

1 System.Action<-T1,-T2,-T3,-T4,-T5> Delegate

```
2 [ILAsm]
3 .class public sealed System.Action`5<-T1,-T2,-T3, -T4, -T5> extends
4 System.MulticastDelegate
5
6 [C#]
7 public delegate void Action<in T1,in T2,in T3,in T4,in T5>(T1 arg1, T2
arg2, T3 arg3, T4 arg4, T5 arg5);
```

8 Assembly Info:

- 9 • *Name:* mscorlib
- 10 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 11 • *Version:* 4.0.0.0
- 12 • *Attributes:*
 - 13 ○ CLSCompliantAttribute(true)

14 Summary

15 Encapsulates a method that has five parameters and does not return a value.

16 Parameters

Parameter	Description
<i>arg1</i>	The first parameter of the method that this delegate encapsulates.
<i>arg2</i>	The second parameter of the method that this delegate encapsulates.
<i>arg3</i>	The third parameter of the method that this delegate encapsulates.
<i>arg4</i>	The fourth parameter of the method that this delegate encapsulates.
<i>arg5</i>	The fifth parameter of the method that this delegate encapsulates.

17 Inherits From: System.MulticastDelegate

18 **Library:** BCL

19 Description

20 You can use the `System.Action`5<T1,T2,T3,T4,T5>` delegate to pass a method as a
21 parameter without explicitly declaring a custom delegate. The encapsulated method
22 must correspond to the method signature that is defined by this delegate. This means
23 that the encapsulated method must have five parameters that are all passed to it by
24 value, and it must not return a value. Typically, such a method is used to perform an

1 operation.
2
3 [*Note:* To reference a method that has five parameters and returns a value, use the
4 generic `System.Func`6<T1, T2, T3, T4, T5, TResult>` delegate instead.
5
6]
7