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
arg1	The first parameter of the method that this delegate encapsulates.
arg2	The second parameter of the method that this delegate encapsulates.
arg3	The third parameter of the method that this delegate encapsulates.
arg4	The fourth parameter of the method that this delegate encapsulates.
arg5	The fifth parameter of the method that this delegate encapsulates.
arg6	The sixth parameter of the method that this delegate encapsulates.

Inherits From: System.MulticastDelegate

Library: BCL

Description

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

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

]