

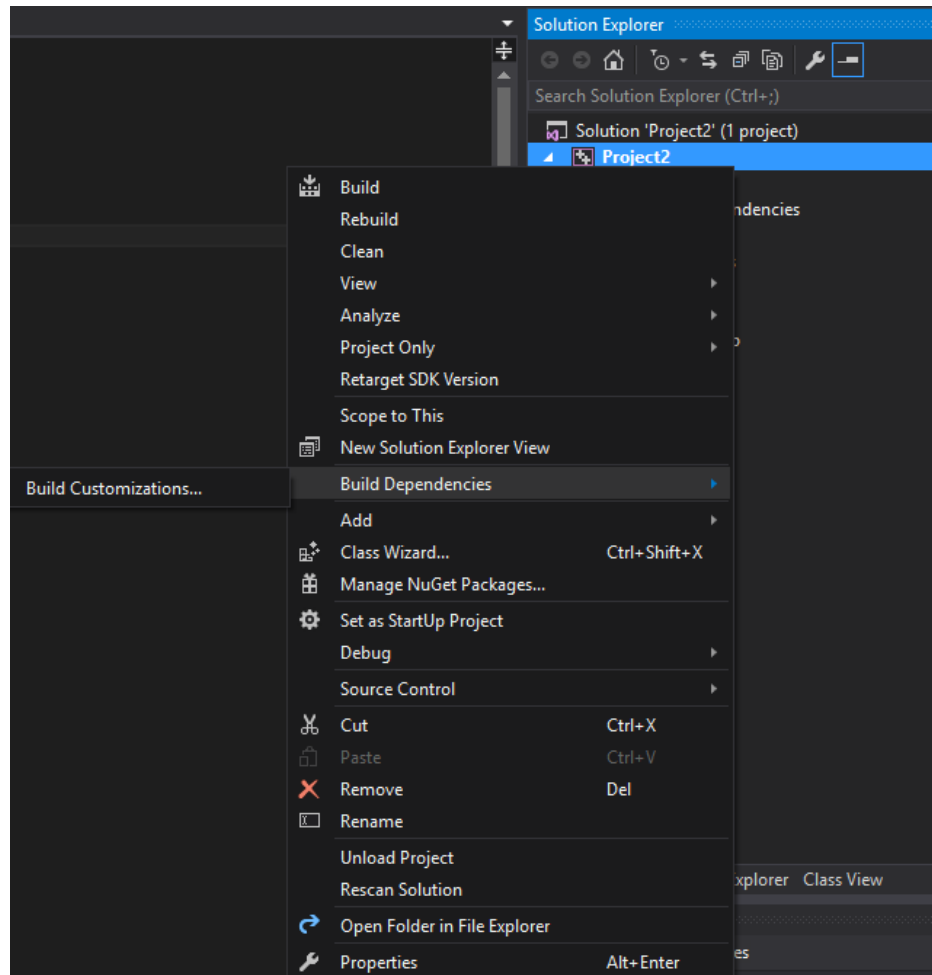
Politechnika Częstochowska
Instytut Inteligentnych Systemów Informatycznych

Laboratorium
Programowanie Niskopoziomowe

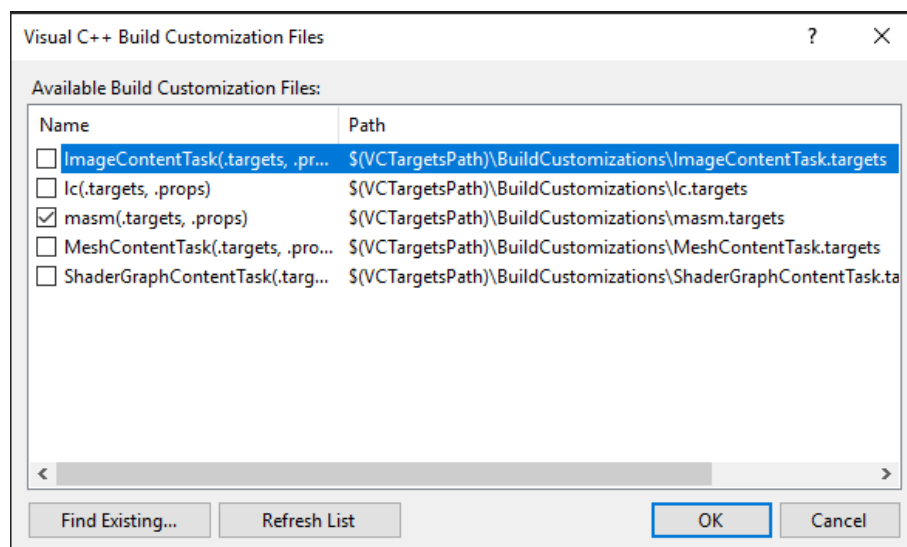
Instrukcja 2b

1. Przygotowanie projektu 64-bitowego

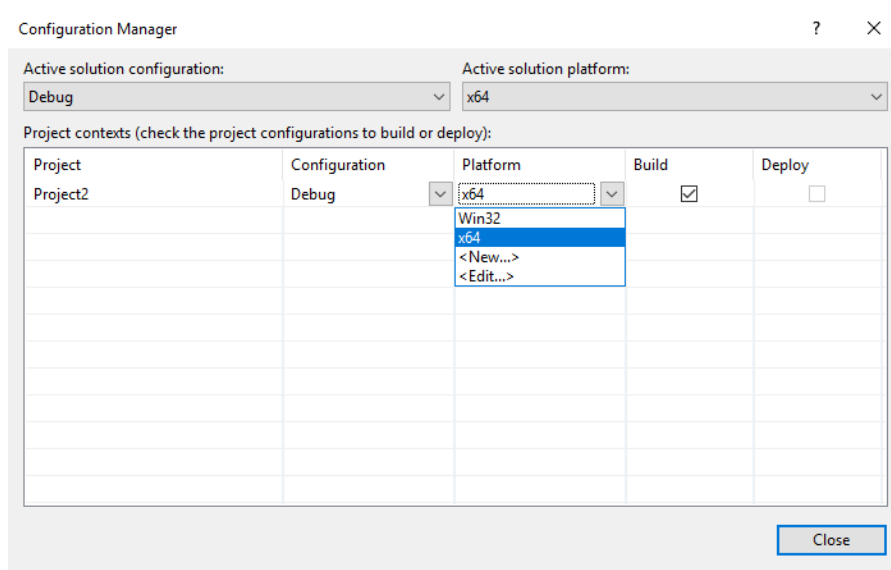
Kliknij prawym przyciskiem myszy nazwę twojego projektu i wybierz "Build Dependencies", a następnie "Build Customizations..."



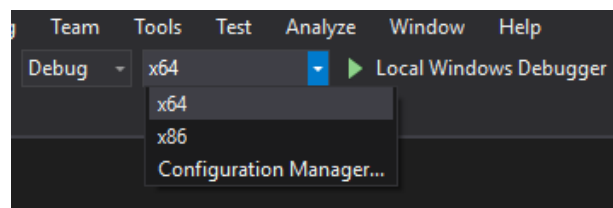
Ustaw obsługę plików MASM, zaznaczając opcję "masm (.targets, .props)":



Wybierz konfigurację "x64" z okna dialogowego "Build / Configuration Manager":



Lub z menu



2. Pisanie kodu w assemblerze x64

Poniższy szablon pozwala umieszczać w pliku .asm dane i kod w odpowiednich segmentach:

_DATA SEGMENT

_DATA ENDS

_TEXT SEGMENT

_TEXT ENDS

END

Dane do odczytu / zapisu definiuje się w segmencie _DATA. Dane tylko do odczytu i kod programu w segmencie _TEXT.

3. Przykłady

3.1. Suma dwóch zmiennych

Kod w assemblerze:

```
_DATA SEGMENT
_DATA ENDS

_TEXT SEGMENT

PUBLIC suma
suma PROC

    movsxd    rax, ecx
    movsxd    rdx, edx
    add       rax, rdx
    ret
suma ENDP

_TEXT ENDS
END
```

Kod w C++:

```
#include <iostream>

using namespace std;

// deklaracja funkcji zewnętrznej
extern "C" __int64 suma(int a, int b);

int main(int argc, char * argv[])
{
    int a = 10;
    int b = 20;
    int c;
    c = suma(a, b);
    cout << "Suma=" << c << endl;
    system("PAUSE");
    return 0;
}
```

3.2. Okno komunikatu

Kod w assemblerze:

```
_DATA SEGMENT
_DATA ENDS

_TEXT SEGMENT
hello_msg db "Hello world!", 0 ;dekl. danych - zestawu bajtów
info_msg db "Info", 0 ;łańcuchy znaków zakończ zerem

EXTERN MessageBoxA: PROC ;dekl. zewnętrznej funkcji
EXTERN GetForegroundWindow: PROC

PUBLIC hello_world_asm

hello_world_asm PROC ;deklaracja podprogramu
push rbp ;zabezpiecz rej. rbp wskaźnik ramy
mov rbp, rsp ;ustaw nowy wskaźnik ramy stosu
sub rsp, 8 * (4 + 2) ;alokacja miejsca dla parametrów
;wywołania (co najmniej 4)
;+ 2 QWORD dla wyrównania stosu
and rsp, not 15 ;wyrównanie stosu do 16 bajtów przed
;wywołaniem funkcji API

call GetForegroundWindow ;Pobierz uchwyt okna
mov rcx, rax

;Sposób wywołania funkcji MessageBoxA:
; WINUSERAPI int WINAPI MessageBoxA(
; RCX => _In_opt_ HWND hWnd,
; RDX => _In_opt_ LPCSTR lpText,
; R8 => _In_opt_ LPCSTR lpCaption,
; R9 => _In_ UINT uType);
mov rdx, offset hello_msg
mov r8, offset info_msg
mov r9, 0 ; MB_OK
call MessageBoxA

mov rsp, rbp ;przywrócenie wskaźnika stosu
pop rbp ;przywrócenie wskaźnika ramy stosu
ret ;powrót z podprogramu
hello_world_asm ENDP

_TEXT ENDS
END
```

Kod w C++:

```
extern "C" void hello_world_asm();

int main(int argc, char * argv[])
{
    hello_world_asm();
    return 0;
}
```