

Fejlesztés Visual Studio-val

- **Irvine project betöltése**
 - `Irvine\Examples\Project_sample`
 - `Project.sln` megnyitása
- **Fordítás, futtatás**
 - *Build / Build Solution* (F7)
 - `Debug\Project.exe` fájl áll elő
 - *Debug / Start Debugging* (F5)
 - *Debug / Start Without Debugging* (Ctrl + F5)

Fejlesztés Visual Studio-val

- **Más forráskód fordítása**
 - Másoljuk a forráskódot a **Project** könyvtárba
 - *Solution Explorer* panelen távolítsuk el az előző forráskódot
 - Kiválasztás, majd *Delete* gomb és *Remove*
 - Új forráskód hozzáadása
 - **Project** kiválasztása a *Solution Explorer* listában
 - *Jobb egérgomb / Add / Existing Item*

Fejlesztés Visual Studio-val

- **Nyomkövetés**

- Töréspont elhelyezése és törlése

- Bal egérgomb kattintás a sor előtti szürke részen vagy **F9** lenyomása a kérdéses sorban
 - Egy vörös kör jelenik meg

- Indítás

- Program indítása (**F5**), nyomkövetés vége (**Shift + F5**)
 - A töréspontnál megáll

- Továbblépés

- Folytatás **F5**, Step Into **F11**, Step Over **F10**

Fejlesztés Visual Studio-val

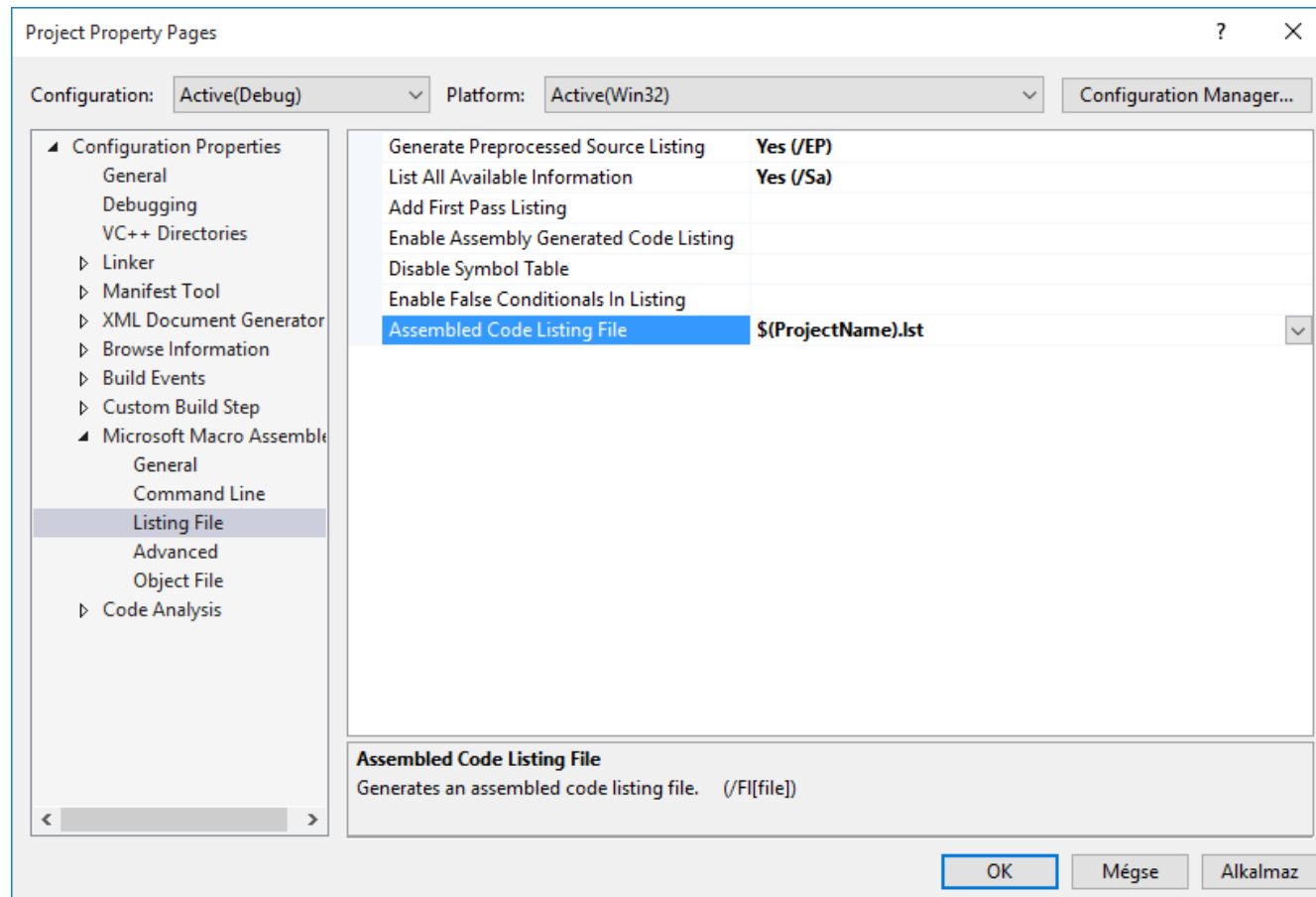
- **Nyomkövetés információk**
 - Nyomkövetés közben érhetőek el
 - *Debug / Windows* menü elemei
 - Regiszter értékek (*Debug / Windows / Registers*)
 - vörös színnel az aktuális utasításban megváltozottak
 - FLAG bitek (*Debug / Windows / Flags*)
 - Kifejezések értékei (*Watch*), ...
 - *Disassembly* kódlista
 - Nagyon hasznos eljárásívási direktívák kifejtésének vizsgálatakor
 - ...

Fejlesztés Visual Studio-val

- **Részletes fordítási lista**
 - Ha beállítjuk, a fordító kész egy lista fájlt, amiben elérhető
 - az összes definíció, makró
 - a lefordított tárgykód,
 - kigyűjtött címkék,
 - ...

Fejlesztés Visual Studio-val

- Részletes fordítási lista beállítása



Nyomkövetés (VS 2015)

The screenshot displays the Microsoft Visual Studio Express 2015 for Windows Desktop interface during a debugging session. The main window shows the disassembly of the assembly file `HelloAsm.asm`. The code includes a title, a data section, and a code section with a `main` procedure. The `main` procedure is currently executing, and the instruction `CALL WriteString` is highlighted, indicating the current execution point. The `Registers` window shows the current state of the CPU registers, and the `Call Stack` window shows the current call stack frame.

Disassembly:

```
TITLE Hello, Assembly! (32 bites)
INCLUDE Irvine32.inc

.data
HelloTxt BYTE "Hello, Assembly!", 0ah, 0
A        DWORD 5

.code
; Belépési pont
main PROC
MOV     EDX,OFFSET HelloTxt
CALL   WriteString    ;2ms elapsed

MOV     EAX,7
ADD     EAX,A
CALL   WriteDec

INVOKE ExitProcess,0
main ENDP

END main
```

Registers:

```
EAX = 31409870 EBX = 7FFDE000 ECX = 00401055 EDX = 00406000
ESI = 00401055 EDI = 00401055 EIP = 004035E5 ESP = 0019FF84
EBP = 0019FF94 EFL = 00000246

OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0
```

Call Stack:

Name	Lang
Project.exe!main() Line 12	Unkr
[External Code]	
[Frames below may be incorrect and/or missing, no symbols loaded for ntd]	

Registers:

Register	Value
EAX	31409870
EBX	7FFDE000
ECX	00401055
EDX	00406000
ESI	00401055
EDI	00401055
EIP	004035E5
ESP	0019FF84
EBP	0019FF94
EFL	00000246

Nyomkövetés

The screenshot displays the Microsoft Visual Studio Express 2015 for Windows Desktop interface during a debugging session. The main window shows the disassembly of the assembly file `HelloAsm_masm32.asm`. The assembly code includes a data section for `HelloTxt` and a code section for the `main` procedure. The `main` procedure starts with `MOV EDX, OFFSET HelloTxt`, followed by `INVOKE GetStdHandle, STD_OUTPUT_HANDLE`, `INVOKE WriteConsole, EAX, ADDR HelloTxt, SIZEOF HelloTxt, 0, 0`, `MOV EAX, 7`, `ADD EAX, A`, and `INVOKE ExitProcess, 0`.

A **Memory 1** window is open, showing a memory dump starting at address `0x00404000`. The first row of the dump shows the string `Hello, Assembly!.....` at address `0x00404000`.

The **Registers** window shows the following register values:

- EAX = D6D2CDF6
- EBX = 7FFDE000
- ECX = 00401005
- EDX = 00404000
- ESI = 00401005
- EDI = 00401005
- EIP = 00401015
- ESP = 0019FF84
- EBP = 0019FF94
- EFL = 00000246

The **Call Stack** window shows the current call stack frame:

- Project.exe!main() Line 21
- [External Code]
- [Frames below may be incorrect and/or missing, no symbols loaded for ntd]

The **Output** window at the bottom shows the text `Hello, Assembly!` printed to the console.

Disassembly (VS 2015)

Project (Debugging) - Microsoft Visual Studio Express 2015 for Windows Desktop

File Edit View Project Build Debug Team Tools Test Window Help

Process: [8256] Project.exe Lifecycle Events Thread: [8260] Main Thread Stack Frame: main

Disassembly - HelloAsm.asm

Address: main(void)

Viewing Options

```
004035DE int 3
004035DF int 3
--- c:\Irvine\Examples\Project_sample\HelloAsm.asm -----
MOV     EDX,OFFSET HelloTxt
004035E0 mov     edx,406000h
CALL   WriteString
004035E5 call   _WriteString@0 (04010BEh)

MOV     EAX,7
004035EA mov     eax,7
ADD     EAX,A
004035EF add     eax,dword ptr ds:[406012h]
CALL   WriteDec
004035F5 call   _WriteDec@0 (04010DC h)

INVOKE ExitProcess,0
004035FA push   0
004035FC call   _ExitProcess@4 (0403536h)
--- No source file ---
00403601 int 3
00403602 int 3
```

Registers

EAX = 31409870 EBX = 7FFDE000 ECX = 00401055 EDX = 00406000
ESI = 00401055 EDI = 00401055 EIP = 004035E5 ESP = 0019FF84
EBP = 0019FF94 EFL = 00000246

OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0

Call Stack

Name	Lang
Project.exe!main() Line 12	Unkr
[External Code]	
[Frames below may be incorrect and/or missing, no symbols loaded for ntd]	

Solution Explorer

Search Solution Explorer (Ctrl+E)

Solution 'Project' (1 project)

- Project
 - External Dependencies
 - References
 - HelloAsm.asm

Output Locals Watch 1 Registers Call Stack Exception Settings Immediate Window

Ready