

©2009 Microsoft Corporation. All rights reserved.

C# Programming Guide

How to: Create and Use C# DLLs (C# Programming Guide)

A dynamic linking library (DLL) is linked to your program at run time. To demonstrate building and using a DLL, consider the following scenario:

- MathLibrary.DLL: The library file that contains the methods to be called at run time. In this example, the DLL contains two methods, Add and Multiply.
- Add.cs: The source file that contains the method Add(long i, long j). It returns the sum of its parameters.

 The class AddClass that contains the method Add is a member of the namespace UtilityMethods.
- Mult.cs: The source code that contains the method Multiply(long x, long y). It returns the product of its parameters. The class MultiplyClass that contains the method Multiply is also a member of the namespace UtilityMethods.
- TestCode.cs: The file that contains the Main method. It uses the methods in the DLL file to calculate the sum and the product of the run-time arguments.

Example

```
C#
 HILLETTI IS GOLDAN
 QDP LD SDEHIDWEAU IMCROV
 HIRVAGE FOLIAND GROWING
 HIRIBERLOU IMPEZIONO IS GOTROLI LIDAGO INT
 BIBLION I
 HIRIBINI - - - CHUI MI
 Ш
C#
 THE LETTER AND ADDRESS.
QCP LN SOLHIOMEN CIMERONI
 HIRSALOR IL GRAVIU MASSINS GRAV
 HIHIBEKLOL MY-ZONGJOJAKSO MAGJIJOKĆE NII
 HIHIT
 Ш
C#
 BID BITTLE SHARE STREET
жерансым элеккөет
 JOSEPH MERCE
 IIID-MATERIAL CALIMACA DE COMI
 HIRING NAME OF STARTIC TRANSPORTED TO THE LINE START OF A START OF A START
 HIIIIILIII U.∀I ...QJ∹.II IIII
 HIRIBITION NAMES TO SECURIOR IN CONTRACTOR OF THE BOARD O
```

1 of 3 7/15/2009 3:01 PM

```
HINDOOD COME TO SAME THE SAME
```

This file contains the algorithm that uses the DLL methods, <code>Add</code> and <code>Multiply</code>. It starts with parsing the arguments entered from the command line, <code>num1</code> and <code>num2</code>. Then it calculates the sum by using the <code>Add</code> method on the <code>AddClass</code> class, and the product by using the <code>Multiply</code> method on the <code>MultiplyClass</code> class.

Notice that the **using** directive at the beginning of the file enables you to use the unqualified class names to reference the DLL methods at compile time, as follows:

C#

```
U AMERIKA DIPOLAMBO NGME TINGME TIT
```

Otherwise, you have to use the fully qualified names, as follows:

C#

```
UNDARGO REMOVIDIAMBO O DOVIDIAMBO IQAZI FIIQME ETE
```

Execution

To run the program, enter the name of the EXE file, followed by two numbers, as follows:

```
TestCode 1234 5678
```

Output

```
Calling methods from MathLibrary.DLL: 1234 + 5678 = 6912 1234 * 5678 = 7006652
```

Compiling the Code

To build the file MathLibrary.DLL, compile the two files Add.cs and Mult.cs using the following command line:

```
csc /target:library /out:MathLibrary.DLL Add.cs Mult.cs
```

The /target:library [http://msdn.microsoft.com/en-us/library/e13syb43(VS.80).aspx] compiler option tells the compiler to output a DLL instead of an EXE file. The /out [http://msdn.microsoft.com/en-us/library /bw3t50f3(VS.80).aspx] compiler option followed by a file name is used to specify the DLL file name. Otherwise, the compiler uses the first file (Add.cs) as the name of the DLL.

To build the executable file, TestCode.exe, use the following command line:

```
csc /out:TestCode.exe /reference:MathLibrary.DLL TestCode.cs
```

The **/out** compiler option tells the compiler to output an EXE file and specifies the name of the output file ($_{\text{TestCode.exe}}$). This compiler option is optional. The <u>/reference</u> [http://msdn.microsoft.com/en-us/library /yabyz3h4(VS.80).aspx] compiler option specifies the DLL file or files that this program uses.

See Also

Tasks

<u>How to: Specify a Base Address for a DLL</u> [http://msdn.microsoft.com/en-us/library/w4w1x6a7(VS.80).aspx] **Concepts**

<u>C# Programming Guide</u> [http://msdn.microsoft.com/en-us/library/67ef8sbd(VS.80).aspx] <u>Creating a Class to Hold DLL Functions</u> [http://msdn.microsoft.com/en-us/library/khbsw73t(VS.80).aspx]

2 of 3 7/15/2009 3:01 PM

Tags:



Community Content

3 of 3