

Forum:[Hack Forums](#)

Thread: [\[Tutorial\]How to use the CodeDom Compiler in VB.NET\[Tutorial\]](#)

Skillful - December 19, 2011, 6:51 am

Hello.

I am glad to present my tutorial! :)

This tutorial will teach you to use the CodeDom Compiler in VB.NET

Before you start reading this tutorial I would recommend you to check out [this](#) program so that you know what can you do by reading this tutorial.

Its nice to be able to make programs with your own programs. You can customize it how you want and it dont need to have that big file size like VB.NET Express or Visual Studio. So im going to teach you how to:

What is CodeDom?

Importing the necessary Namespace and declaring the required Variables

Creating a Compiler

Compiler Options

Adding References

Adding Resources

Show installed Programming Languages

Compiling to an Assembly

Example

Conclusion

So lets start! :D

What is CodeDom?

CodeDom is Code Document Object Model. This provides a graphical (in the sense of graph nodes, not of a picture) representation of code in the form of a tree. This represents the hierarchy of nested code elements (namespaces contain classes, classes contain methods, methods contain variable declarations, and so on). You can use CodeDom for generating an in memory code tree, or you can use it to compile ordinary code written in a file (like we will do later). View MSDN to read all about CodeDom elements. With this code tree, you can represent any piece of code independently of a specific language.

Importing the necessary Namespace and declaring the required Variables

To begin with...

Import this namespace at the top of your code:

Code Imports System.CodeDom.Compiler

Inside your class, you need to declare these variables:

Code Private provider As CodeDomProvider

Private Options As New CompilerOptions

In this tutorial we will be using Options as CompilerParameters and Provider as a CodeDomProvider.

Creating a Compiler

To create a compiler, you must choose what language that shall be used, for example VB, CS, CPP, etc.

To do this, use this code:

```
Code If CodeDomProvider.IsDefinedLanguage("vb") Then
```

```
    provider = CodeDomProvider.CreateProvider("vb")
```

```
    'We will use this provider in our code later
```

```
Else
```

```
    MsgBox("The specified language is not defined for this system.")
```

```
End If
```

Of course you can change "vb" to whatever coding language you want, but we will come back to that later in Show installed Programming Languages.

Compiler Options

We will use the class CompilerParameters to set the options for our compiler.

The available options are: CompilerOptions - Parameters for compiling, for example "/optimize".

EmbeddedResources - The .NET Resource files for embedding when compiling.

GenerateExecutable - If true, generates an executable assembly(EXE), not a library(DLL).

GenerateInMemory - If true, generates the compiled output in memory.

IncludeDebugInformation - Specifies whether to create debug information for the executable file.

LinkedResources - The .NET Resource files that is referenced in the source.

MainClass - The main, or startup class.

OutputAssembly - The output file.

ReferencedAssemblies - Assemblies referenced by the current project, for example "System.dll".

TempFiles - Temporary files for the compilation.

TreatWarningsAsErrors - Gets or sets a value indicating whether to treat warnings as errors.

UserToken - Gets or sets the user token to use when creating the compiler process.

WarningLevel - Gets or sets the warning level at which the compiler aborts compilation.

Win32Resource - Gets or sets the file name of a Win32 resource file to link into the compiled assembly.

Adding References

To add references to your project, use the ReferencedAssemblies property.

This example demonstrates how to use this property:

Code Dim OpenReferences As New OpenFileDialog

OpenReferences.Title="Add References"

OpenReferences.MultiSelect=True

If OpenReferences.ShowDialog() = DialogResult.OK Then

Options.ReferencedAssemblies.AddRange(OpenReferences.FileNames) 'Adds a range of files selected in the file dialog

End If

Its smart to always add the System.dll reference because i cant think of any program that don't uses that. Mscorlib.dll is also a smart reference to add. To do so use this:

Code Options.ReferencedAssemblies.Add("System.dll")

Options.ReferencedAssemblies.Add("mscorlib.dll")

Adding Resources

Resources is a group of files compiled into a .NET Resource file. They can be images, sounds, text, or files. They are used by the program in different ways. One way is to display an image in a picture box, another is to change the language of the application. The EmbeddedResources property is mostly files or objects used in the application, while the LinkedResources can be culture options, these will be included in a separate assembly. To check if the assembly supports resources, use this piece of code:

Code If Provider.Supports(GeneratorSupport.Resources) Then

'Add resource files here

End If

To add a resource file, or more use this:

Code Dim OpenResources As New OpenFileDialog

OpenResources.Title="Add Resources"

OpenResources.MultiSelect=True

OpenResources.Filter = "Resource files(*.resource)|*.resource"

If OpenResources.ShowDialog() = DialogResult.OK Then

Options.EmbeddedResources.AddRange(OpenResources.FileNames) 'Adds a range of files selected in the file dialog

End If

You can change 'EmbeddedResources' to 'LinkedResources' if you want the resources to be linked in another assembly.

Show installed Programming Languages

Its handy to know which coding languages that is supported on your computer, here is some code to show it:

Code Dim Languages As String = ""

```
For Each language As CompilerInfo In CodeDomProvider.GetAllCompilerInfo
    For Each codingLanguage In language.GetLanguages()
        Languages &= codingLanguage & vbNewLine
    Next
Next
```

```
MsgBox(Languages)
```

This code will show you a message box with each language on its own line. You can replace "vb" in Creating a Compiler with one of these languages.

Compiling to an Assembly

To compile files to an assembly we will use the CodeDomProvider.CompileAssemblyFromFile method.

This requires 2 parameters CompilerOptions As CompilerParameters and FileNames() As String. We will use Options that we have set earlier in Compiler Options and some filenames that we will get with this code:

```
Code Dim SourceFiles As New OpenFileDialog
SourceFiles.Title="Add Source Files"
SourceFiles.MultiSelect=True
Dim Results As CompilerResults
If SourceFiles.ShowDialog() = DialogResult.OK Then
    Results = Provider.CompileAssemblyFromFile(Options, SourceFiles.FileNames)
End If
```

This code opens a dialog for selecting files, then it tries to compile the filenames with the specified CompilerOptions and returns the result to the Results variable. Later on we can check if there were any errors with the Results.Errors property.

Example

I will now include a full example for compiling code.

[This](#) will demonstrate the different properties we have been through.

Also remember [this](#) project to see what you can make with the CodeDom compiler.

Conclusion

I have spent a lot of valuable time writing this tutorial so that other members or newbies can benefit from it.

A thanks would be appreciated.

Regards,
Skillful

daniellook - December 19, 2011, 7:00 am

Great tutorial.

I dont know if im ever gonna use it but I bookmarked it.

Good job!

????? - December 19, 2011, 7:00 am

Nice tutorial, but you might want to explain to other's exactly what CodeDom is and what it does, how to add icons, parameters etc etc. Nonetheless, good tutorial man. Good job.

RiperFish - December 19, 2011, 7:03 am

(Dec 19, 2011 08:00 AM)????? Wrote: [CITE](#) Nice tutorial, but you might want to explain to other's

exactly what CodeDom is and what it does, how to add icons, parameters etc etc. Nonetheless, good tutorial man. Good job.

They can simply search on Msdn.Great tut Dude :).

????? - December 19, 2011, 7:05 am

(Dec 19, 2011 08:03 AM)L.I ?aGnifiCo Wrote: [\[CITE\]](#) They can simply search on Msdn.Great tut Dude :).

But wouldn't that be defeating the purpose of a tutorial in the first place?

I mean it doesn't bother me, i know codedom. Just trying to help the quality of the tutorial that's all.

Skillful - December 19, 2011, 7:08 am

Thanks for your suggestion ?????! :D

I've updated the thread with the 'missing information' :D

Weakky. - December 19, 2011, 7:10 am

Nice tutorial man :) ! Keep it up ?

????? - December 19, 2011, 7:12 am

(Dec 19, 2011 08:08 AM)Skillful Wrote: [\[CITE\]](#) Thanks for your suggestion ?????! :D

I've updated the thread with the 'missing information' :D

=> i meant no disrespect, just trying to help. Great tutorial man.

Tedd - December 19, 2011, 7:14 am

Great tutorial. Bookmarked for reference! :)

Skillful - December 19, 2011, 7:15 am

Glad that you'll liked this ?

blackperuvian - December 19, 2011, 7:21 am

Thanks Very usefull tutorial , HQ tut

intrigued1980 - December 28, 2011, 5:50 pm

Nice tutorial man! Many thanks for this share.

ProfitCoder - March 16, 2013, 1:16 pm

Hey ;)

This is a very nice tutorial

Thanks

hansbald - May 21, 2013, 11:53 am

Wow thanks thats what i needed!

Nice work!

gungunfebrianza - May 22, 2013, 7:06 am

Sir... can you help me to hide this cmd form :

[img]http://i564.photobucket.com/albums/ss86/febrianza/1-13_zps823e8f6c.png[/img]

PHP Code

<?php

this is my c# code, in i love to learning your tutorial.

using System.Windows.Forms;

```
class Test
{
    static void Main(string[] args)
    {
        MessageBox.Show("Hello World!", "NScript");
    }
};
```

and i need your help how to add some resource(VBS file) and embed it into an compiled executable
? so when i execute it the vbs executed :)

LastSoul - May 22, 2013, 7:39 am

^^thx alot its very good tutorial^^

ProfitCoder - May 22, 2013, 5:39 pm

Great Tutorial ;)

Thanks for the share

C3oI - May 22, 2013, 5:46 pm

(May 22, 2013 07:06 AM)gungunfebrianza Wrote: [\[CITE\]](#) Sir... can you help me to hide this cmd form
:

and i need your help how to add some resource(VBS file) and embed it into an compiled executable
? so when i execute it the vbs executed :)

To CompilerParameters add CompilerOptions /target:winexe

ChuckTestaLolz - July 21, 2013, 10:13 am

Thanks for the tutorial, really well done!