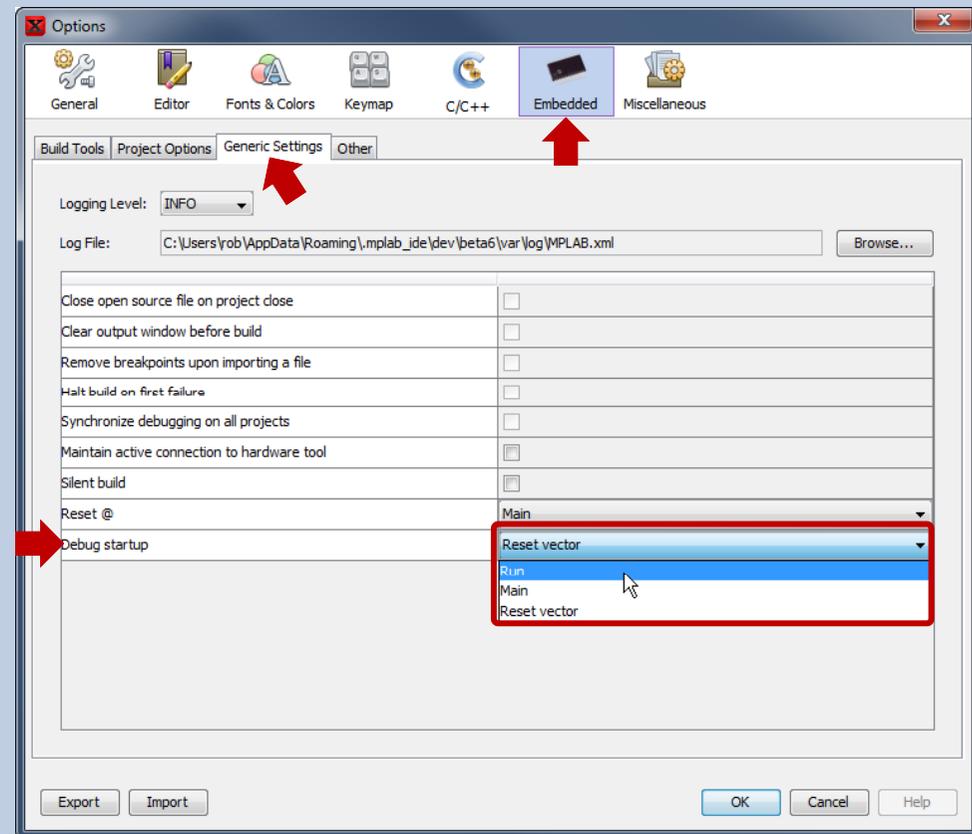
# How to build applications

## Working with a Debugger



If you don't want the debugger to automatically start running:

- 1 Select from the main menu: **Tools** ▶ **Options**
- 2 Select the **Embedded** icon
- 3 Select the **Generic Settings** tab
- 4 For the **Debug startup** setting choose either **Main** or **Reset vector** instead of **Run**



# How to build applications

## Working with a Programmer

### MPLAB® IDE 8

Release Release Mode



### MPLAB X IDE



#### Program Target Project

- Release Mode
- Make
- Program Target Device



Optionally, you may toggle the Hold in Reset button which will allow the target to run without unplugging it from the programmer.

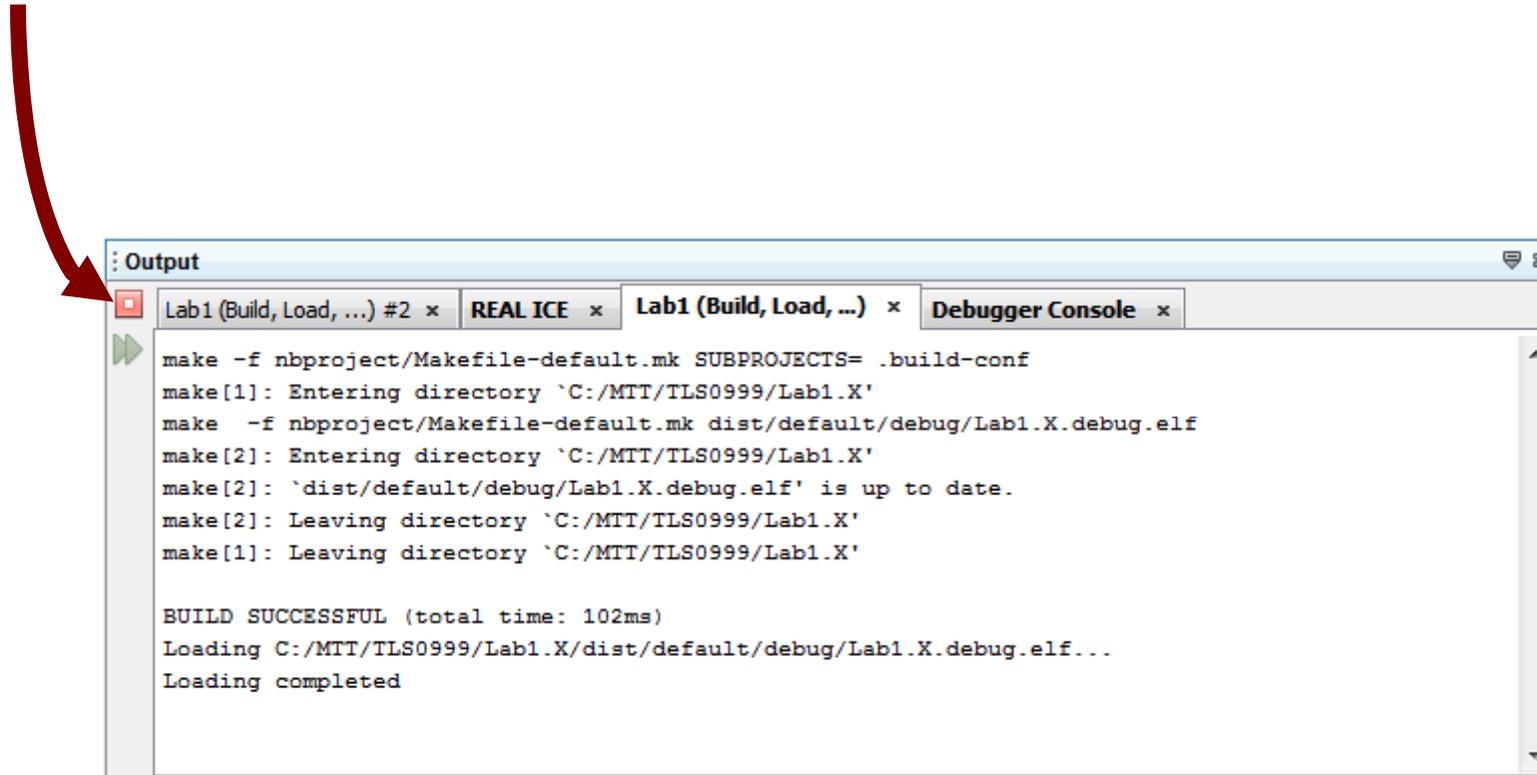


In most situations, it is not necessary to hit the "Build" or "Clean and Build" button before hitting this button.

# How to build applications

## Stopping builds before they finish

- Click on the  icon in the left margin of the Output window



# How to build applications

## Gotchas – for now...

- If code or header files are referenced in code via `#includes`, but are not in the project tree, `make` will not currently detect changes to those files, so a **Clean and Build** must be executed before **Debug** or **Program**

# How to build applications

## Gotchas – platform related errors

- **Relative paths should use '/' instead of '\\':**  
`#include "Graphics/Graphics.h"`
  - '/' works on ALL platforms, including Windows
  - '\\' works ONLY on Windows
  - Compiler on Linux cannot find files with paths using '\\', which is an escape character on Linux
- **Linux is case sensitive**
  - `#include <p24fj128ga010.h>` won't be found because filename is `p24FJ128GA010.h`



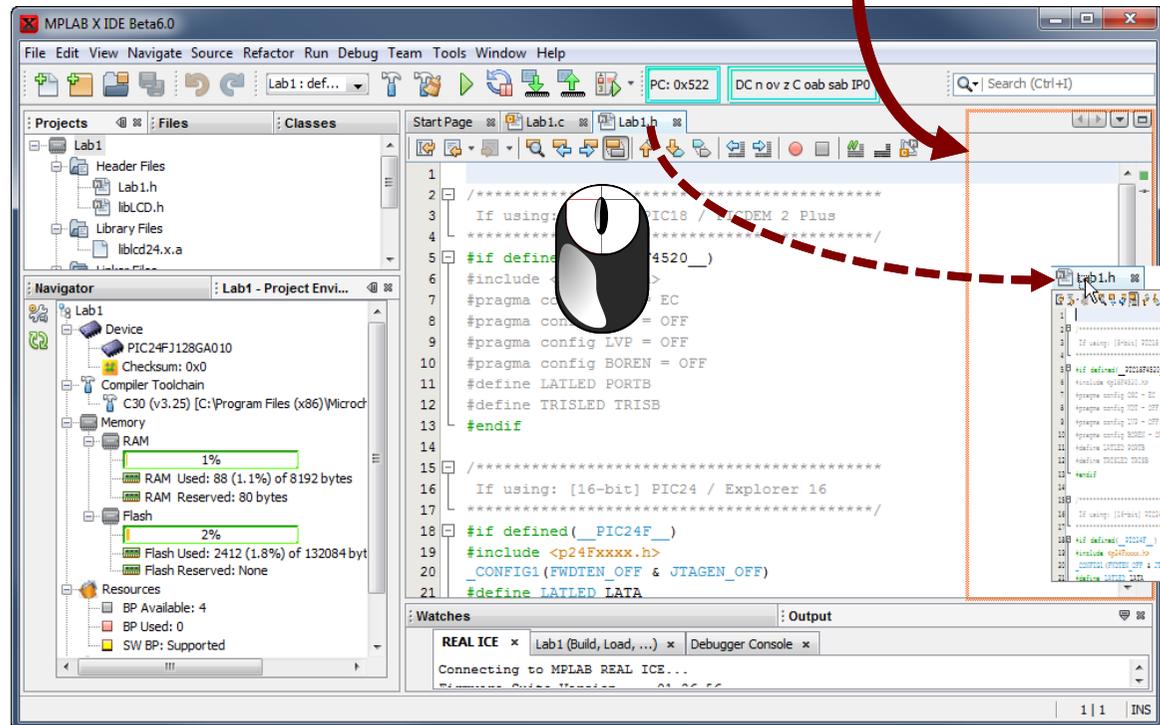
# How to Customize the IDE

# How to view two files side-by-side

## 1 Click on tab and drag into desired position

As you drag a tab away from the top of the editor, notice the red outline that appears behind it. This outline indicates where the new tab will be displayed.

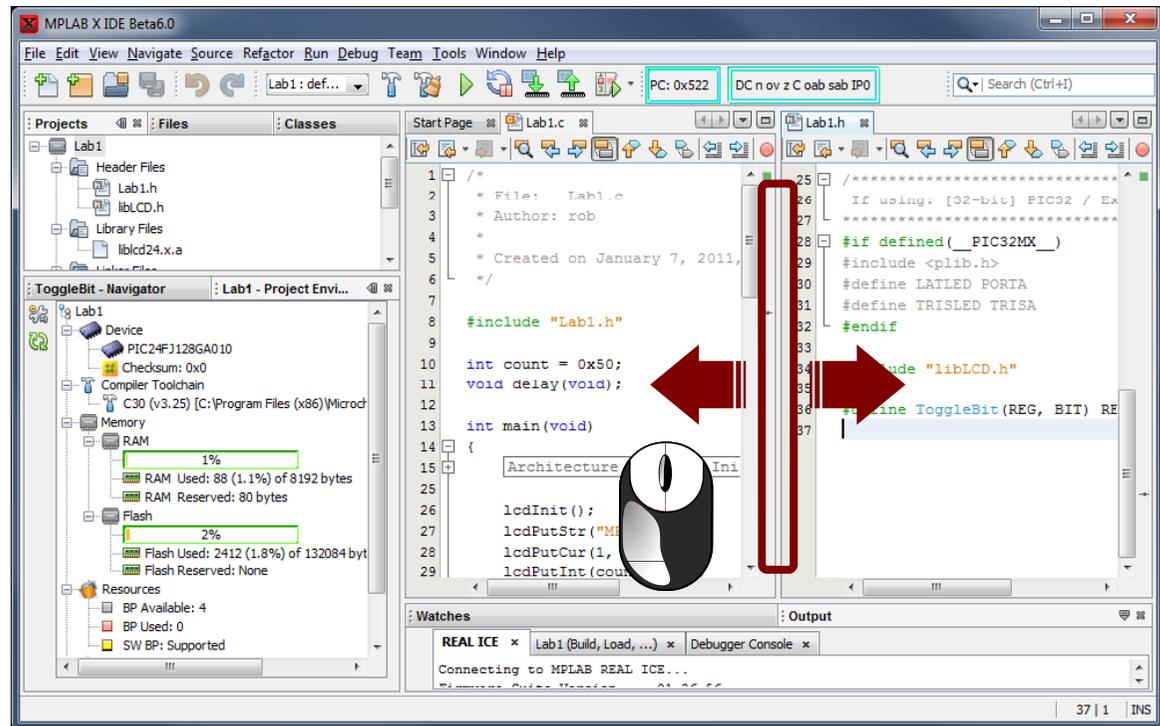
Red box indicates location where tab will "snap in" when mouse button is released



# How to view two files side-by-side

## 2 Release mouse to dock window

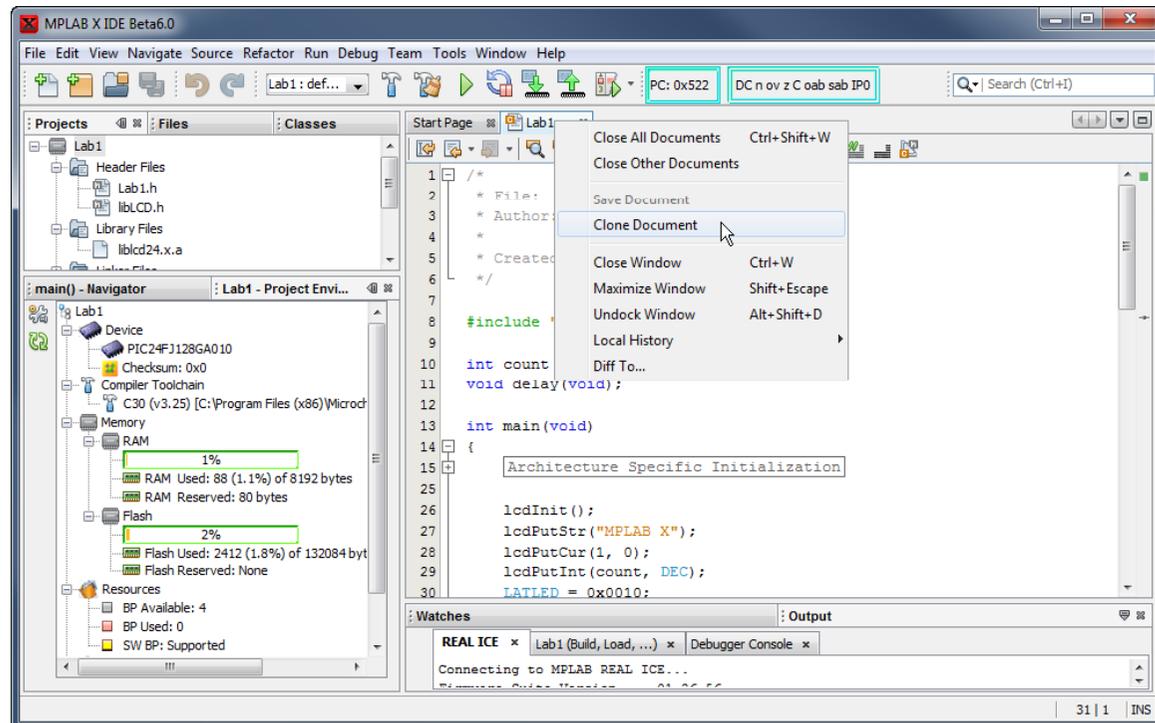
The width of the windows may be adjusted by clicking and dragging in the bar between them.



# How to split view a file

## 1 Right click on tab and select Clone Document

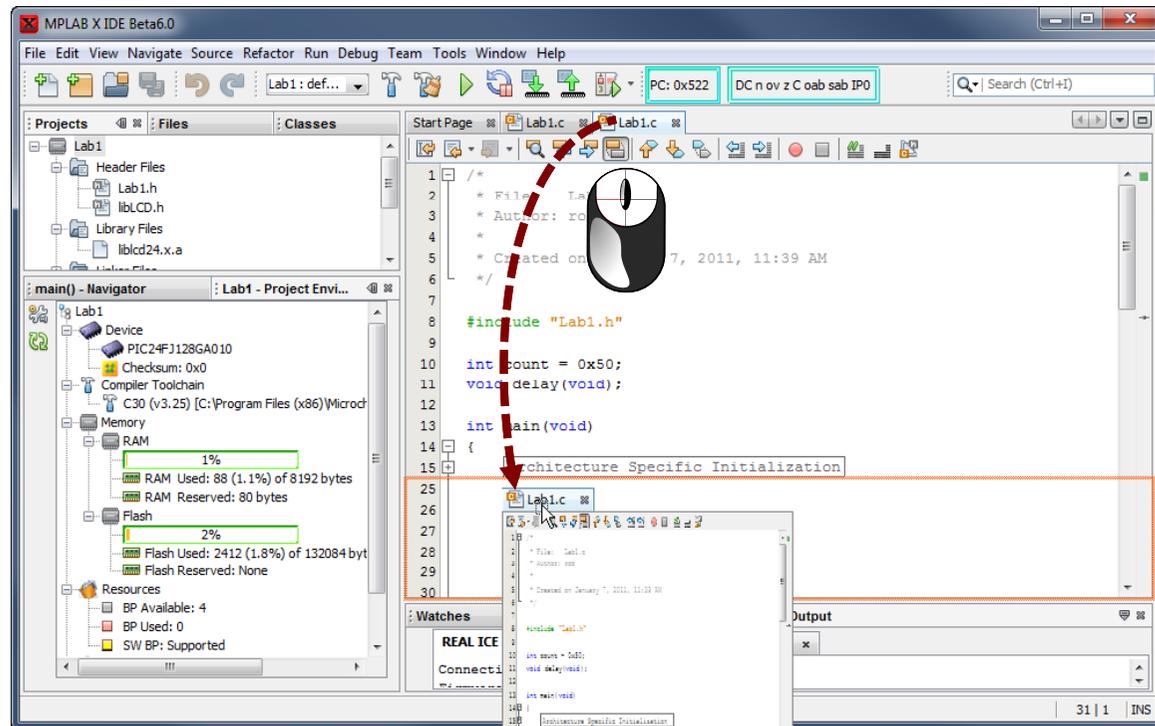
Only one instance of a tab may be opened normally. Cloning a document creates a second instance which allows you to view the same file from different locations at the same time.



# How to split view a file

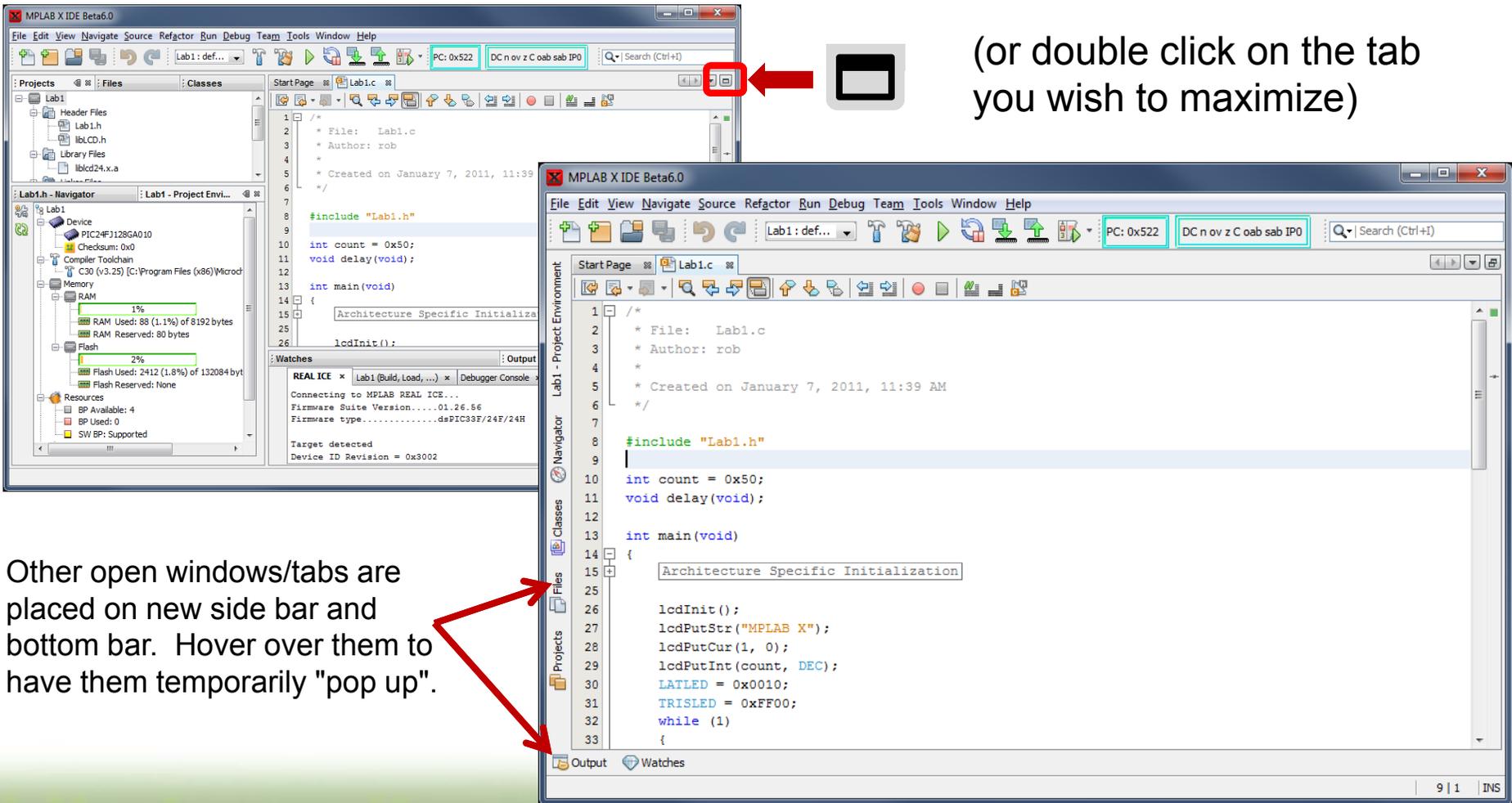
## 2 Click on tab and drag into desired position

Once a second instance is open, it may be moved and docked like any other tab.



# How to maximize the editor

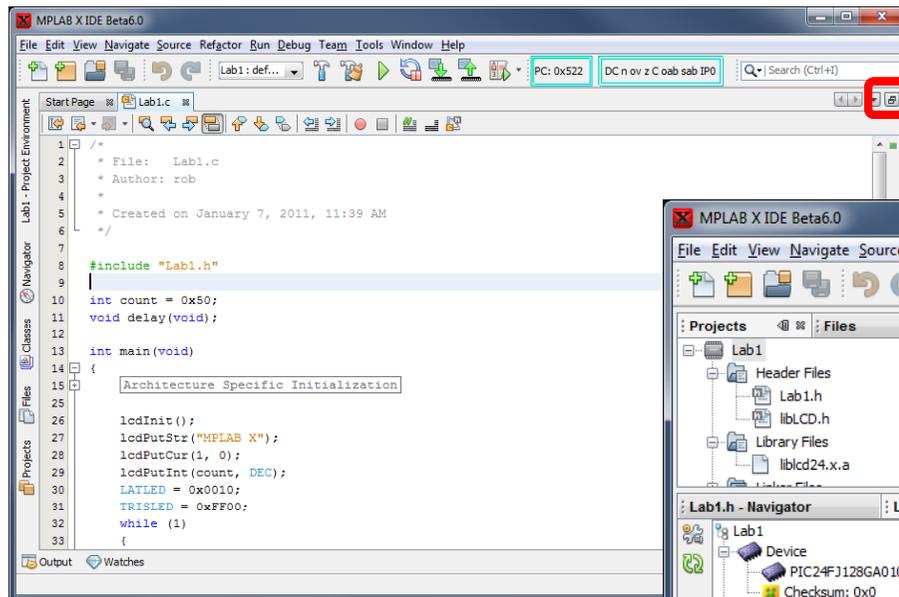
**1** Click on the editor's maximize window button (or double click on the tab you wish to maximize)



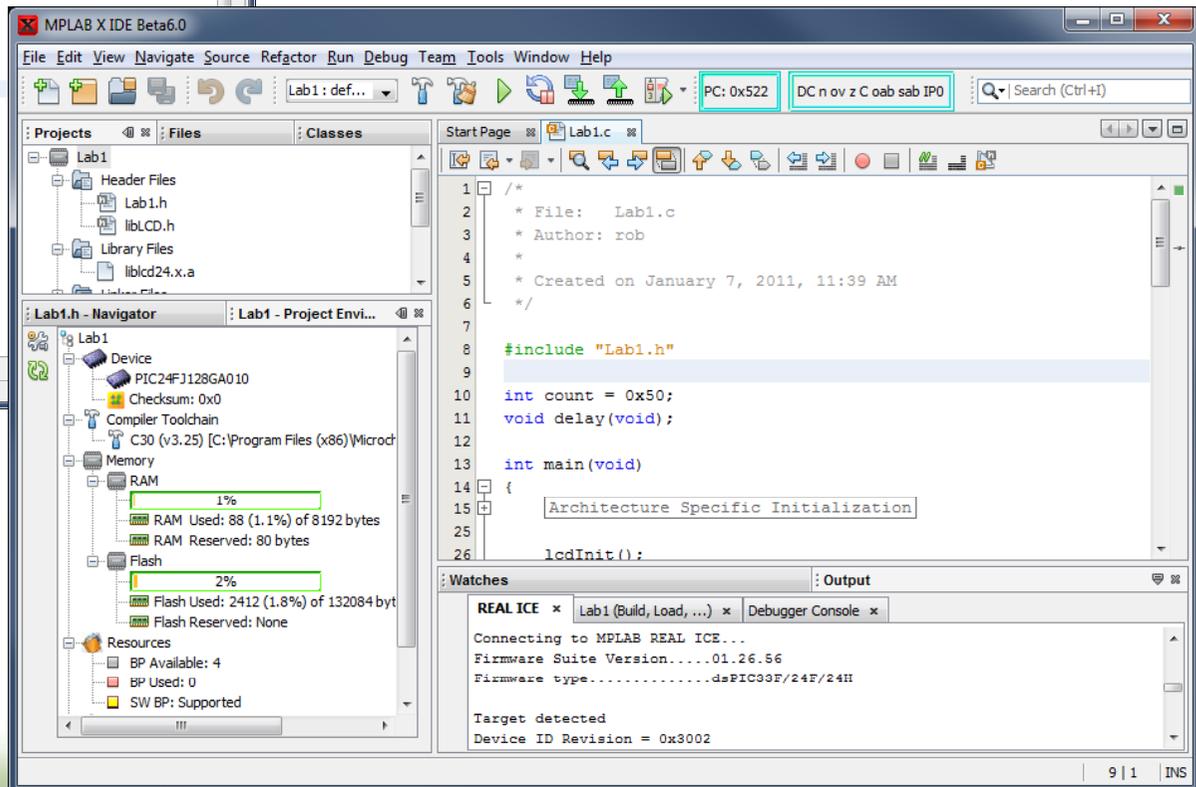
Other open windows/tabs are placed on new side bar and bottom bar. Hover over them to have them temporarily "pop up".

# How to minimize the editor

1 Click on the editor's minimize window button



(or double click on the tab to restore the original layout)

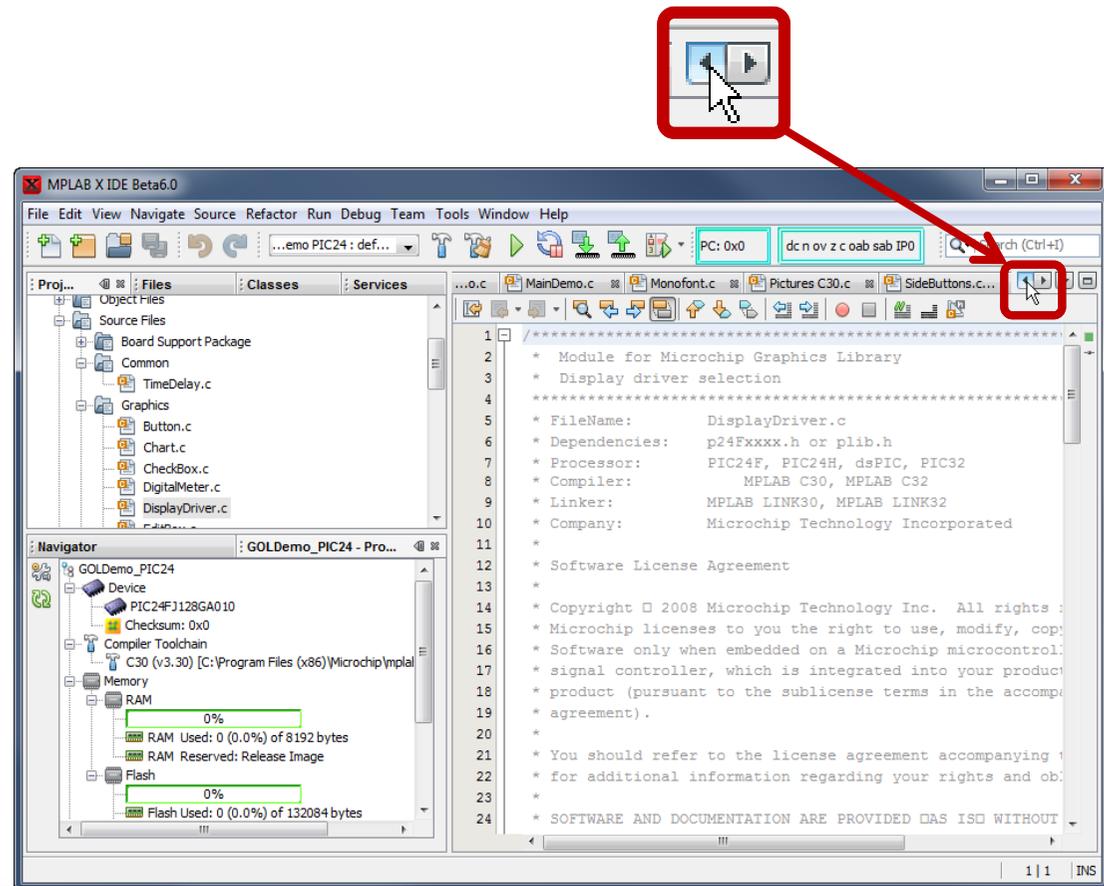


# How to navigate through tabs

## Method 1

### 1 Use the arrow buttons to scroll through tabs at top

When you have more documents open than there is space available to display their tabs, the arrow buttons may be used to "slide" the tabs left or right so that hidden tabs become visible.



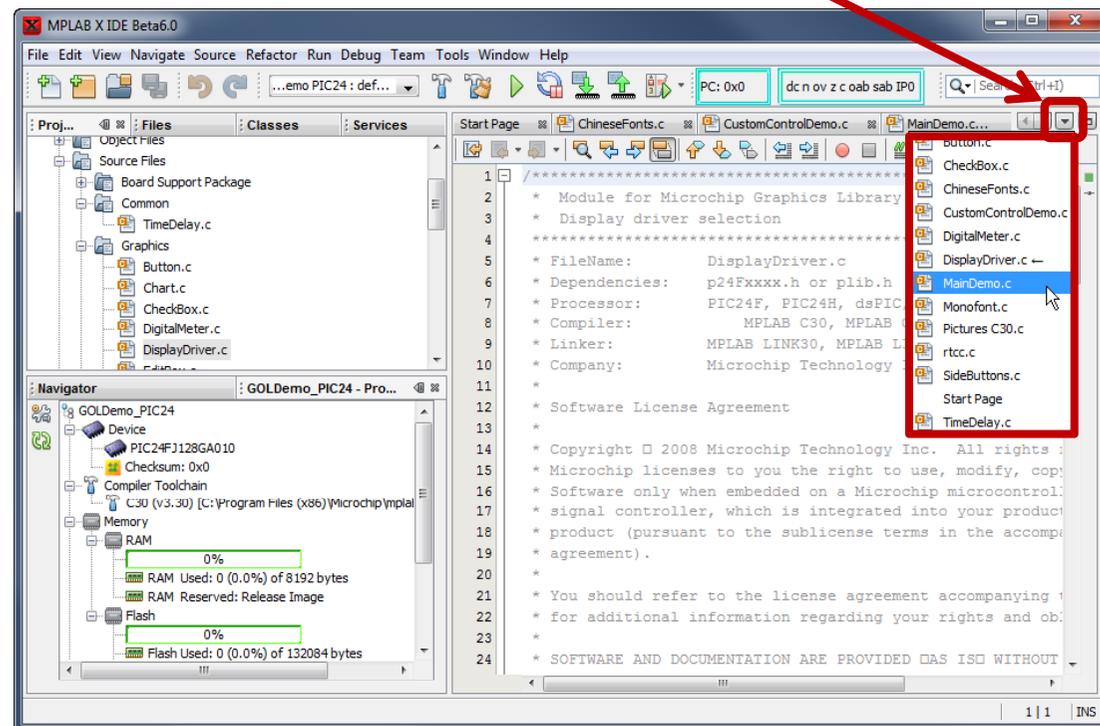
# How to navigate through tabs

## Method 2

### 1 Use the tab list to directly jump to the desired tab

The tab list shows all open tabs so that you can simply select the one you wish to make active in the editor.

The currently active tab has a small arrow to the right of its name.





# How to work with the main toolbar

# How to work with the main toolbar

## Default Buttons



New File



New Project



Open Project



Save All



Undo



Redo



Hold in Reset



Build



Rebuild (Clean and Build)



Program Target



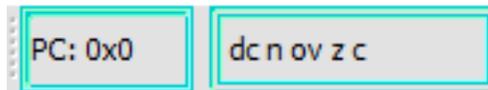
Read Target



Debug (Build, Program, Run)



**Project  
Configuration**



**Program Counter and  
Status Bits**



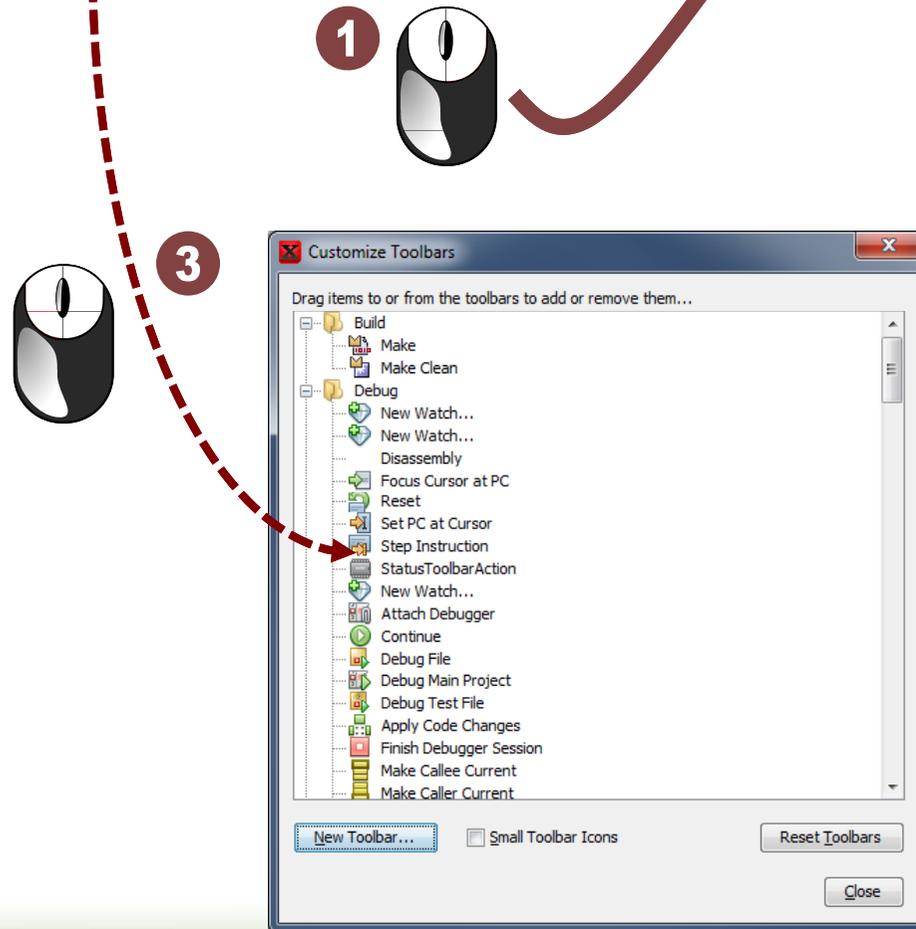
**Quick Search**

# How to work with the main toolbar

## How to Customize the Toolbars



- 1 Right-click on a toolbar
- 2 Select **Customize...** from the popup menu
- 3 Drag items from list to desired location on toolbar



 To remove a button, drag it from the toolbar back to the list. To move a button, remove it first, then add it where you want it.



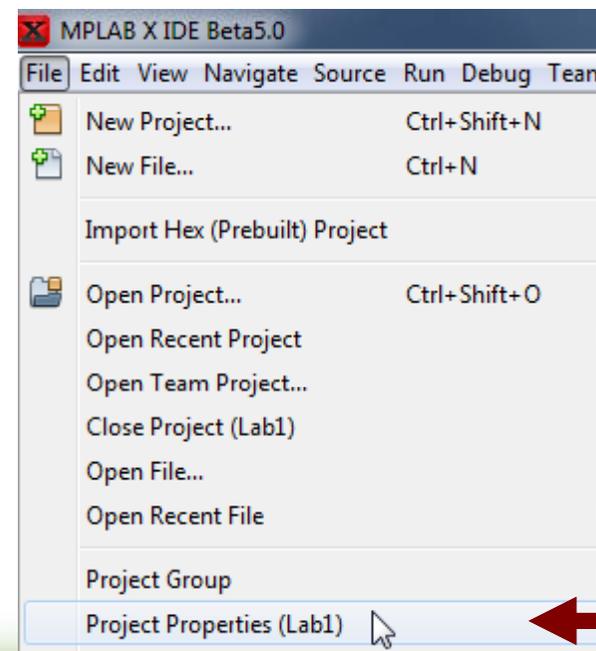
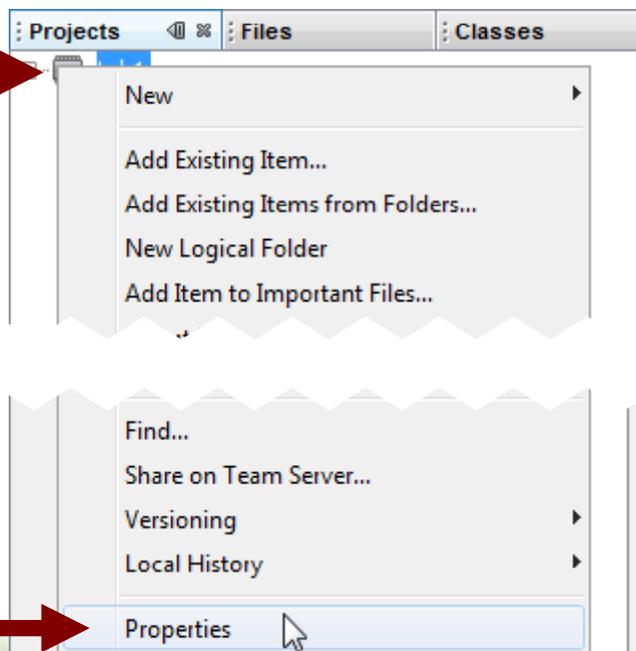
# How to work with projects

# How to change project properties

- Right click on the top node of the project in the project tree and select **Properties** from the popup menu – OR –
- Select **File ► Project Properties (*project name*)**



Bottom of world's longest popup menu



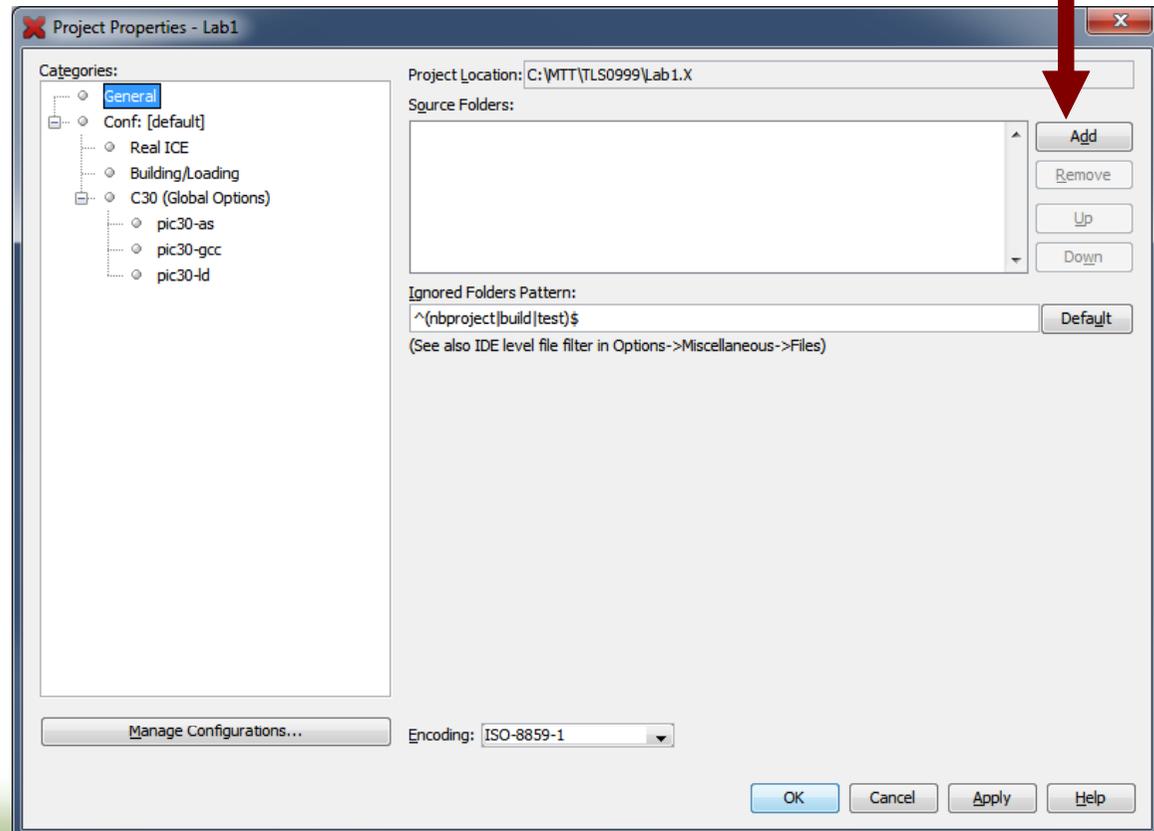
# How to change project properties

## General

**File ► Project Properties  
General**

If using source files from outside the project directory, paths to their directories should be added here.

**Add source folders  
other than the  
project folder**



# How to change project properties

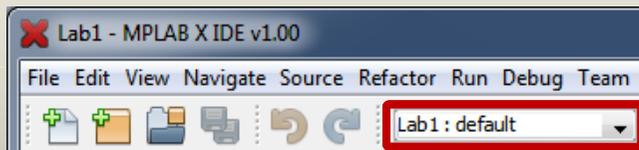
## Device, Build Tools, and Debuggers

### File ► Project Properties Conf

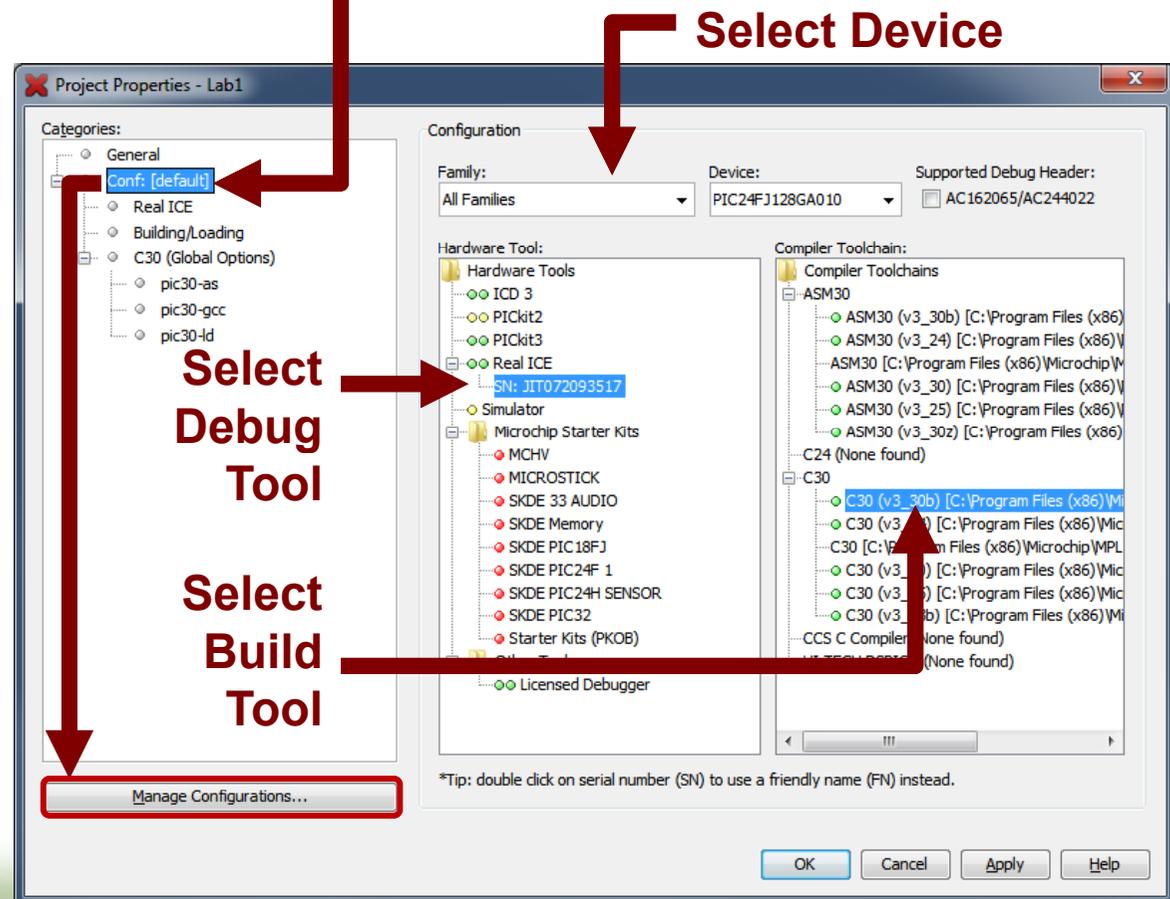
Options for current project configuration including:

- Device
- Hardware Tool
- Compiler Toolchain

Additional configurations may be created for quick swapping of frequently changed options



### Set options for selected configuration



**Select Device**

**Select Debug Tool**

**Select Build Tool**

**Manage Configurations...**

# How to change project properties

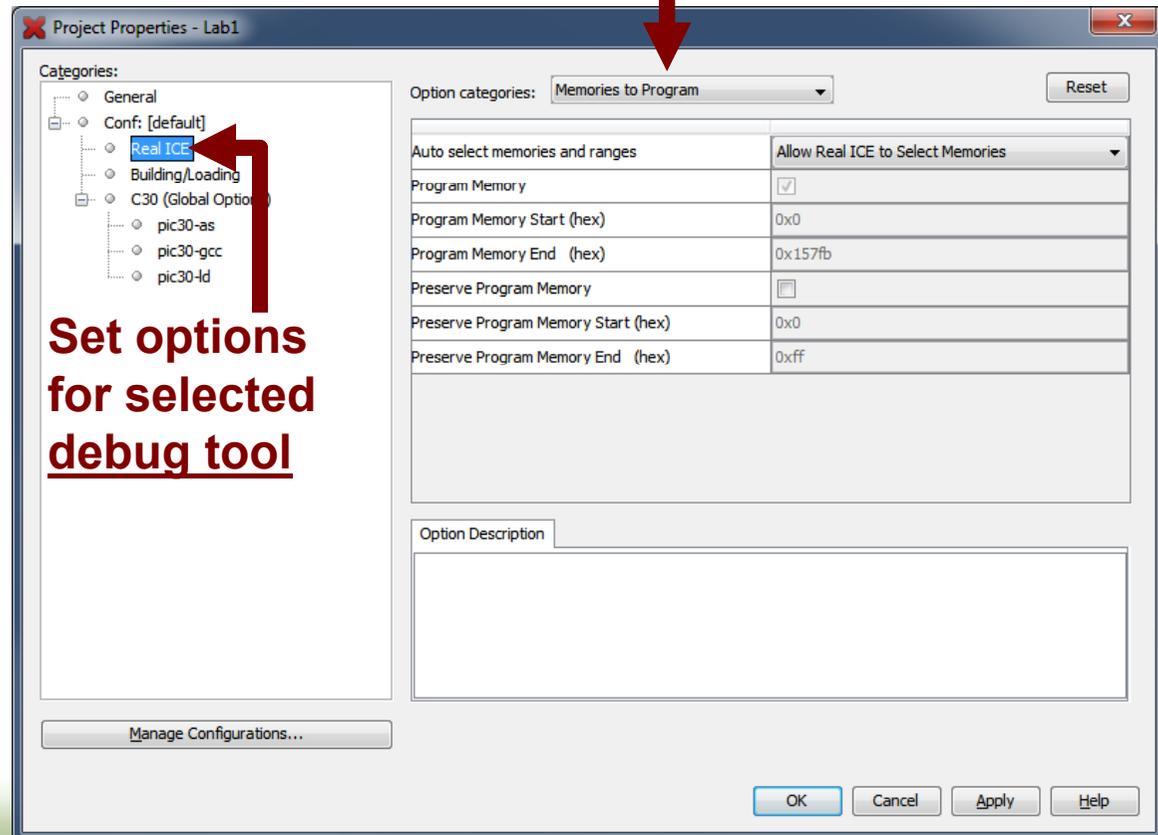
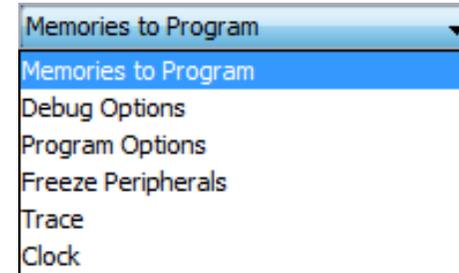
## Debug Tool Properties

File ► Project Properties  
Debugger-Programmer

Debugging and  
Programming  
options including:

- Memory Ranges
- Memory Preservation
- Freeze Peripherals
- Trace
- Clock

**Categories**  
(Same as MPLAB® IDE 8)



**Set options  
for selected  
debug tool**

# How to change project properties

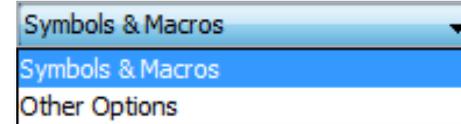
## Assembler Properties

File ► Project Properties  
Assembler

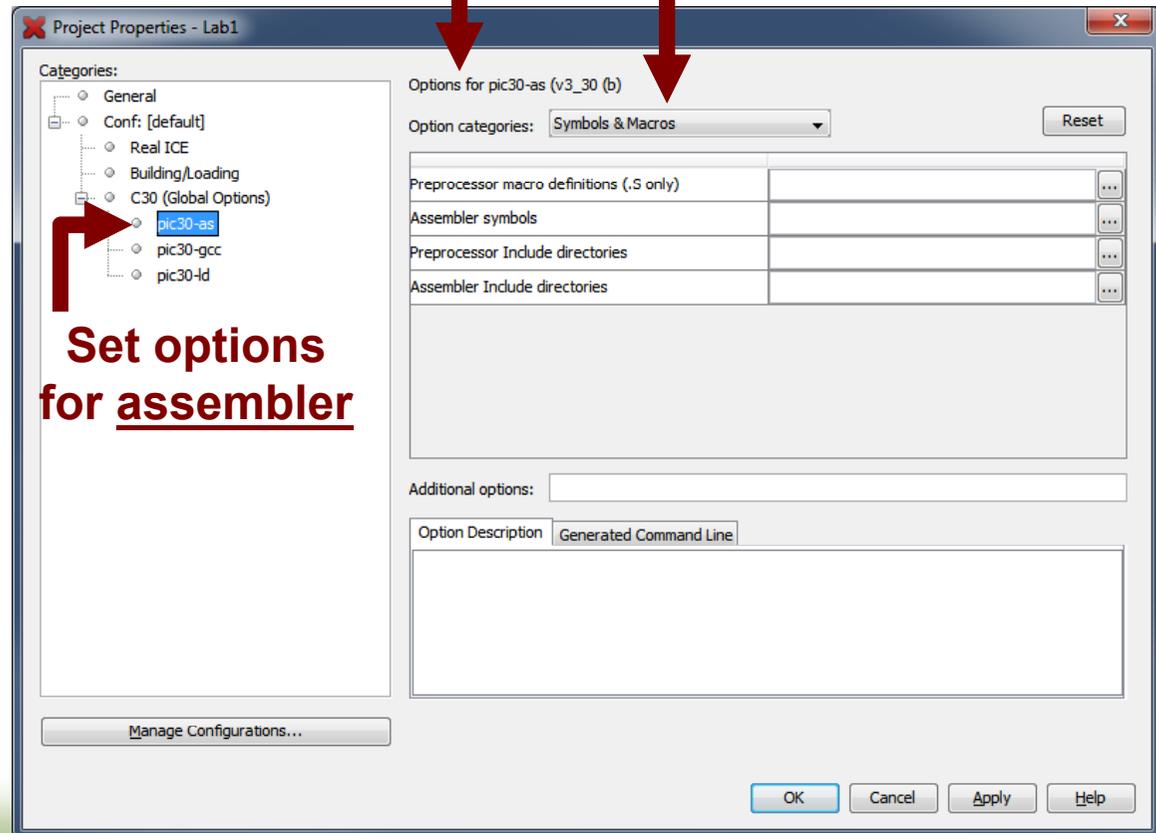
Assembler options including:

- Assembler Includes
- Macro Expansion

**Categories**  
(Same as MPLAB® IDE 8)



**Version**



**Set options  
for assembler**

# How to change project properties

## Compiler Properties

File ► Project Properties  
*Compiler*

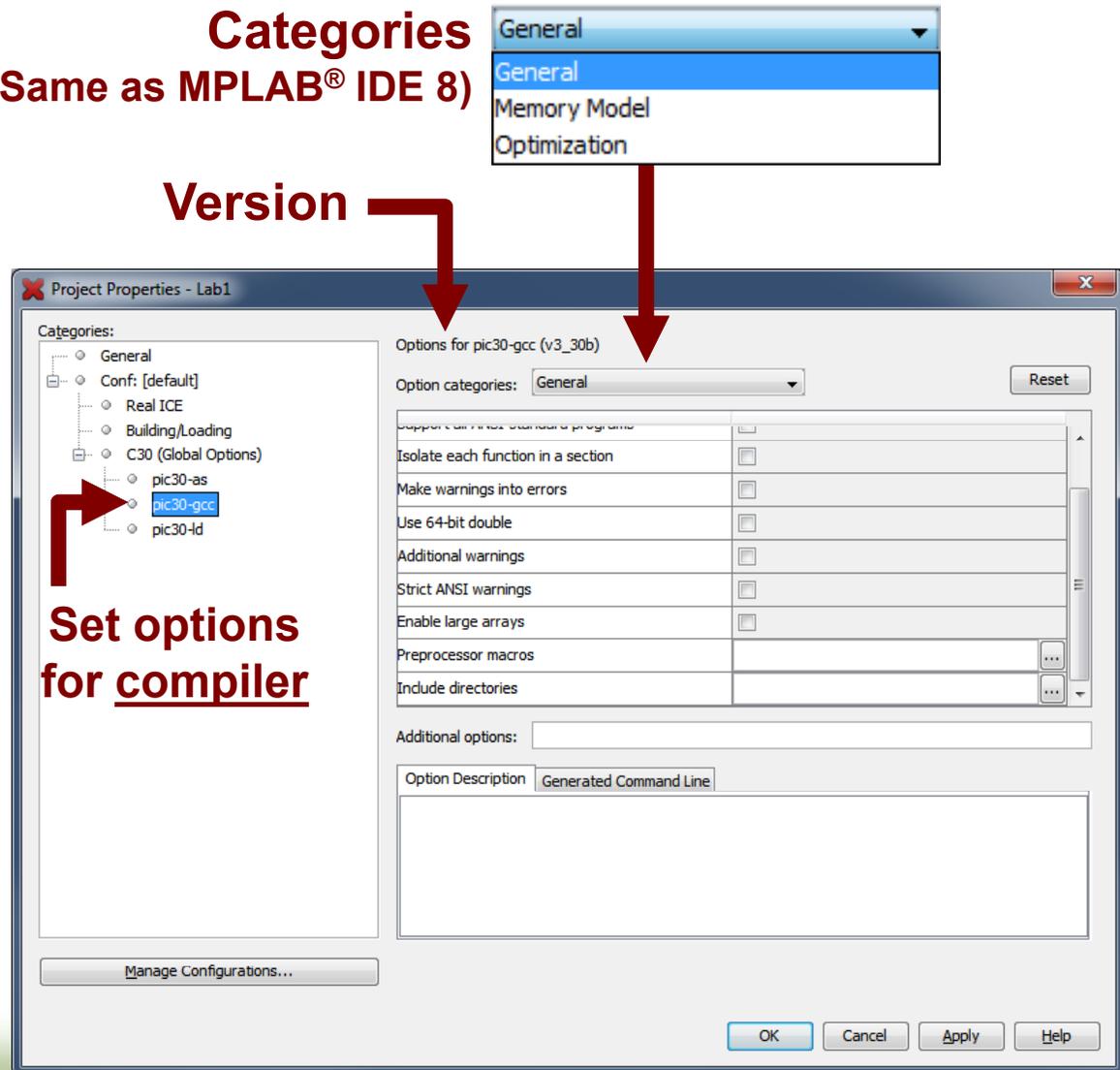
Compiler options including:

- Memory Model
- Optimization
- ANSI Compliance

**Categories**  
(Same as MPLAB® IDE 8)

**Version**

**Set options for compiler**



# How to change project properties

## Linker Properties

File ► Project Properties  
*Linker*

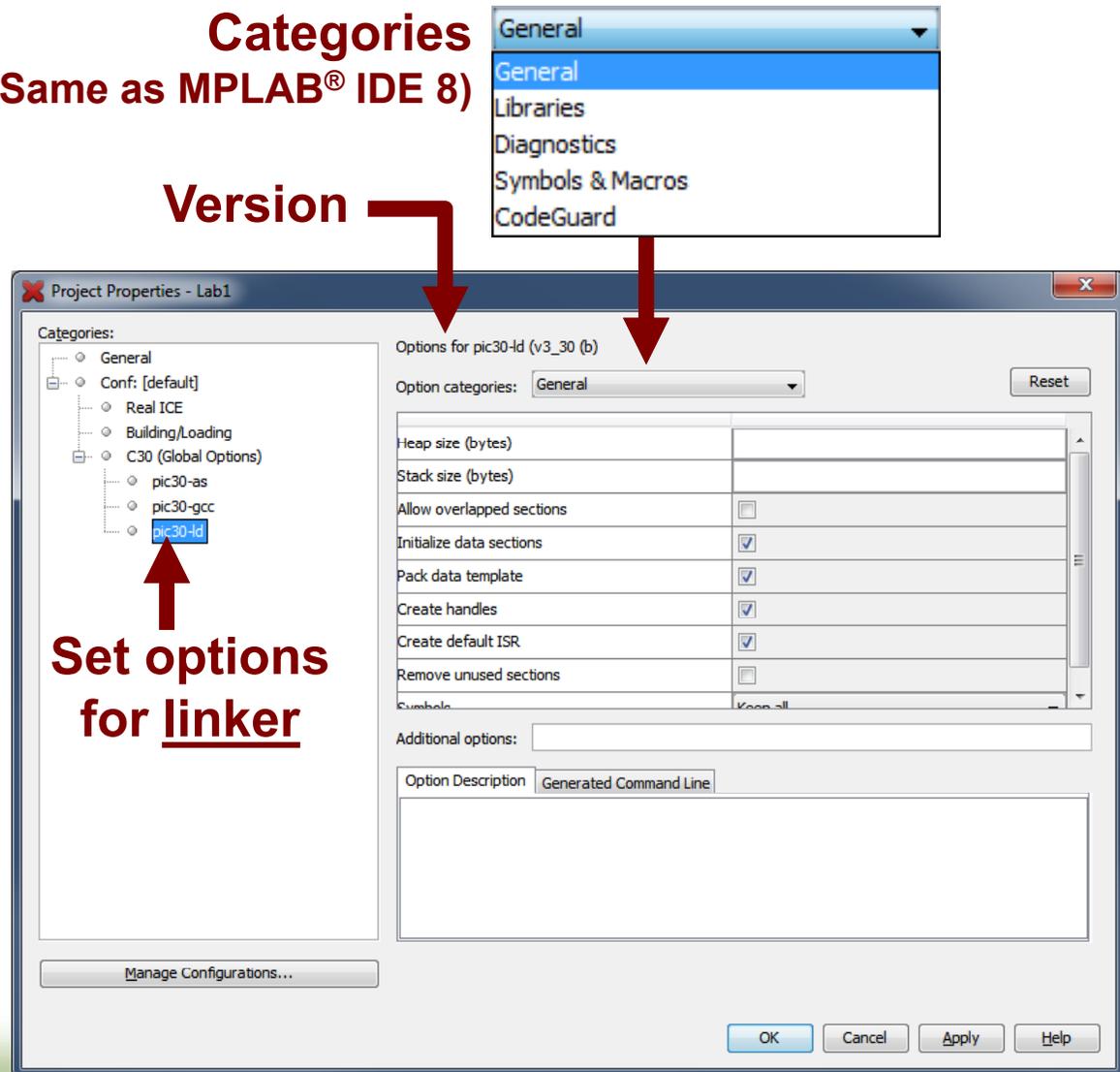
Linker options including:

- Heap Size
- Stack Size
- Library Paths
- Symbols

**Categories**  
(Same as MPLAB® IDE 8)

**Version**

**Set options for linker**



# How to view a program's statistics Dashboard Window

- 1 From the menu select **Window ► Dashboard**
- 2 Select the project in the project tree
- 3 See **Dashboard** tab
  - Device
  - Checksum
  - Memory Usage
  - Available Breakpoints
  - Debug Tool

Navigator Lab1 - Dashboard

- Lab1
  - Device: PIC24FJ128GA010
  - Checksum: Debug Image
  - Compiler Toolchain: C30 (v3\_30b) [C:\Program Files (x86)\Microchip\mplab30\v3.30b\bin]
  - Memory
    - RAM 8192 (0x2000) bytes: 1%
    - RAM Used: 88 (0x58) Free: 8024 (0x1F58)
    - RAM Reserved: 80 bytes
    - Flash 44030 (0xABFE) words: 2%
    - Flash Used: 799 (0x31F) Free: 43231 (0xA8DF)
    - Flash Reserved: None
  - Resources
    - Program BP Used: 0 Free: 4
    - Data BP Used: 0 Free: 4
    - Data Capture BP Used: 0 Free: 4
    - SW BP: Disabled
  - Debug Tool
    - Active Connection: Real ICE: JIT072093517
    - Press Refresh for Tool Status

DC n ov Z C oab sab IP0

```

LCD24.h
k() lcdPutSCmd(LCD_CURSOR_BACK)
t() lcdPutSCmd(LCD_PAN_LEFT)
ht() lcdPutSCmd(LCD_PAN_RIGHT)

types
);
ar cmd);
har cmd); //Used only by macros above
har ch);
ar *str);
ar row, char col);
ng num, char radix);

void lcdPutUInt(unsigned long num, char radix);
void lcdPutFloat(float fnum, char precision);
void lcdClear(void);
  
```

Output Lab1 (Build, Load) x Lab1 (Build, Load, ...) x Debugger Console x REAL ICE x

```

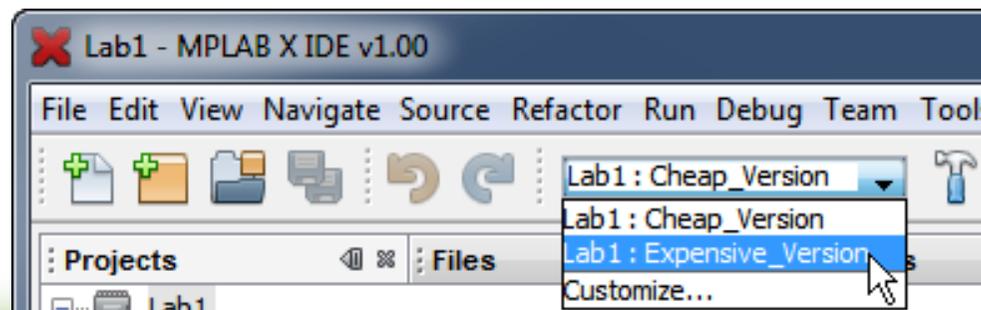
Target detected
Device Revision = 0x3002
Programming...
Programming/Verify complete
Running
Halting...
Target Halted
  
```



opens the  
project properties  
window

# Project Configurations

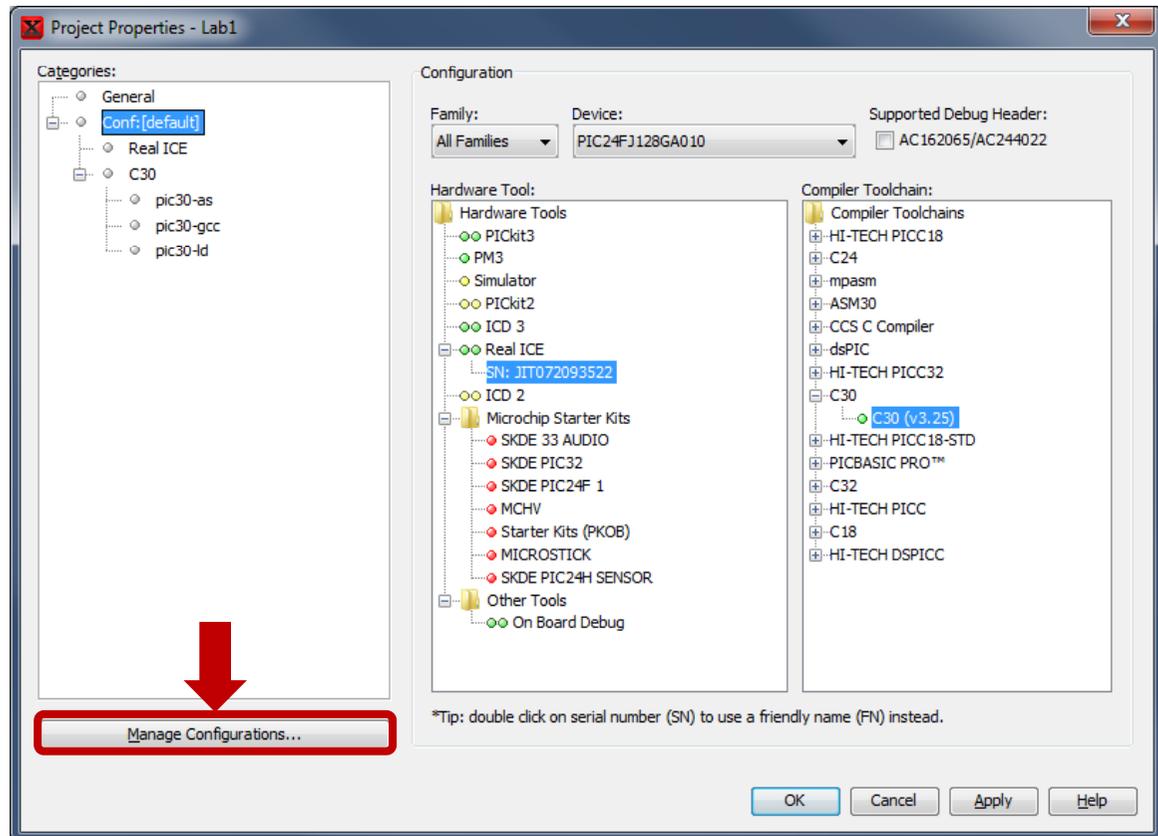
- **A single project can have multiple configurations for same code base:**
  - Different target devices
  - Different debug tools
  - Different compiler settings
  - Anything in project settings window may be different
- **Configurations may be selected quickly from the combo box on the toolbar:**



# How to create a configuration

## 1 Manage Configurations

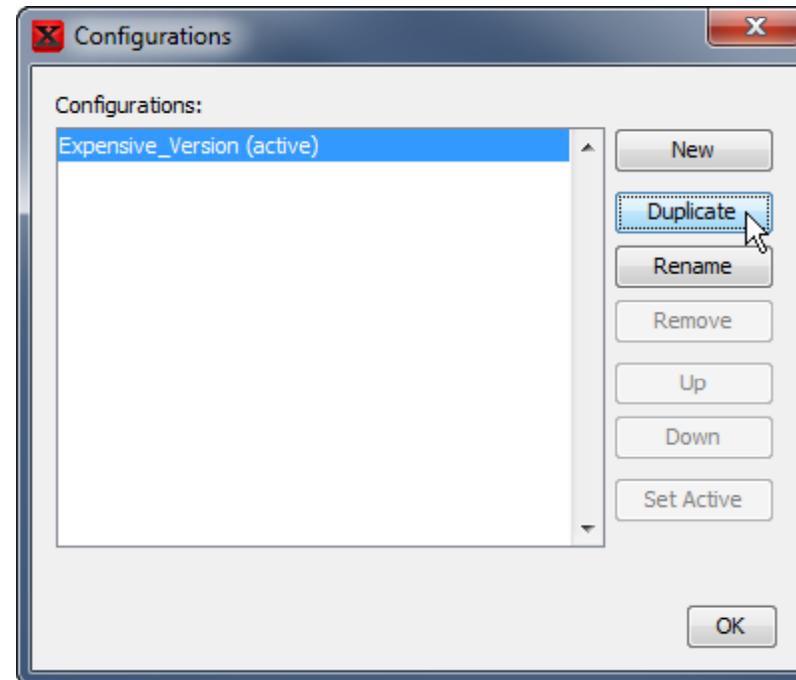
In the **Project Properties** window, click on the **Manage Configurations...** button.



# How to create a configuration

## 2 Create new or duplicate existing configuration

A new configuration will require almost as many steps as creating a new project. Duplicating a configuration will create a copy of the selected configuration from which you can make any desired modifications.



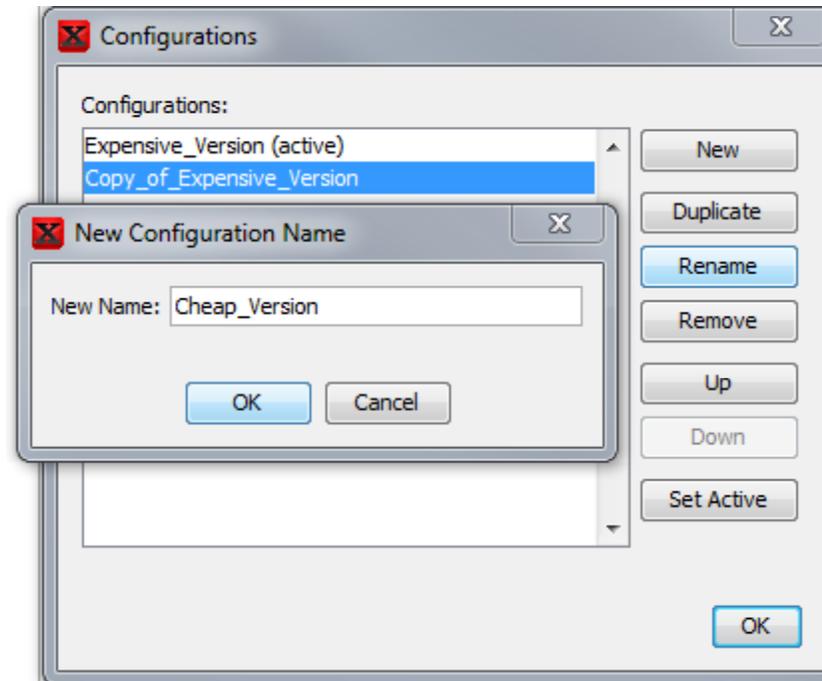
# How to create a configuration

## 3 Give your configurations meaningful names

Select the new configuration and click the **Rename** button.

In the dialog box that opens, enter the new name and click **OK**.

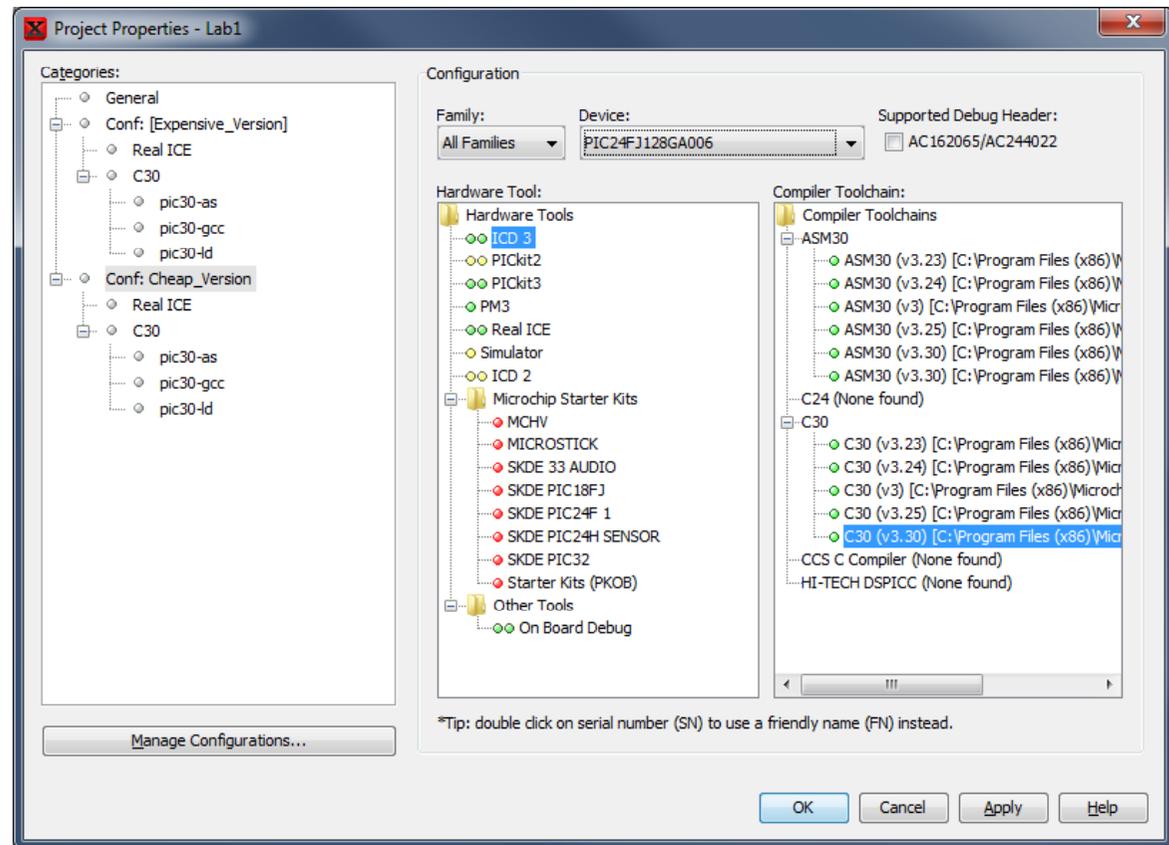
Click OK in the **Configurations** window and you will have a new configuration in the **Project Properties** window.



# How to create a configuration

## 4 Modify the new configuration to suit your needs

Each configuration provides the exact same set of options that may be uniquely configured between them.



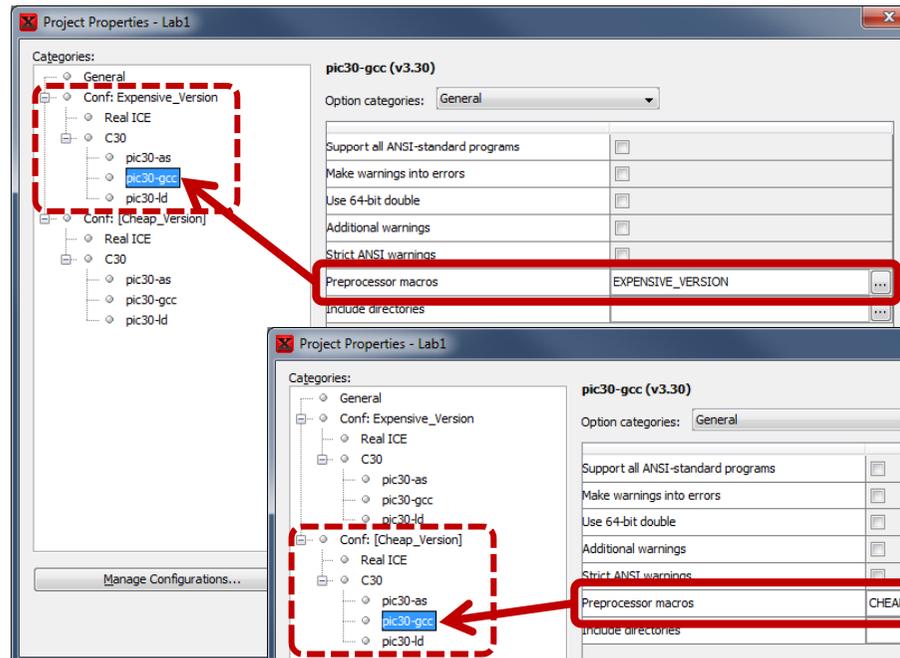
# How to create a configuration

## 5 Create Preprocessor Macros to identify configuration

Preprocessor macros may be used in `#ifdef` directives to identify blocks of code that should be compiled specifically for a particular configuration.

Select the compiler node (the compiler itself, not the suite header) and in the **General** category, add a label to **Preprocessor macros**

This works for assembly too if you add preprocessor macros to the assembler's settings.

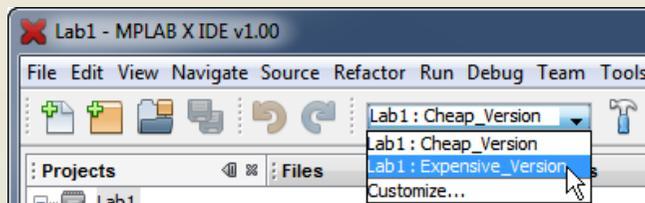


# How to create a configuration

## 6 Use preprocessor macros in code

Use the preprocessor macros you just defined to create configuration specific blocks of code.

Use the combo box in the toolbar to switch between configurations.



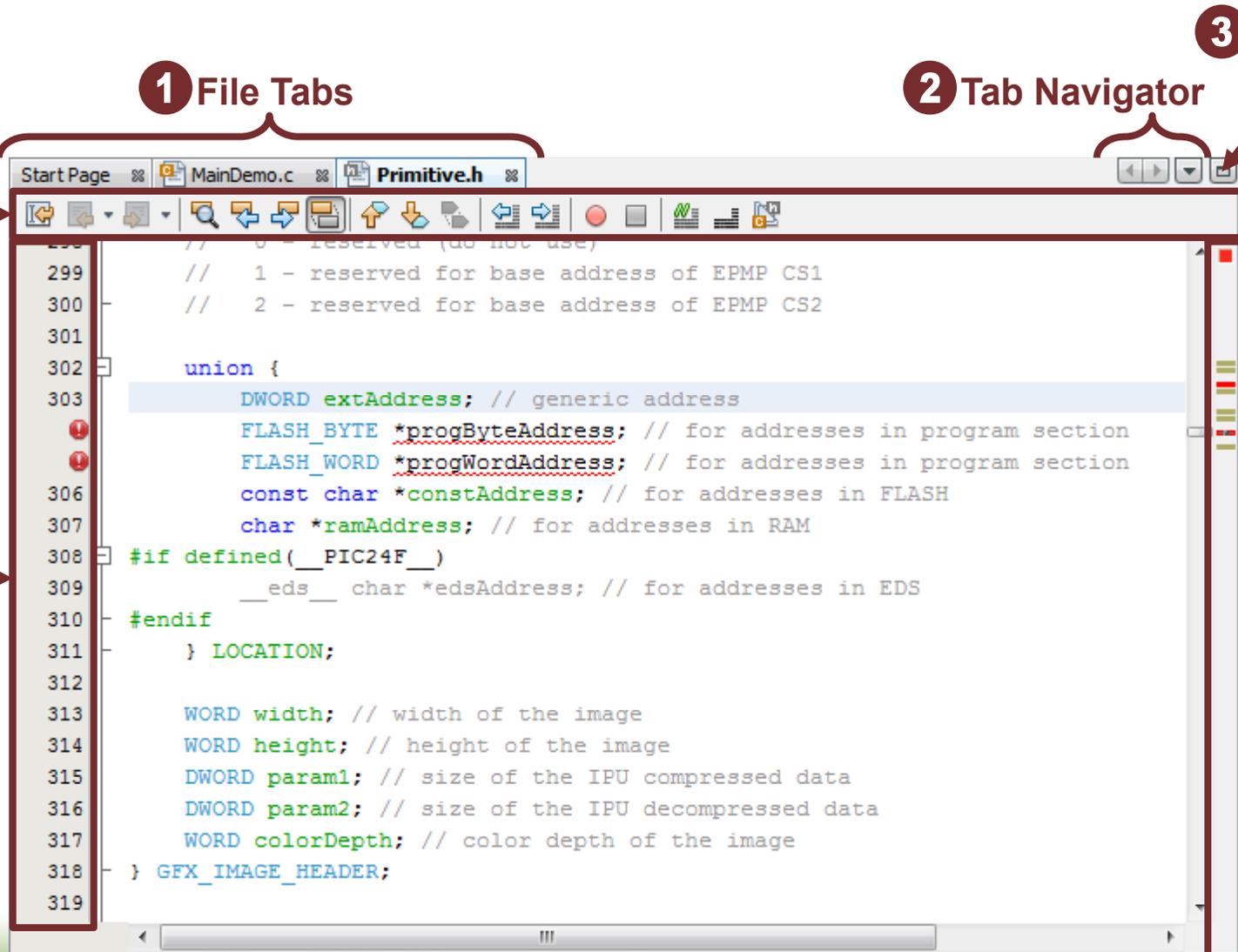
```
#ifdef EXPENSIVE_VERSION
    lcdPutStr( "Expensive" );
#endif
#ifdef CHEAP_VERSION
    lcdPutStr( "Cheap" );
#endif
```



# How to work with the Editor

# How to work with the Editor

## Parts of the Editor



**1** File Tabs

**2** Tab Navigator or Restore Editor

**3** Maximize or Restore Editor

**4** Editor Toolbar

**5** Glyph Margin

**6** Error Stripe

```
299 // 1 - reserved for base address of EPMP CS1
300 // 2 - reserved for base address of EPMP CS2
301
302 union {
303     DWORD extAddress; // generic address
304     FLASH_BYTE *progByteAddress; // for addresses in program section
305     FLASH_WORD *progWordAddress; // for addresses in program section
306     const char *constAddress; // for addresses in FLASH
307     char *ramAddress; // for addresses in RAM
308 #if defined(__PIC24F__)
309     __eds__ char *edsAddress; // for addresses in EDS
310 #endif
311     } LOCATION;
312
313     WORD width; // width of the image
314     WORD height; // height of the image
315     DWORD param1; // size of the IPU compressed data
316     DWORD param2; // size of the IPU decompressed data
317     WORD colorDepth; // color depth of the image
318 } GFX_IMAGE_HEADER;
319
```

# How to work with the Editor

## Features: Editor Toolbar



- |  |   |
|--|---|
|  <b>Last Edit</b> – Cycle through edits |  <b>Shift Line Left (Indent Less)</b>  |
|  <b>Previous Edit</b>                   |  <b>Shift Line Right (Indent More)</b> |
|  <b>Next Edit</b>                       |  <b>Start Macro Recording</b>          |
|  <b>Find Selection</b>                  |  <b>Stop Macro Recording</b>           |
|  <b>Find Previous Occurrence</b>        |  <b>Comment</b>                        |
|  <b>Find Next Occurrence</b>            |  <b>Uncomment</b>                      |
|  <b>Toggle Highlight Search</b>       |  <b>Go to Header/Source</b>          |
|  <b>Previous Bookmark</b>             |   |
|  <b>Next Bookmark</b>                 |   |
|  <b>Toggle Bookmark</b>               |   |

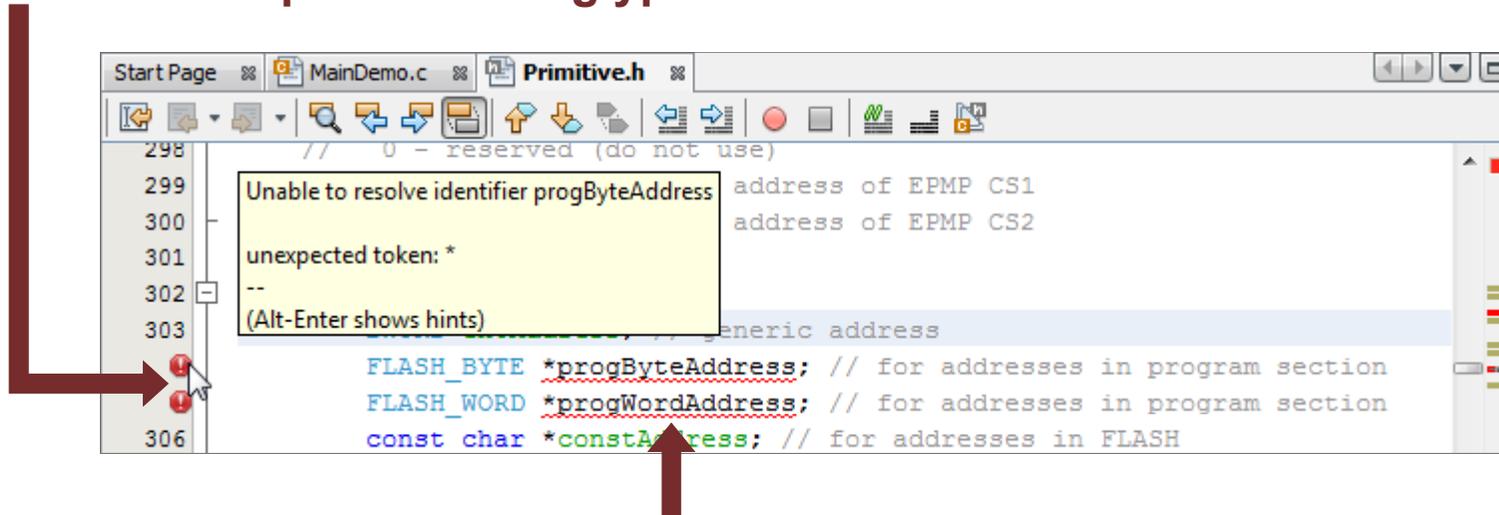
\* Next and Previous Edit buttons become active after using **Navigate ▶ Go to X...**

# How to work with the Editor

## Features: Live Parsing

- **Editor parses code as you type and flags errors immediately**

Hover mouse pointer over glyph for information about error

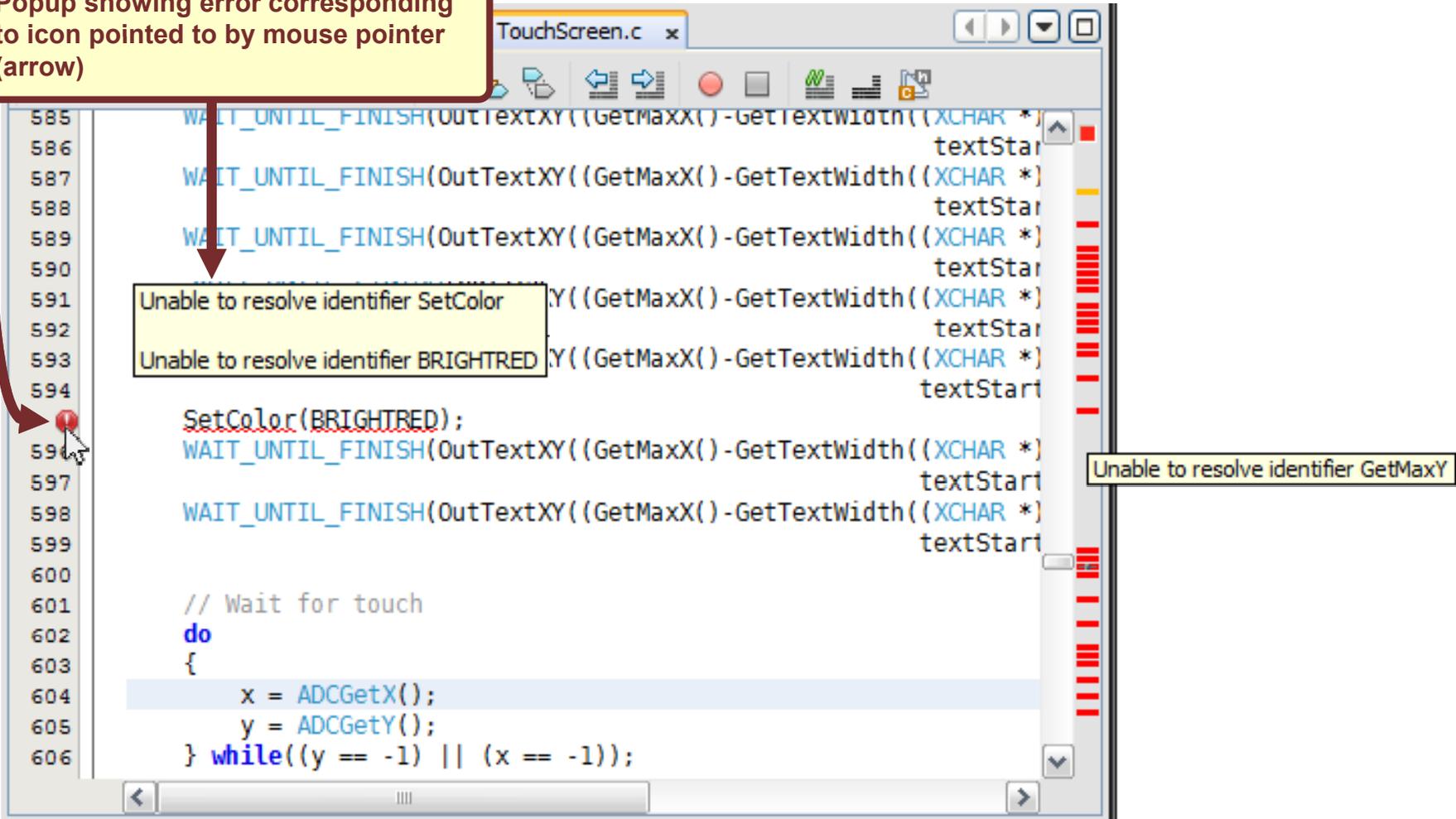


Errors are underlined in red (like misspelled words in word processor)

# How to work with the Editor

## Features: Glyph Margin

Popup showing error corresponding to icon pointed to by mouse pointer (arrow)

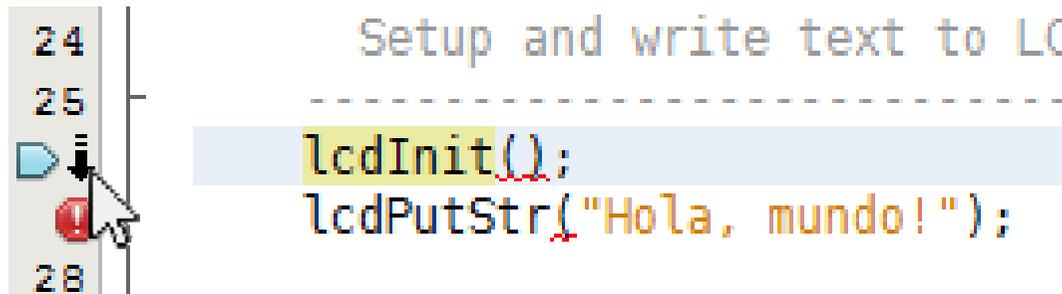
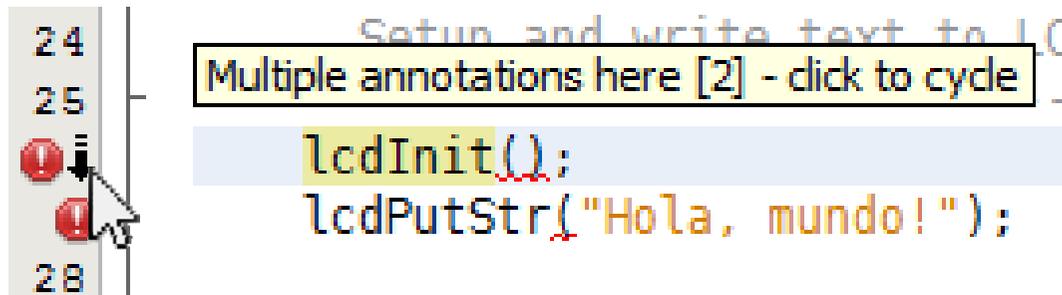


```
585 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX() - GetTextWidth((XCHAR *)
586 textStar
587 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX() - GetTextWidth((XCHAR *)
588 textStar
589 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX() - GetTextWidth((XCHAR *)
590 textStar
591 Unable to resolve identifier SetColor Y((GetMaxX() - GetTextWidth((XCHAR *)
592 textStar
593 Unable to resolve identifier BRIGHTRED Y((GetMaxX() - GetTextWidth((XCHAR *)
594 textStar
595 SetColor(BRIGHTRED);
596 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX() - GetTextWidth((XCHAR *)
597 textStar Unable to resolve identifier GetMaxY
598 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX() - GetTextWidth((XCHAR *)
599 textStar
600
601 // Wait for touch
602 do
603 {
604 x = ADCGetX();
605 y = ADCGetY();
606 } while((y == -1) || (x == -1));
```

# How to work with the Editor

## Features: Glyph Margin

### Error and Bookmark on the same line



**When multiple glyphs are required on a line, a small black arrow pointing down appears to the right of one glyph.**

**Clicking on the arrow will cycle through the glyphs.**

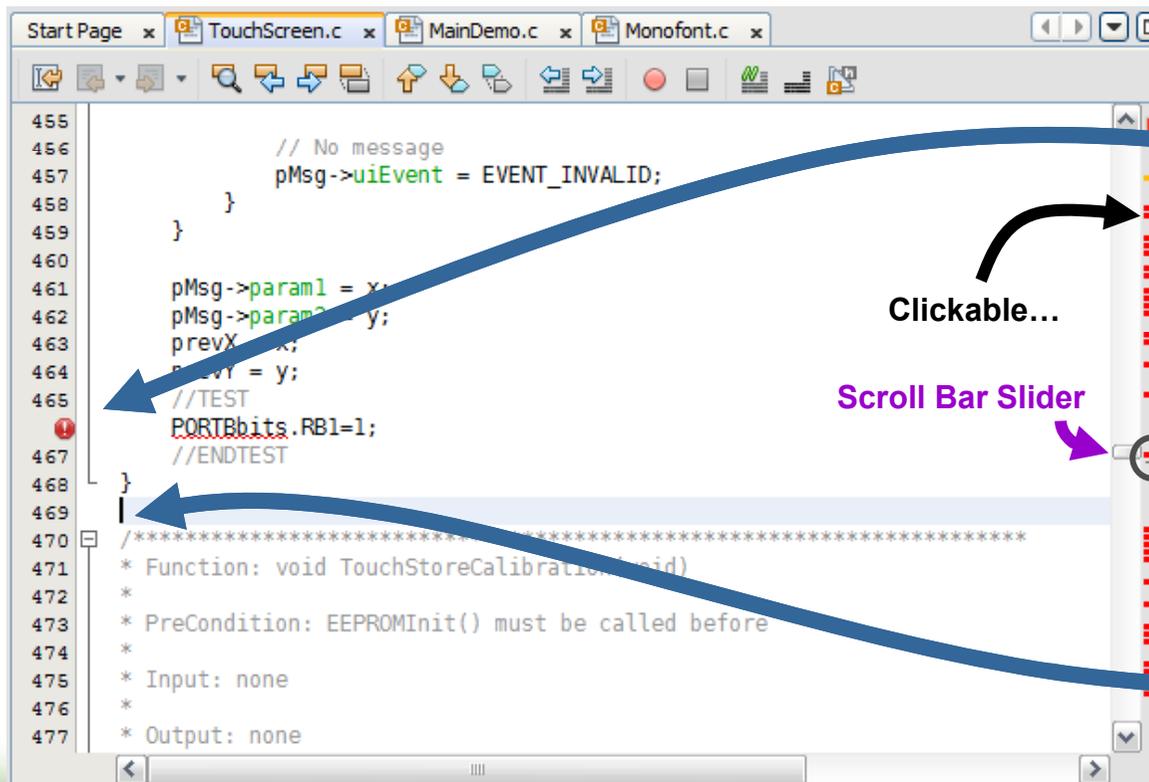
# How to work with the Editor

## Navigation: Error Stripe

- Represents full length of file
- Shows relative position of errors, warnings and occurrences of currently highlighted item

### Color Code

		Warnings
		Errors
		Occurrences
		Breakpoint
		Bookmark
		Suggestion
		Multiple Items



```

455
456 // No message
457 pMsg->uiEvent = EVENT_INVALID;
458 }
459 }
460
461 pMsg->param1 = x;
462 pMsg->param2 = y;
463 prevX = x;
464 prevY = y;
465 //TEST
466 PORTBbits.RB1=1;
467 //ENDTEST
468 }
469
470 /*****
471 * Function: void TouchStoreCalibration(void)
472 *
473 * PreCondition: EEPROMInit() must be called before
474 *
475 * Input: none
476 *
477 * Output: none

```

Clickable...

Scroll Bar Slider

Error Stripe

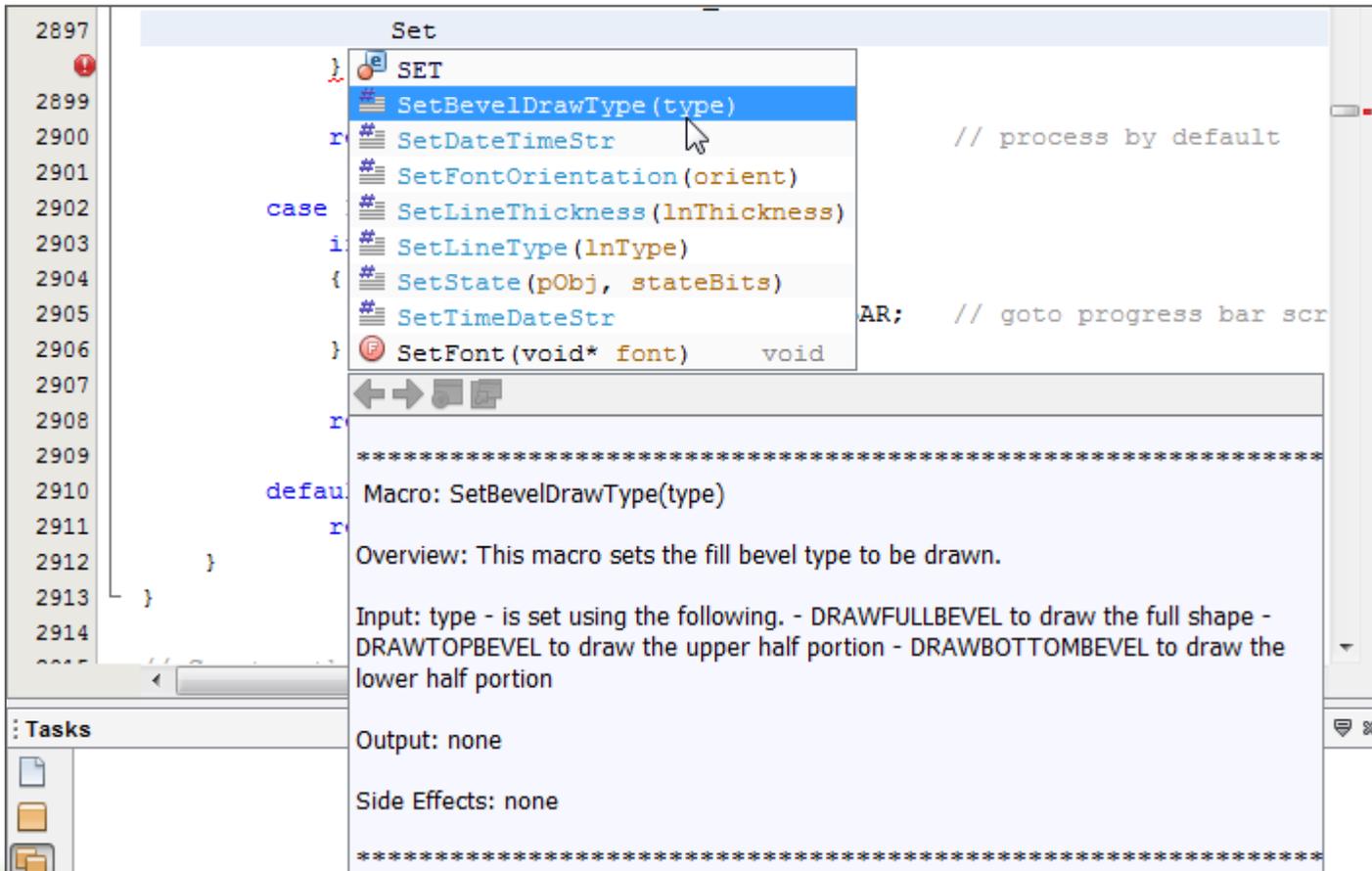
Error

Cursor Position

# How to work with the Editor

## Features: Smart Code Completion

- Hit **Ctrl** + **Alt** + *Space* after part of an identifier



```
2897 Set
2898 }
2899 SET
2900 # SetBevelDrawType (type) // process by default
2901 # SetDateTimeStr
2902 # SetFontOrientation (orient)
2903 case # SetLineThickness (lnThickness)
2904 i # SetLineType (lnType)
2905 { # SetState (pObj, stateBits)
2906 # SetTimeDateStr AR; // goto progress bar scr
2907 } # SetFont (void* font) void
2908 r
2909
2910 default Macro: SetBevelDrawType(type)
2911 r
2912 Overview: This macro sets the fill bevel type to be drawn.
2913 Input: type - is set using the following. - DRAWFULLBEVEL to draw the full shape -
2914 DRAWTOPBEVEL to draw the upper half portion - DRAWBOTTOMBEVEL to draw the
2915 lower half portion
2916
2917 Output: none
2918
2919 Side Effects: none
2920
2921 *****
```

# How to work with the Editor

## Features: Documentation Pop-ups

- Editor shows function parameter list in a tool tip as you type (  +  to display)

- 2 Tool tip should pop-up automatically, but may be manually displayed by hitting *Ctrl+P*

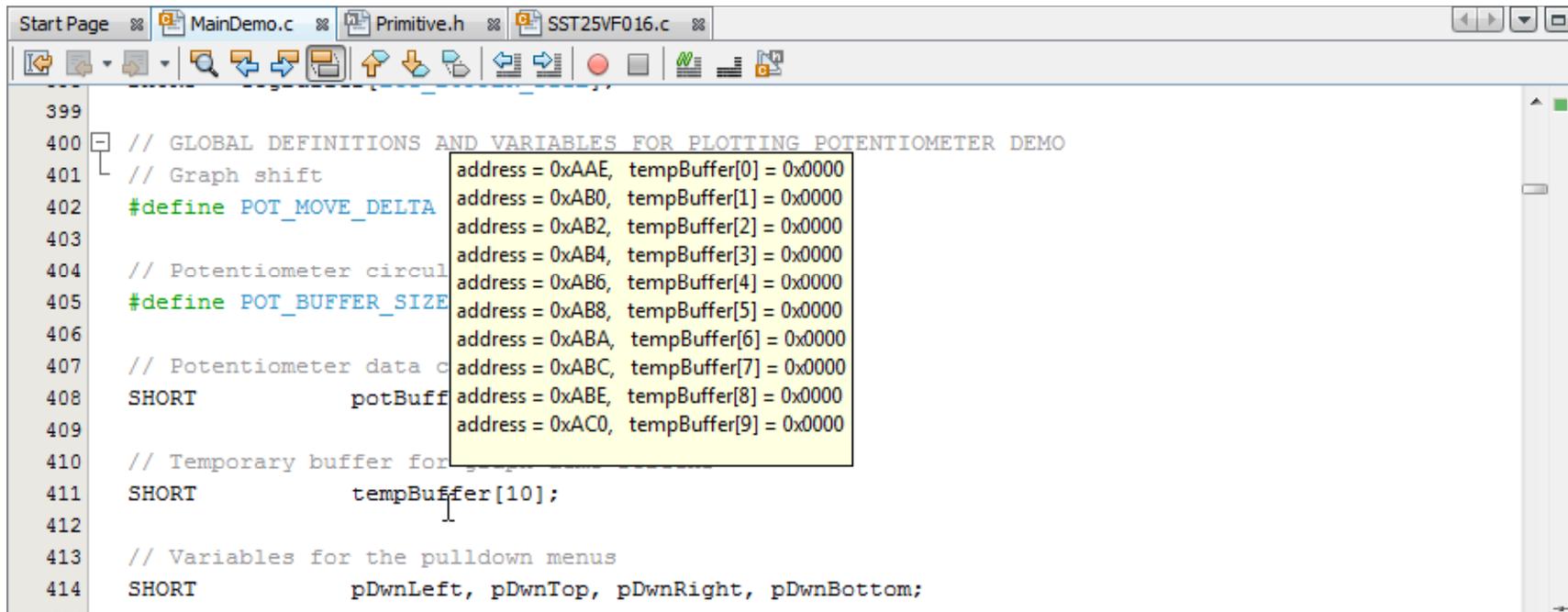
```
2895     {  
2896         screenSetFont( void* font, CREATE_EDITBOX;           // goto edit box screen  
2897         SetFont ( )
```

- 1 Type function name with open parenthesis '('

# How to work with the Editor

## Features: View a variable's contents

- Hover mouse over a register or variable...
- Debug session must be active



The screenshot shows an IDE window with several tabs: Start Page, MainDemo.c, Primitive.h, and SST25VF016.c. The code editor displays the following code:

```
399
400 // GLOBAL DEFINITIONS AND VARIABLES FOR PLOTTING POTENTIOMETER DEMO
401 // Graph shift
402 #define POT_MOVE_DELTA
403
404 // Potentiometer circul
405 #define POT_BUFFER_SIZE
406
407 // Potentiometer data c
408 SHORT          potBuff
409
410 // Temporary buffer for
411 SHORT          tempBuffer[10];
412
413 // Variables for the pulldown menus
414 SHORT          pDwnLeft, pDwnTop, pDwnRight, pDwnBottom;
```

A tooltip is displayed over the variable `tempBuffer`, showing its contents:

address = 0xAAE,	tempBuffer[0] = 0x0000
address = 0xAB0,	tempBuffer[1] = 0x0000
address = 0xAB2,	tempBuffer[2] = 0x0000
address = 0xAB4,	tempBuffer[3] = 0x0000
address = 0xAB6,	tempBuffer[4] = 0x0000
address = 0xAB8,	tempBuffer[5] = 0x0000
address = 0xABA,	tempBuffer[6] = 0x0000
address = 0xABC,	tempBuffer[7] = 0x0000
address = 0xABE,	tempBuffer[8] = 0x0000
address = 0xAC0,	tempBuffer[9] = 0x0000

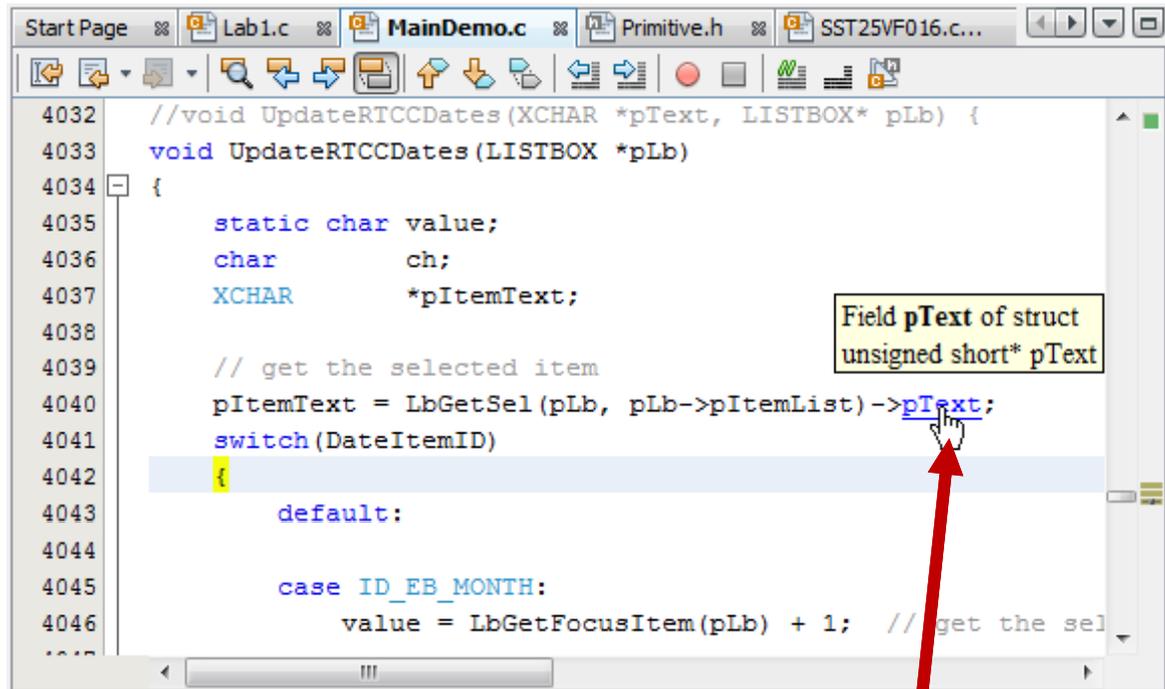
# How to work with the Editor

## Navigation: Hyperlinked Identifiers

### To show the link:

Hover mouse pointer over an identifier and click the **Ctrl** key

Click the [link](#) to jump to the identifier's declaration in the editor



```
4032 //void UpdateRTCCDates(XCHAR *pText, LISTBOX* pLb) {
4033 void UpdateRTCCDates (LISTBOX *pLb)
4034 {
4035     static char value;
4036     char      ch;
4037     XCHAR     *pItemText;
4038
4039     // get the selected item
4040     pItemText = LbGetSel(pLb, pLb->pItemList)->pText;
4041     switch(DateItemID)
4042     {
4043     default:
4044
4045     case ID_EB_MONTH:
4046         value = LbGetFocusItem(pLb) + 1; // get the sel
4047 }
```

Field pText of struct unsigned short\* pText

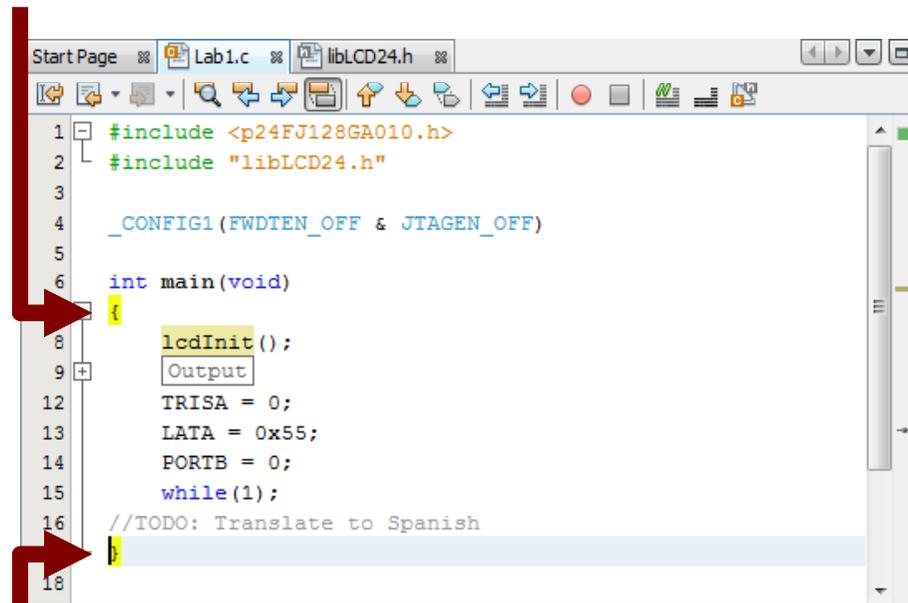


# How to work with the Editor

## Navigation: Automatic Brace Matching

Select any brace '{' or '}' and it will automatically be highlighted along with its match

### Matching Brace



```
1 #include <p24FJ128GA010.h>
2 #include "libLCD24.h"
3
4 _CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
5
6 int main(void)
7 {
8     lcdInit();
9     Output
10
11
12     TRISA = 0;
13     LATA = 0x55;
14     PORTB = 0;
15     while(1);
16     //TODO: Translate to Spanish
17 }
18
```

### Selected Brace

# How to work with the Editor

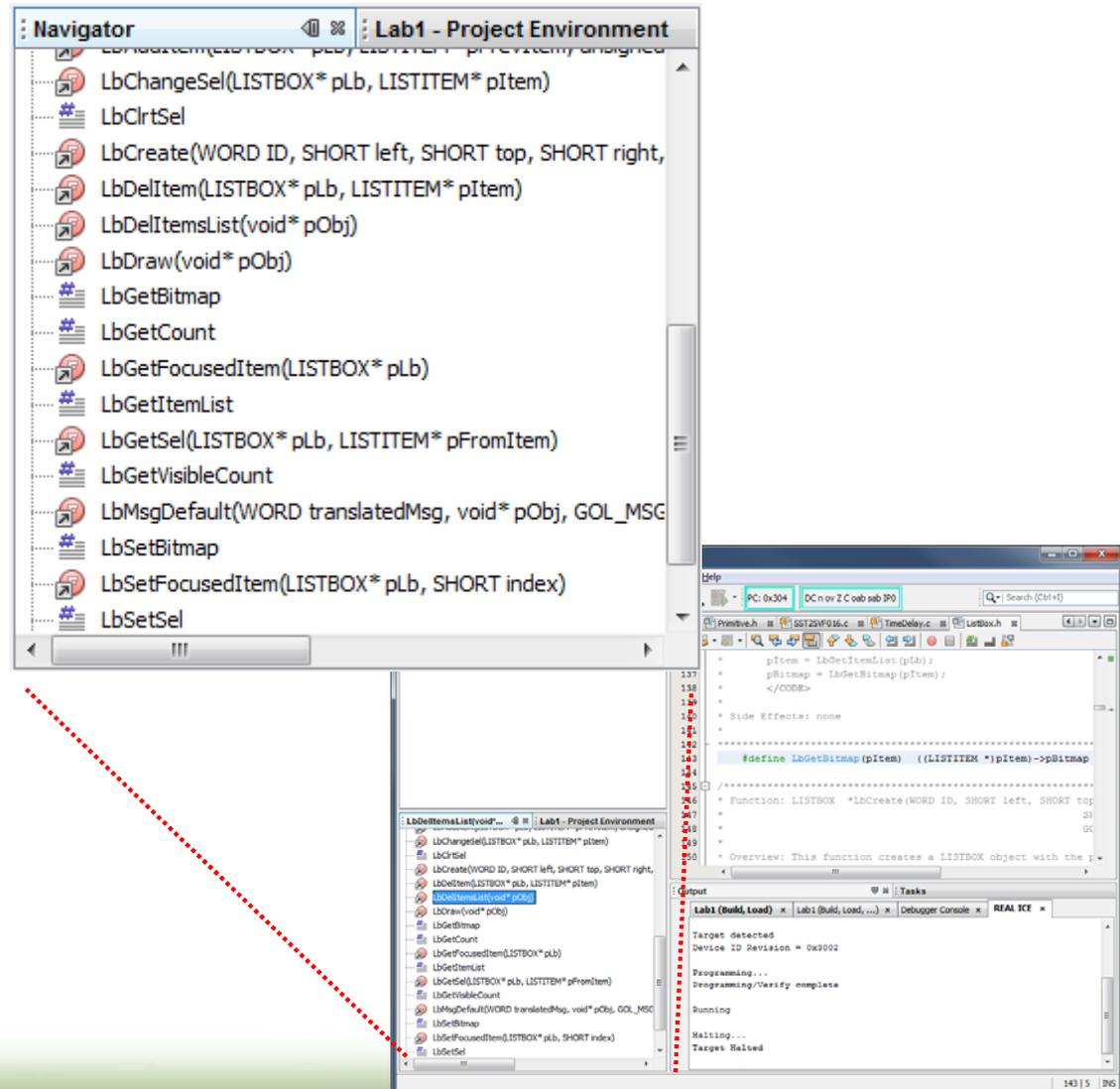
## Navigation: Go To...

- **Go To Declaration**
- **Go To Source**
- **Go To Type**
- **Go To File**
- **Go To Symbol**
- **Go To Line**

# How to work with the Editor

## Navigation: The Navigator Window

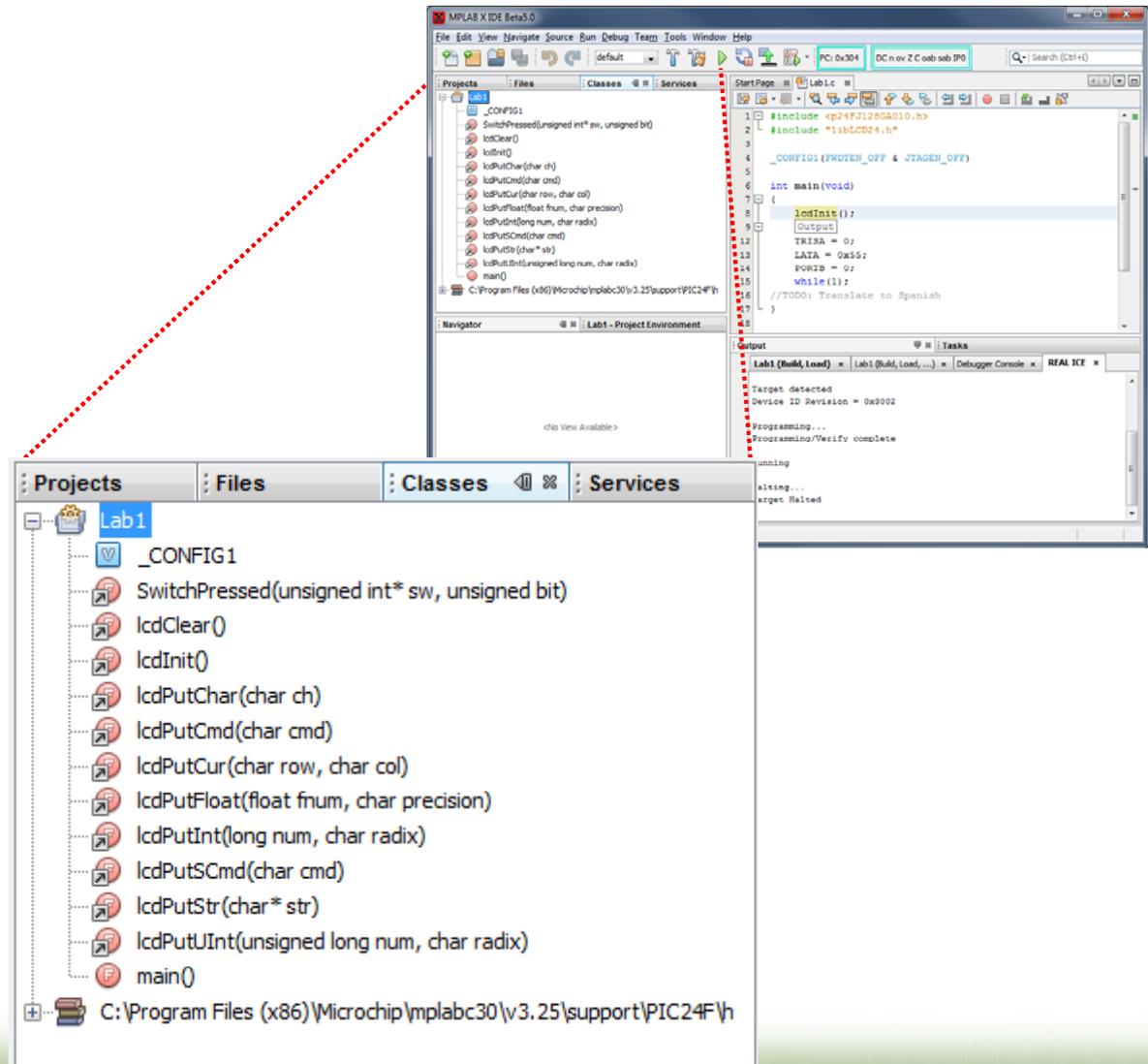
- Lists identifiers and symbols for current file
- Double click on an item to jump to its declaration
- Right click for additional options



# How to work with the Editor

## Navigation: The Classes Window

- Lists identifiers for the project
- Double click to jump to declaration
- Right click for additional options



# How to work with the Editor

## Features: Custom code folds

- To create code folds without { }:

### Syntax

```
//<editor-fold defaultstate="collapsed" desc="Descriptive Text">  
Your Code Here  
//</editor-fold>
```

### Unfolded / Expanded

```
6 int main(void)  
7 {  
8 // <editor-fold defaultstate=  
9 lcdInit();  
10 // </editor-fold>  
11 lcdPutStr("Hello, world!");  
12 TRISA = 0;  
13 LATA = 0x55;  
14 while(1);
```

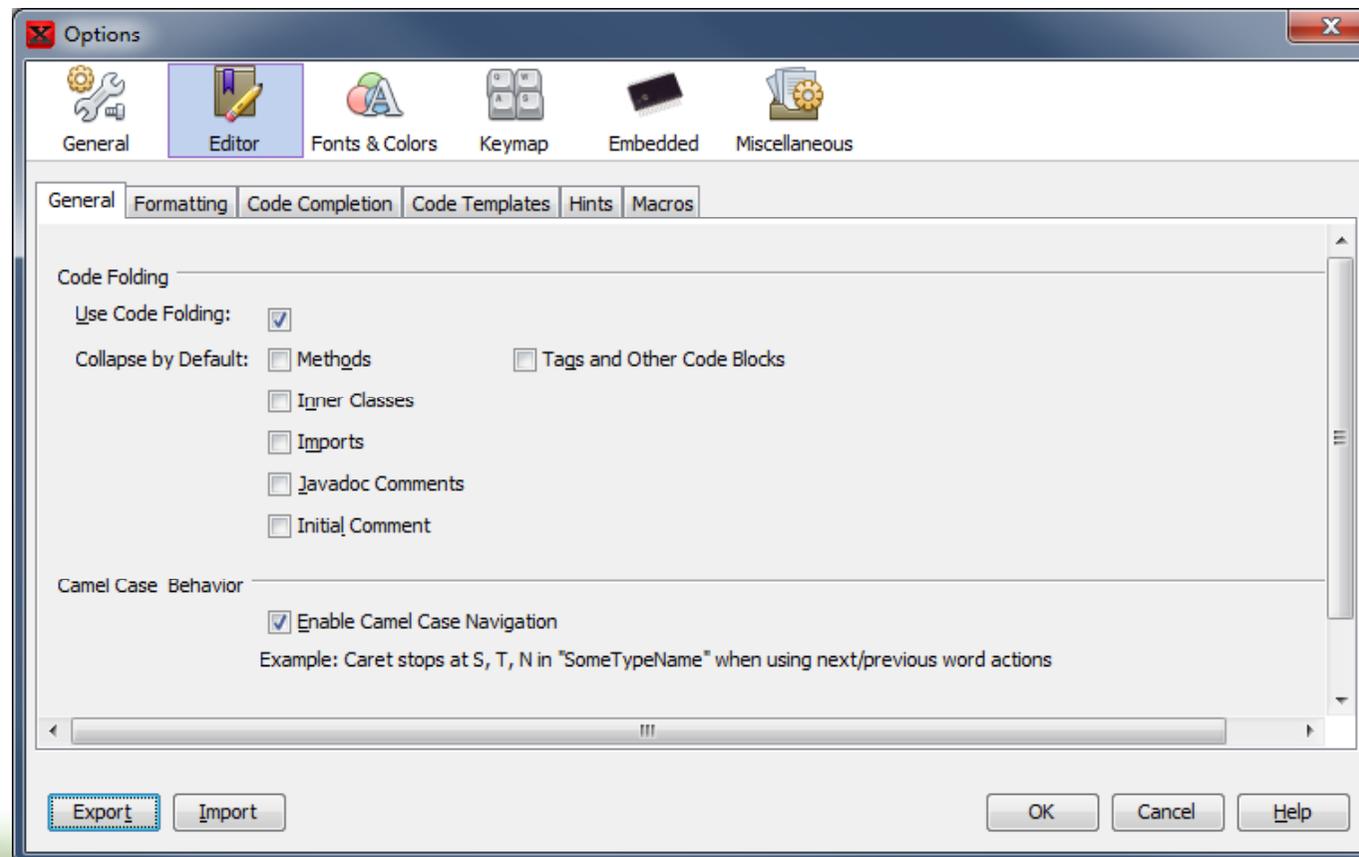
### Folded / Collapsed

```
6 int main(void)  
7 {  
8 Initialization Code  
11 lcdPutStr("Hello, world!");  
12 TRISA = 0;  
13 LATA = 0x55;  
14 while(1);  
15 }  
16
```

# How to work with the Editor

## How to configure the editor's settings

- Select **Tools** ▶ **Options** from the menu, then select  **Editor** from the icons on top

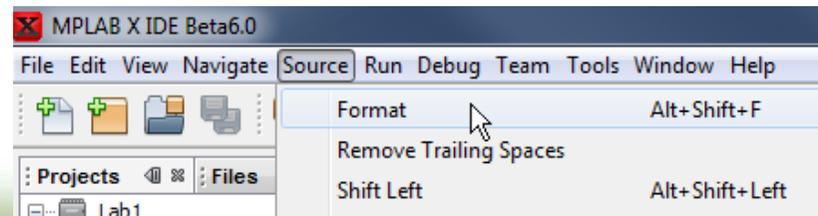
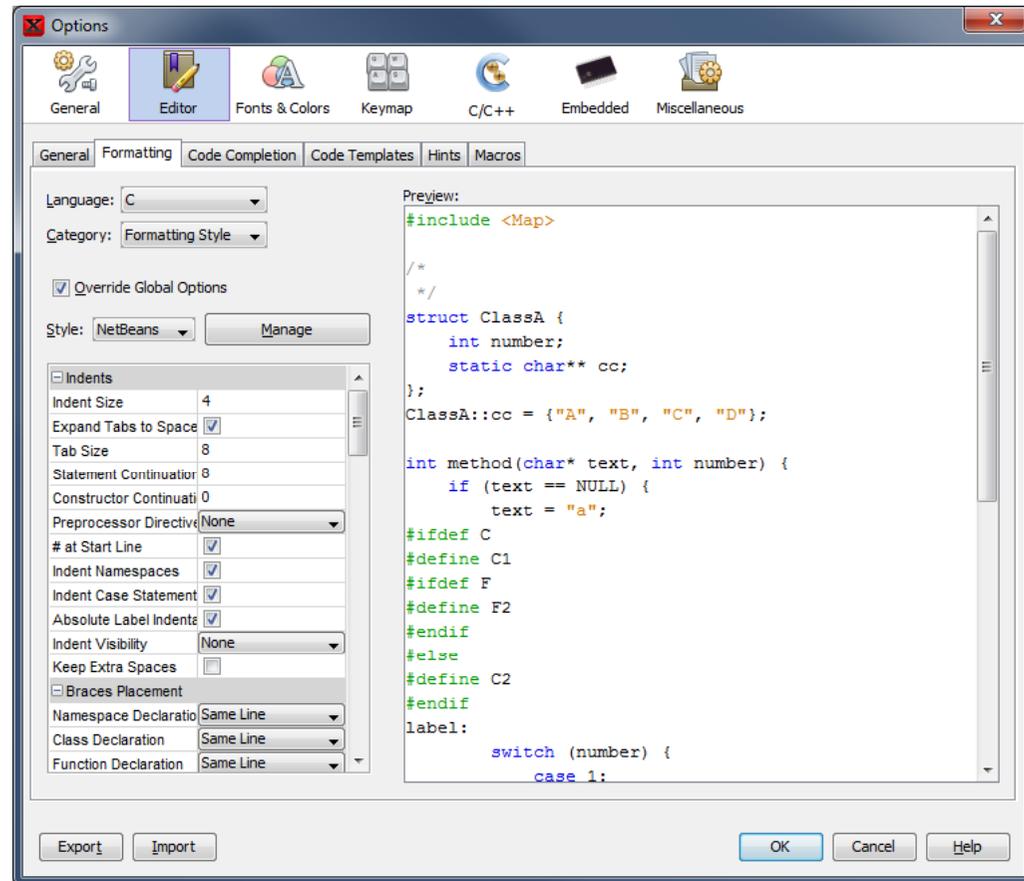


# How to work with the Editor

## Automatic Code Formatting

Rules for C code formatting may be defined in **Tools ▶ Options ▶ Editor ▶ Formatting**. Choose **C** from the Language combo box.

To apply the rules to the file currently selected in the editor, select from the main menu **Source ▶ Format**

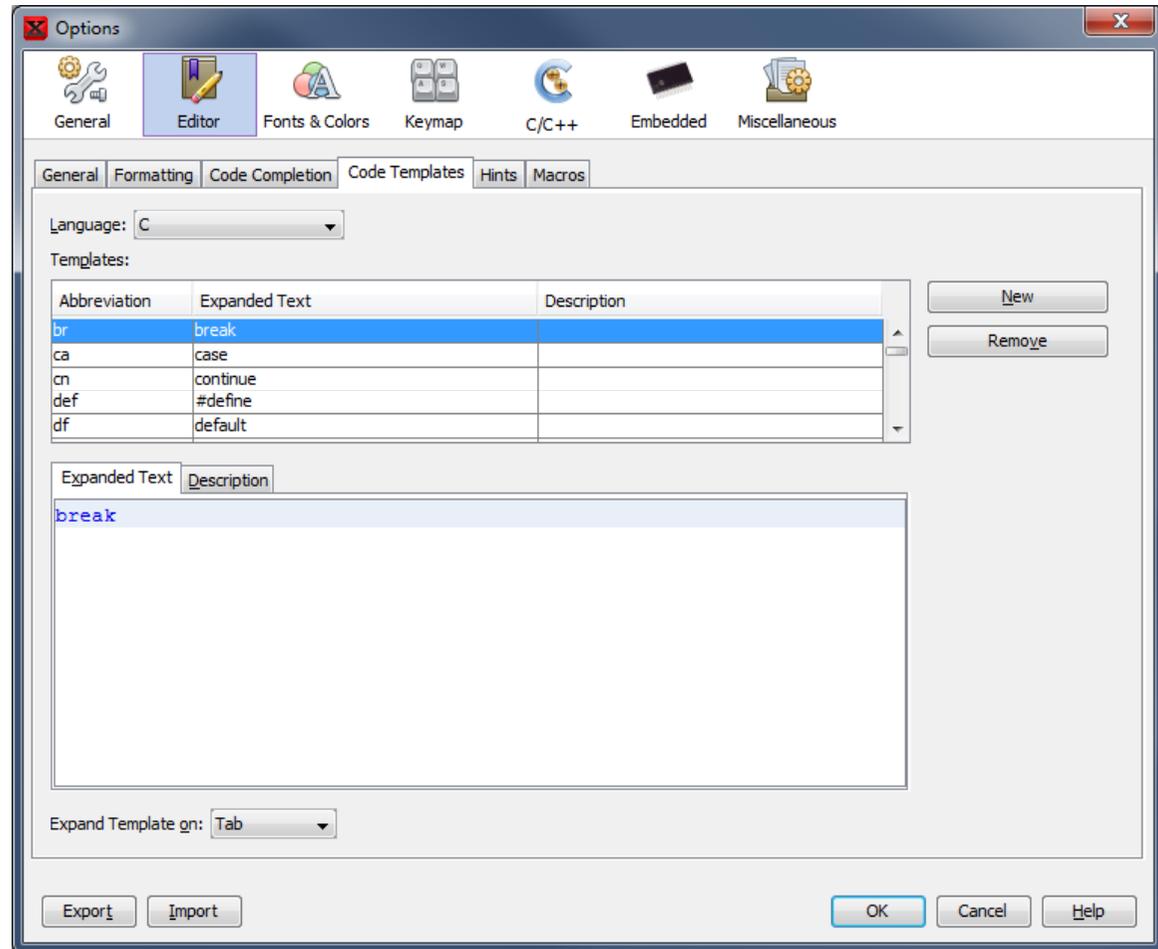


# How to work with the Editor

## Code Templates

Code templates provide a shorthand notation for code snippets that will be expanded by the editor.

Code templates can take parameters that make it easy to enter values in code without having to move the cursor.



# How to work with the Editor

## Code Templates: Example

### while Template Definition

```
while (${EXP default="exp"}) {  
    ${selection line}${cursor}  
}
```

You type into editor and press **Tab** key:

```
wh
```

Editor expands this into:

```
while (exp) {  
  
}
```

Type in a value for `exp` and press **Enter** key. The cursor will be automatically placed inside the braces.

# How to work with the Editor

## Tasks List

### 1 Create ToDo patterns for comment tagging

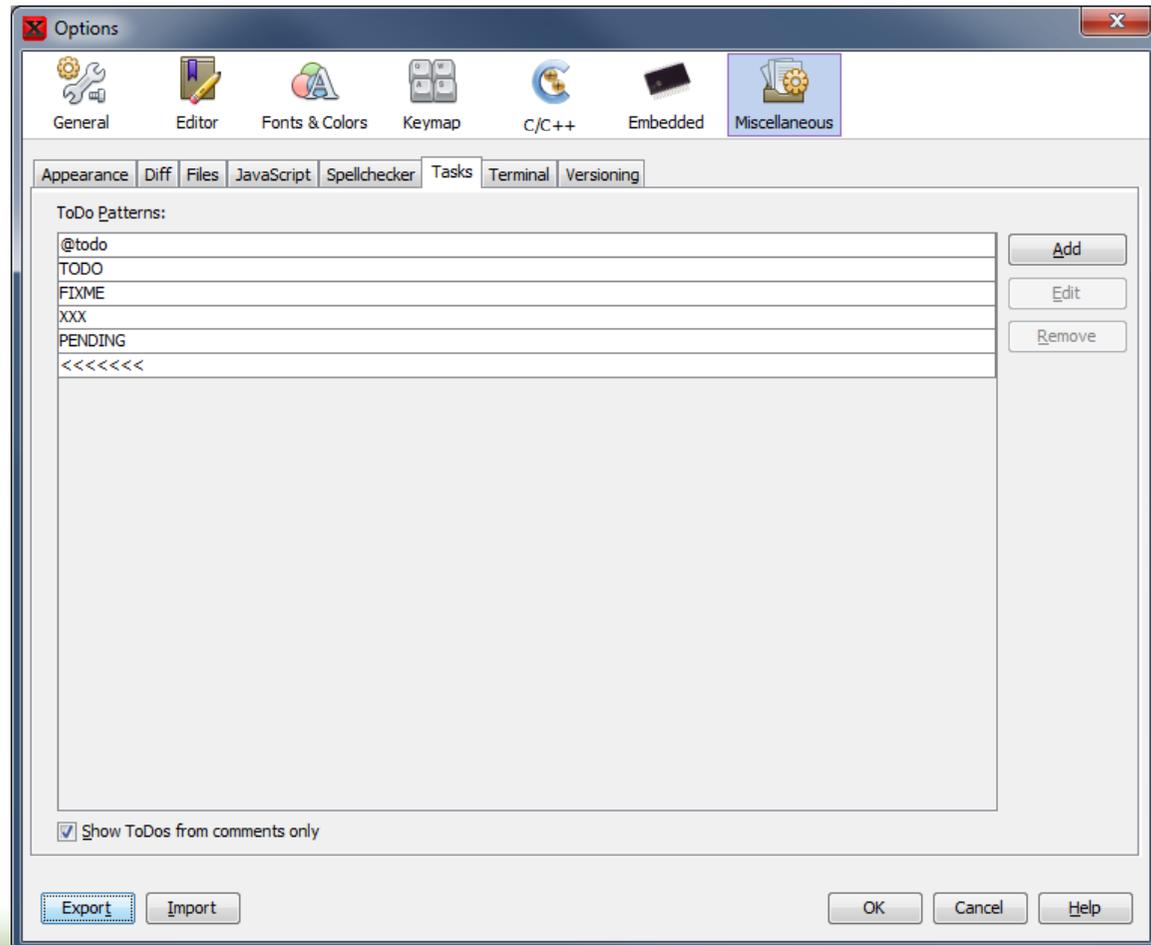
From the main menu select:  
**Tools ▶ Options**

Select **Miscellaneous**

Select **Tasks**

Add a custom ToDo Pattern,  
or just use the default list

Click OK when done



# How to work with the Editor

## Tasks List

### 2 Create comments using a ToDo pattern

Start a comment with one of the ToDo patterns from step one. Any text may follow the ToDo pattern.

```
#ifdef EXPENSIVE_VERSION
    lcdPutStr( "Expensive" );
#endif
#ifdef CHEAP_VERSION
    lcdPutStr( "Cheap" );
#endif
```

```
//TODO Translate to Spanish
```

```
lcdPutCur(1, 0);
lcdPutInt(count, DEC);
LATLED = 0x0010;
TRISLED = 0xFF00;
```

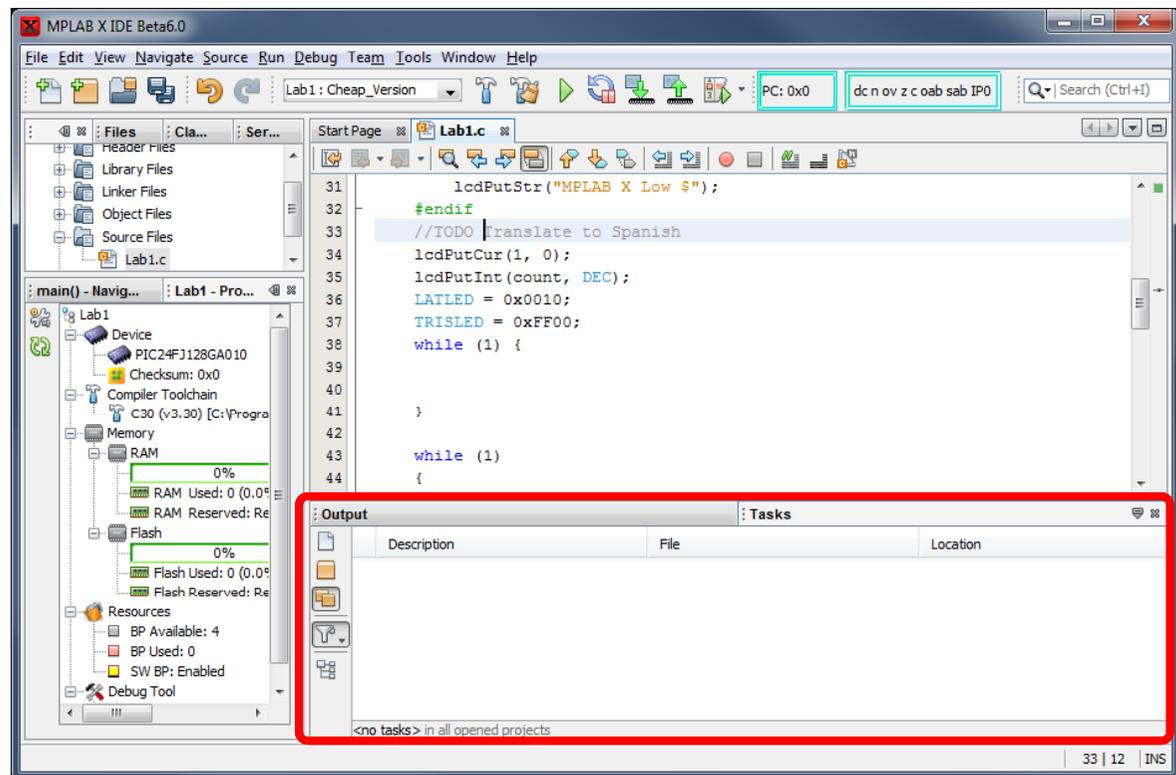
# How to work with the Editor

## Tasks List

### 3 Open the Tasks window

From the main menu select:  
**Window ▶ Tasks**

The Tasks window will open up in the Output area



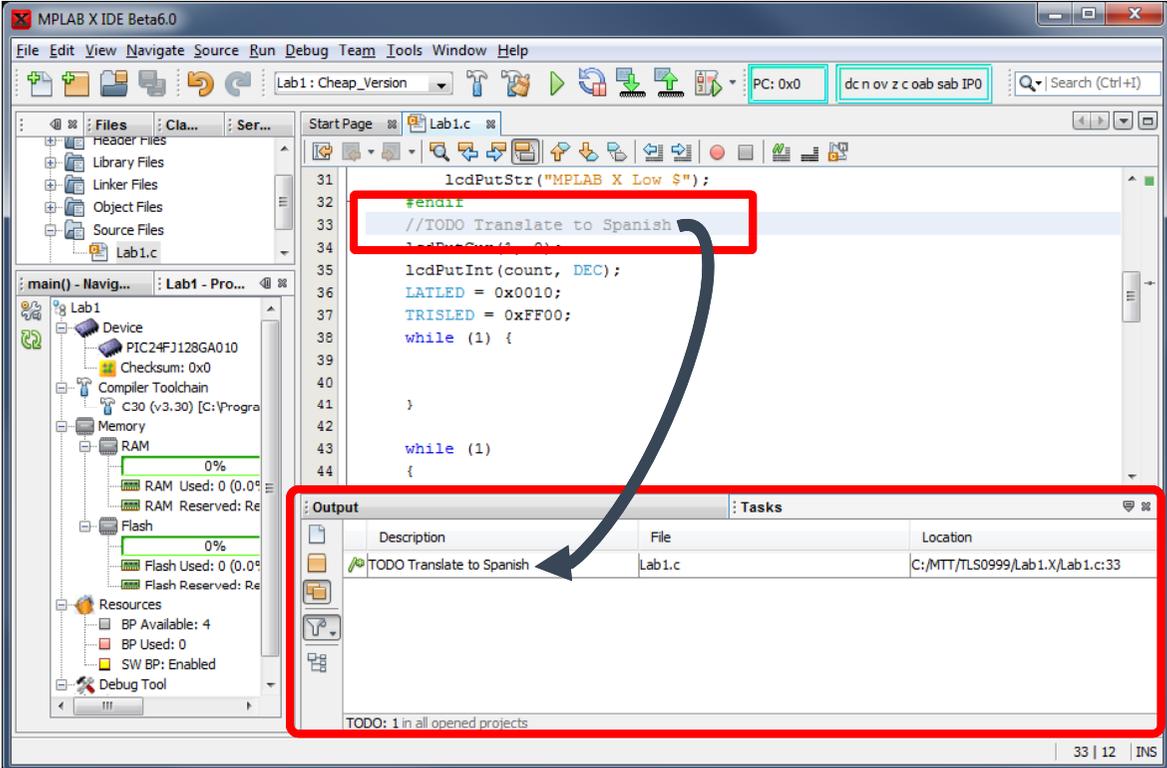
# How to work with the Editor

## Tasks List

### 4 Save the project

Save  the project so that MPLAB® X IDE will parse the files and collect all the specially tagged comments in the tasks window.

Double clicking on a task will take you right to the comment in your source code.



The screenshot shows the MPLAB X IDE interface. The main editor window displays the source code for Lab1.c. A red box highlights a comment line: `//TODO Translate to Spanish`. The Output window at the bottom right shows a table of tasks, with a red box around it and an arrow pointing from the comment in the source code to the task entry.

Description	File	Location
TODO Translate to Spanish	Lab1.c	C:/MTT/TL50999/Lab1.X/Lab1.c:33

Output window also displays: `TODO: 1 in all opened projects`



# How to debug applications

# How to start a debug session



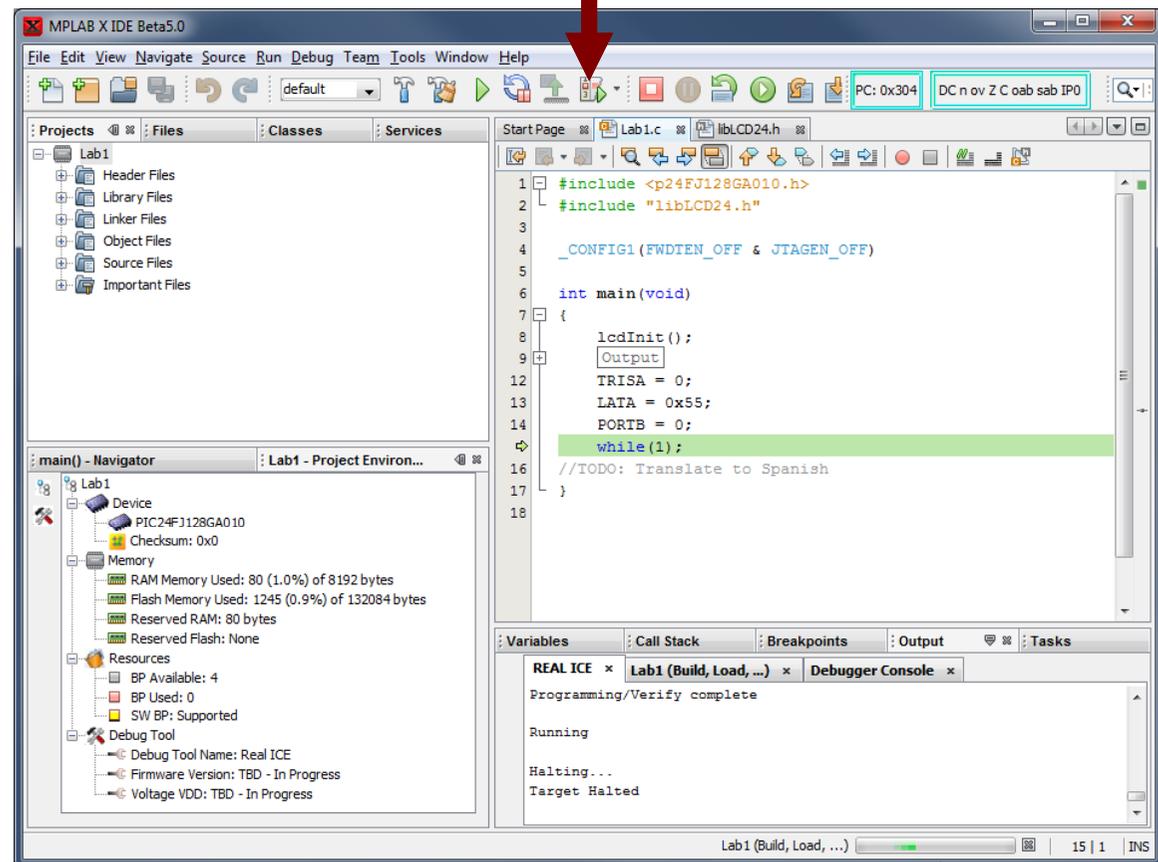
## Debug Project

Starts a debug session.  
Builds project in debug mode.  
If connected to a hardware debugger, this establishes a USB connection, programs the target and runs the program.



## End Debug Session

Ends a debug session.  
If connected to a hardware debugger, this severs the USB connection. This must be done before making changes to code and continuing to debug.



**Session active while  
progress bar visible**

# How to work with the debug toolbar

Function	MPLAB® IDE 8	MPLAB X IDE
End Debug Session	No Equivalent	
Halt / Pause		
Run / Continue		
Animate		No Equivalent
Run To Cursor	In Context Menu	
Step Into		
Step Over		
Reset		
Focus Cursor at PC	No Equivalent	

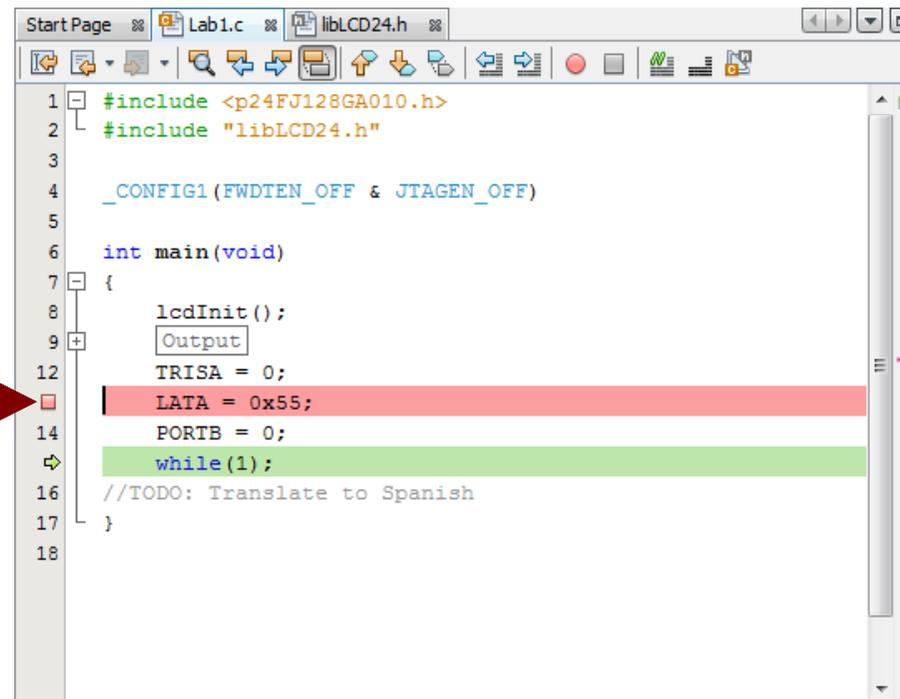
# Breakpoints

## How to set or clear a breakpoint

### Line Breakpoints

Click on a line number in the glyph margin to toggle a breakpoint on that line.

A red square indicates that a breakpoint is set (■). The line of code will also be highlighted in red.



```
1 #include <p24FJ128GA010.h>
2 #include "libLCD24.h"
3
4 _CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
5
6 int main(void)
7 {
8     lcdInit();
9     Output
12 TRISA = 0;
13 LATA = 0x55;
14 PORTB = 0;
15 while(1);
16 //TODO: Translate to Spanish
17 }
18
```

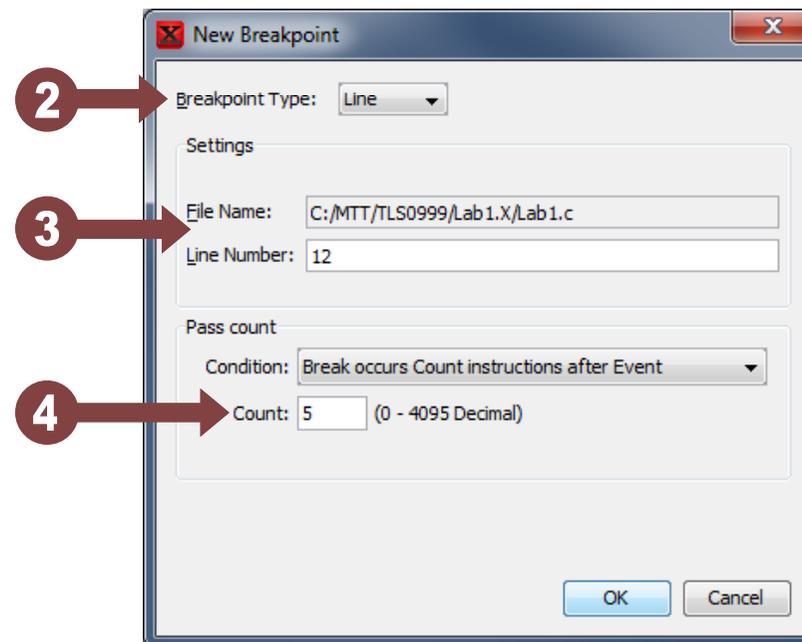
**Click on line number to toggle breakpoint**

# Breakpoints

## How to set a conditional line breakpoint

### Conditional Line Breakpoints

- 1 Select from the menu **Debug** ▶ **New Breakpoint**
- 2 Choose **Line** as the Breakpoint Type
- 3 Specify the data **File** and **Line Number** settings  
(automatically populated from cursor position)
- 4 Specify the **Pass Count Condition**

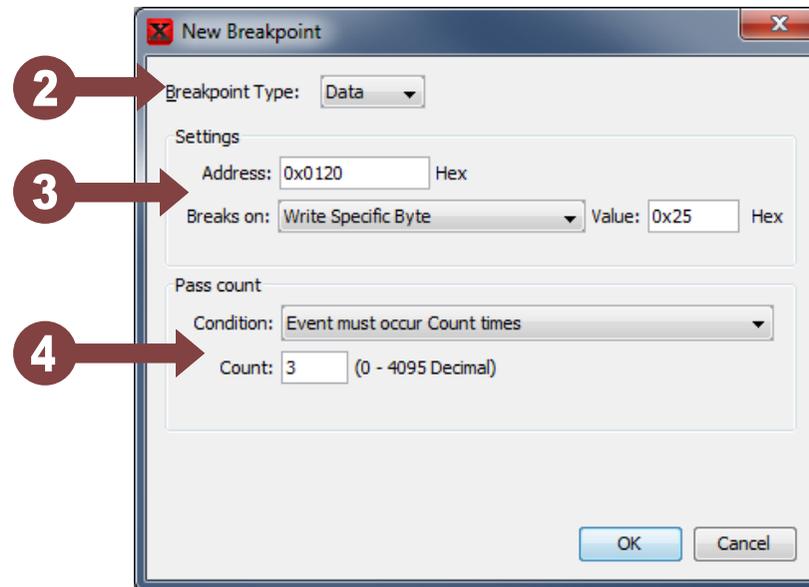


# Breakpoints

## How to set a conditional data breakpoint

### Conditional Data Breakpoints

- 1 Select from the menu **Debug** ▶ **New Breakpoint**
- 2 Choose **Data** as the Breakpoint Type
- 3 Specify the data **Address** and **Breaks on** settings
- 4 Specify the **Pass count** **Condition**

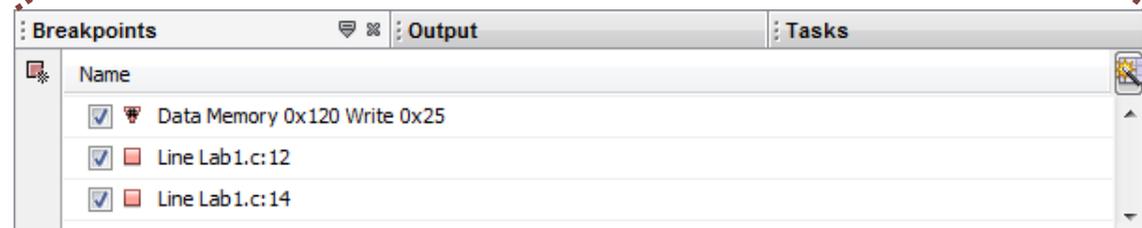
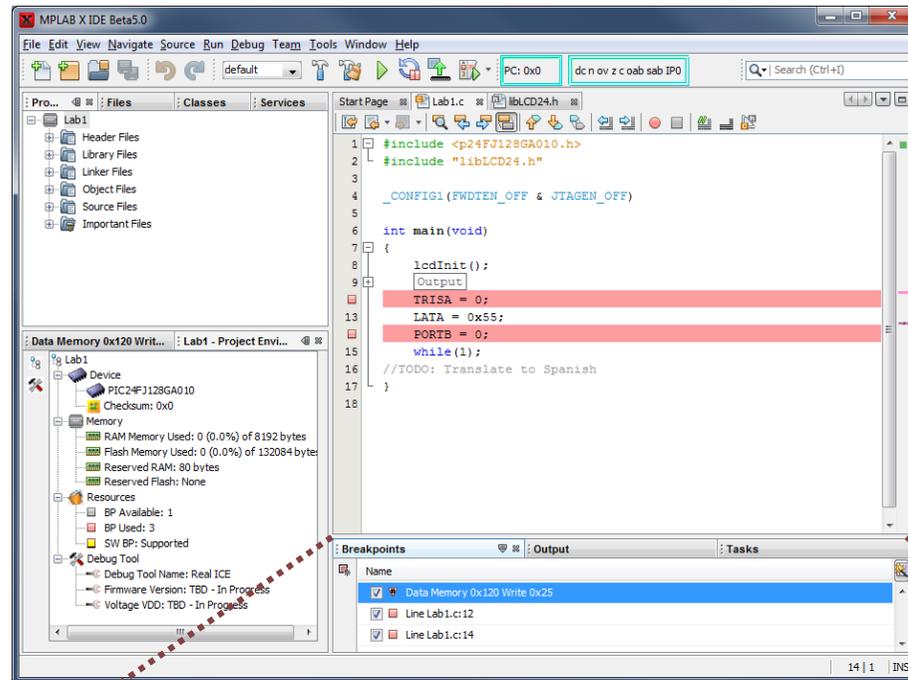


# Breakpoints

## How to view all breakpoints in a project

### Show Breakpoint Window

- 1 Select **Debug** ▶ **New Breakpoint**
- 2 Right click to:
  - Create new breakpoint
  - Enable all breakpoints
  - Disable all breakpoints
  - Delete all breakpoints
  - Group breakpoints (allows enable/disable by group)
  - Change a breakpoint's settings

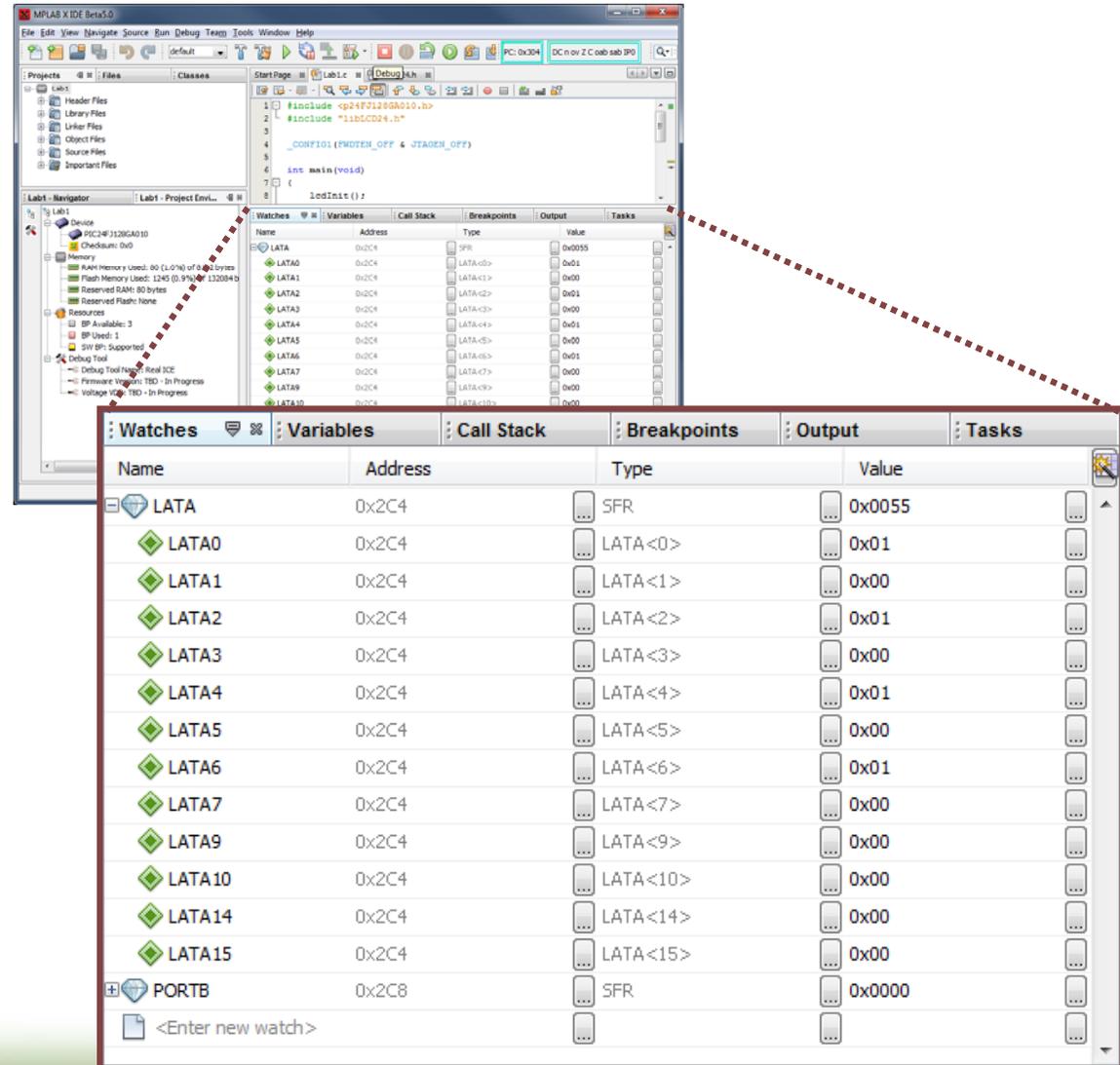
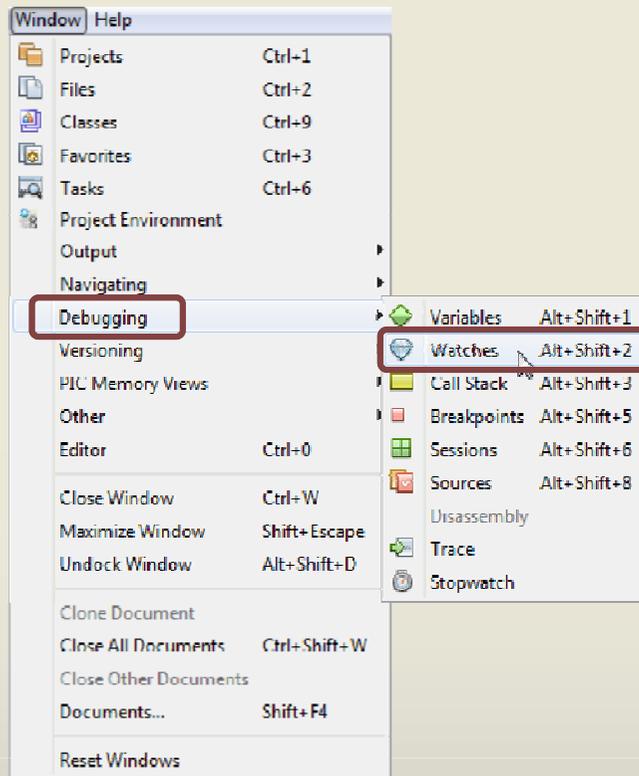


**Double click a breakpoint to go to its source**

# Watches

## How to display the Watches tab

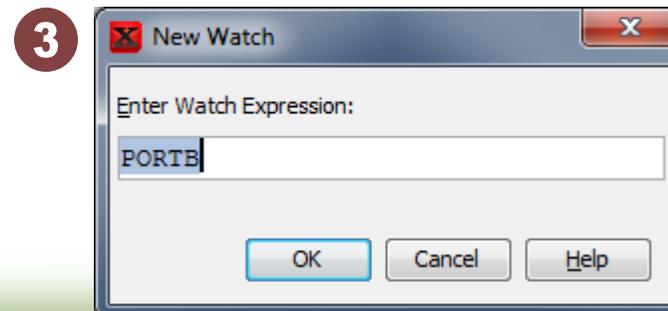
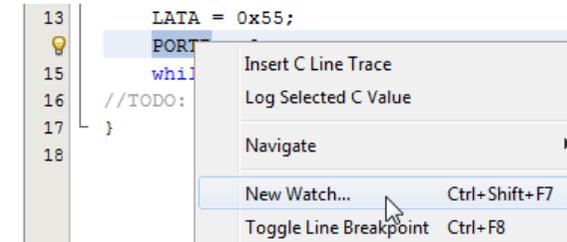
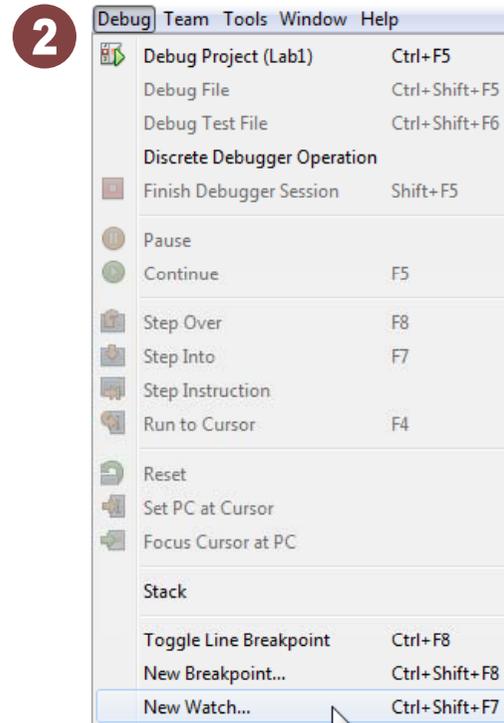
From the main menu select: **Window** ▶  
**Debugging** ▶ **Watches**



# Watches

## How to add a variable to the watches tab

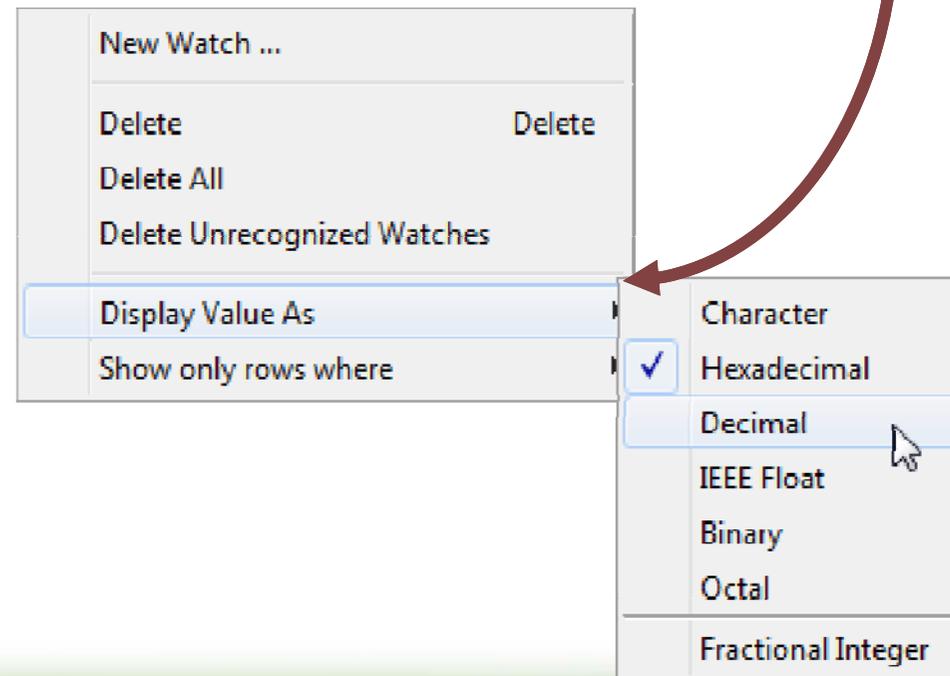
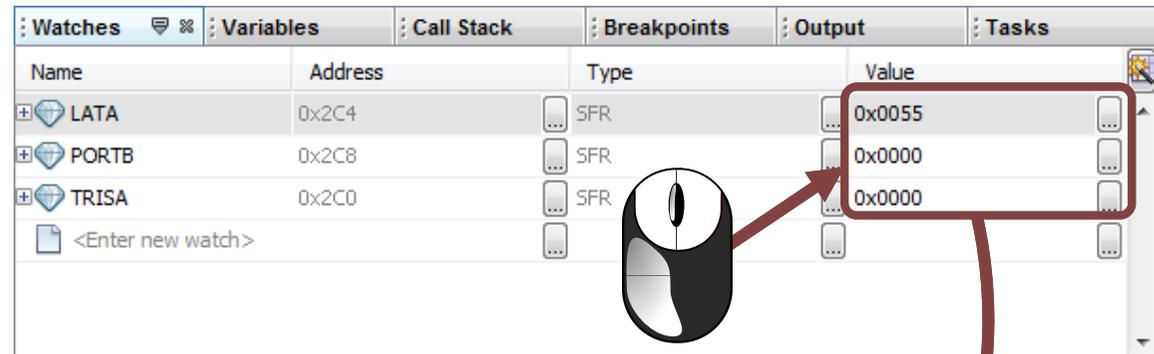
- 1 (Optional) Highlight variable in editor
- 2 Do one of the following:
  - a Right click in editor and select **New Watch...** from the popup menu
  - b Select from main menu: **Debug ▶ New Watch...**
  - c If you highlighted the variable in the editor, click and drag it to the Watches window
  - d 
- 3 Enter variable name or just accept what's there then click 



# Watches

## How to change a watch variable's radix

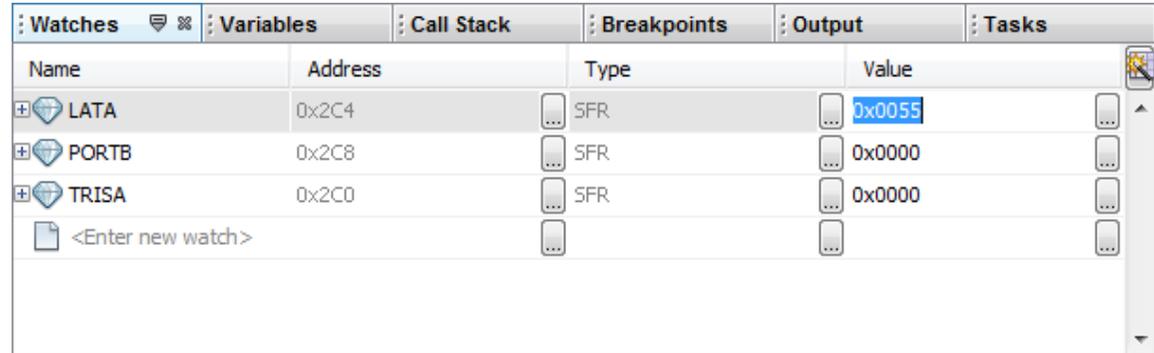
- 1 Right click on a value in the **Value** column
- 2 Select from the popup menu:  
**Display Value As**
- 3 Choose the desired radix / format:
  - Character
  - Hexadecimal
  - Decimal
  - IEEE Float
  - Binary
  - Octal
  - Fractional Integer



# Watches

## How to change a watch variable's value

- 1 Double click on a value in the **Value** column
- 2 Enter a new value and hit the Enter key when done



Name	Address	Type	Value
LATA	0x2C4	SFR	0x0055
PORTB	0x2C8	SFR	0x0000
TRISA	0x2C0	SFR	0x0000
<Enter new watch>			



Values may not be changed on the fly. You must halt execution of your program before a change can be made.

# Watches

## How to sort watch variables

Click on the column headings...

**1** Default  
(Fields Sorted Numerically)

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA0	
0x2C4	LATA1	
0x2C4	LATA2	
0x2C4	LATA3	
0x2C4	LATA4	
0x2C4	LATA5	
0x2C4	LATA6	

**2** 1st Click:  
Sort Ascending (Alphabetically)

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA0	
0x2C4	LATA1	
0x2C4	LATA10	
0x2C4	LATA14	
0x2C4	LATA15	
0x2C4	LATA2	
0x2C4	LATA3	

**3** 2nd Click:  
Sort Descending (Alphabetically)

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA9	
0x2C4	LATA7	
0x2C4	LATA6	
0x2C4	LATA5	
0x2C4	LATA4	
0x2C4	LATA3	
0x2C4	LATA2	

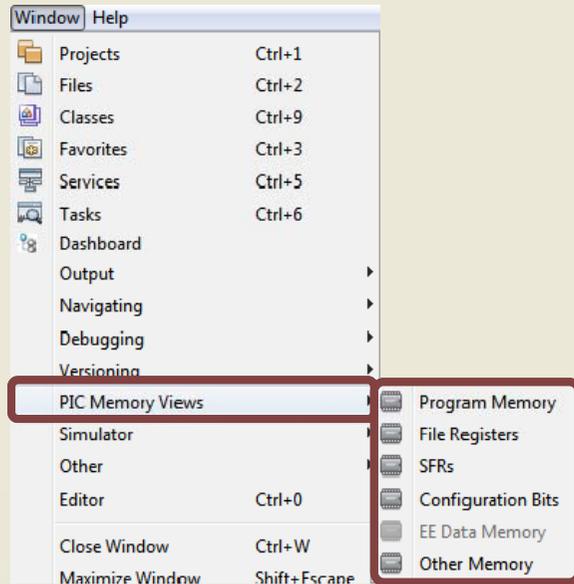
**4** 3rd Click:  
Default

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA0	LAT
0x2C4	LATA1	LAT
0x2C4	LATA2	LAT
0x2C4	LATA3	LAT
0x2C4	LATA4	LAT
0x2C4	LATA5	LAT
0x2C4	LATA6	LAT

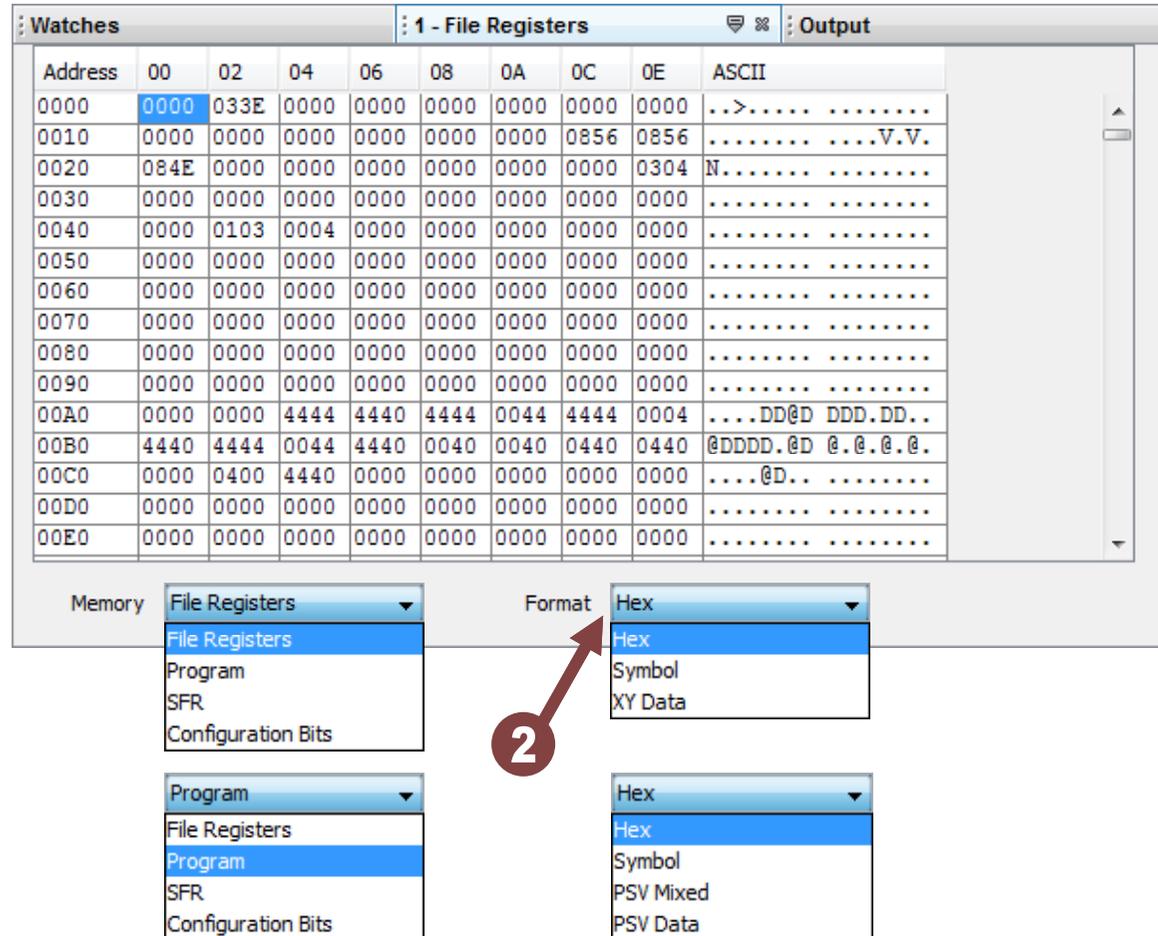
# Memory Windows

## How to view embedded memory

- From the main menu select: **Window** ▶  
**PIC Memory Views** ▶



- Select a display format



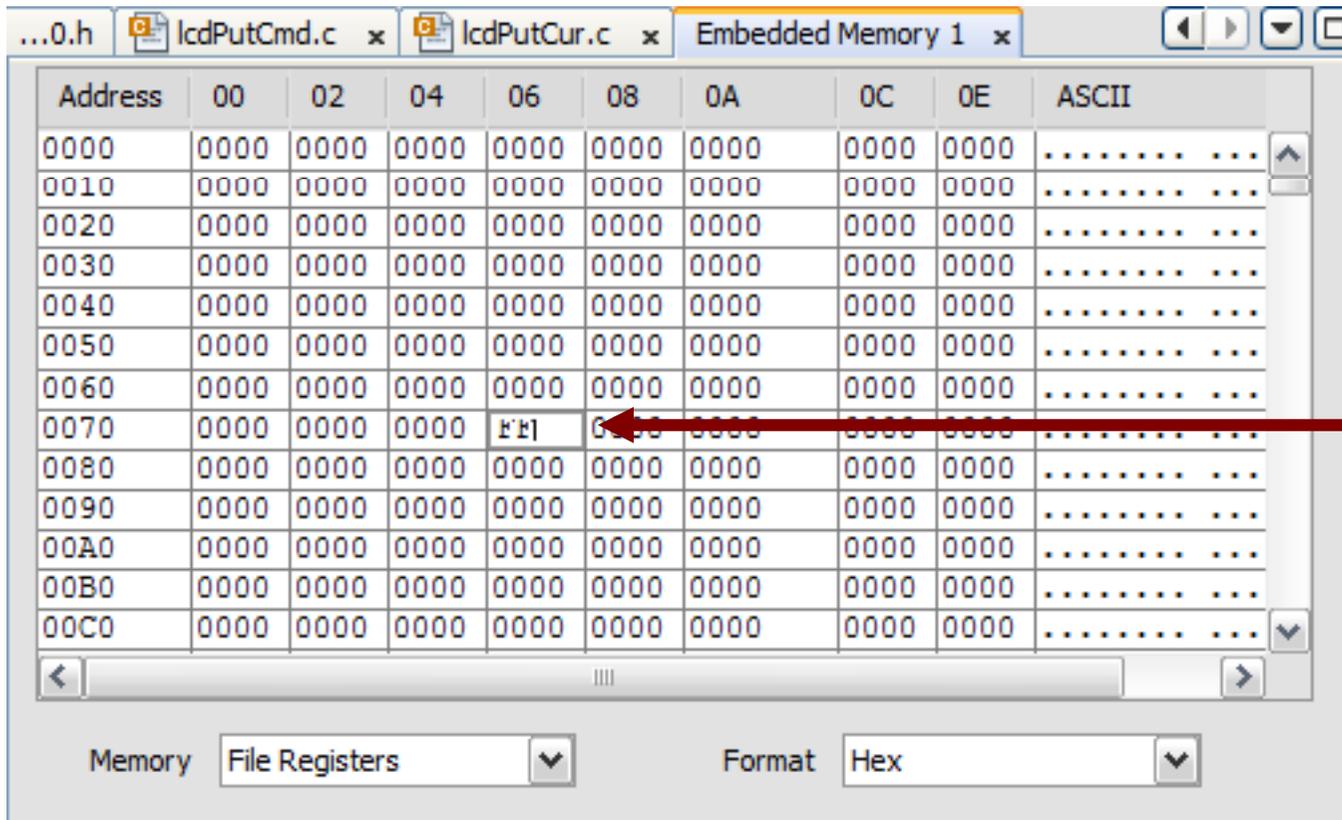
Address	00	02	04	06	08	0A	0C	0E	ASCII
0000	0000	033E	0000	0000	0000	0000	0000	0000	...>.....
0010	0000	0000	0000	0000	0000	0000	0856	0856	.....V.V.
0020	084E	0000	0000	0000	0000	0000	0000	0304	N.....
0030	0000	0000	0000	0000	0000	0000	0000	0000	.....
0040	0000	0103	0004	0000	0000	0000	0000	0000	.....
0050	0000	0000	0000	0000	0000	0000	0000	0000	.....
0060	0000	0000	0000	0000	0000	0000	0000	0000	.....
0070	0000	0000	0000	0000	0000	0000	0000	0000	.....
0080	0000	0000	0000	0000	0000	0000	0000	0000	.....
0090	0000	0000	0000	0000	0000	0000	0000	0000	.....
00A0	0000	0000	4444	4440	4444	0044	4444	0004	...DD@D DDD.DD..
00B0	4440	4444	0044	4440	0040	0040	0440	0440	@DDDD.@D @. @. @. @.
00C0	0000	0400	4440	0000	0000	0000	0000	0000	...@D..
00D0	0000	0000	0000	0000	0000	0000	0000	0000	.....
00E0	0000	0000	0000	0000	0000	0000	0000	0000	.....



Each of the memory view windows may be reconfigured to display any of the supported memory types.

# Memory Windows

## How to change a memory location's value



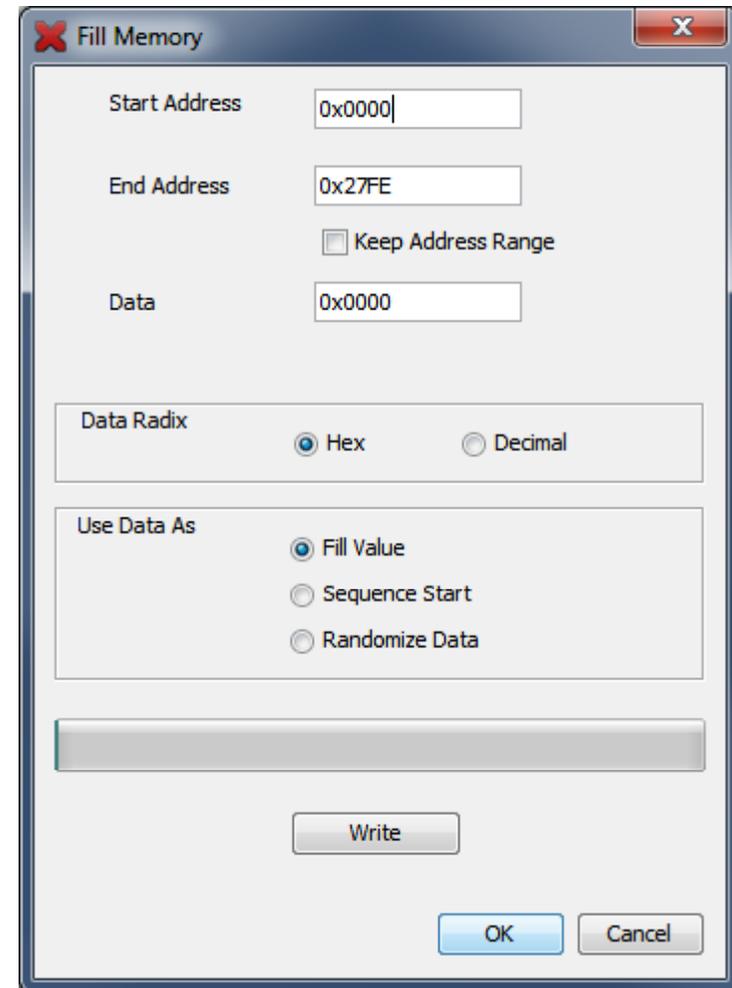
Address	00	02	04	06	08	0A	0C	0E	ASCII
0000	0000	0000	0000	0000	0000	0000	0000	0000	.....
0010	0000	0000	0000	0000	0000	0000	0000	0000	.....
0020	0000	0000	0000	0000	0000	0000	0000	0000	.....
0030	0000	0000	0000	0000	0000	0000	0000	0000	.....
0040	0000	0000	0000	0000	0000	0000	0000	0000	.....
0050	0000	0000	0000	0000	0000	0000	0000	0000	.....
0060	0000	0000	0000	0000	0000	0000	0000	0000	.....
0070	0000	0000	0000	FF	0000	0000	0000	0000	.....
0080	0000	0000	0000	0000	0000	0000	0000	0000	.....
0090	0000	0000	0000	0000	0000	0000	0000	0000	.....
00A0	0000	0000	0000	0000	0000	0000	0000	0000	.....
00B0	0000	0000	0000	0000	0000	0000	0000	0000	.....
00C0	0000	0000	0000	0000	0000	0000	0000	0000	.....

Double click on a cell and start typing the new value – hit Enter when done

# Memory Windows

## How to fill memory

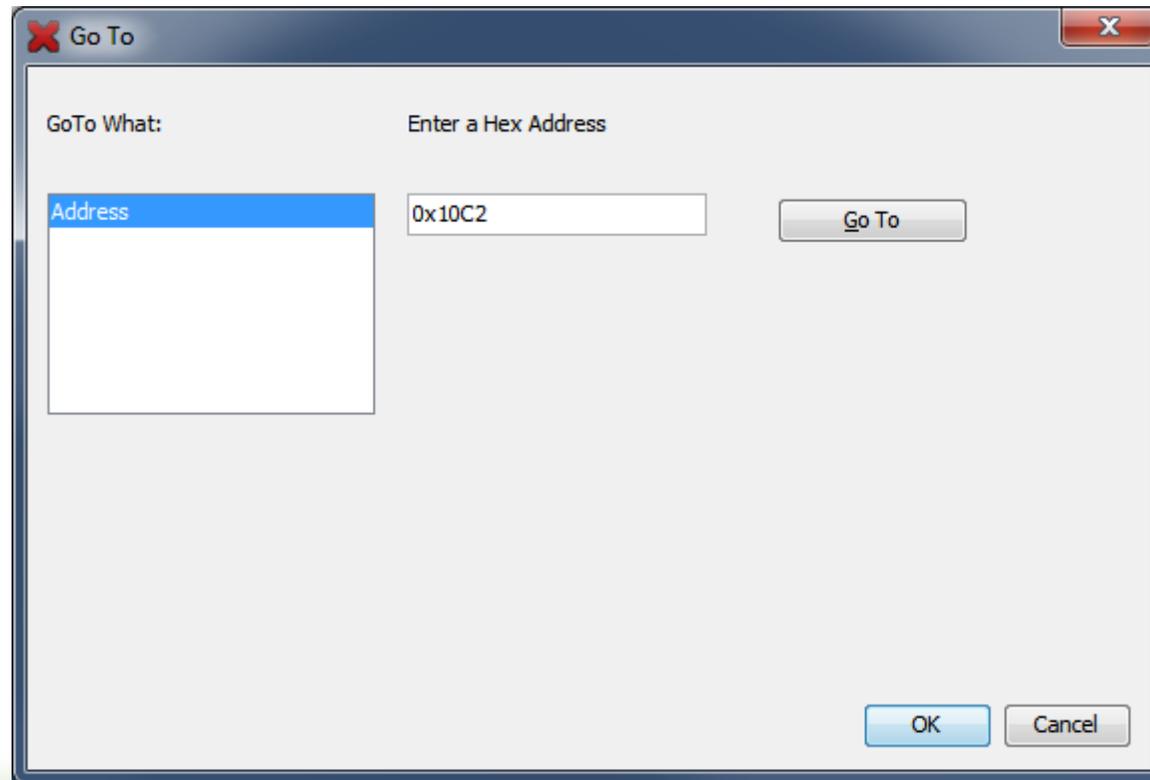
- Right click in memory window and select **Fill Memory...** from the popup menu



# Memory Windows

## How to quickly jump to a specific address

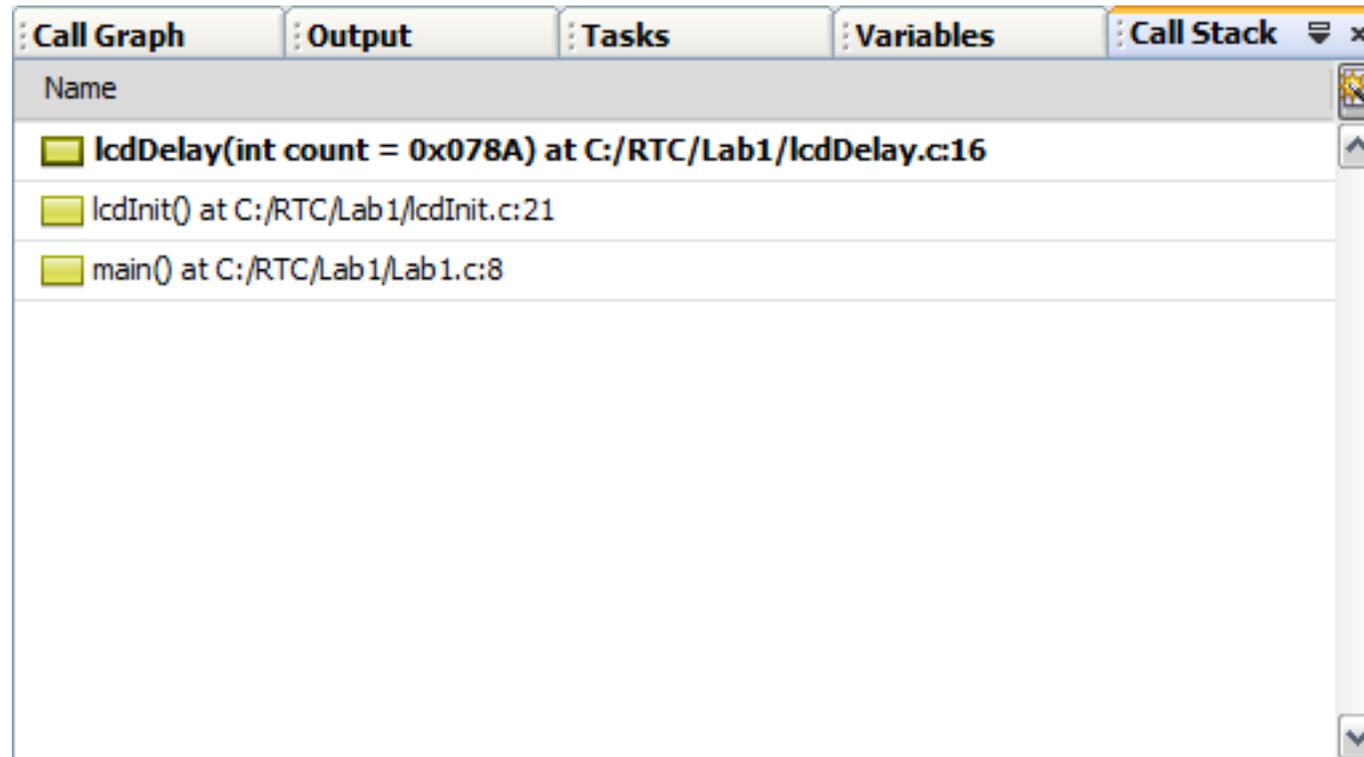
- Right click in the memory window and select **Go To...** from the popup menu



# Call Stack

## How to view the call stack

- **Select Window ► Debugging ► Call Stack**



Double click or right click on a stack level to make the editor display the line of code that made the call.



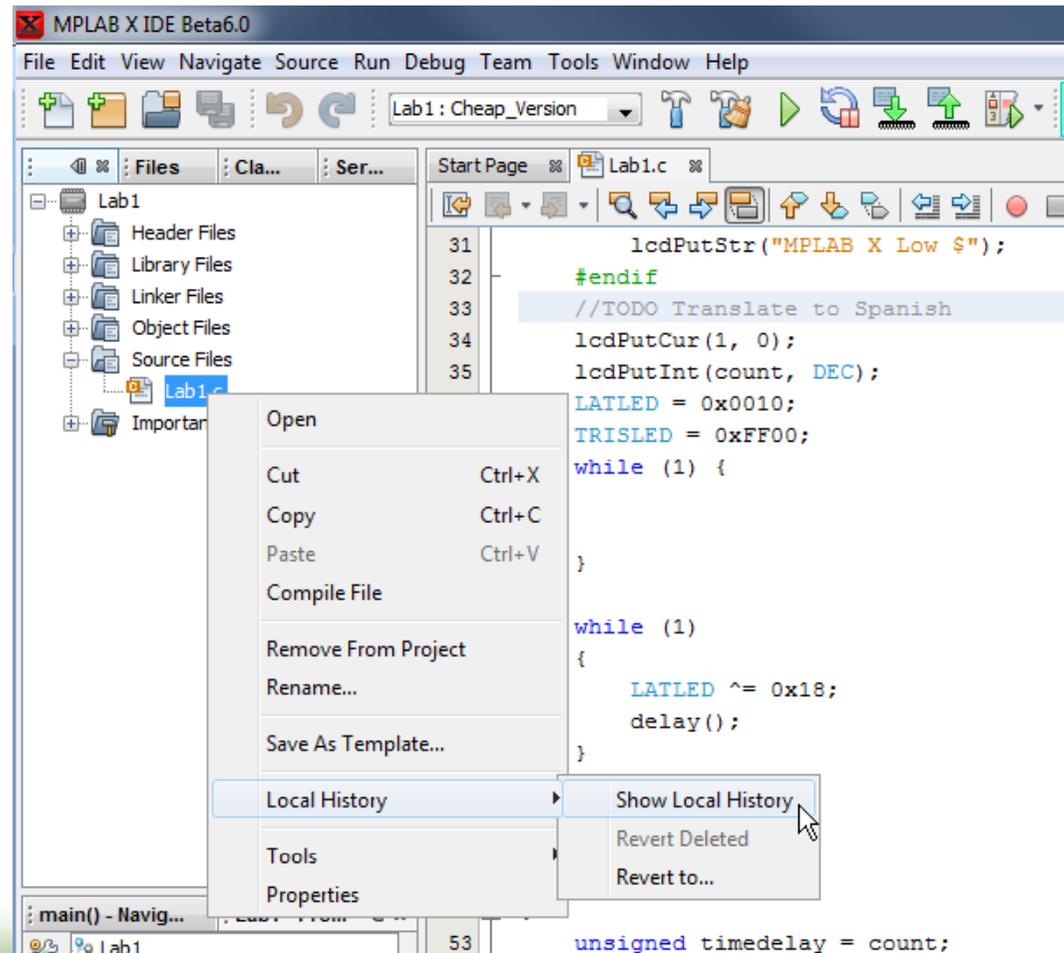
# How to work with the Local History

# How to work with the Local History

## 1 Select a source file and display its local history

Right click on a source file in the project tree and select from the popup menu:

**Local History ▶**  
**Show Local History**



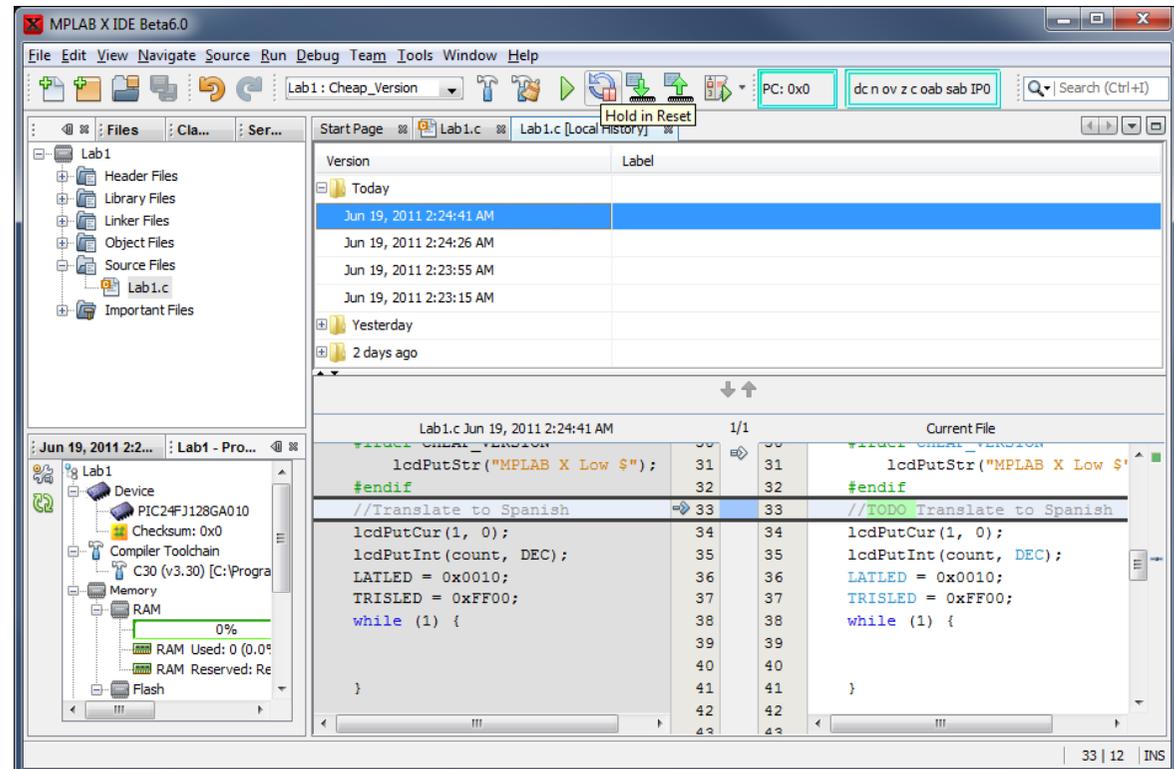
# How to work with the Local History

## 2 Select a previous version in the upper pane

A new tab will open up in the editor area.

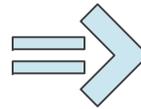
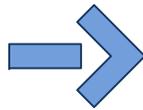
The upper pane shows a list of previous versions of the selected file.

The lower pane shows a diff comparison between the current version (right) and the previous version (left) selected above.



# How to work with the Local History

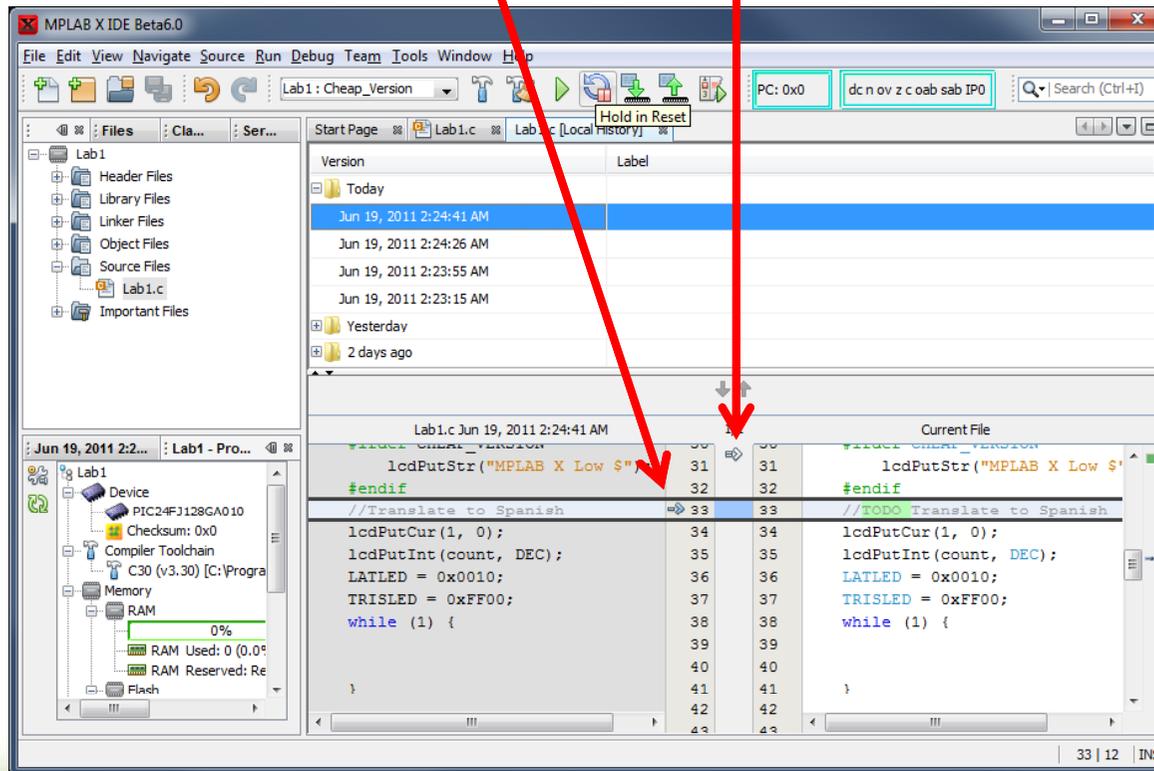
Add item from previous version to current version.



Revert entire file to previous version (replace current with previous).



Remove item from current version to match previous version.



MPLAB X IDE Beta6.0

Lab1: Cheap\_Version

PC: 0x0 dc n ov z c oab sab IP0

Local History

Version	Label
Today	
Jun 19, 2011 2:24:41 AM	
Jun 19, 2011 2:24:26 AM	
Jun 19, 2011 2:23:55 AM	
Jun 19, 2011 2:23:15 AM	
Yesterday	
2 days ago	

Lab1.c Jun 19, 2011 2:24:41 AM

```

//MPLAB X LOCAL HISTORY
lcdPutStr("MPLAB X Low $");
#endif
//Translate to Spanish
33 //TODO Translate to Spanish
lcdPutCur(1, 0);
lcdPutInt(count, DEC);
LATLED = 0x0010;
TRISLED = 0xFF00;
while (1) {
}

```

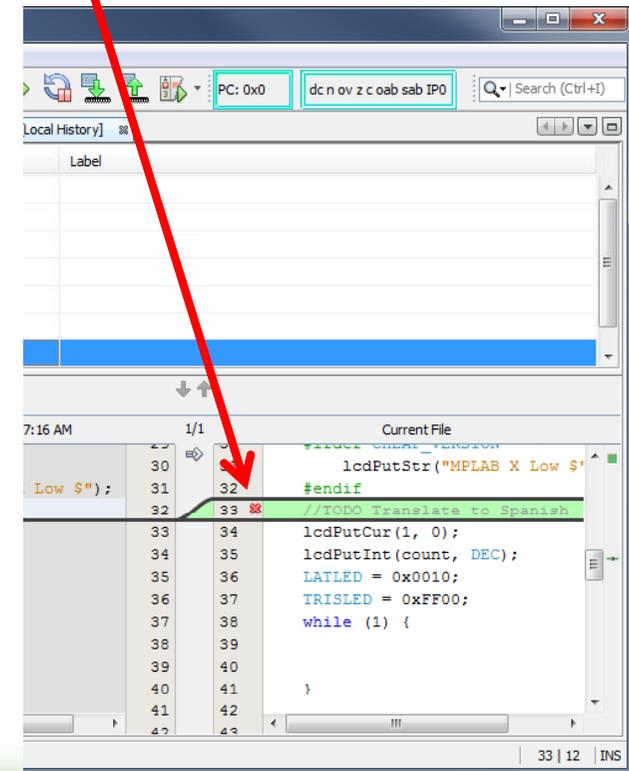
Current File

```

//MPLAB X LOCAL HISTORY
lcdPutStr("MPLAB X Low $");
#endif
//TODO Translate to Spanish
33 //TODO Translate to Spanish
lcdPutCur(1, 0);
lcdPutInt(count, DEC);
LATLED = 0x0010;
TRISLED = 0xFF00;
while (1) {
}

```

33 | 12 | INS



Local History

Version	Label
7:16 AM	
1/1	

Current File

```

//MPLAB X LOCAL HISTORY
30 lcdPutStr("MPLAB X Low $");
31 #endif
32 //TODO Translate to Spanish
33 //TODO Translate to Spanish
34 lcdPutCur(1, 0);
35 lcdPutInt(count, DEC);
36 LATLED = 0x0010;
37 TRISLED = 0xFF00;
38 while (1) {
39
40
41
42
43

```

33 | 12 | INS

# How to work with the Local History

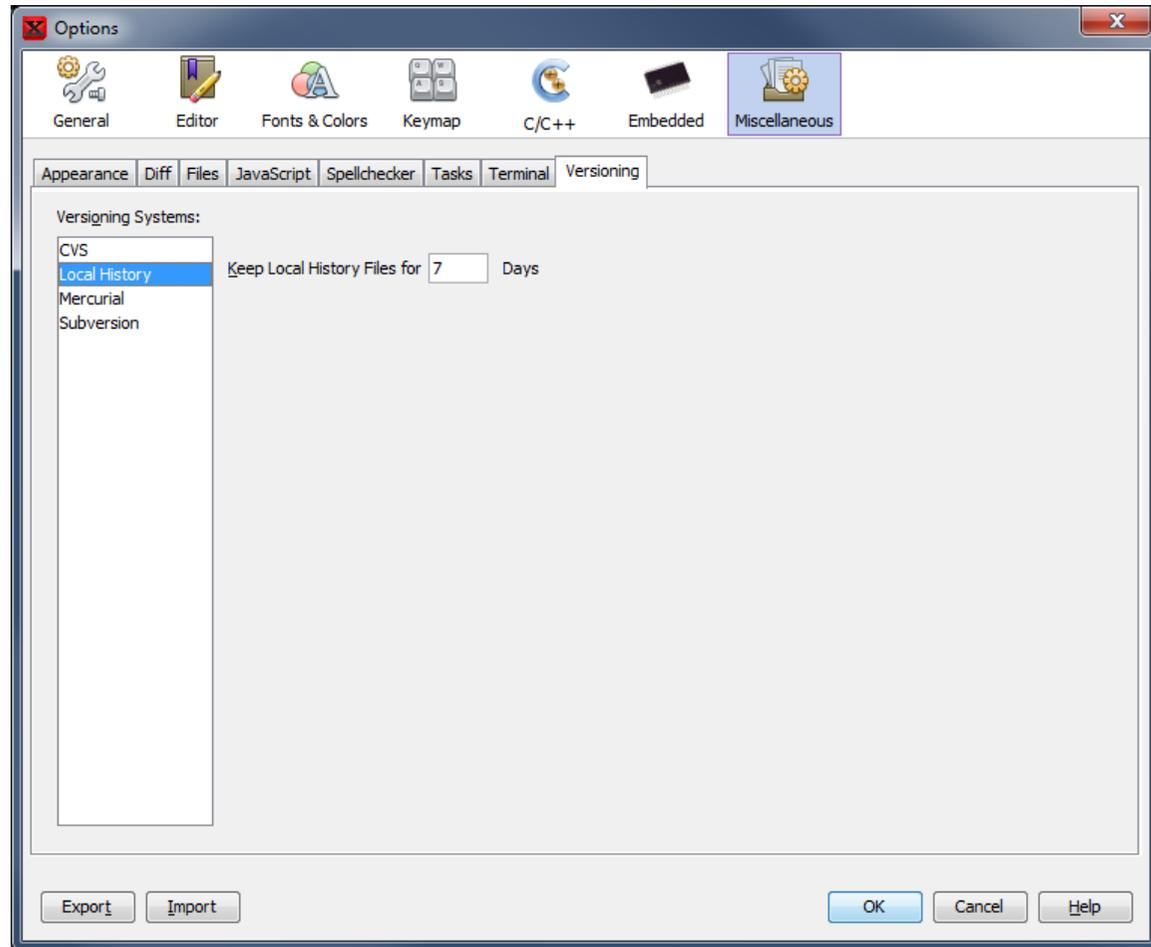
To alter the length of time Local History will keep track of files, select from the main menu:

**Tools ▶ Options**

Choose **Miscellaneous** category

Select **Versioning** Tab

Select **Local History** under **Versioning Systems** at left





**MICROCHIP**

**MASTERS Conference**

# **MASTERS** 2012

The premier technical training conference for embedded control engineers

## **How to convert an existing MPLAB® IDE 8 Project**

# How to prepare for conversion

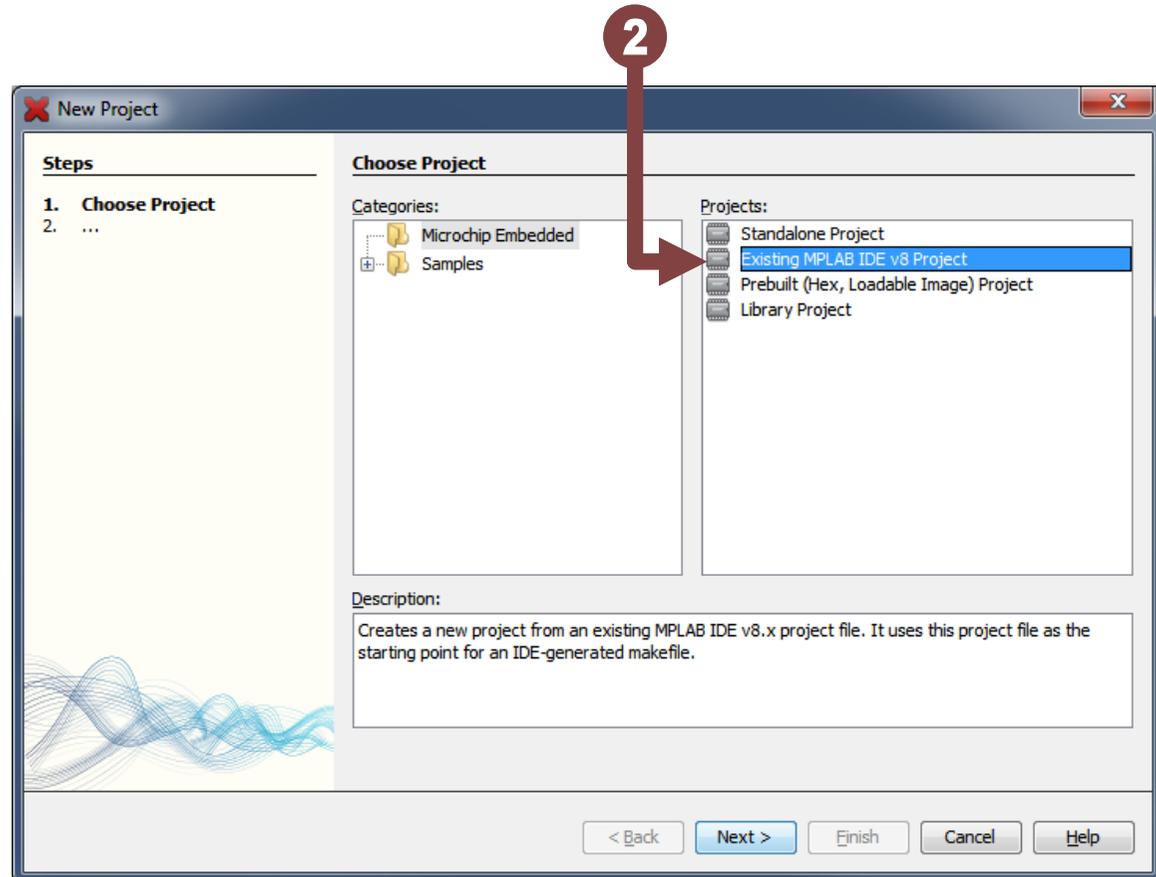
- **Ensure project builds in MPLAB® IDE v8**
- **Ensure version is 8.14.03A or later**
- **Ensure all header files are in the project tree – not just in code via `#include`**
- **If converting for use on Linux or Mac:**
  - Ensure `#include` filenames use correct case  
e.g. "p24FJ128GA010.h" – NOT "p24fj128ga010.h"
  - Use '/' in relative paths instead of '\\'  
e.g. `#include "Graphics/graphics.h"`

# After Conversion

- **Source files are shared between MPLAB® IDE 8 and MPLAB X projects**
- **Project may be worked on from either IDE**
  - Changes to source files reflected in each IDE
  - Changes to project settings only in one IDE
- **MPLAB X IDE project files in subdirectory of MPLAB IDE 8 project directory**

# How to convert a legacy project

- 1 Start a new project by clicking on the new project icon: 
- 2 Choose Existing MPLAB® IDE v8 Project under Projects
- 3 Follow the remaining steps in the Wizard, which are very similar to those for creating a new standalone project.





# How to work with multiple projects

# How to work with multiple projects

## Active Projects vs. Main Projects

### Active Project

The *Active Project* is the one that is currently selected in the project tree. All project related actions will be applied to the Active Project until a different project is selected.

### Main Project

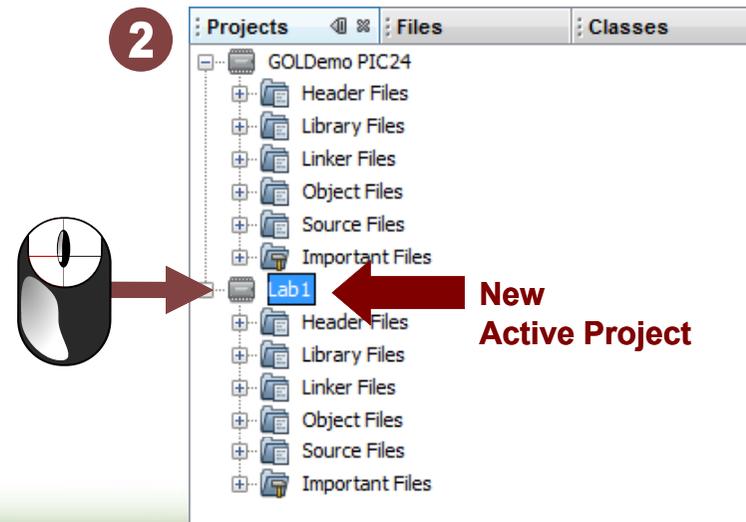
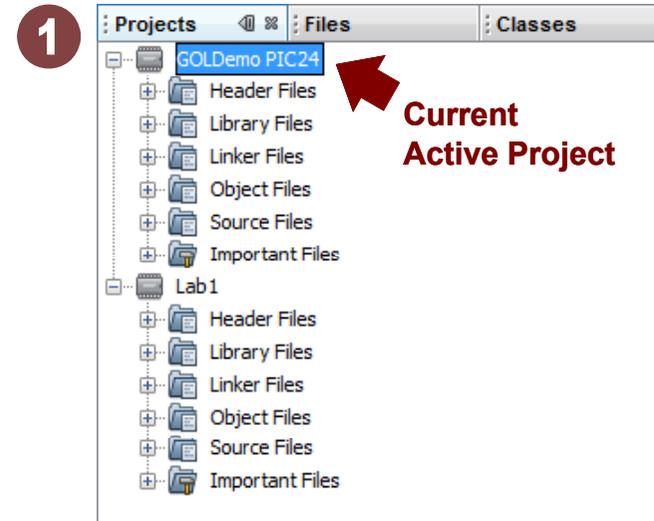
A *Main Project* is explicitly selected from the right click menu. Once selected, all project related actions will apply to the Main Project, regardless of which project is selected in the project tree.

# How to work with multiple projects

## How to select the active project

In the Project tree, left click on the project you want to make active.

Once selected, all displays will reflect data for the newly selected project and all actions will be applied to that project.

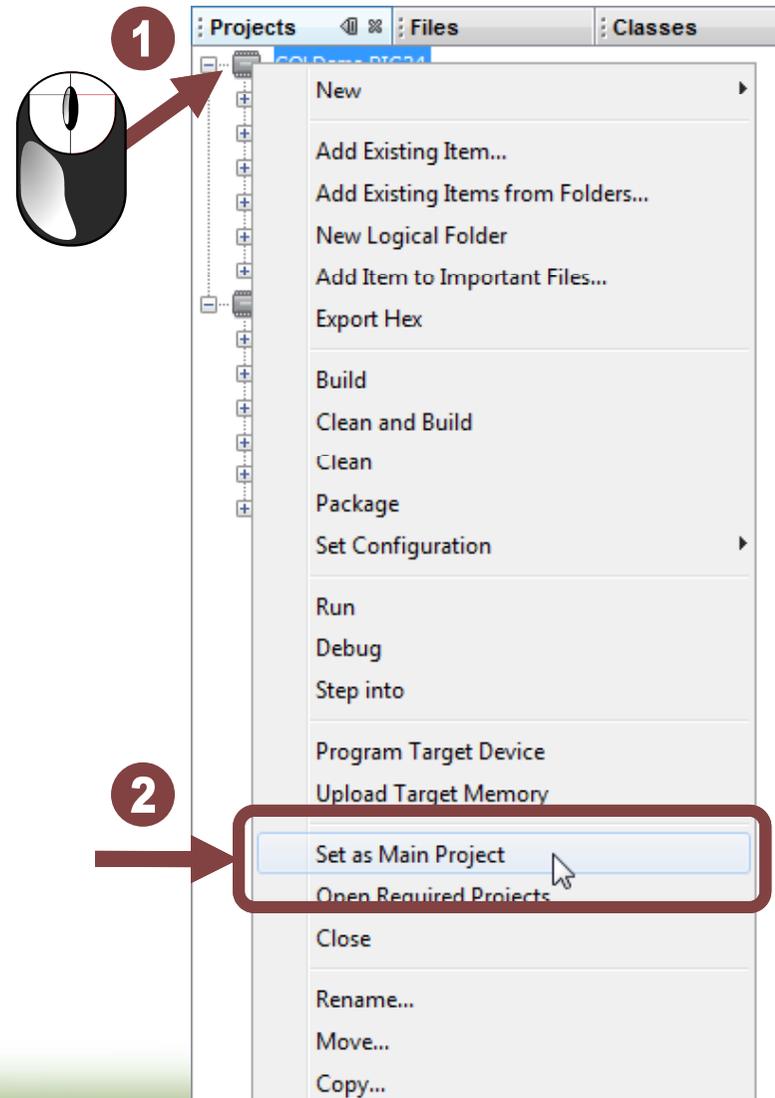


# How to work with multiple projects

## How to set a Main project

In the Project tree, right click on the project you want to be the Main project, then select **Set as Main Project** from the popup menu.

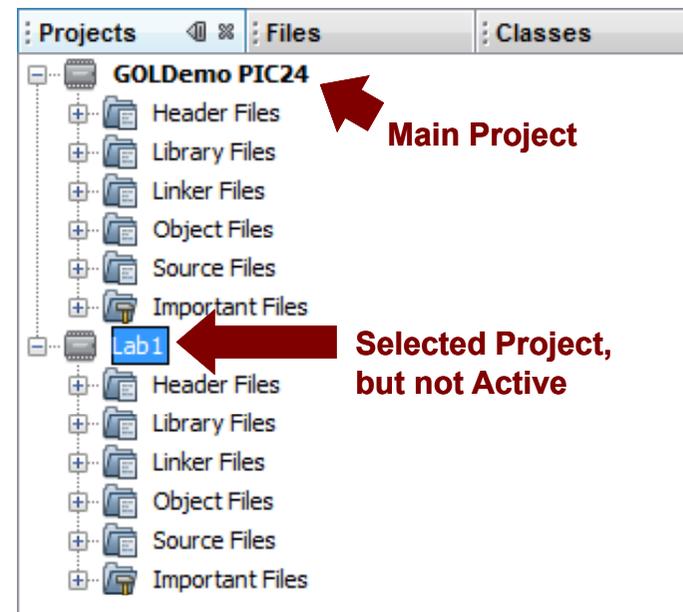
All displays will reflect data for the main project and all actions will be applied to the main project only.



# How to work with multiple projects

## The Main Project

The Main Project will always appear with a bold typeface to indicate that it is active regardless of which project is selected in the project tree.





***Thank You!***

# Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KeeLoq, KeeLoq logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A. All other trademarks mentioned herein are property of their respective companies.

© 2012, Microchip Technology Incorporated, All Rights Reserved.