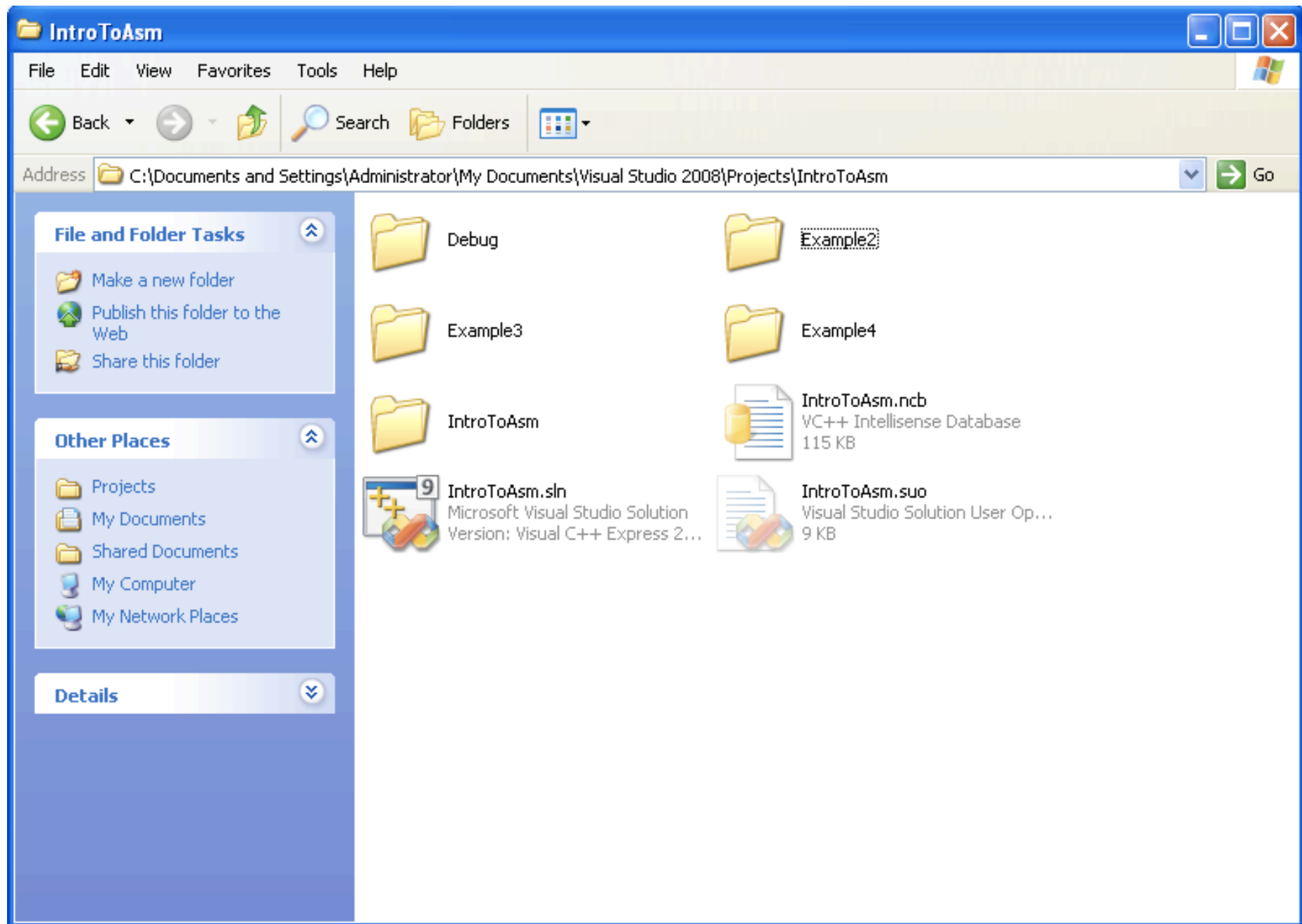
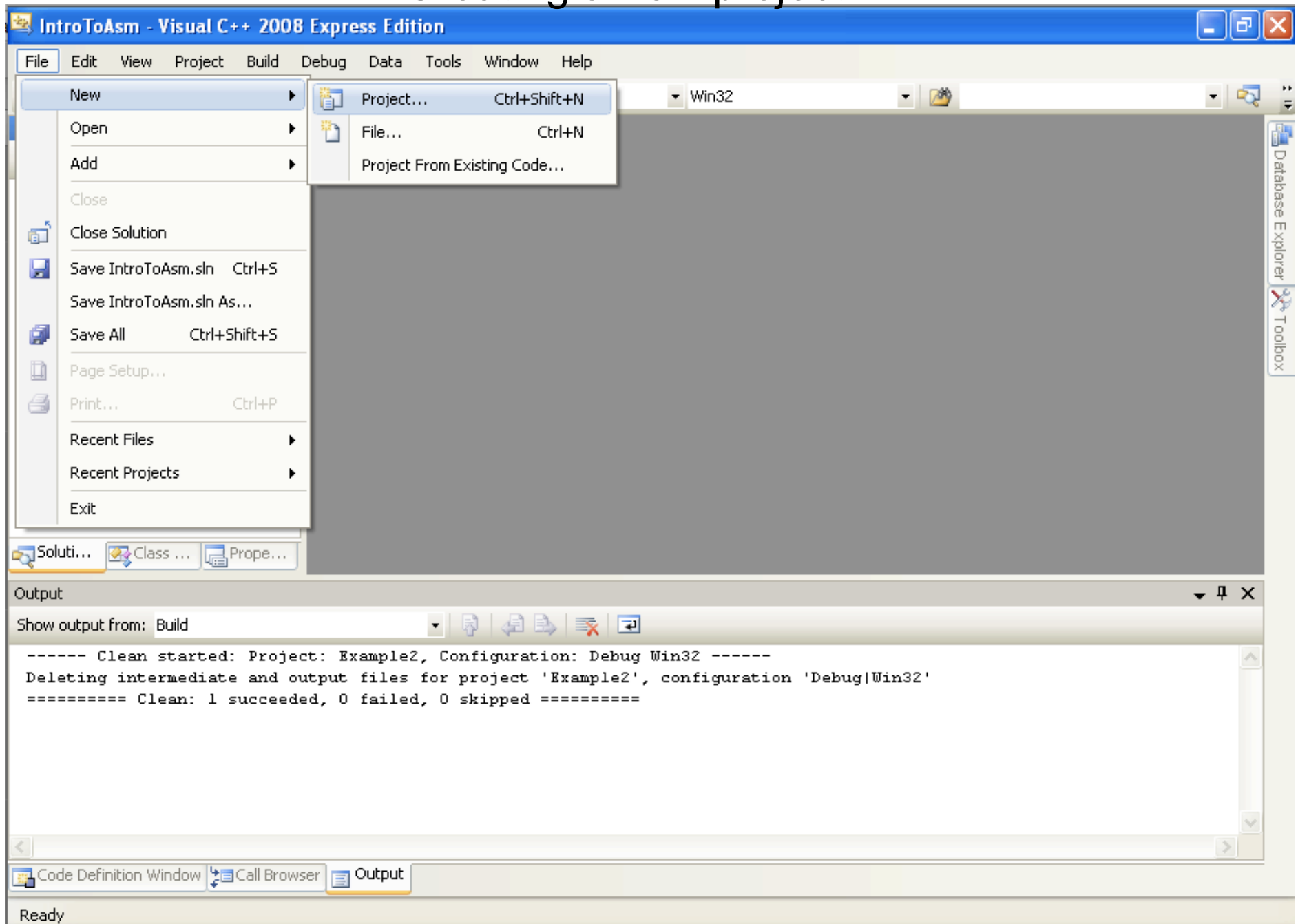


# Let's do that in a tool

- Visual C++ 2008 Express Edition (which I will shorthand as “VisualStudio” or VS)
- Standard Windows development environment
- Available for free, but missing some features that pro developers might want
- Can't move applications to other systems without installing the “redistributable libraries”



# Creating a new project - 1



## Creating a new project - 2

**New Project**

Project types:

- Visual C++
  - CLR
  - Win32
  - General

Templates:

**Visual Studio installed templates**

- Empty Project
- Makefile Project

**My Templates**

- Search Online Templates...

An empty project for creating a local application

Name:

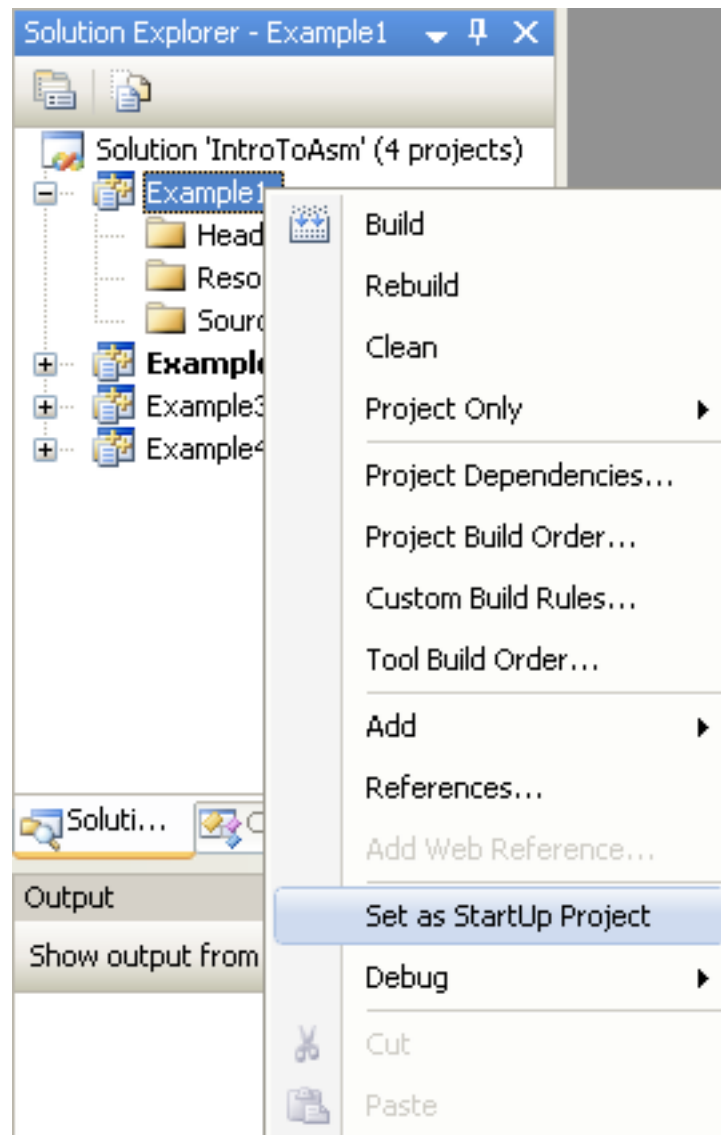
Location:

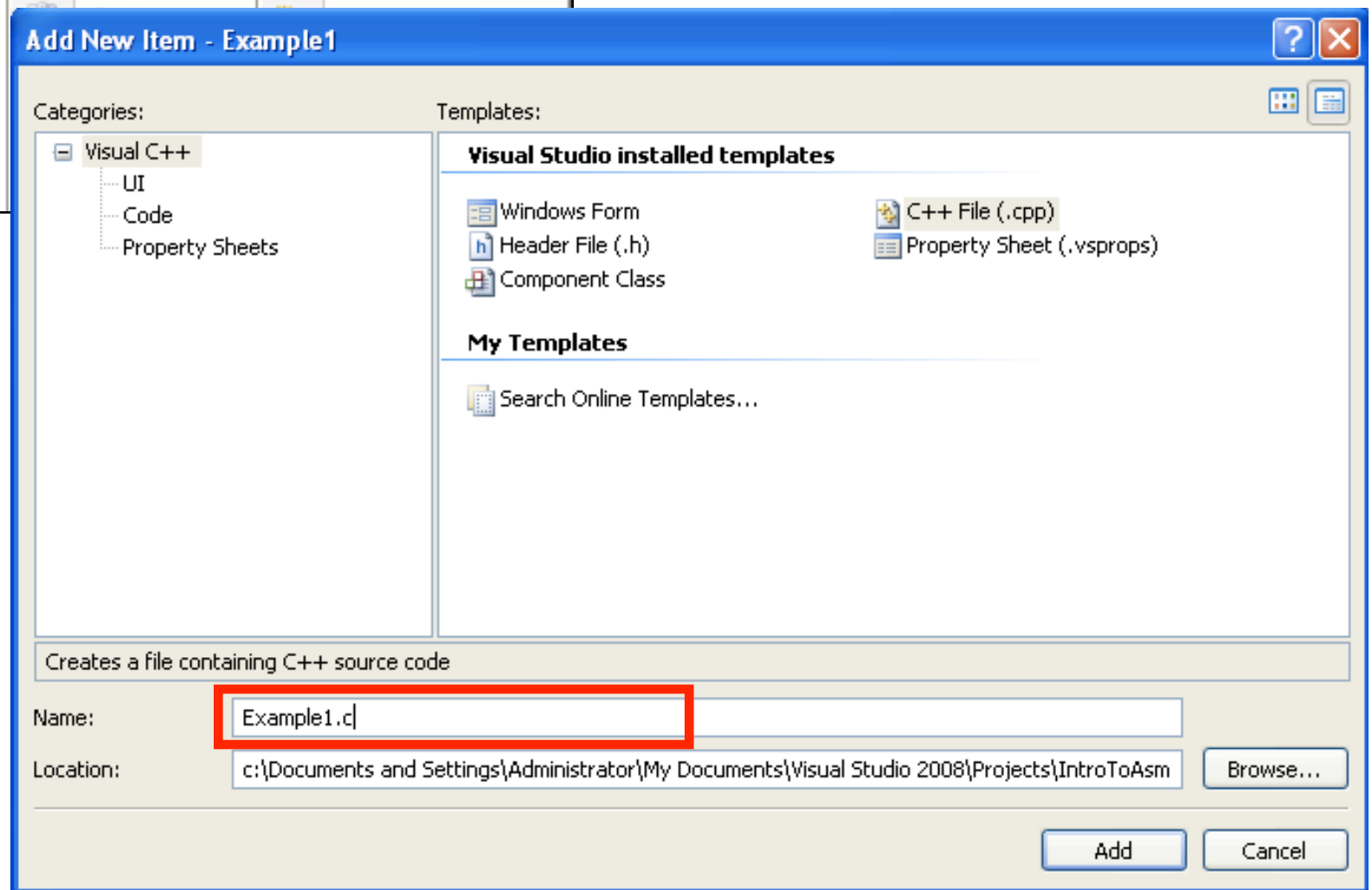
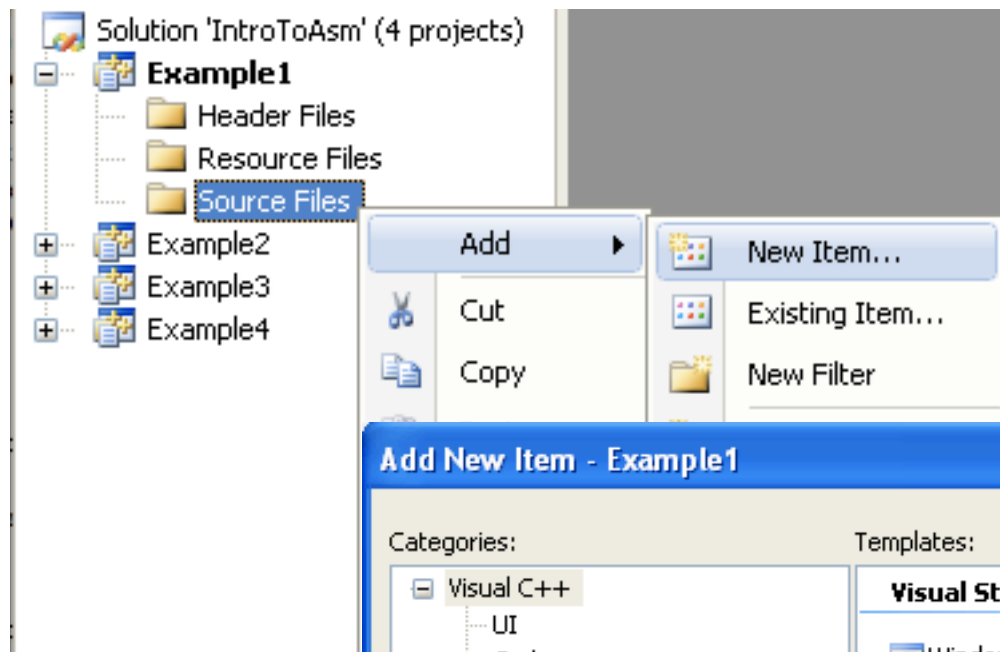
Solution:

☐ Create directory for solution

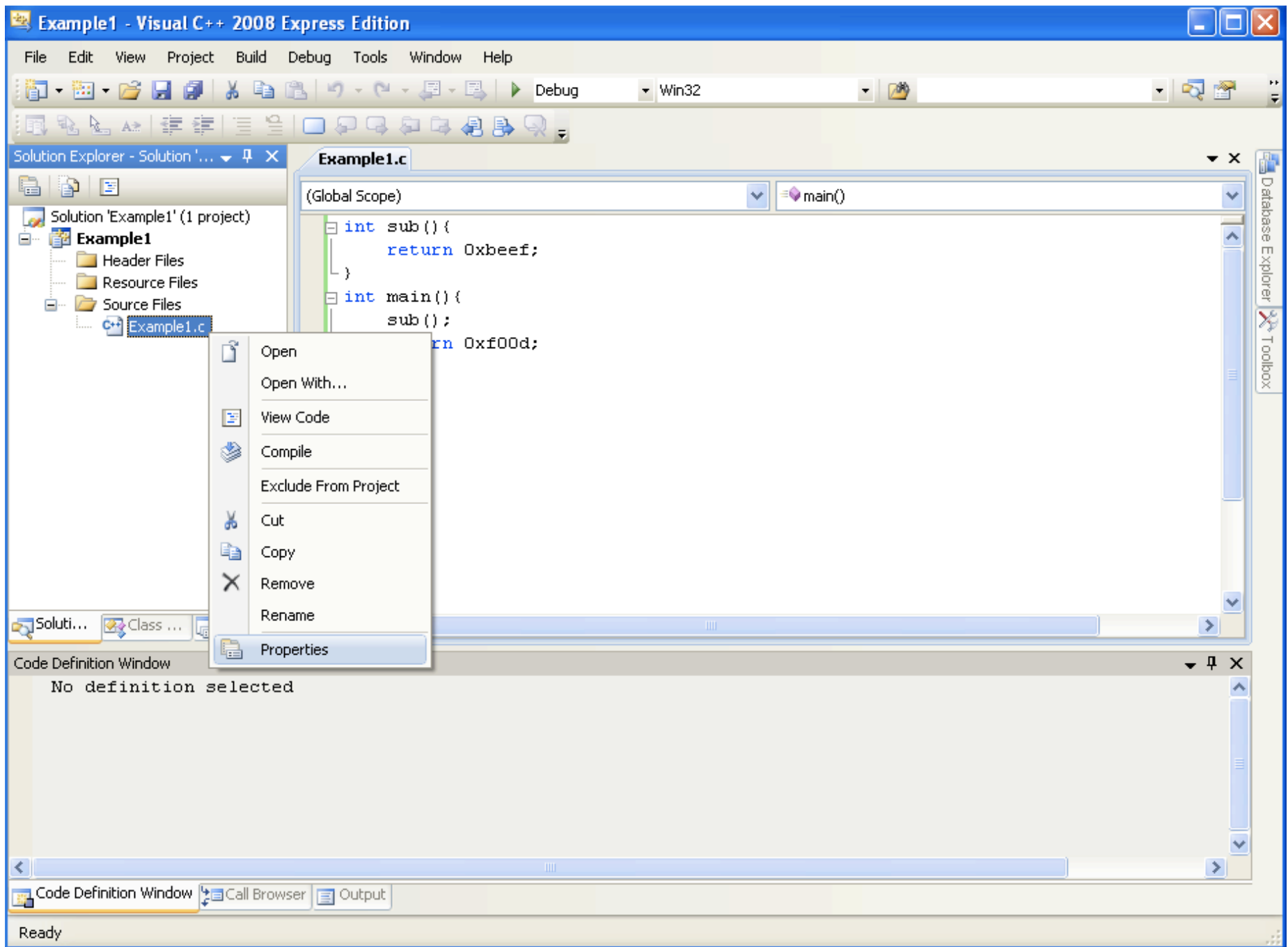
Solution Name:

## Creating a new project - 3

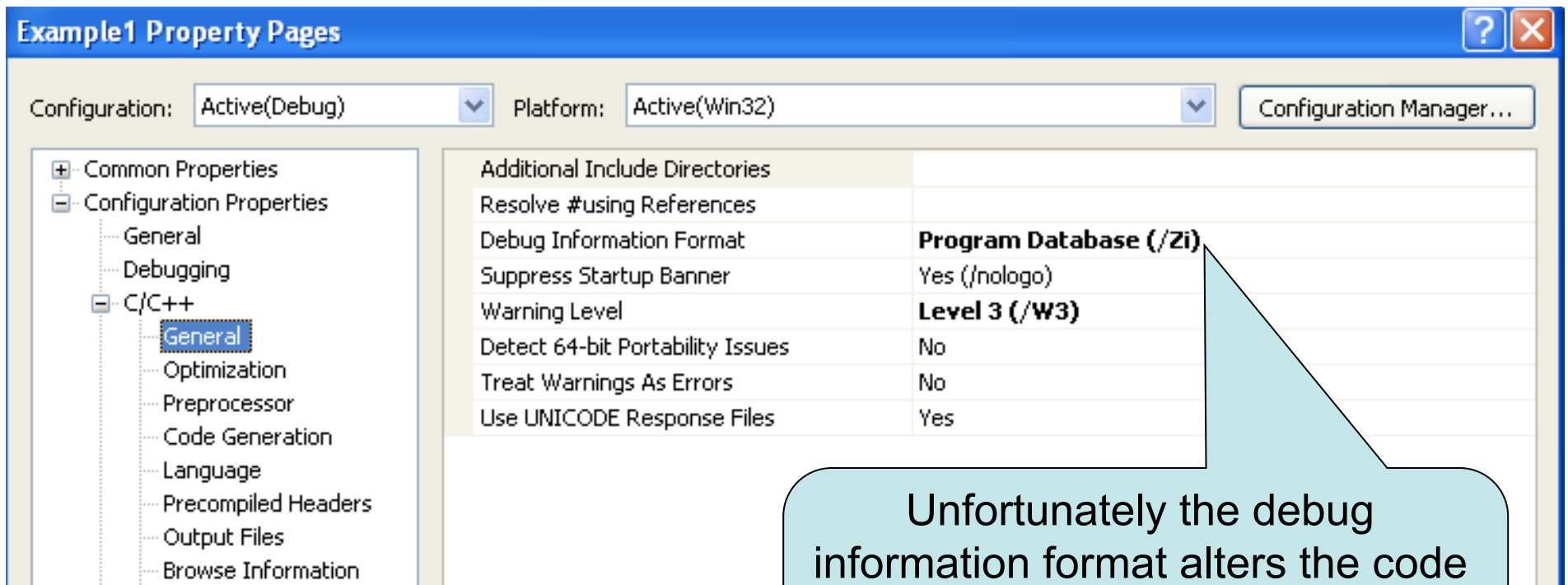




# Setting project properties - 1



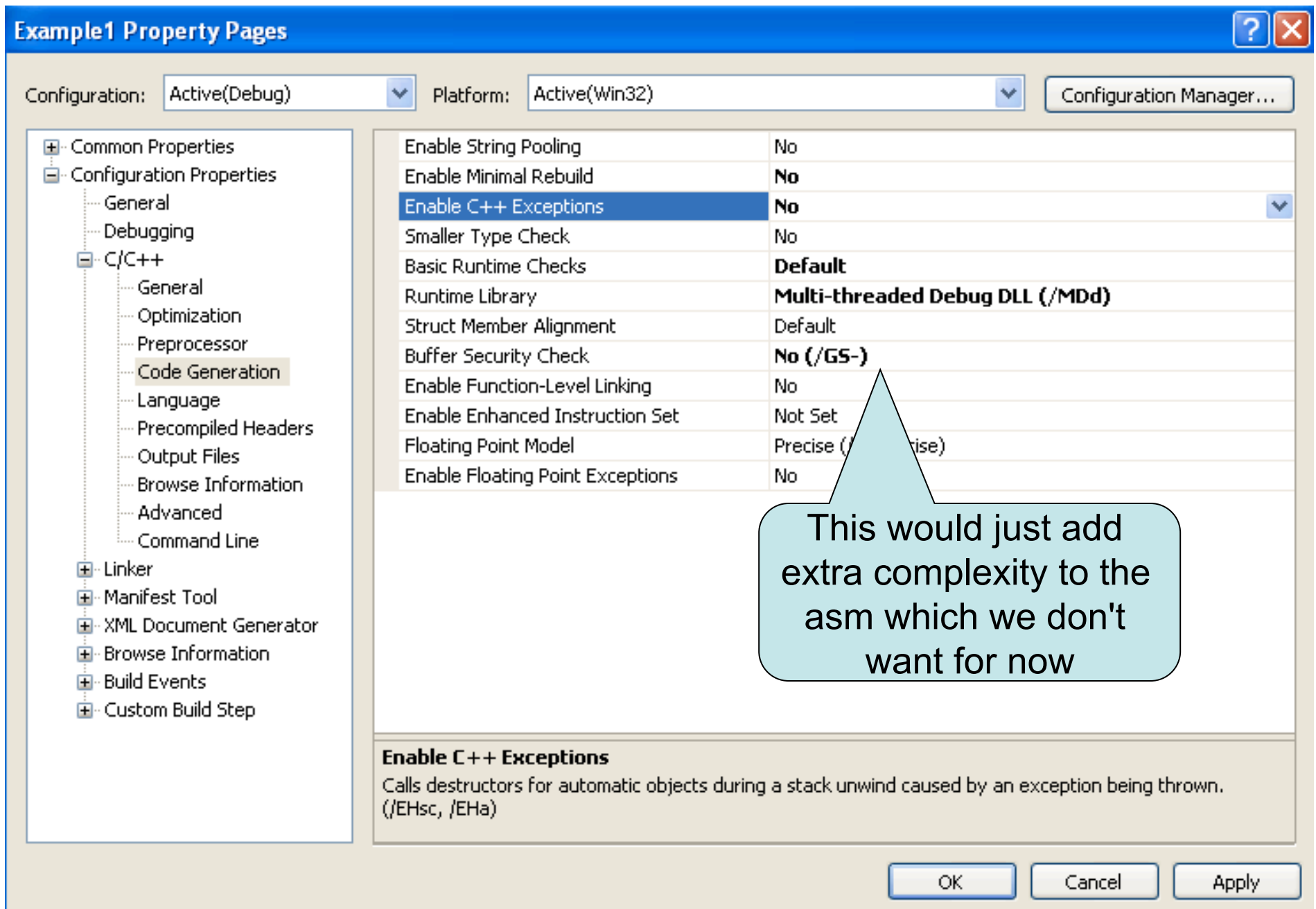
## Setting project properties - 2



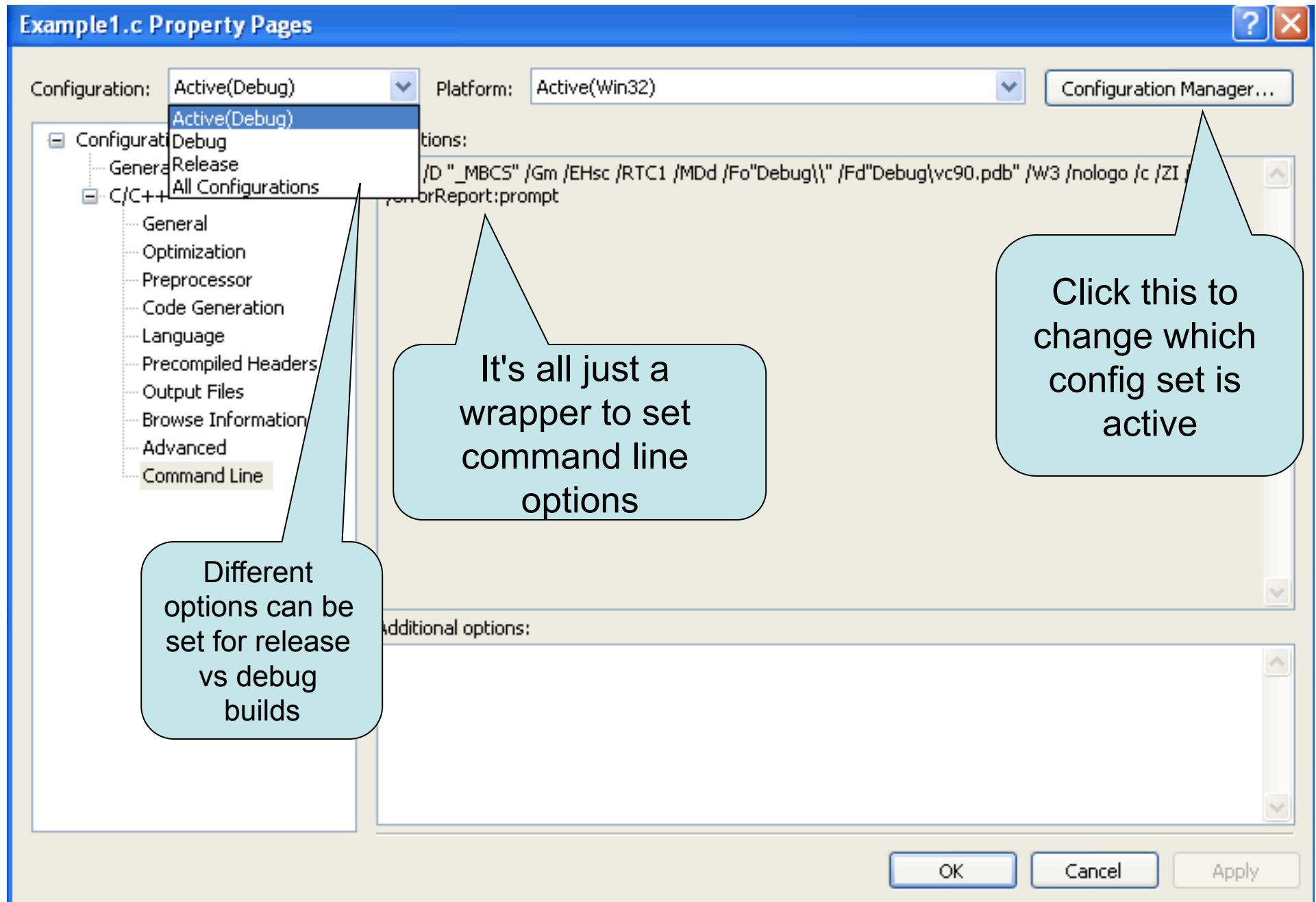
Unfortunately the debug information format alters the code which gets generated too much, making it not as simple as I would like for this class.



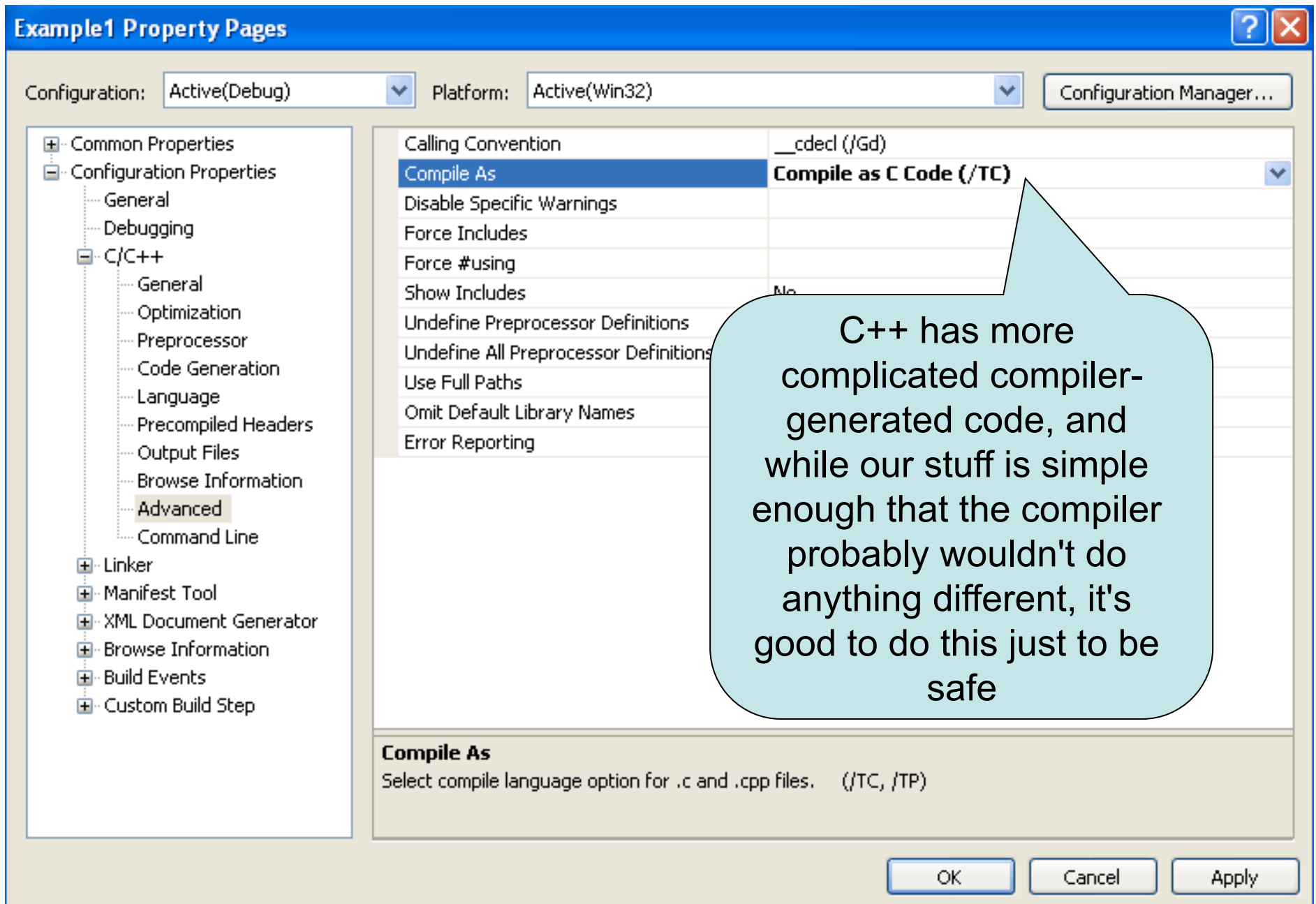
## Setting project properties - 3



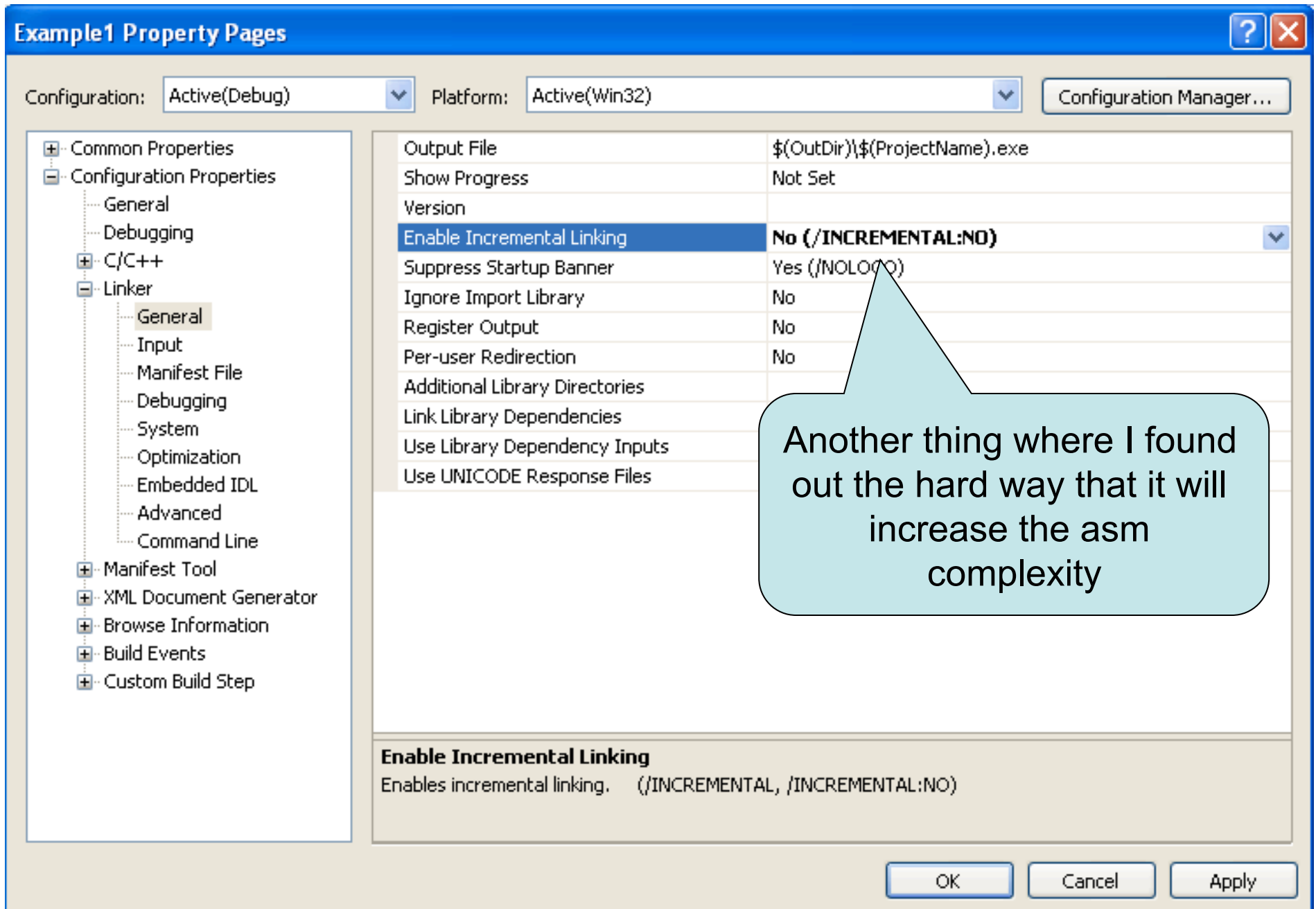
## Setting project properties - 4



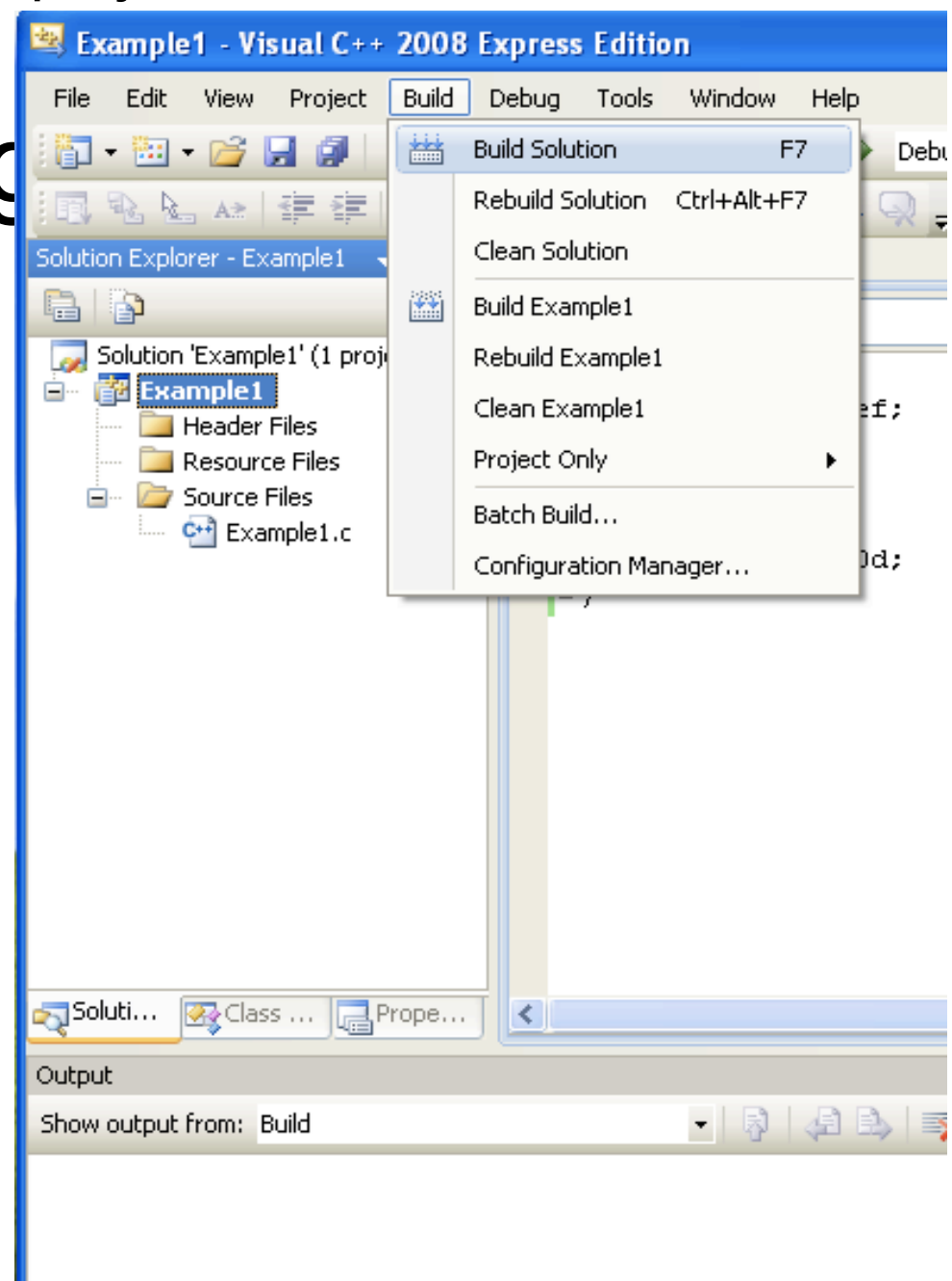
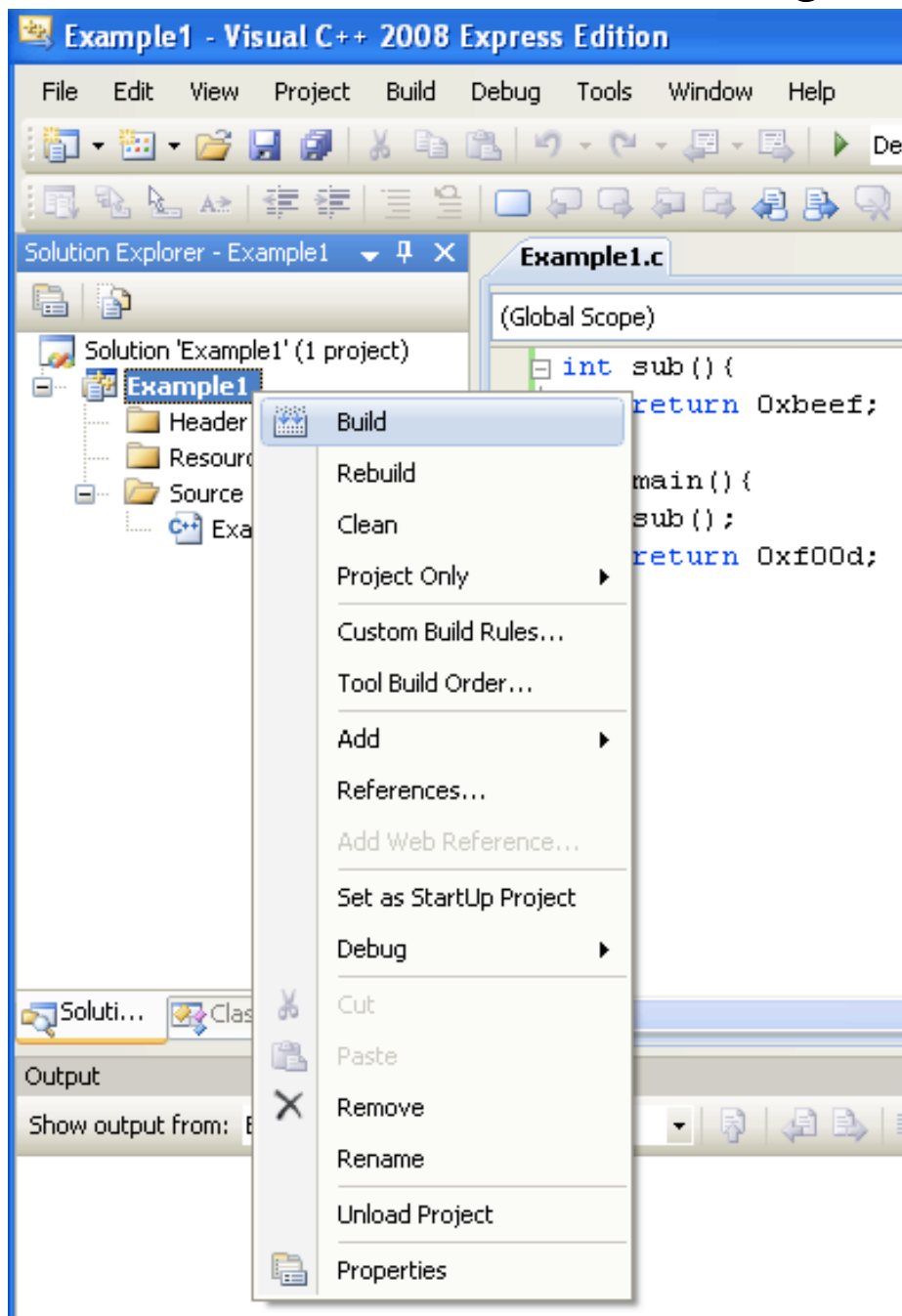
# Setting project properties - 5



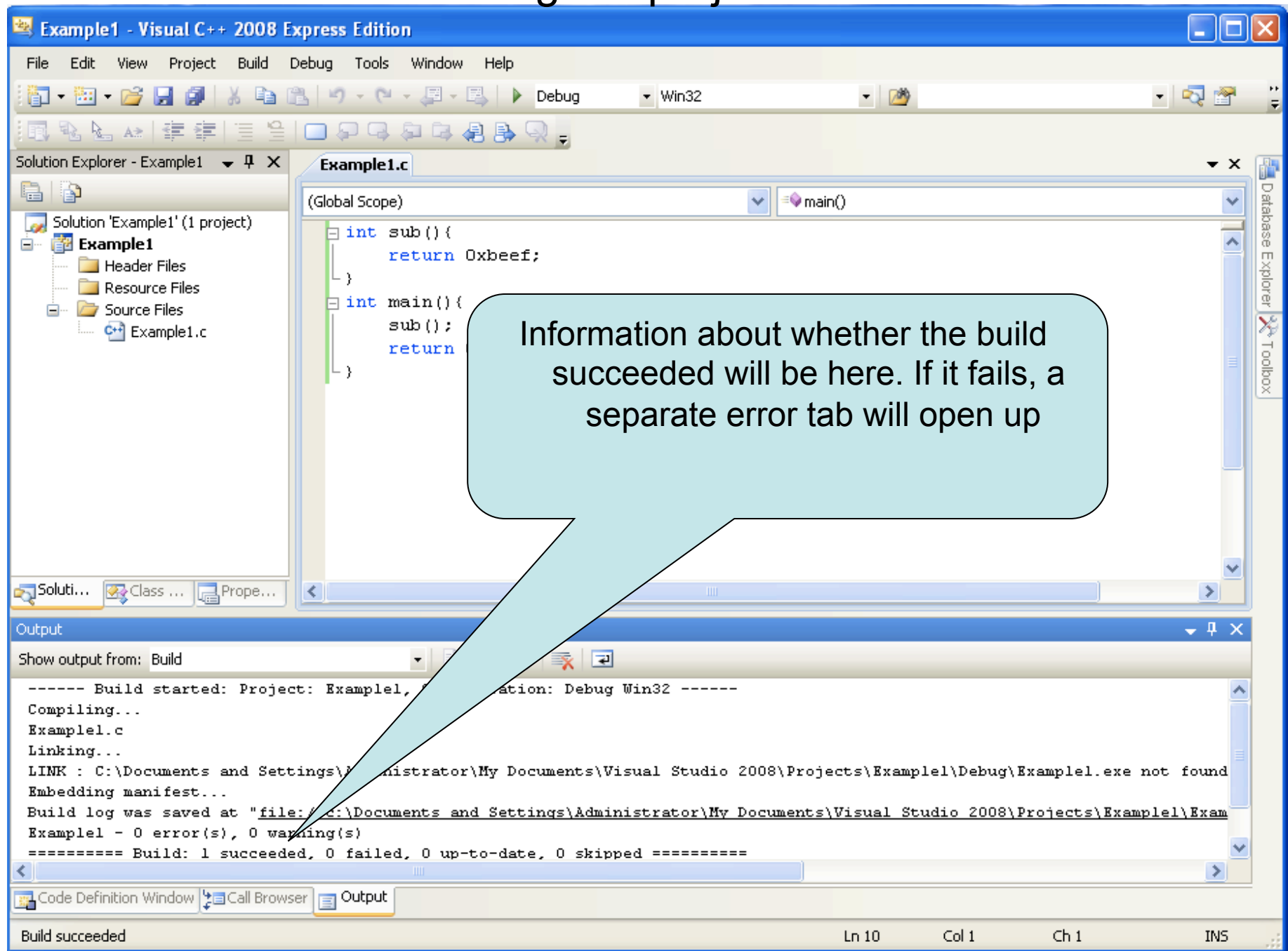
## Setting project properties - 6



# Building the project - 1

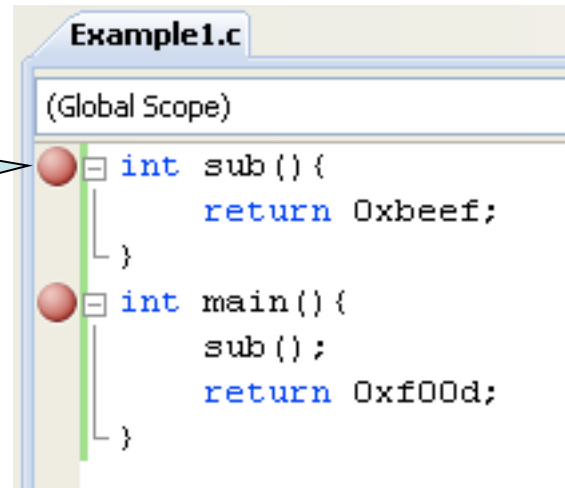


## Building the project - 2



# Setting breakpoints & start debugger

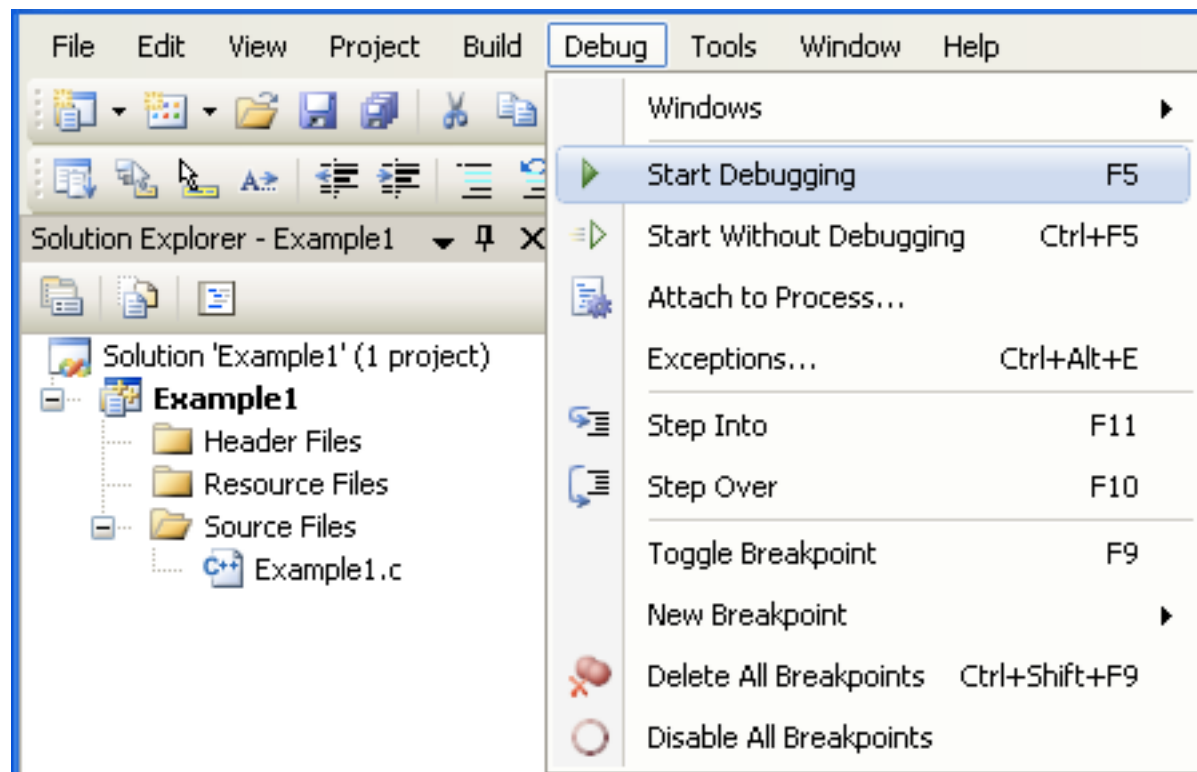
Click to the left of the line to break at.

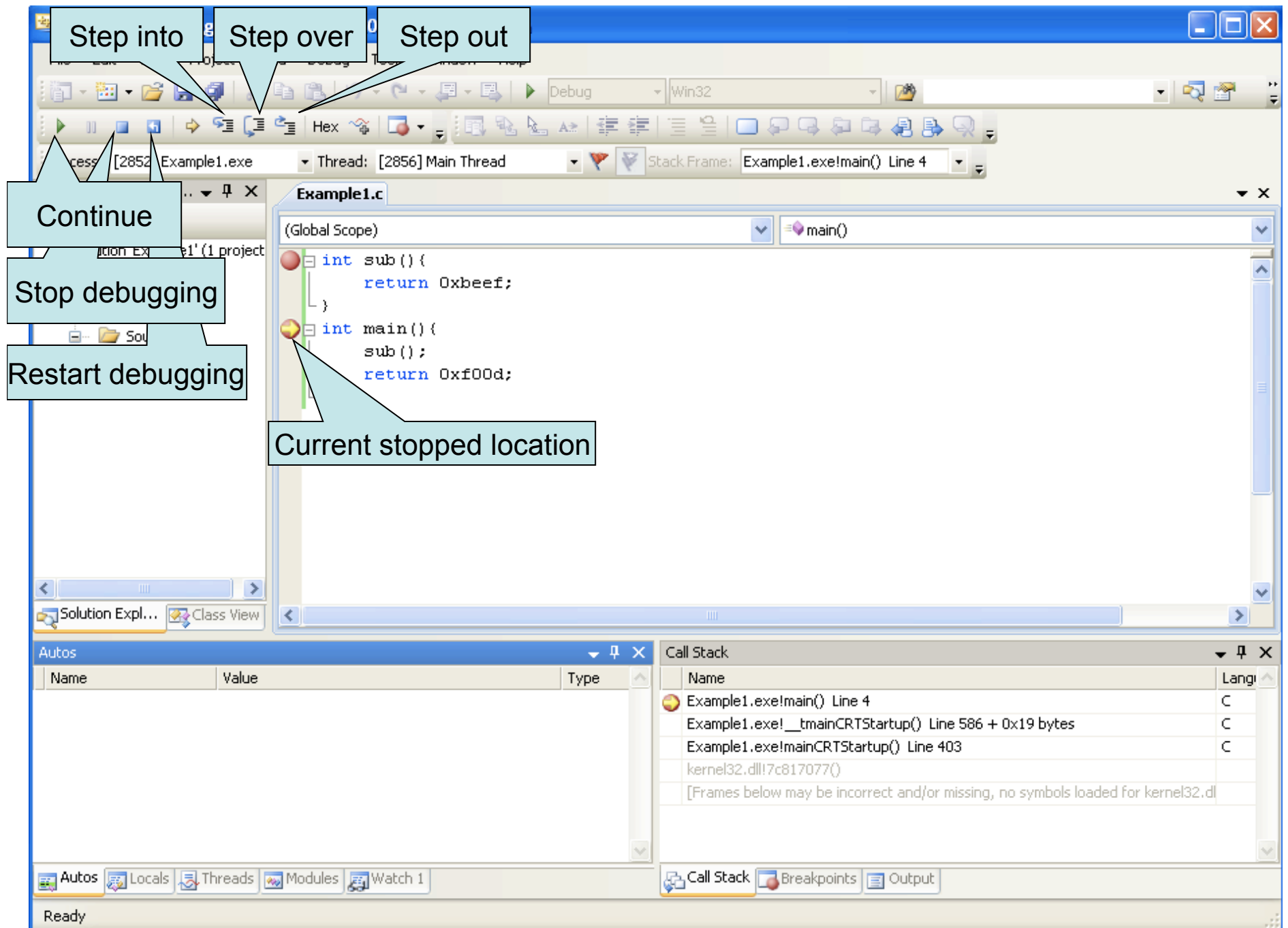


```
Example1.c
(Global Scope)

int sub() {
    return 0xbeef;
}

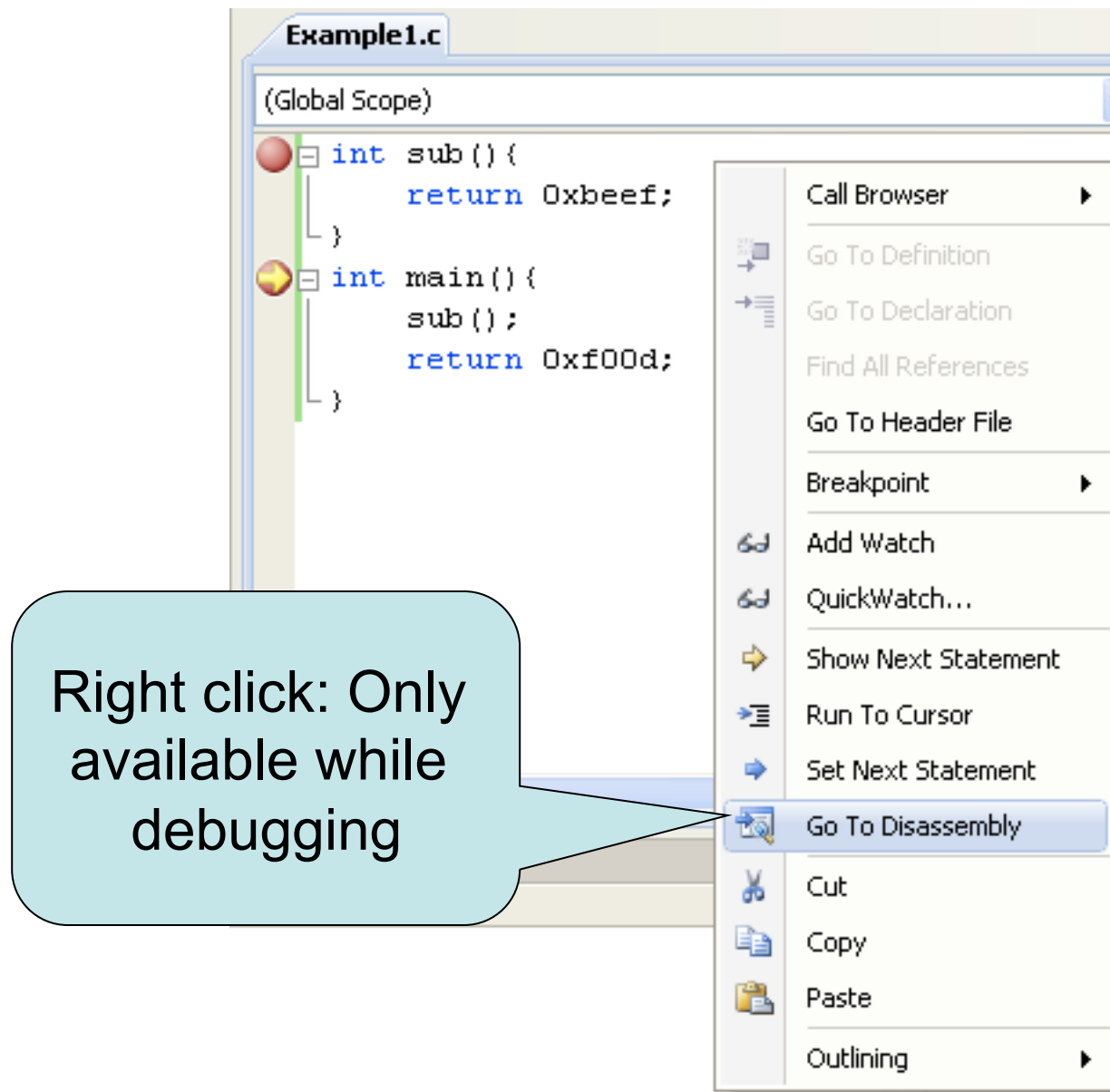
int main() {
    sub();
    return 0xf00d;
}
```







# Showing assembly



IntroToAsm (Debugging) - Visual C++ 2008 Express Edition

File Edit View Project Build Debug Tools Window Help

Debug Win32

Process: [0xF1C] Example1.exe Thread: [0x844] Main Thread Stack Frame: Example1.exe!main() Line 4

Solution Explorer - Ex...

Solution 'IntroToAsm' (4 projects)

- Example1
- Example2
- Example3
- Example4

Disassembly Example1.c

Address: main(void)

```
0040102C int 3
0040102D int 3
0040102E int 3
0040102F int 3
--- c:\documents and settings\administrator\my documents\visual studio 2008\projects\introtoasm
int main(){
00401030 push ebp
00401031 mov ebp,esp
sub();
00401033 call @ILT+0(_sub) (401005h)
return 0xf00d;
00401038 mov eax,0F00Dh
}
0040103D pop ebp
```

Autos

Name	Value	Type
EBP	0012FFB8	

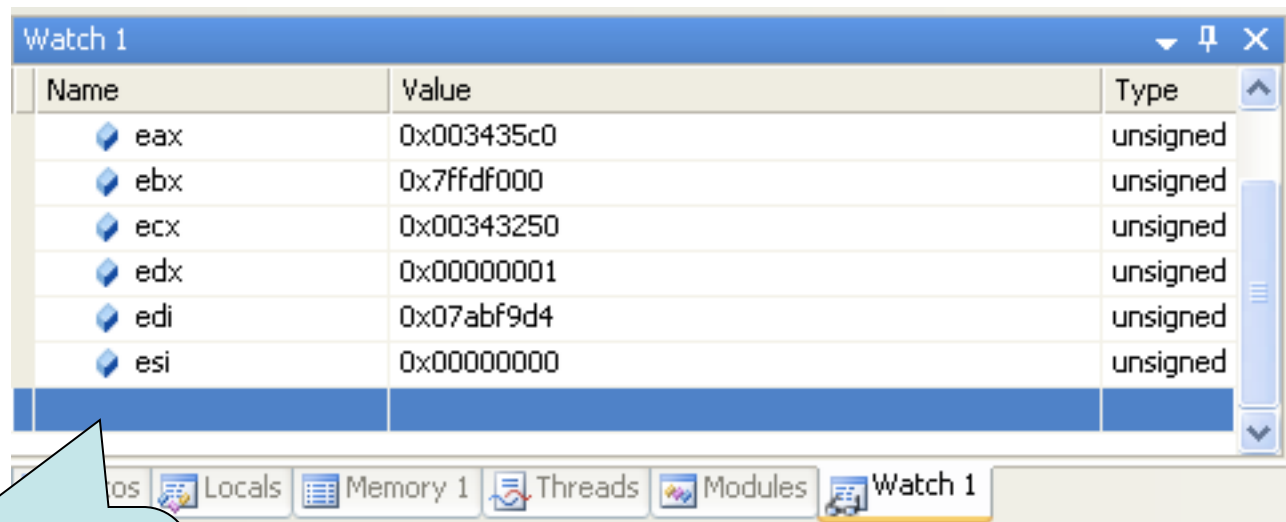
Call Stack

Name	Language
Example1.exe!main() Line 4	C
Example1.exe!__tmainCRTStartup() Line 586 + 0x19 bytes	C
Example1.exe!mainCRTStartup() Line 403	C
kernel32.dll!7c817077()	
[Frames below may be incorrect and/or missing, no symbols loaded for k...]	

Note that it knows the ebp register is going to be used in this instruction

Ready

# Showing registers

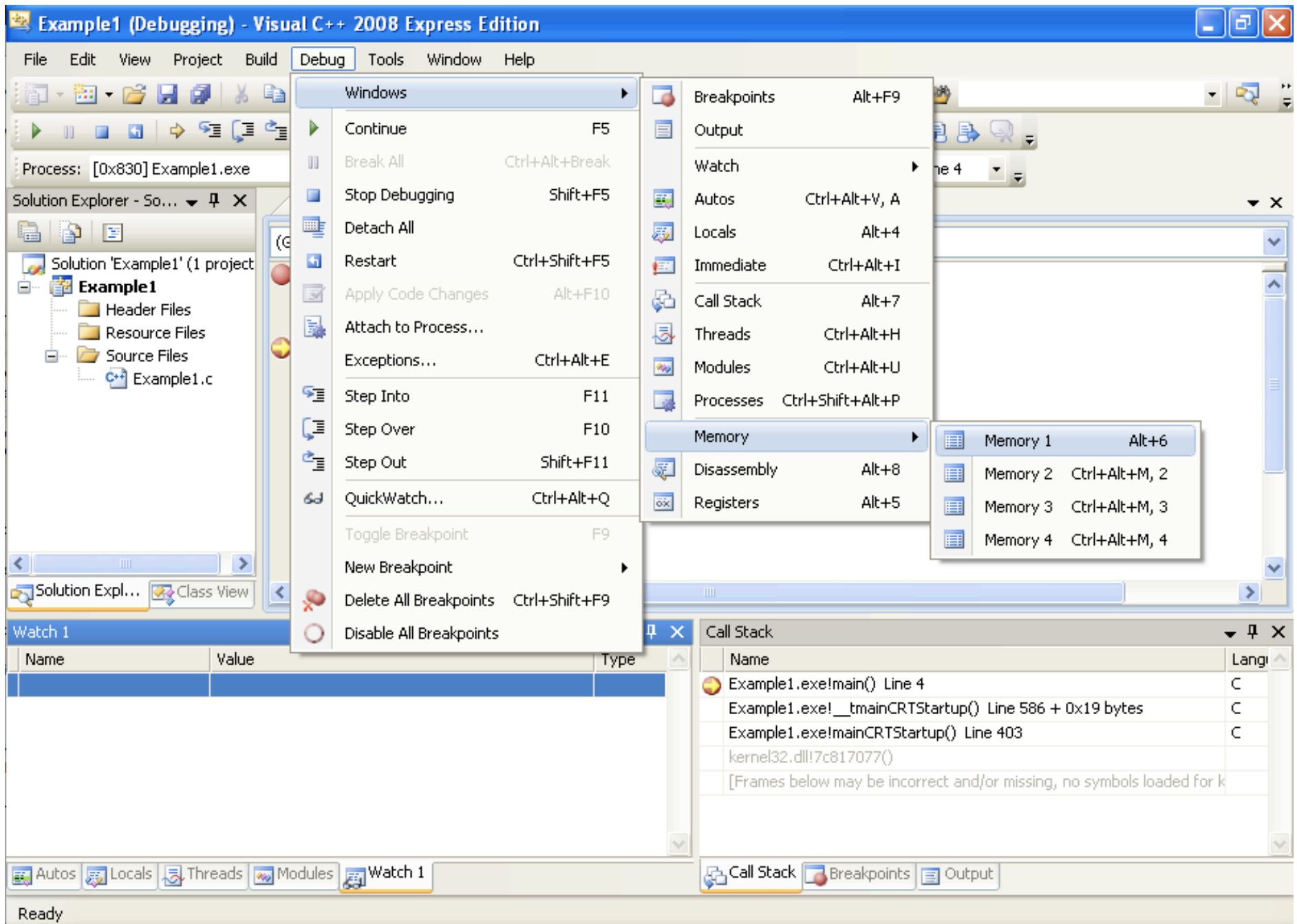


Name	Value	Type
eax	0x003435c0	unsigned
ebx	0x7ffdf000	unsigned
ecx	0x00343250	unsigned
edx	0x00000001	unsigned
edi	0x07abf9d4	unsigned
esi	0x00000000	unsigned

os Locals Memory 1 Threads Modules Watch 1

Here you can  
enter register  
names or variable  
names

# Watching the stack change - 1



## Watching the stack change - 2

Set address to esp (will always be the top of the stack)

Right click on the body of the data in the window and make sure everything's set like this

Set to 1

Click "Reevaluate Automatically" so that it will change the display as esp changes

Memory 1

Address: esp

0x0012FF64	0
0x0012FF68	0
0x0012FF6C	0
0x0012FF70	0
0x0012FF74	00343250 P24.
0x0012FF78	00343690 .64.
0x0012FF7C	2dc16b40 @kÁ-

Memory 1 Memory 2

Columns: 1

86

