**如何：合并委托（多路广播委托）（C# 编程指南）**

本示例演示如何创建多播委托。 [委托](https://msdn.microsoft.com/zh-cn/library/900fyy8e.aspx)对象的一个有用属性是：可以使用 **+** 运算符将多个对象分配给一个委托实例。多播委托包含已分配委托的列表。在调用多播委托时，它会按顺序调用列表中的委托。只能合并相同类型的委托。

**-** 运算符可用于从多播委托中移除组件委托。

using System;

// Define a custom delegate that has a string parameter and returns void.

delegate void CustomDel(string s);

class TestClass

{

// Define two methods that have the same signature as CustomDel.

static void Hello(string s)

{

System.Console.WriteLine(" Hello, {0}!", s);

}

static void Goodbye(string s)

{

System.Console.WriteLine(" Goodbye, {0}!", s);

}

static void Main()

{

// Declare instances of the custom delegate.

CustomDel hiDel, byeDel, multiDel, multiMinusHiDel;

// In this example, you can omit the custom delegate if you

// want to and use Action<string> instead.

//Action<string> hiDel, byeDel, multiDel, multiMinusHiDel;

// Create the delegate object hiDel that references the

// method Hello.

hiDel = Hello;

// Create the delegate object byeDel that references the

// method Goodbye.

byeDel = Goodbye;

// The two delegates, hiDel and byeDel, are combined to

// form multiDel.

multiDel = hiDel + byeDel;

// Remove hiDel from the multicast delegate, leaving byeDel,

// which calls only the method Goodbye.

multiMinusHiDel = multiDel - hiDel;

Console.WriteLine("Invoking delegate hiDel:");

hiDel("A");

Console.WriteLine("Invoking delegate byeDel:");

byeDel("B");

Console.WriteLine("Invoking delegate multiDel:");

multiDel("C");

Console.WriteLine("Invoking delegate multiMinusHiDel:");

multiMinusHiDel("D");

}

}

/\* Output:

Invoking delegate hiDel:

Hello, A!

Invoking delegate byeDel:

Goodbye, B!

Invoking delegate multiDel:

Hello, C!

Goodbye, C!

Invoking delegate multiMinusHiDel:

Goodbye, D!

\*/