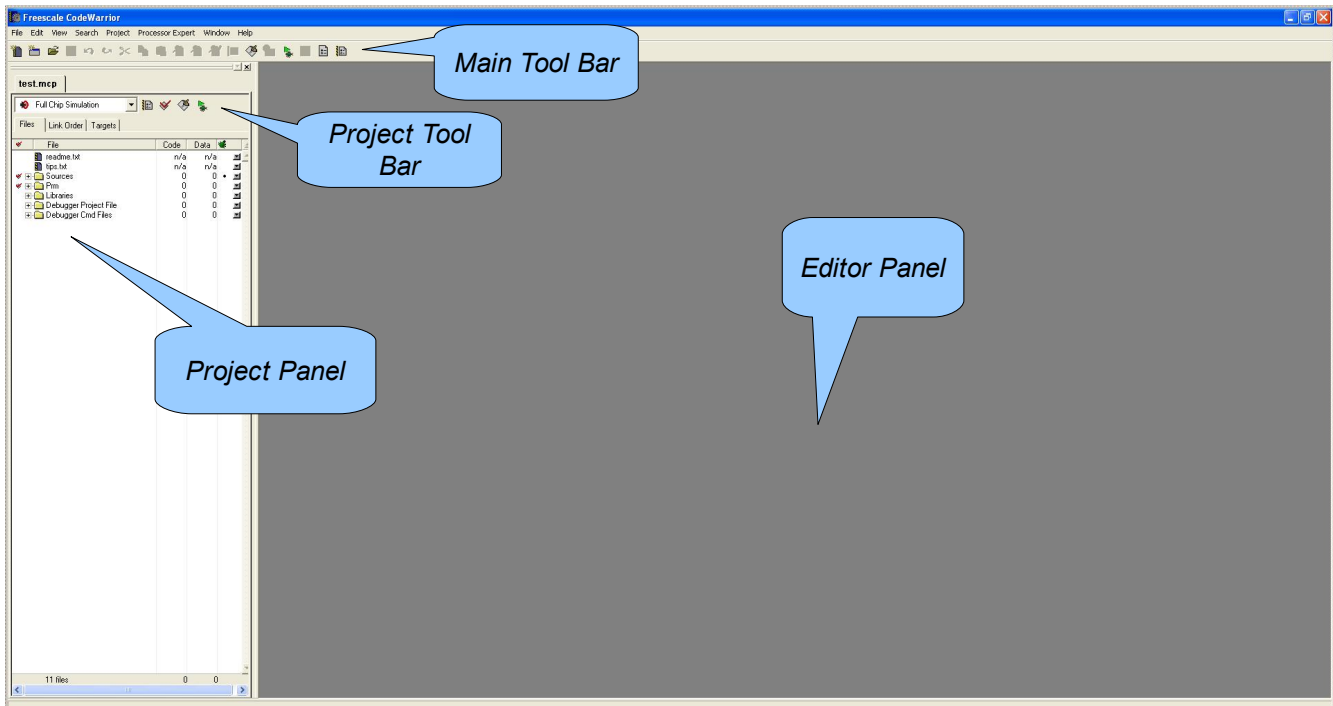


Quick Guide to Using Code Warrior

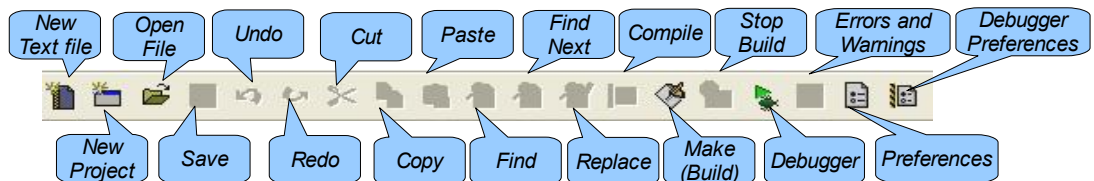
So now that you have created a project, you are going to want to write an assembly program and assemble (compile) it.

When you create a project a new panel appears on the left of the screen, this is the project panel. The project panel contains all the files and settings required to ensure your program compiles properly for the right hardware device (in our case the MC9S12XDP512 processor).



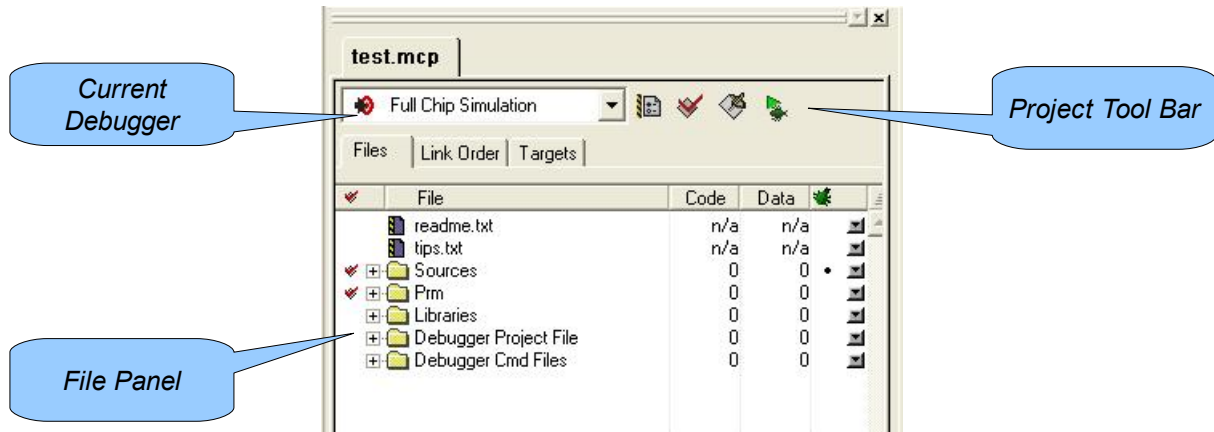
Main Tool Bar

The main tool bar allows you to create new projects and files, it also allows you to open and save your files, as well as cut, copy and paste from one file to another. The main tool bar also allows you compile, build and debug your programs. The figure below explains the function of each button on the main tool bar, buttons are gray if they are unavailable.



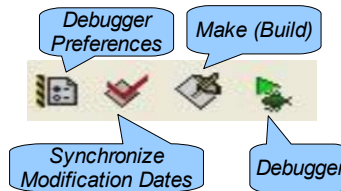
Project Panel

The project panel contains everything associated with your project. It contains a project tool bar and a file panel.



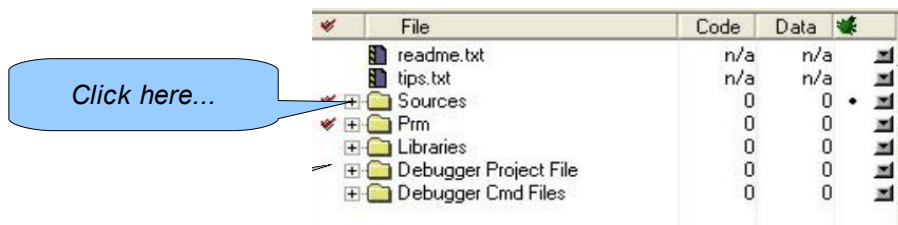
Project Tool bar

The project tool bar allows you to easily check, build and debug your programs.



Project File Panel

The project file panel contains all the files associated with your project. These files are categorized into sub folders depending on their purpose. The **sources** directory contains all your source code, all your assembly files are contained within this directory. The **prm** directory contains ????? and the **Libraries** directory contains any library files your project may be using to support the various devices (subsystems) built into the processor. The two remaining folders (**Debugger Project File**, **Debugger Cmd Files**) contain configuration data for the debugger.



Step 1: How to Modify the example assembly file

If you click you click the plus sign next to the sources directory, it will expand showing you all your source files.



If you double click "**main.asm**", then Code warrior will open the text file "**main.asm**" inside the editor panel, as shown in following figure.

The editor panel where the “*main.asm*” code appears is simply a text editor. This panel allows you to modify and save your code. You must first modify the example code in this panel so that you can use it for your own programs. Below is a listing of “*main.asm*”, the lines that are colored red should be removed, the lines colored blue should be added and the lines colored black should be left alone.

```

;*****
;* This stationery serves as the framework for a          *
;* user application (single file, absolute assembly application) *
;* For a more comprehensive program that                 *
;* demonstrates the more advanced functionality of this   *
;* processor, please see the demonstration applications   *
;* located in the examples subdirectory of the           *
;* Freescale CodeWarrior for the HC12 Program directory  *
;*****
; export symbols
        XDEF Entry          ; export 'Entry' symbol
        ABSENTRY Entry      ; for absolute assembly: mark this as application entry
point

; include derivative specific macros
        INCLUDE 'mc9s12xdp512.inc'

ROMStart EQU $4000 ; absolute address to place my code/constant data

; variable/data section

        ORG RAMStart
; Insert here your data definition.
Counter DS.W 1
FiboRes DS.W 1

; code section
        ORG ROMStart
Entry:
        LDS #RAMEnd+1      ; initialize the stack pointer
        CLI                ; enable interrupts
mainLoop:
        LDX #1             ; X contains counter
couterLoop:
        STX Counter        ; update global.
        BSR CalcFibo
        STD FiboRes        ; store result
        LDX Counter
        INX
        CPX #24            ; larger values cause overflow.
        BNE couterLoop
        BRA mainLoop      ; restart.

CalcFibo: ; Function to calculate fibonacci numbers. Argument is in X.
        LDY #$00          ; second last
        LDD #$01          ; last
        DBEQ X,FiboDone   ; loop once more (if X was 1, were done already)
FiboLoop:
        LEAY D,Y          ; overwrite second last with new value
        EXG D,Y           ; exchange them -> order is correct again
        DBNE X,FiboLoop
FiboDone:
        RTS                ; result in D
Spin:   BRA Spin

;*****
;* Interrupt Vectors                                     *
;*****
        ORG $FFFE
        DC.W Entry        ; Reset Vector

```

When you have finished modifying “main.asm” it should look like this.

```

;*****
;* This stationery serves as the framework for a          *
;* user application (single file, absolute assembly application) *
;* For a more comprehensive program that                 *
;* demonstrates the more advanced functionality of this   *
;* processor, please see the demonstration applications   *
;* located in the examples subdirectory of the            *
;* Freescale CodeWarrior for the HC12 Program directory   *
;*****

; export symbols
XDEF Entry          ; export 'Entry' symbol
ABSENTRY Entry      ; for absolute assembly: mark this as application entry

point

; include derivative specific macros
INCLUDE 'mc9s12xdp512.inc'

ROMStart EQU $4000 ; absolute address to place my code/constant data

; variable/data section

ORG RAMStart
; Insert here your data definition.

; code section
ORG ROMStart
Entry:
LDS #RAMEnd+1      ; initialize the stack pointer
CLI                ; enable interrupts

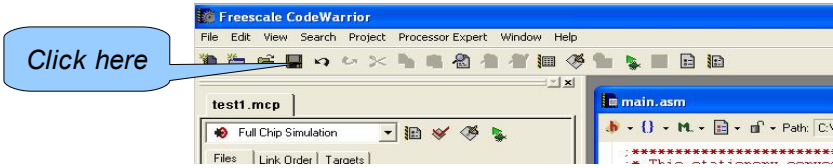
mainLoop:
Spin: BRA Spin

;*****
;* Interrupt Vectors                                     *
;*****
ORG $FFFE
DC.W Entry          ; Reset Vector
    
```

Variables go here...

Your assembly code goes here...

Once you have made the changes to “main.asm” save it.



The assembler source file “main.asm” is now ready to be modified for a particular problem.

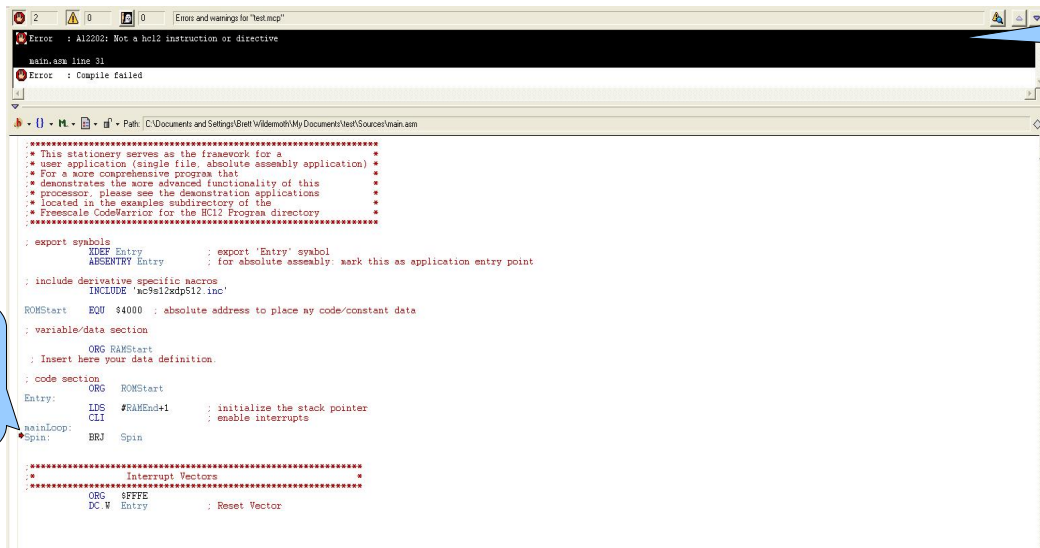
Step 2: How to Make (Build) your modified Code

To ensure that the assembly file “main.asm” is free of any errors you should click the “make” button on the project panel tool bar.



If the assembly code is error free then nothing will appear on the screen. However if an error is detected the

following panel will appear above the “main.asm” code.



Error Panel

Arrow showing line in code where error appears.

Now that you are sure that the code is error free, it is time to run it on the Adapt9S12X microcontroller. This process is outlined in the final document in this series “Using the Debugger inside Code Warrior” (CW-07-003.pdf).