

به نام خدا

نحوه ی ایجاد یک کتابخانه و استفاده از آن

Programming Environment:

Visual Studio 2008(team system)

C++ for DLL / C# for Program



SalehJG
اسفند 1388

از جنس پیشگفتار

شاید برایتان پیش آمده باشد که برنامه ای بنویسید که نیازمند سرعت بیشتری باشد اگر شما با ++C راحت نیستید و به طور کامل بدان مسلط نیستید ، بهترین گزینه استفاده از یک کتابخانه ای است که الگوریتم خود را در آن پیاده سازی کرده باشید و آن را به بدنه ی برنامه ی اصلی خود که در یک زبان کندتر و با سرعت توسعه بیشتر نوشته شده، رفرنس بدهید. اگر شما هم مثل من در رشته های برنامه نویسی تحصیل نمی کنید ، پیشنهاد میکنم این مقاله را تا آخر مطالعه بفرمایید.

اگر پیشنهادی،انتقادی دارید اینجانب را بی بهره نگذارید .

saleh_jamali@ymail.com

صالح جمالی
(salehjpg)

محیط های برنامه نویسی

در این مقاله از ویژوال استودیو 2008 استفاده شده است که فقط دو زبان C++ و C# بکار گرفته شده اند. ما قصد داریم از C++ برای نوشتن کتابخانه و از C# برای نوشتن بدنه (پوسته ی برنامه) استفاده کنیم.

توضیحاتی در مورد آرایه ها در C++ و C#

تعریف آرایه ها در C++ کمی با تعریف آرایه ها در C# متفاوت است که قصد داریم آنها را مورد بررسی قرار دهیم.

C# . 1

برای تعریف یک آرایه ی تک بعدی (تا آخر مقاله با آرایه های تک بعدی کار خواهیم کرد.) چنین عمل کنید:

```
ArrayType [] ArrayName= new ArrayType[ArrayLength];
```

C++ . 2

برای تعریف یک آرایه ی تک بعدی در C++ نیز چنین عمل می کنیم:

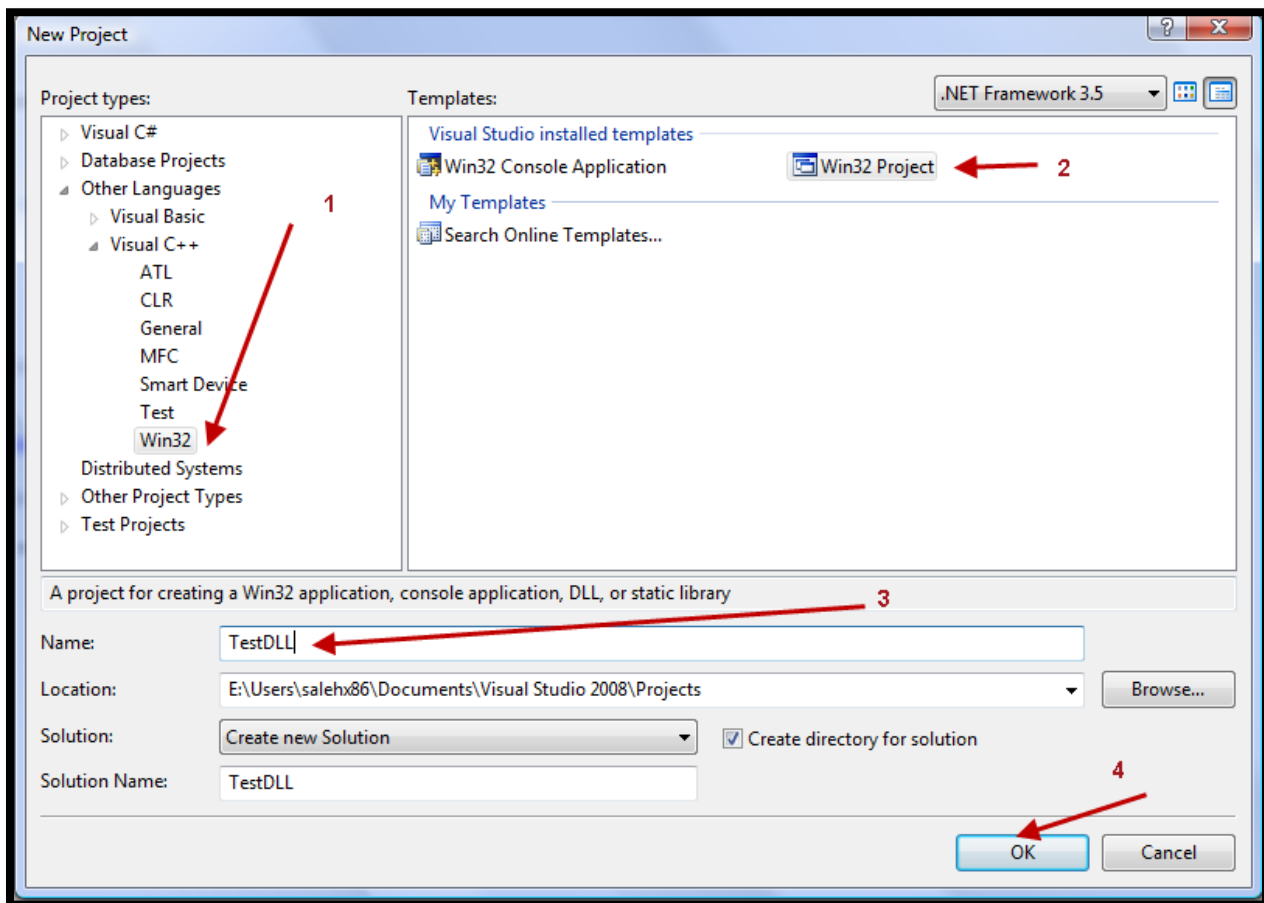
```
ArrayType ArrayName [ArrayLength];
```

البته در تعریف آرایه ها در **Prototype** ها از تعریف طول آرایه خود داری می کنیم و جای آن را خالی میگذاریم. مثلا:

```
Int test(int Array[]);
```

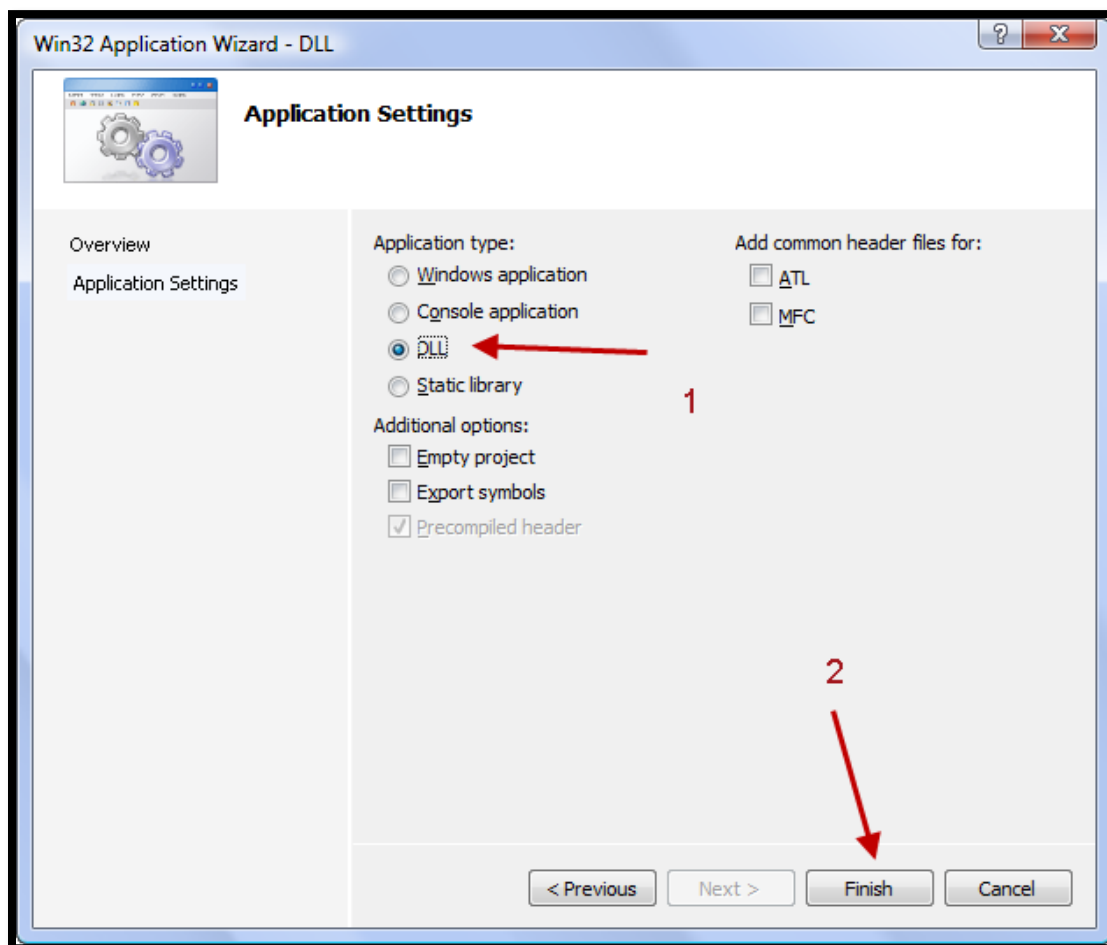
ساختن یک DLL ساده در C++ (برای استفاده در سایر زبان ها)
 برای ساختن یک کتابخانه در قسمت **New Project** به ترتیب زیر عمل می کنیم.

-1



2 - سپس در پنجره ی باز شده **Next** را بزنید

3- سپس در این پنجره در قسمت **Radio Box** ها **DLL** را انتخاب کرده و **Finish** را بزنید.



حالا شما یک پروژه ی تقریبا آماده DLL دارید که باید:

- توابع خود را درون یک فایل CPP تعریف کنید.
- یک فایل (.DEF) Module Definition اضافه کرده و نام توابعی را که قصد استفاده در سایر برنامه های خود دارید، در آن درج کنید.

توضیحات:

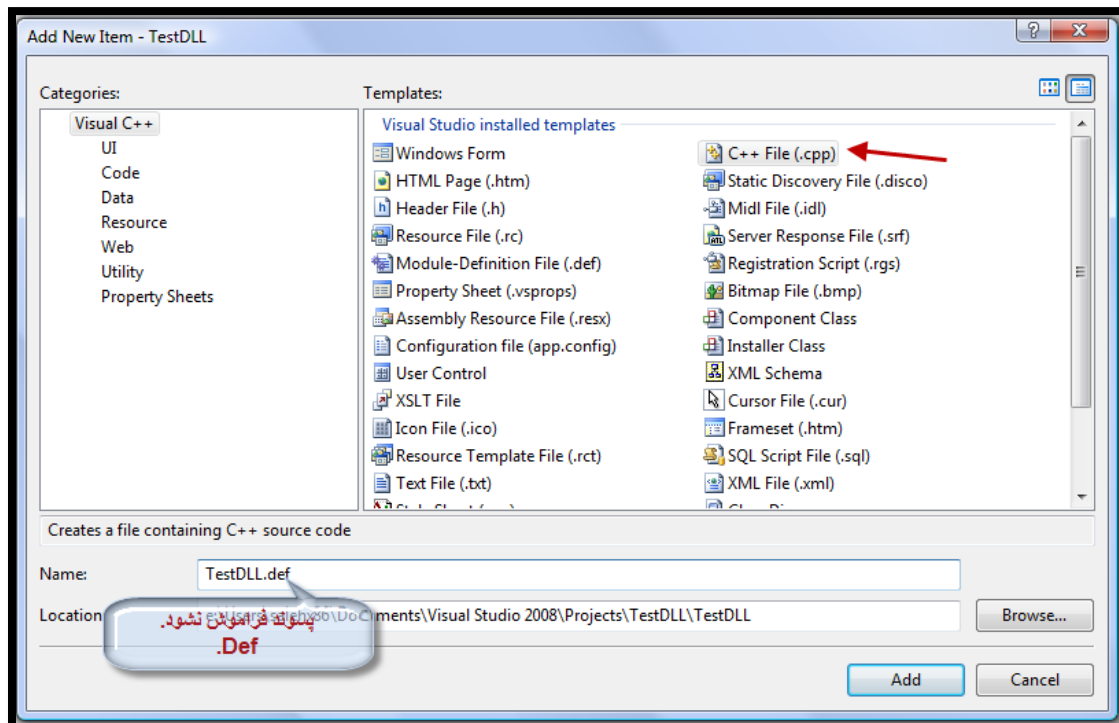
در هنگام تعریف توابع خود باید پس از تعریف نوع داده ی بازگشتی عبارت `__stdcall` را اضافه کنید. مثل:

```
int __stdcall sum(int num1,int num2)
{
return num1+num2;
}
```

فرقی نمی کند که تابع خود را در کدام فایل CPP تعریف می کنید شما در این مورد می توانید توابع خود را درون فایل های مختلف با پسوند CPP تعریف کنید.

پس از همه ی این کارها فقط دو مرحله باقی مانده است.

1. افزودن یک فایل `.def`.



داخل فایل def نام توابع خود را مثل مثال (توابعی که می خواهید Export شوند) درج کنید:

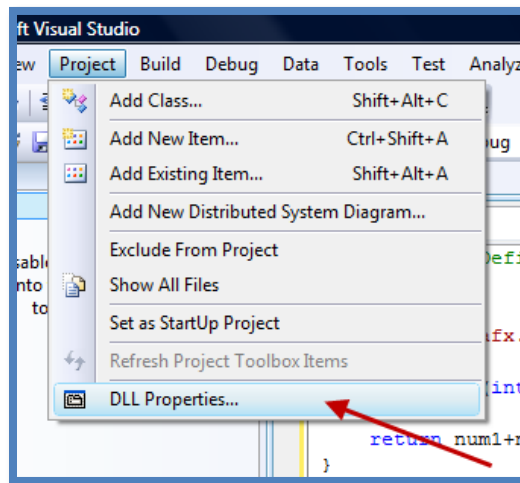
```
LIBRARY "TestDLL"  
EXPORTS  
  
sum
```

2. تعریف کردن "Def" نام فایل "در مسیر:

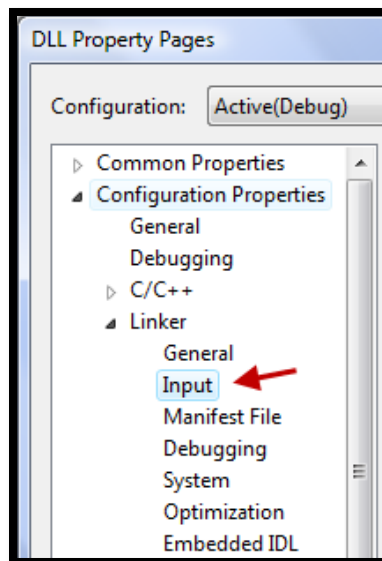
Project > [Project Name] Property > Configuration > Linker > Input
=> Module Definition File

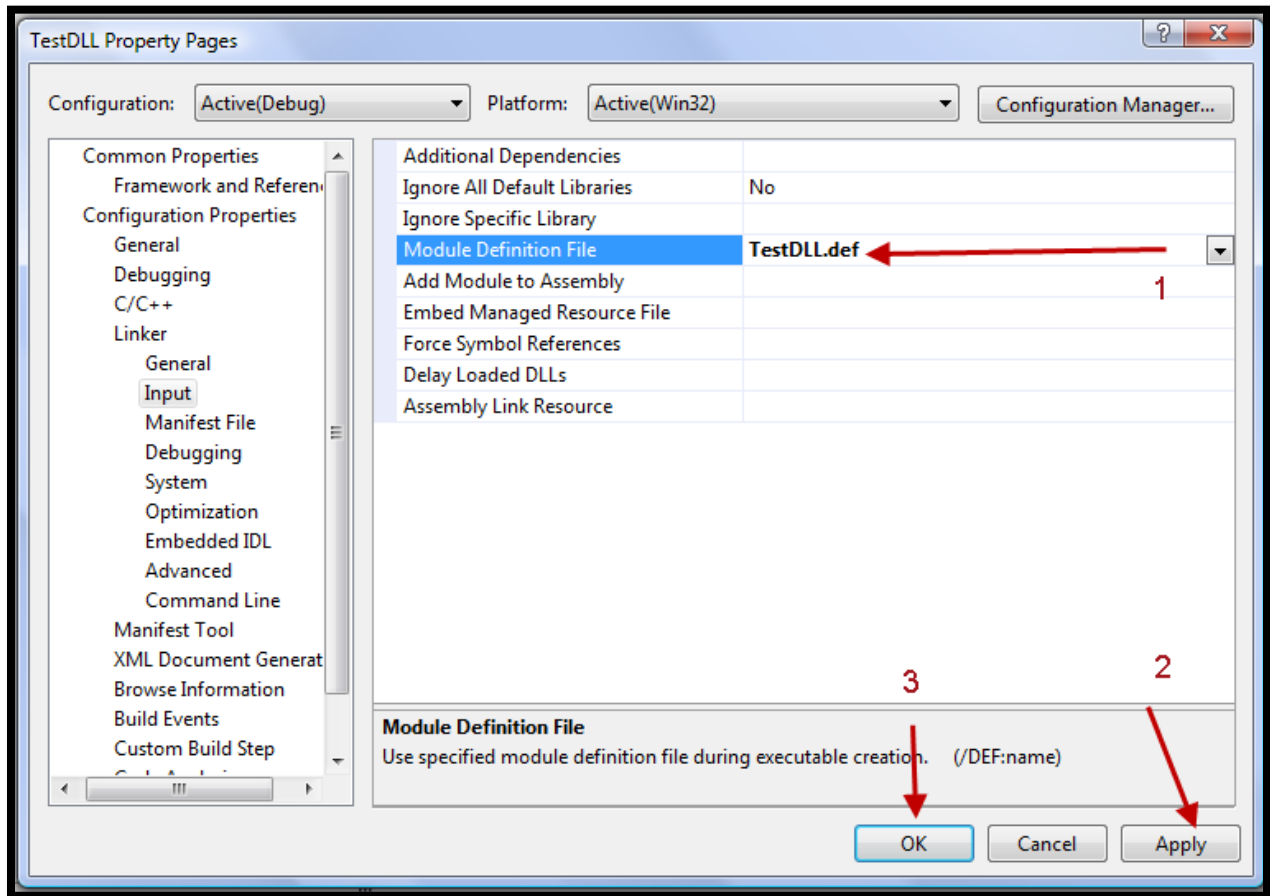
تصاویر:

1.



2.

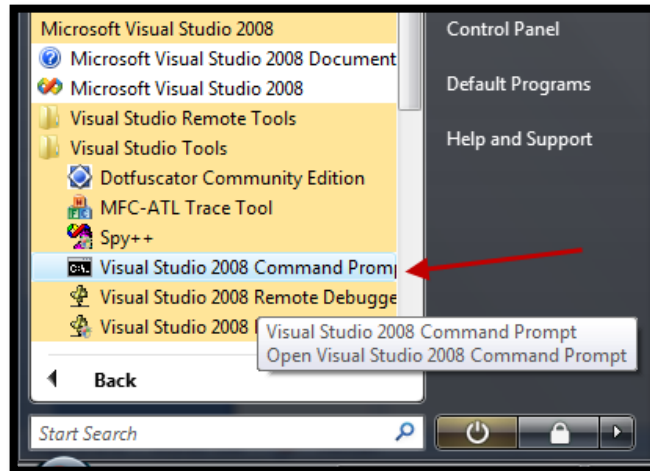




حال DLL شما آماده کامپایل شدن می باشد. آن را روی Release کامپایل کنید.

مشاهده توابع Export شده در یک DLL

به کمک یکی از ابزار های Visual Studio که Visual Studio 2008 Command Prompt نام دارد ، آدرس DLL خود را در Dumpbin وارد می کنیم.



DumpBin "File Address" /EXPORTS

```

Administrator: Visual Studio 2008 Command Prompt
E:\Windows\system32>DUMPBIN "E:\Users\salehx86\Documents\Visual Studio 2008\Projects\TestDLL\Debug\TestDLL.dll" /EXPORTS
Microsoft (R) COFF/PE Dumper Version 9.00.21022.08
Copyright (C) Microsoft Corporation. All rights reserved.

Dump of file E:\Users\salehx86\Documents\Visual Studio 2008\Projects\TestDLL\Debug\TestDLL.dll
File Type: DLL

Section contains the following exports for TestDLL.dll

00000000 characteristics
47A37A1D time date stamp Fri Feb 01 11:59:25 2008
0.00 version
1 ordinal base
1 number of functions
1 number of names

ordinal hint RVA      name
1 0 0001104B sum = @ILT+70(<sum@PYGHHHCZ>)

Summary
1000 .data
1000 .idata
2000 .rdata
1000 .reloc
1000 .rsrc
4000 .text
10000 .textbss
  
```

ایجاد یک برنامه ی کاربردی و استفاده از TestDLL.dll در C#

در این مرحله شما با نحوه ی استفاده از DLL تولید شده آشنا می شوید.

نحوه ی کار از این قرار است که:

- یک پروژه ایجاد شود
- اضافه کردن سرویس Interop بوسیله ی کد:
- تعریف تابع در برنامه بصورت extern با DllImport :

```
using System.Runtime.InteropServices;
```

```
[DllImport("TestDLL.dll")]
public static extern int sum(int num1, int num2);
```

مثال:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace TestDLLPRG
{
    public partial class Form1 : Form
    {
        [DllImport("TestDLL.dll")]
        public static extern int sum(int num1, int num2);

        public Form1()
        {
            InitializeComponent();
        }

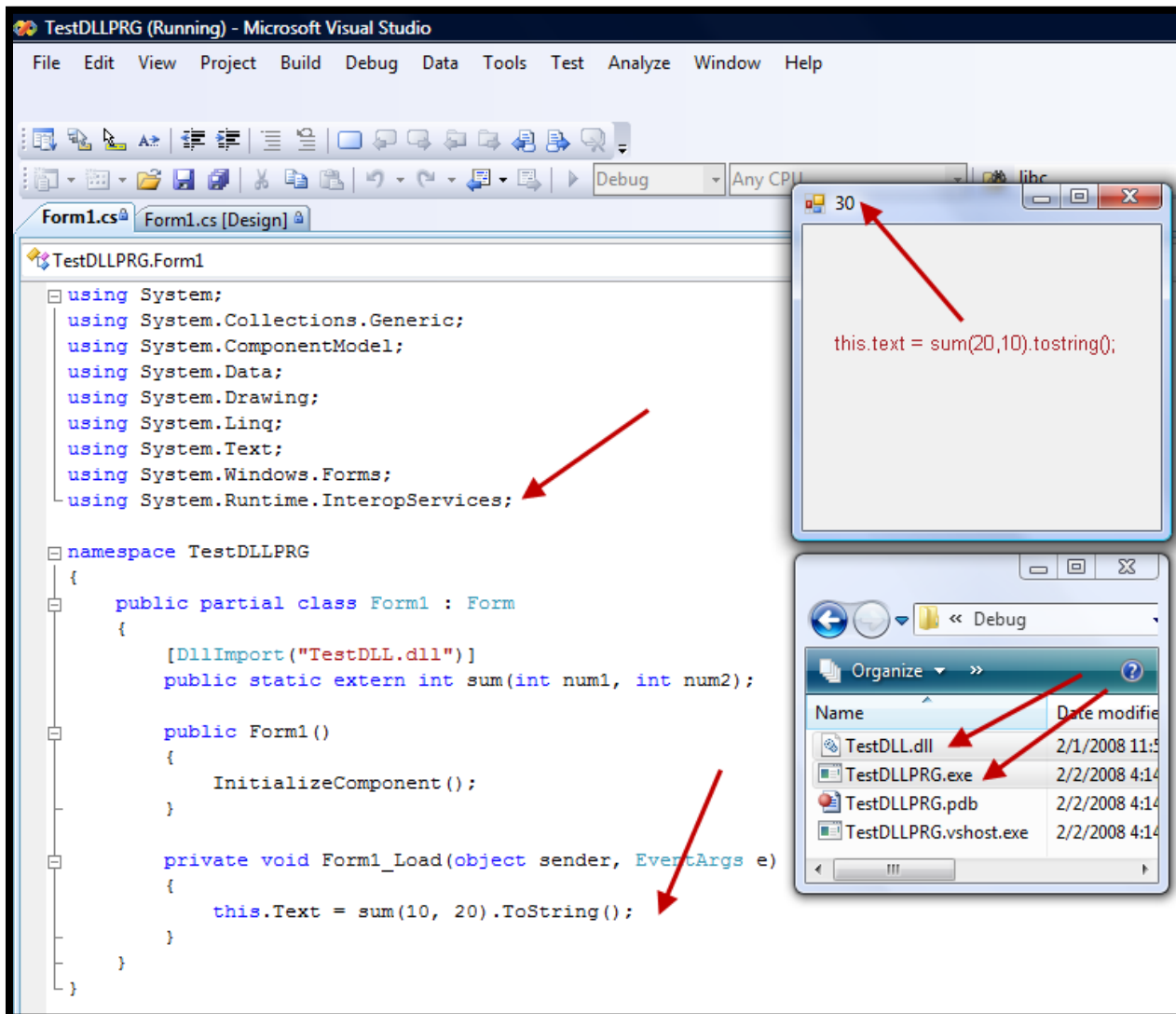
        private void Form1_Load(object sender, EventArgs e)
        {
            this.Text = sum(10, 20).ToString();
        }
    }
}
```

نتیجه ی کار:

تابع sum که درون فایل TestDLL.dll قرار دارد از طریق برنامه ی TestDLL_PRG.exe فراخوانی شده و مقدار بازگشتی آن در عنوان پنجره نمایش داده می شود.

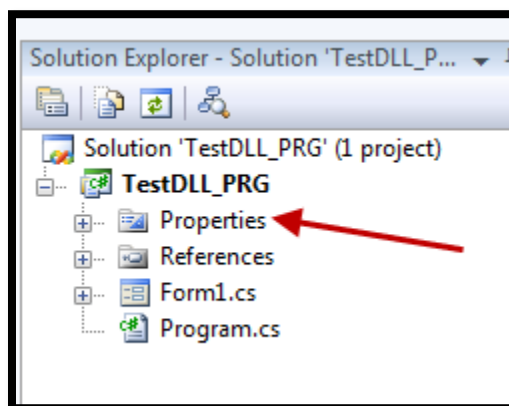
فقط یادتان باشد که TestDLL.dll را در مسیر فایل اجرایی یا درون System32 کپی نمایید.

عکس برنامه:

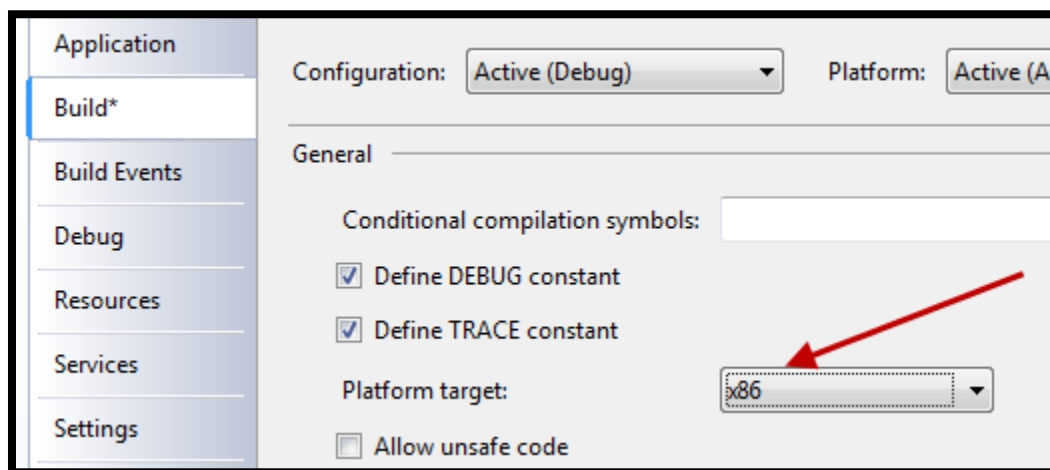


تذکره:

اگر از سیستم عامل های 64 بیتی استفاده می کنید یادتان نرود که:



در قسمت **Properties** پروژه از قسمت **Build** پلتفرم را از **Any CPU** به **x86** تغییر دهید.



چون DLL شما 32 بیتی است پس باید برنامه ی کاربردی شما نیز 32 بیتی باشد در غیر این صورت در سیستم عامل های x64 برنامه دچار مشکل خواهد شد.

و حتما هر دو (برنامه و DLL) را در حالت Release کامپایل کنید، در غیر این صورت برنامه ی شما در سایر ویندوزهایی که Visual Studio نصب نیست، اجرا نخواهد شد.

یک نکته:

اگر بخواهید فایلی را با استفاده از کتابخانه های Framework باز کنید (منظورم این نسخه از فریم ورک هست از سایرین اطلاعی ندارم) در هنگام تایپ کردن آدرس فایل بجای یک عدد / از دو عدد (//) استفاده می کنیم.

مثال:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.BackgroundImage = Image.FromFile("E:\\ Pictures\\t1.bmp");
}
```

در آینده

از جمله نرم افزارهایی که نیاز به سرعت بالای اجرا دارد ، نرم افزارهای پردازش تصویر می باشد.

در سری بعدی این مقاله سراغ پردازش تصویر در C# & C++ می رویم که در آن خواهید آموخت:

- چگونه رنگ پیکسل های یک عکس را در آرایه قرار دهید. (C#)
- چگونه فیلترها را بکار بگیرید.
- چگونه سرعت برنامه ی خود را افزایش دهید.
- چگونه تصویر را از آرایه استخراج کنید.

امیدوارم که این نوشته ی کوتاه را از نظرات و انتقادات خود بی بهره نگذارید.

SalehJG

Email: saleh_jamali@ymail.com

<http://forum.iranled.com/>

<http://pudn.com/>

<http://codeproject.com/>

به امید موفقیتان