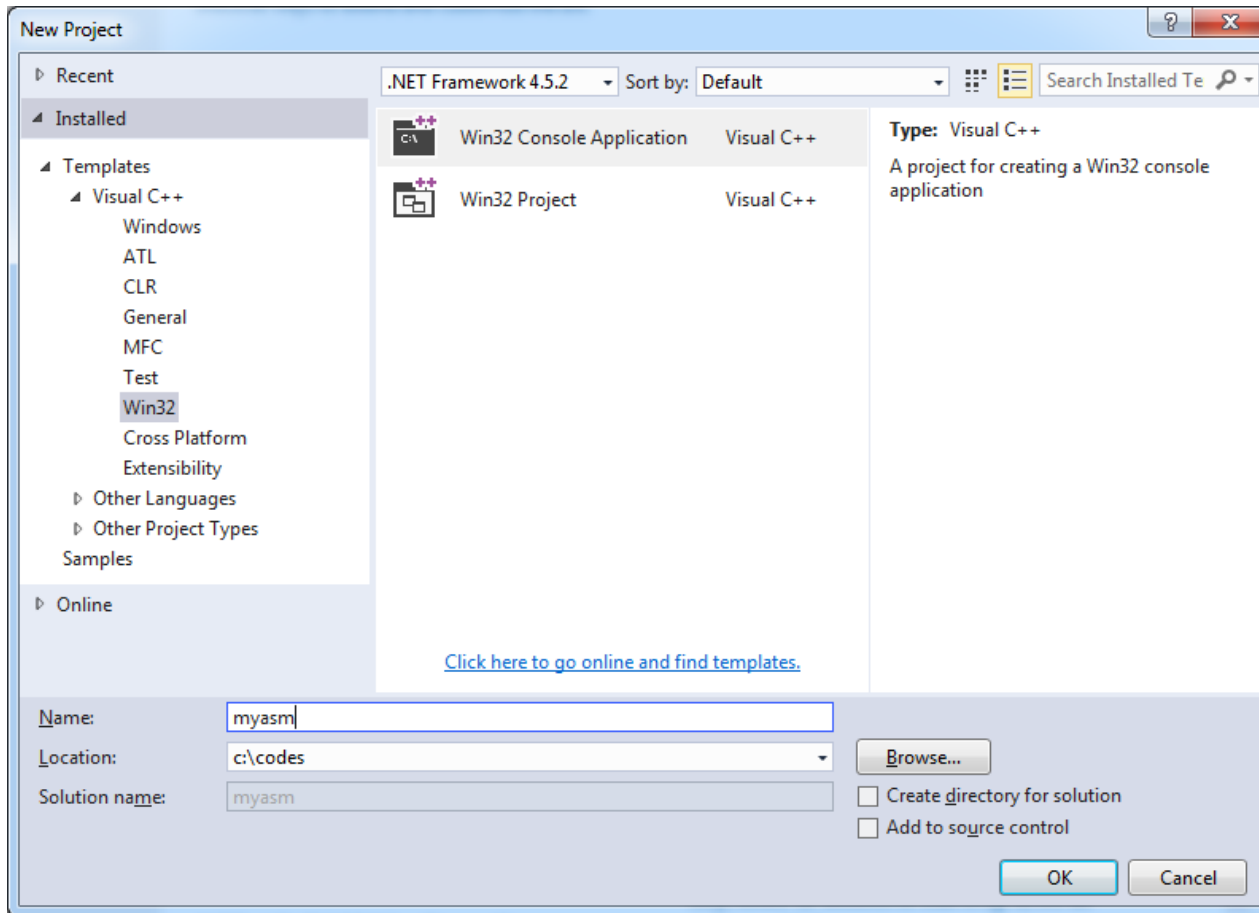


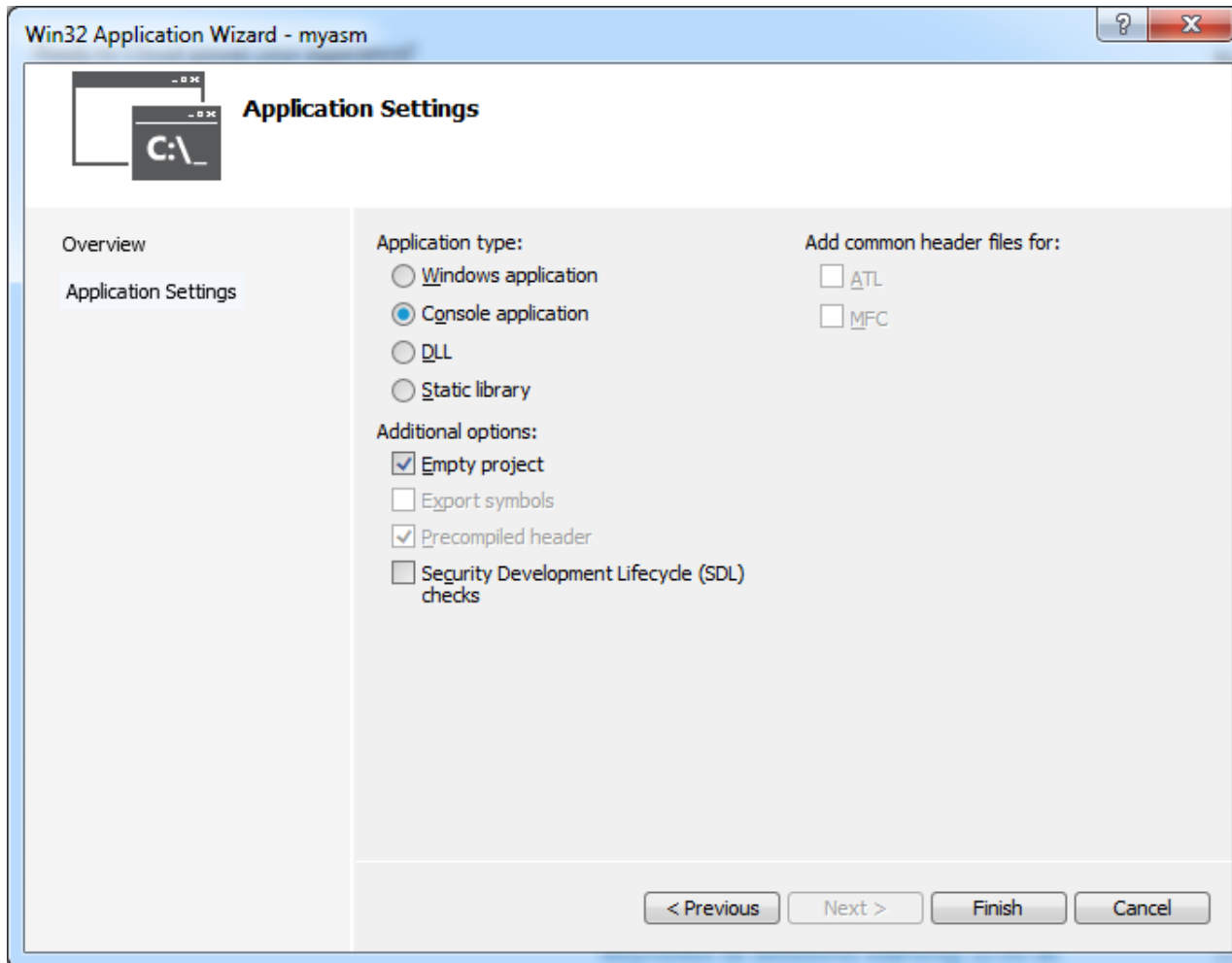
Coding and debugging Assembly in Visual Studio

Add new project



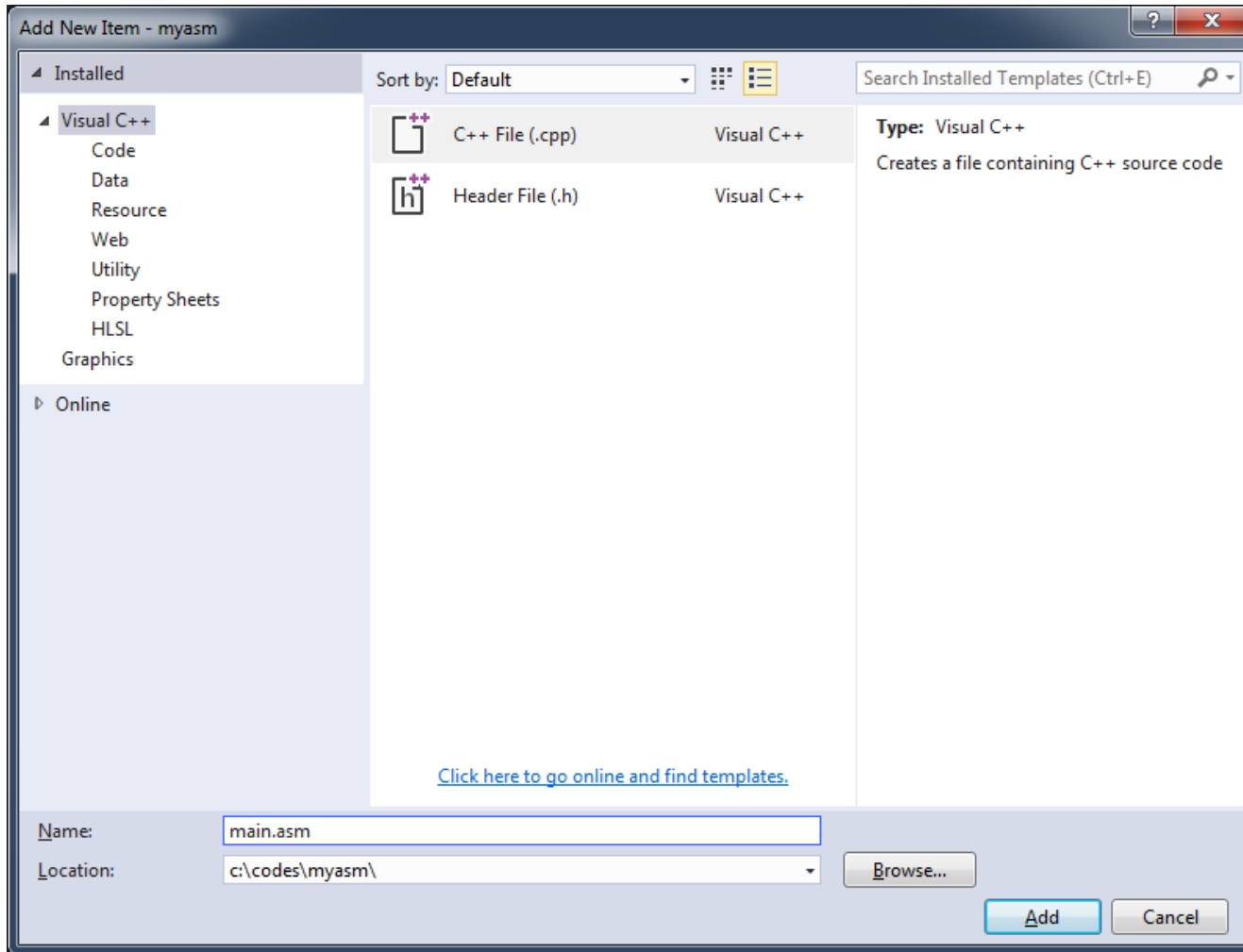
- Locate Visual C++ template
- Win32 Console Application
- Enter project name
- Press OK

Project settings



- Check 'Empty project'
- Uncheck 'Security Development Lifecycle'
- Press OK to proceed

Add new item



- Right click on source in Solution Explorer
- Add -> New Item
- Enter filename – file_name.asm
- Press Add

Code template

```
.386
.model flat, stdcall
.stack 4096
;-----
; include and lib files
;-----
include      msvcrt.inc
includelib   msvcrt.lib
include      kernel32.inc
include      user32.inc
;-----
; Data section
;-----
.data
    msg_text    db  "Hello, world! %d times", 10, 0      ; 10 is New Line in Ascii table
    var1        dd  1000
    var2        dd  3000
;-----
; Code section
;-----
.code

main proc

    mov     eax, [var1]
    add     eax, [var2]

    ; call printf function manually
    push    eax
    push    offset msg_text
    call    [crt_printf] ; calling a C (CDECL) function
    add     esp, 8        ; caller clean the stack frame

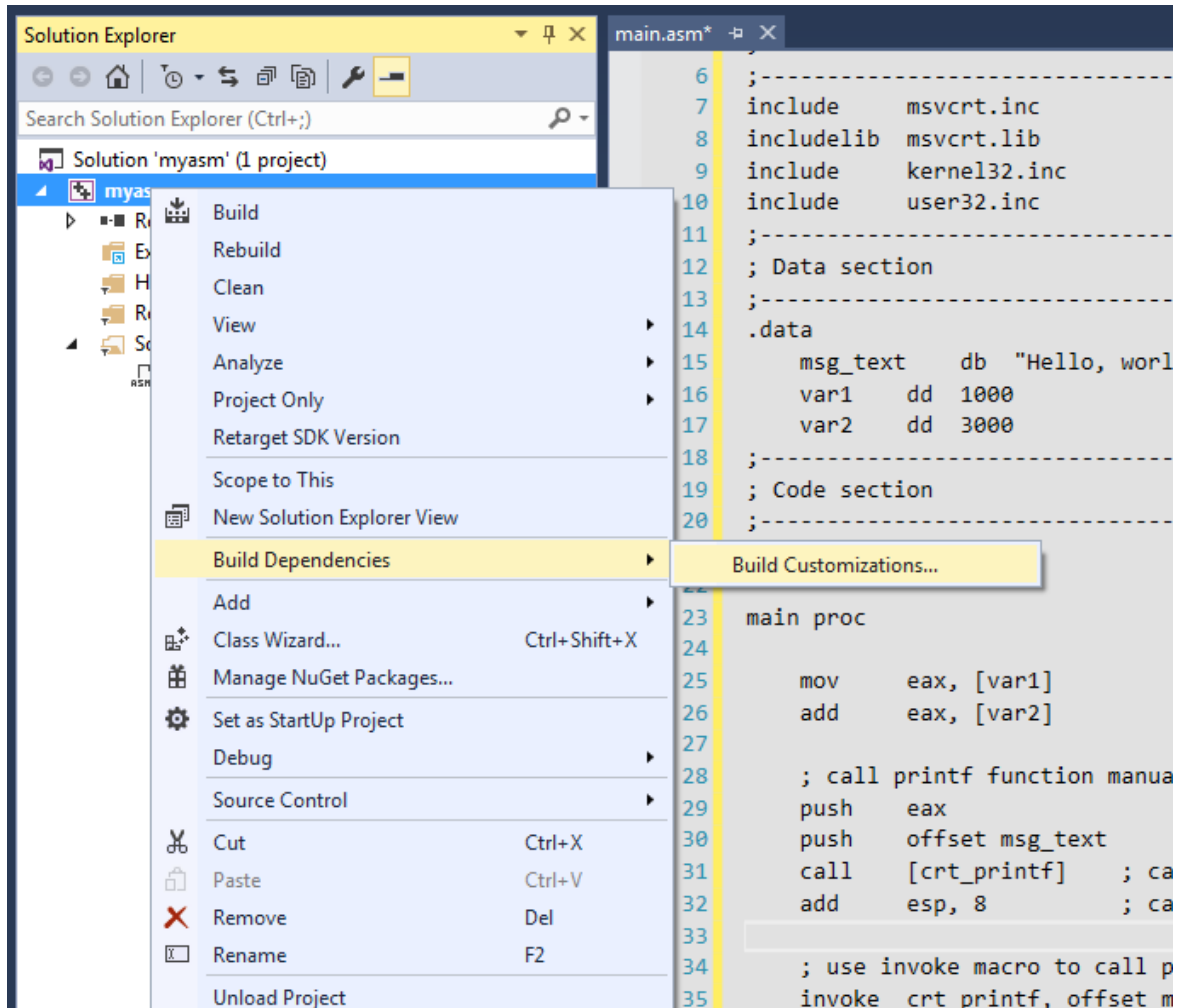
    ; use invoke macro to call printf function
    invoke  crt_printf, offset msg_text, var2
    ; add     esp, 8 // this line will be added by the macro

    invoke  ExitProcess, 0 ; use invoke macro to call ExitProcess API
    ret

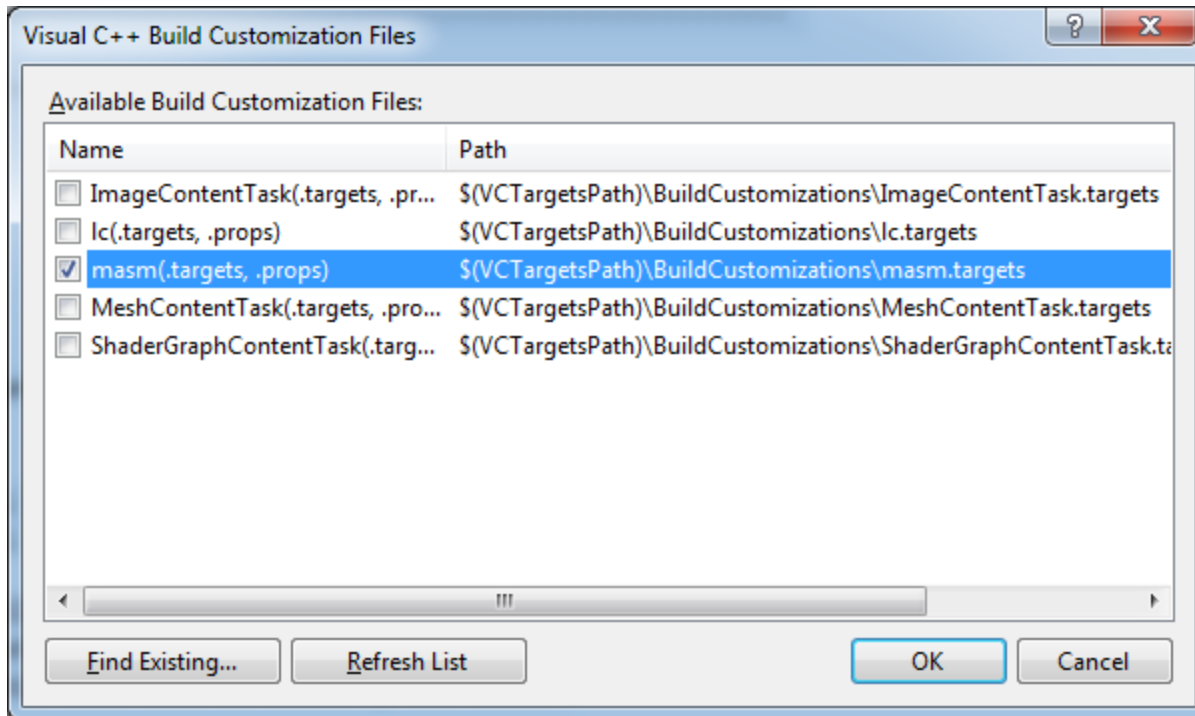
main endp

end main
;-----
```

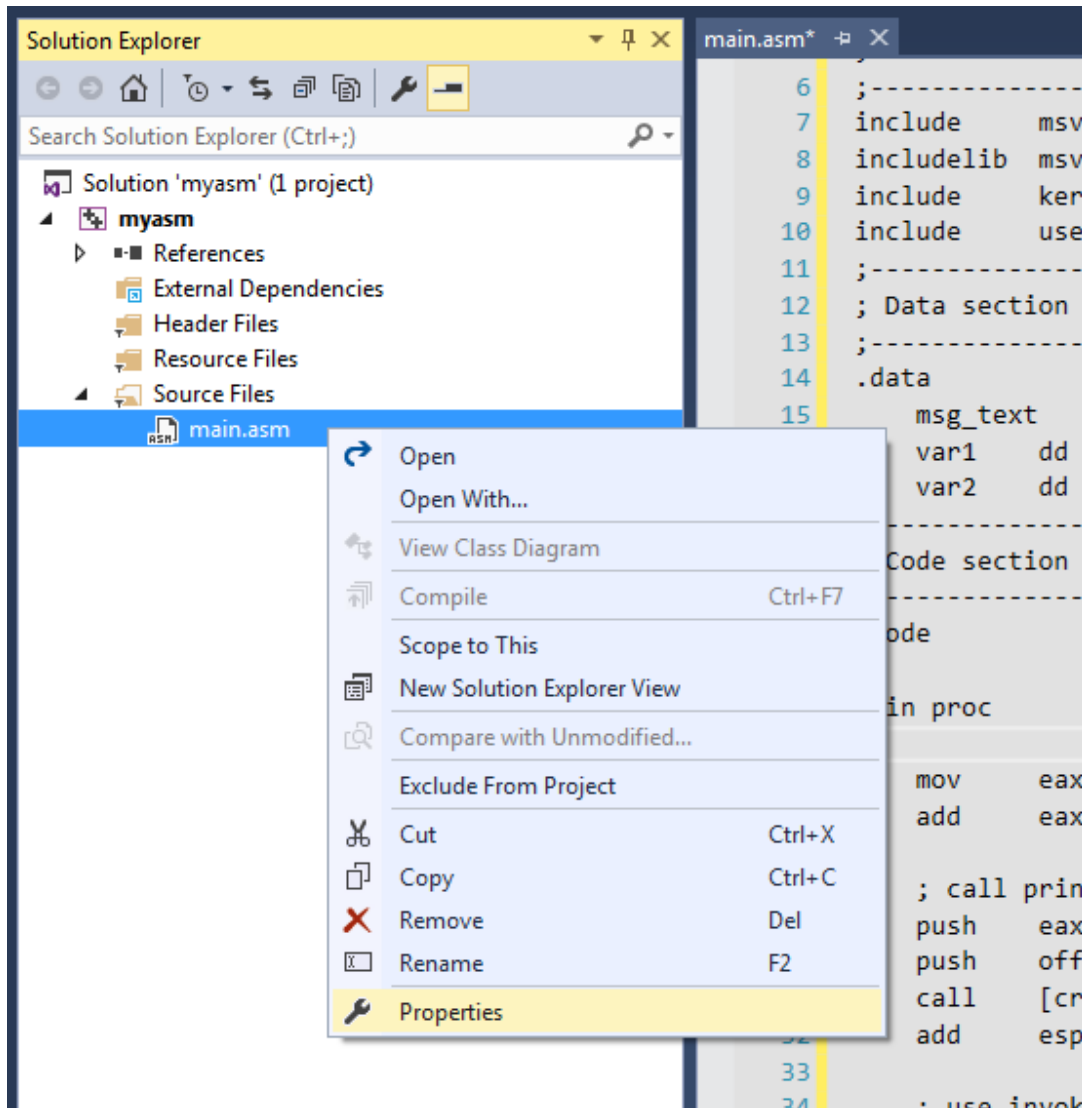
Build Customization



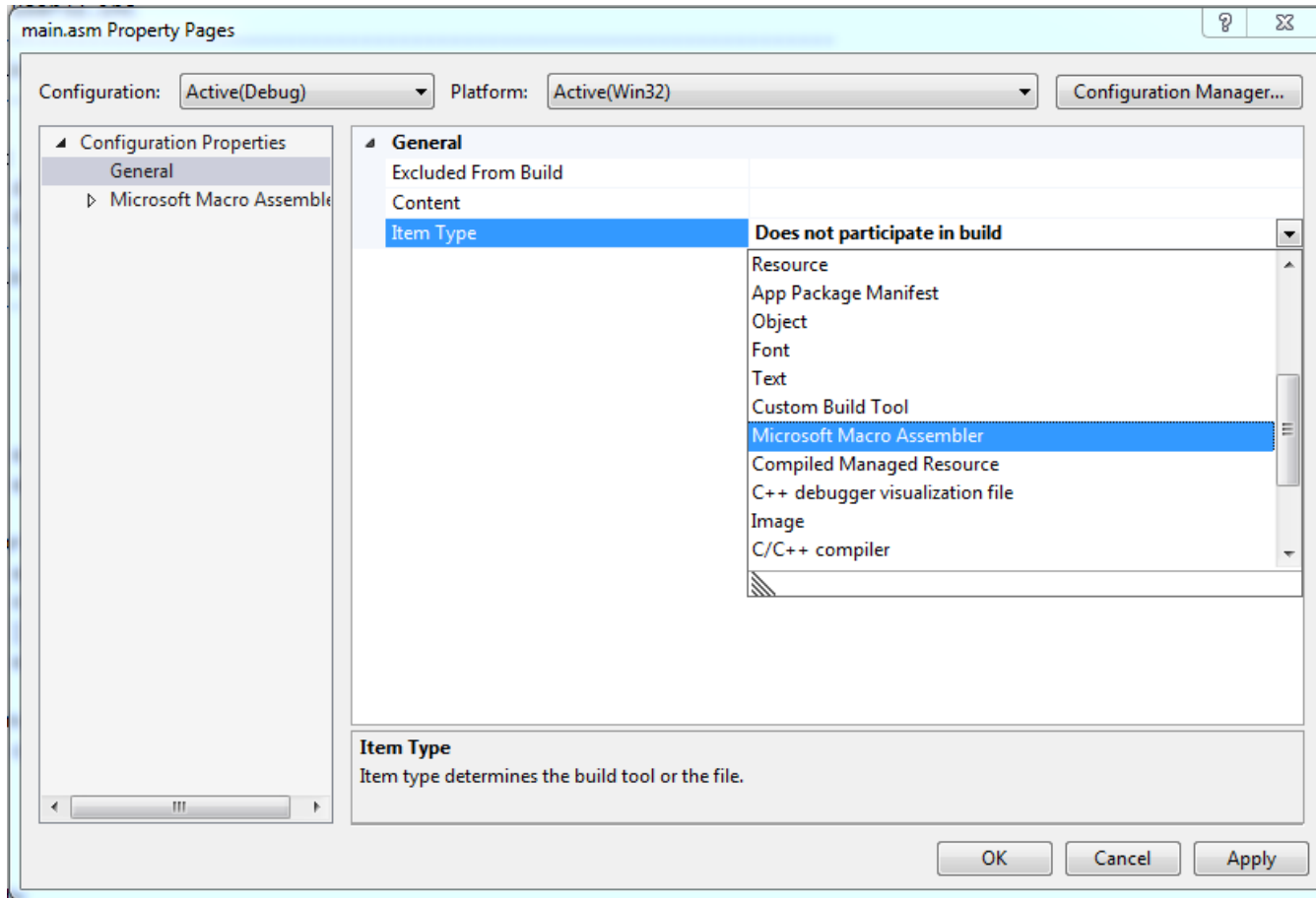
Build target MASM



main.asm properties



Build with MASM



- Copy the files below into project directory
 - msvcrt.inc
 - msvcrt.lib
 - kernel32.inc
 - user32.inc

Debugging

myasm (Debugging) - Microsoft Visual Studio (Administrator)

File Edit View VAssistX Project Build Debug Team Tools Test Analyze Window Help

Process: [0x2AB4] myasm.exe Lifecycle Events Thread: [0x250C] Main Thread

Solution Explorer

Search Solution Explorer (Ctrl+K)

Solution 'myasm' (1 project)

- myasm
 - References
 - External Dependencies
 - Header Files
 - Resource Files
 - Source Files
 - main.asm

main.asm

```
16 var1 dd 1000
17 var2 dd 3000
18 ;-----
19 ; Code section
20 ;-----
21 .code
22
23 main proc
24
25 mov eax, [var1]
26 add eax, [var2]
27
28 ; call printf function manually
29 push eax < 1ms elapsed
30 push offset msg_text
31 call [crt_printf] ; calling a C (cdecl) function
32 add esp, 8 ; caller clean the stack frame
33
34 ; use invoke macro to call printf function
35 invoke crt_printf, offset msg_text, var2
36 ; add esp, 8 // this line will be added by the macro
37
```

Registers

EAX = 0000FA0 EBX = 7EFDE000 ECX = 00000000
EDX = 00DD1005 ESI = 00000000 EDI = 00000000
EIP = 00DD2027 ESP = 0039FCFC EBP = 0039FD04
EFL = 00000214

Memory1

Address: &var1

Address	Value	Comment
0x00DD5018	e8 03 00 00 b8 0b 00 00 00 00	è.....
0x00DD5022	00 00 00 00 00 00 00 00 00 00
0x00DD502C	00 00 00 00 00 00 00 00 00 00
0x00DD5036	00 00 00 00 00 00 00 00 00 00
0x00DD5040	00 00 00 00 00 00 00 00 00 00
0x00DD504A	00 00 00 00 00 00 00 00 00 00
0x00DD5054	00 00 00 00 00 00 00 00 00 00
0x00DD505F	00 00 00 00 00 00 00 00 00 00

Solution Explorer Class View Autos Locals Registers Threads Modules Watch 1 Call Stack Breakpoints Exception Settings Output Memory1

Ready