Ask a question

Search related threads   Search forum questions

Quick access

Answered by:

5
Points

dustbin1_uk
Partner   Joined Jan 2006

dustbin1_uk's thre...

4   Show activity

Top related threads

Using a C++ DLL in VB.Net 2.0

Using C++ dll in VB.Net

how to invoke method in a class from a C++ dll in vb.net

Calling C++ DLL from VB.net

Interop C++ dll from vb.net

C++ DLL in VB.NET

Visual Studio Languages , .NET Framework > Visual Basic

Question

Hey all! Hope you are well!

Googled around for an answer to this and getting very confused!

0

Sign in
to vote

I have followed this tutorial HERE on creating a C++ DLL, very simple process, love it!

Now, I can call this DLL from any C++ program (tested and works great!), but I wanted to learn how to get it working in VB .NET. In other words I wanted to make a multi functional DLL that can be called from VB .NET or C++.

I created a new project in VB.NET saved it and copied the MathFuncsDll.dll to my bin Debug folder, and I went to my project properties -> references and I pressed the Add button, navigated to the bin Debug folder and tried to add the DLL. But I get an error:

"A reference to 'c:\Documents and Settings\mark\My Documents\Visual Studio 2008\TestDll\TestDLL\bin\Debug\MathFuncsDll.dll' could not be added. Please make sure that the file is accessible, and that it is a valid assembly or COM component."

Ok I just now that I'm going to feel VERY silly when someone tells me what I am doing wrong. So if anyone would be so kind as to educate me, I'd be grateful. There's the satisfaction of making me feel silly in it for you! :)

Thanks,

M

Friday, February 05, 2010 10:23 AM

Reply | Quote

dustbin1_uk (Partner)   5 Points

Answers

1

Sign in
to vote

OK!!!! Got it working at last, and I didn't need to trim out the Namespace or the Class from the DLL. I knew there was a way to do it!

But thanks to all the people who tried to help!

But thanks to all the people who tried to help!

So here is the solution:

I created the DLL in C++ as I said before:

```cpp
// MathFuncsDll.h
namespace MathFuncs
{
    class MyMathFuncs
    {
    public:
        // Returns a + b
        static __declspec(dllexport) double Add(double a, double b);

        // Returns a - b
        static __declspec(dllexport) double Subtract(double a, double b);

        // Returns a * b
        static __declspec(dllexport) double Multiply(double a, double b);

        // Returns a / b
        // Throws DivideByZeroException if b is 0
        static __declspec(dllexport) double Divide(double a, double b);
    };
}
```

I then used this tool (click for link) to view the DLL and get the exported function definition. In this case the Add function is: ?Add@MyMathFuncs@MathFuncs@@SANNN@Z

Now, I also realised that I was being really dumb in defining the function in VB.net using Decimals, not doubles, so I changed that.
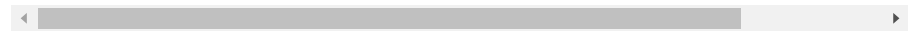
In the end the working code to call the DLL and use it's Add function inside the MyMathsFunc Class inside the MathFuncs C++ namespace is:

```vbnet
Public Class Form1

    Declare Function Add Lib "MathFuncsDll.dll" _
    Alias "?Add@MyMathFuncs@MathFuncs@@SANNN@Z" _
    (ByVal mynum As Double, ByVal mynum2 As Double) As Double

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Syst
    Handles Button1.Click
        MsgBox(Add(12, 7))
    End Sub

End Class
```

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

Marked as answer by dustbin1_uk   Tuesday, February 09, 2010 1:15 PM

Tuesday, February 09, 2010 1:09 PM

dustbin1_uk (Partner)   5 Points

All replies

0

Sign in
to vote

Don't put the DLL in the Bin folder. Put it in the project folder. When you add it to your project (Add Existng Item) you can highlight it in the solution explorer list, and set the Copy to Output setting to Copy if newer or Copy always. That way, it will always be copied to where the EXE expects to find it at run time.

From your description. it seems that you will not be able to add the DLL as a reference, but you will need to import the functions using importDLL - something like ths:

<System.Runtime.InteropServices.DllImport("MyDll.dll", _
SetLastError:=False)> _
Public Shared Function myFunction(ByVal myString As String) As String
End Function

See:
http://msdn.microsoft.com/en-
us/library/system.runtime.interopservices.dllimportattribute.aspx

Proposed as answer by   DiegoCattaruzza   Friday, February 05, 2010 12:08 PM

Friday, February 05, 2010 11:43 AM

Reply | Quote

Acamar (MCC, MVP)   108,225 Points

0

Sign in
to vote

Almost, but I get an error in VB.NET
"Unable to find an entry point named 'Add' in DLL 'MathFuncsDll.dll'."

The function is inside a class, and that's inside a namespace, so I just need to know how I define the entry point in such a situation:

```
// MathFuncsDll.h

namespace MathFuncs
{
    class MyMathFuncs
    {
    public:
        // Returns a + b
        static __declspec(dllexport) double Add(double a, double b);

        // Returns a - b
        static __declspec(dllexport) double Subtract(double a, double b);

        // Returns a * b
```

```
        // Returns a * b
        static __declspec(dllexport) double Multiply(double a, double b);

        // Returns a / b
        // Throws DivideByZeroException if b is 0
        static __declspec(dllexport) double Divide(double a, double b);
    };
}
```

And here is the code in VB .NET:

```
Public Class Form1

    <System.Runtime.InteropServices.DllImport("MathFuncsDll.dll", _
    SetLastError:=False)> _
    Public Shared Function Add(ByVal mynum As Decimal, ByVal mynum2 As Decim

    End Function


    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.
        TextBox1.Text = Add(12, 3).ToString()
    End Sub

End Class
```

◄ |◄ ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓ | ►

Friday, February 05, 2010 12:39 PM

Reply | Quote

dustbin1_uk (Partner)   5 Points

0

Sign in
to vote

Anyone??

Monday, February 08, 2010 8:52 AM

Reply | Quote

dustbin1_uk (Partner)   5 Points

0

Probably-

    Public Shared Function MyMathFuncs.Add(ByVal mynum As Decimal, ByVal mynum2 As Decimal) As Decimal

But AFAIK there's no reason that you wouldn't fully qualify it:

    Public Shared Function MathFuncs.MyMathFuncs.Add(ByVal mynum As Decimal, ByVal mynum2 As Decimal) As Decimal

Monday, February 08, 2010 10:48 AM

Reply | Quote

Acamar (MCC, MVP)   108,225 Points

0

Nope don't work

That's what I thought would be the obvious thing to do so I tried both of those and I get a "End of Statement Expected Error" from the first full stop sign in either case.

Monday, February 08, 2010 10:54 AM

Reply | Quote

dustbin1_uk (Partner) 5 Points

0

You forget to add Extern "C" to your library. So that you can access it outside

extern "C"
{

You can put all your function inside this block

}

or do it individual

```
 // Returns a + b
    extern "C" static __declspec(dllexport) double Add(double a, double b);

    // Returns a - b
    extern "C" static __declspec(dllexport) double Subtract(double a, double b);

    // Returns a * b
    extern "C" static __declspec(dllexport) double Multiply(double a, double b);

    // Returns a / b
    // Throws DivideByZeroException if b is 0
    extern "C" static __declspec(dllexport) double Divide(double a, double b);
```

kaymaf

If that what you want, take it. If not, ignored it and no complain
CODE CONVERTER SITE : http://www.carlosag.net/Tools/CodeTranslator/.

Monday, February 08, 2010 12:46 PM

Reply | Quote

kaymaf   44,530 Points

0

Ungh! Still not working. I know I am being thick, but I just can't figure it out!

I still get the
"Unable to find an entry point named 'Add' in DLL 'MathFuncsDll.dll'."

Sigh! I'm sure this is down to how I have the program set up in VB.net, but can I figure it out?

NO! :(

Here is my C++ header code:

NOTE: the fact there is a namespace and Class in there.... if that's my issue?

```cpp
// MathFuncsDll.h

namespace MathFuncs
{
        extern "C"
        {
        class MyMathFuncs
    {
    public:
        // Returns a + b
        static __declspec(dllexport) double Add(double a, double b);

        // Returns a - b
        static __declspec(dllexport) double Subtract(double a, double b);

        // Returns a * b
        static __declspec(dllexport) double Multiply(double a, double b);

        // Returns a / b
        // Throws DivideByZeroException if b is 0
        static __declspec(dllexport) double Divide(double a, double b);
    };
        }
}
```

VB.NET code:

```vbnet
Imports System.Runtime.InteropServices

Public Class Form1
    <System.Runtime.InteropServices.DllImport("MathFuncsDll.dll", _
  SetLastError:=False)> _
    Public Shared Function Add(ByVal mynum As Decimal, ByVal mynum2 As Decim
    End Function

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.
        TextBox1.Text = Add(12, 3).ToString()
    End Sub

End Class
```

◀ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

Monday, February 08, 2010 12:59 PM

0

If i have to do this, i will do it this way

```
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

extern "C"
{


// Returns a + b
        __declspec(dllexport) double Add(double a, double b)

    {

      Returns a + b
     }
      __declspec(dllexport) double Subtract(double a, double b)

    {

      Returns a - b
     }
     __declspec(dllexport) double Multiply(double a, double b)
     {

      Returns a * b
     }

     // Throws DivideByZeroException if b is 0
      __declspec(dllexport) double Divide(double a, double b)
     {
       Returns a / b
     }

}
```
kaymaf

If that what you want, take it. If not, ignored it and no complain
CODE CONVERTER SITE : http://www.carlosag.net/Tools/CodeTranslator/ .

Monday, February 08, 2010 1:09 PM

Reply | Quote

kaymaf   44,530 Points

0

Ok. So putting these in a namespace and then a class is overcomplicating things?

Let me try it and get back to you.

Monday, February 08, 2010 1:26 PM

Reply | Quote

dustbin1_uk (Partner)   5 Points

1

OK!!!! Got it working at last, and I didn't need to trim out the Namespace or the Class from the DLL. I knew there was a way to do it!

But thanks to all the people who tried to help!

So here is the solution:

I created the DLL in C++ as I said before:

```
// MathFuncsDll.h
namespace MathFuncs
{
    class MyMathFuncs
    {
    public:
        // Returns a + b
        static __declspec(dllexport) double Add(double a, double b);

        // Returns a - b
        static __declspec(dllexport) double Subtract(double a, double b);

        // Returns a * b
        static __declspec(dllexport) double Multiply(double a, double b);

        // Returns a / b
        // Throws DivideByZeroException if b is 0
        static __declspec(dllexport) double Divide(double a, double b);
    };
}
```

I then used this tool (click for link) to view the DLL and get the exported function definition. In this case the Add function is: ?Add@MyMathFuncs@MathFuncs@@SANNN@Z

Now, I also realised that I was being really dumb in defining the function in VB.net using

Now, I also realised that I was being really dumb in defining the function in VB.net using Decimals, not doubles, so I changed that.

In the end the working code to call the DLL and use it's Add function inside the MyMathsFunc Class inside the MathFuncs C++ namespace is:

```
Public Class Form1

    Declare Function Add Lib "MathFuncsDll.dll" _
    Alias "?Add@MyMathFuncs@MathFuncs@@SANNN@Z" _
    (ByVal mynum As Double, ByVal mynum2 As Double) As Double

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Syst
    Handles Button1.Click
        MsgBox(Add(12, 7))
    End Sub

End Class
```

Marked as answer by dustbin1_uk   Tuesday, February 09, 2010 1:15 PM

Tuesday, February 09, 2010 1:09 PM

Reply | Quote

dustbin1_uk (Partner)   5 Points

0

Sign in
to vote

I would add to your comment that __stdcall is a MUST HAVE when trying to use it in VB/VB.net

```
static __declspec(dllexport) double __stdcall Add(double a, double b);
```

Regards,

rickygzz

Thursday, September 03, 2015 5:39 AM

Reply | Quote

RickGzz Home   0 Points