

System.Action<-T1,-T2,-T3,-T4,-T5,-T6>

Delegate

```
[ILAsm]
.class public sealed System.Action`6<-T1,-T2,-T3, -T4, -T5, -T6> extends
System.MulticastDelegate

[C#]
public delegate void Action<in T1,in T2,in T3,in T4,in T5,in T6>(T1 arg1,
T2 arg2, T3 arg3, T4 arg4, T5 arg5, T6 arg6);
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 4.0.0.0
- *Attributes:*
 - CLSCompliantAttribute(true)

Summary

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

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.
<i>arg6</i>	The sixth parameter of the method that this delegate encapsulates.

Inherits From: System.MulticastDelegate

Library: BCL

Description

1 You can use the `System.Action`6<T1,T2,T3,T4,T5,T6>` delegate to pass a method as
2 a parameter without explicitly declaring a custom delegate. The encapsulated method
3 must correspond to the method signature that is defined by this delegate. This means
4 that the encapsulated method must have six parameters that are all passed to it by
5 value, and it must not return a value. Typically, such a method is used to perform an
6 operation.

7
8 [*Note:* To reference a method that has six parameters and returns a value, use the
9 generic `System.Func`7<T1,T2,T3,T4,T5,T6,TResult>` delegate instead.

10
11]
12