

Forum:[Hack Forums](#)

Thread: [How do I create a Builder?](#)

riksel - March 2, 2016, 5:11 am

Hey guys,

I coded my first project in C# and I want to have an external application where a user can enter 4 strings and then the "builder" should make a new .exe. Since I have never done that I have no idea. I heard that I should use something like EOF facepalm or CodeDOM or something (no idea lol) I've already looked up some tutorials here from HF but couldn't really do anything with it.

Can anyone help me on how to make the external application add the userinput into my main code from the other / real project. (Just like in any virus or idk)

Thanks in advance

Jacob ?

Nick (Name) - March 2, 2016, 5:53 am

EOS! Use EOS! Everyone nowadays uses End of Shit! Dropped a deuce on my application and it came out lovely!

Actually, there are more tutorials out there than just on HF about CodeDom, and by EOS you mean EOF which is End of File, which is something that you should never use. EOF triggers AV's like crazy now. If you look on MSDN you can find almost everything to know about .NET. I advise reading MSDN's references than any tutorial. That's just my opinion.

IAmBabyFinn - March 2, 2016, 6:49 am

Here you go.

```
Code //using System;
//using System.CodeDom.Compiler;
//using Microsoft.CSharp;
public static void asd(){
    CSharpCodeProvider provider = new CSharpCodeProvider();
    CompilerParameters comp = new CompilerParameters();
    comp.ReferencedAssemblies.Add("System.dll");//All the Dll Files what your project needs
    comp.ReferencedAssemblies.Add("System.IO.dll");
    comp.ReferencedAssemblies.Add("System.Net.dll");
    comp.ReferencedAssemblies.Add("System.Windows.Forms.dll");
    comp.GenerateExecutable = true;
    comp.WarningLevel = 0;
    comp.CompilerOptions = @"//win32icon:"; //Icon
    comp.CompilerOptions = "/target:winexe";//Output the file as exe
    comp.OutputAssembly = "stub.exe";
    comp.GenerateInMemory = false;
    CompilerResults cr = provider.CompileAssemblyFromFile(comp,"YOUR SOURCE FILE
HERE");
}
```

So if you want to input some strings to your source i used this way

```
Code File.WriteAllText(PathWhereToWrite,Properties.Resource.MySource);
```

```
string Source=File.ReadAllText(PathWhereToWrite);
```

```
Source=Source.Replace("MYSETTING1",TextBox1.Text);So lets say your source file has
```

a string called
string option1="MYSETTING1";
So MYSETTING1 gets replaced with the textbox input user typed

riksel - March 2, 2016, 7:28 am

Thanks for the answers, edited the "End of Shit" xD Gonna try out BabyFinn's method :)

Hoodie+ - March 2, 2016, 7:33 am

You would have a resource and the builder would edit the resource.

riksel - March 2, 2016, 8:01 am

(Mar 2, 2016 07:49 AM)IAmBabyFinn Wrote: [\[CITE\]](#) Here you go.

```
Code //using System;
//using System.CodeDom.Compiler;
//using Microsoft.CSharp;
public static void asd(){
    CSharpCodeProvider provider = new CSharpCodeProvider();
    CompilerParameters comp = new CompilerParameters();
    comp.ReferencedAssemblies.Add("System.dll");//All the Dll Files what your project needs
    comp.ReferencedAssemblies.Add("System.IO.dll");
    comp.ReferencedAssemblies.Add("System.Net.dll");
    comp.ReferencedAssemblies.Add("System.Windows.Forms.dll");
    comp.GenerateExecutable = true;
    comp.WarningLevel = 0;
    comp.CompilerOptions = @"/win32icon:";//Icon
    comp.CompilerOptions = "/target:winexe";//Output the file as exe
    comp.OutputAssembly = "stub.exe";
    comp.GenerateInMemory = false;
    CompilerResults cr = provider.CompileAssemblyFromFile(comp,"YOUR SOURCE FILE
HERE");
}
```

So if you want to input some strings to your source i used this way
Code File.WriteAllText(PathWhereToWrite,Properties.Resource.MySource);
string Source=File.ReadAllText(PathWhereToWrite);
Source=Source.Replace("MYSETTING1",TextBox1.Text);So lets say your source file has
a string called
string option1="MYSETTING1";
So MYSETTING1 gets replaced with the textbox input user typed
I made it like this.

```
Code File.WriteAllText("C:\\Users\\Jacob\\Documents\\Visual Studio
2015\\Projects\\Cake\\Cake\\Form1.cs", Properties.Resources.MySource);
```

but it shows me an error at "Properties.Resources.MySource" should I input something for MySource?

IAmBabyFinn - March 2, 2016, 8:07 am

Oh no

I assume you are using visual studio so take properties from your project and find tab called resources.

From there add file or make a new text file where you copy your source.

So you can call that resource wit "Properties.Resource.the name of your resource name"

[/code]

rikxel - March 2, 2016, 8:37 am

(Mar 2, 2016 09:07 AM)IAmBabyFinn Wrote: [\[CITE\]](#) Oh no

I assume you are using visual studio so take properties from your project and find tab called resources.

From there add file or make a new text file where you copy your source.

So you can call that resource wit "Properties.Resource.the name of your resource name"

[/code]

It seems kinda stupid but I still can't get it. Or is it and I am just missing an exe output? I tried it like this:

```
Code using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.CodeDom.Compiler;
using Microsoft.CSharp;
```

```
namespace Cake_Builder
```

```
{
    public partial class Form1 : Form
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void btnAdd_Click(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
        private void btnBuild_Click(object sender, EventArgs e)
```

```
//public static void
```

```
compile()
```

```
{
```

```
        File.WriteAllText("C:\\Users\\Jacob\\Documents\\Visual Studio
2015\\Projects\\Cake_Builder\\done.txt", Properties.Resources.sourcedes);
        string Source = File.ReadAllText("C:\\Users\\Jacob\\Documents\\Visual Studio
2015\\Projects\\Cake_Builder\\sourcedes.txt");
        Source = Source.Replace("12ujBpLpE1AUefXvSUexgKmSEa7TZLBcyP",
TXtadress1.Text);
        Source = Source.Replace("1255oyKBN2CUtJtPjeQaSXd4VJarHQ325X",
```

```

TXTadress2.Text);
    Source = Source.Replace("1KK9FEpwsaXLR69S2pLosEEX49sWpWnoFS",
TXTadress3.Text);
    Source = Source.Replace("1Nwp99oXtH3iPuFHCMKuZdMouwsgB3RwmP",
TXTadress4.Text);

```

```

    CSharpCodeProvider provider = new CSharpCodeProvider();
    CompilerParameters comp = new CompilerParameters();
    comp.ReferencedAssemblies.Add("System.dll"); //All the Dll Files
what your project needs
    comp.ReferencedAssemblies.Add("System.IO.dll");
    comp.ReferencedAssemblies.Add("System.Net.dll");
    comp.ReferencedAssemblies.Add("System.Windows.Forms.dll");
    comp.GenerateExecutable = true;
    comp.WarningLevel = 0;
    comp.CompilerOptions = @"/win32icon:"; //Icon
    comp.CompilerOptions = "/target:winexe"; //Output the file as exe
    comp.OutputAssembly = "stub.exe";
    comp.GenerateInMemory = false;
    CompilerResults cr = provider.CompileAssemblyFromFile(comp,
"C:\\Users\\Jacob\\Documents\\Visual Studio 2015\\Projects\\Cake_Builder\\sourcedes.txt");

    }

}
}

```

btw. sourcedes has been added as a .txt I just copied the source from the main application and put it in the builder folder as a .txt also and call it from there.

IAmBabyFinn - March 2, 2016, 8:57 am

EDIT!

This here is working code I tested .

```

Code //using System.CodeDom.Compiler;
//using Microsoft.CSharp;

```

```

try {
    CSharpCodeProvider provider = new CSharpCodeProvider();
    CompilerParameters comp = new CompilerParameters();
    comp.ReferencedAssemblies.Add("System.dll"); // Here you put dll needed in your
project what your project needs
    comp.ReferencedAssemblies.Add("System.Linq.dll"); // Here you put dll needed in
your project what your project needs
    comp.ReferencedAssemblies.Add("Microsoft.CSharp.dll"); // Here you put Dll needed in
your project what your project needs
    comp.GenerateExecutable = true;
    comp.WarningLevel = 0;
    comp.CompilerOptions = "/target:winexe";
    comp.OutputAssembly = @"C:\users\Karl\Desktop\stub.exe";
    comp.GenerateInMemory = false;
    CompilerResults cr = provider.CompileAssemblyFromFile(comp,
@"C:\Users\karl\Desktop\asd.txt");

```

```

        foreach (CompilerError CompErr in cr.Errors) { //This here lists all errors your project might
counter !
            Console.WriteLine("Line number " + CompErr.Line + ", Error Number: " +
CompErr.ErrorNumber + ", " + CompErr.ErrorText + ";" + Environment.NewLine +
Environment.NewLine);
        }
        Console.ReadLine();
    } catch (Exception e) {

        Console.WriteLine(e);
        Console.Read();
    } This should work

```

rikxel - March 2, 2016, 9:55 am

(Mar 2, 2016 09:57 AM)IAmBabyFinn Wrote: [\[CITE\]](#) EDIT!

THis here is working code I tested .

```

Code //using System.CodeDom.Compiler;
//using Microsoft.CSharp;

```

```

try {
    CSharpCodeProvider provider = new CSharpCodeProvider();
    CompilerParameters comp = new CompilerParameters();
    comp.ReferencedAssemblies.Add("System.dll");          // Here you put dll needed in your
project what your project needs
    comp.ReferencedAssemblies.Add("System.Linq.dll");      // Here you put dll needed in
your project what your project needs
    comp.ReferencedAssemblies.Add("Microsoft.CSharp.dll"); // Here you put Dll needed in
your project what your project needs
    comp.GenerateExecutable = true;
    comp.WarningLevel = 0;
    comp.CompilerOptions = "/target:winexe";
    comp.OutputAssembly = @"C:\users\Karl\Desktop\stub.exe";
    comp.GenerateInMemory = false;
    CompilerResults cr = provider.CompileAssemblyFromFile(comp,
@"C:\Users\karl\Desktop\asd.txt");
    foreach (CompilerError CompErr in cr.Errors) { //This here lists all errors your project might
counter !
        Console.WriteLine("Line number " + CompErr.Line + ", Error Number: " +
CompErr.ErrorNumber + ", " + CompErr.ErrorText + ";" + Environment.NewLine +
Environment.NewLine);
    }
    Console.ReadLine();
} catch (Exception e) {

    Console.WriteLine(e);
    Console.Read();
} This should work

```

Wait, this is Console, right? If so I still need to import the source. Or else it shows me that no Startingpoint was found.

IAmBabyFinn - March 2, 2016, 10:43 am

It does not matter if its console or not. Il pm you my skype and we can talk there

