# System.Action<-T1,-T2,-T3,-T4,-T5,-T6,-T7,-T8> Delegate

```
[ILAsm]
.class public sealed System.Action`8<-T1,-T2,-T3, -T4, -T5, -T6, -T7, -T8>
extends System.MulticastDelegate

[C#]
public delegate void Action<in T1,in T2,in T3,in T4,in T5,in T6,in T7,in
T8>(T1 arg1, T2 arg2, T3 arg3, T4 arg4, T5 arg5, T6 arg6, T7 arg7, T8
arg8);
```

**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:*
    - o CLSCompliantAttribute(true)

**Summary**

Encapsulates a method that has eight 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. |
| *arg7* | The seventh parameter of the method that this delegate encapsulates. |
| *arg8* | The eighth parameter of the method that this delegate encapsulates. |

**Inherits From: System.MulticastDelegate**

1

2 **Library:** BCL

3

4 **Description**

5 You can use the `System.Action`8<T1,T2,T3,T4,T5,T6,T7,T8>` delegate to pass a
6 method as a parameter without explicitly declaring a custom delegate. The encapsulated
7 method must correspond to the method signature that is defined by this delegate. This
8 means that the encapsulated method must have eight parameters that are all passed to
9 it by value, and it must not return a value. Typically, such a method is used to perform
10 an operation.
11
12 [*Note:* To reference a method that has eight parameters and returns a value, use the
13 generic `System.Func`9<T1,T2,T3,T4,T5,T6,T7,T8,TResult>` delegate instead.
14
15 ]

16