

1 System.MulticastDelegate Class

```
2 [ILAsm]  
3 .class public abstract serializable beforefieldinit MulticastDelegate  
4 extends System.Delegate  
  
5 [C#]  
6 public abstract class MulticastDelegate: Delegate
```

7 Assembly Info:

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

13 Summary

14 Represents a multicast delegate; that is, a delegate that can have more than one
15 element in its invocation list.

16 Inherits From: System.Delegate

17

18 **Library:** BCL

19

20 Description

21 `System.MulticastDelegate` is a special class. Compilers and other tools can derive
22 from this class, but you cannot derive from it explicitly. The same is true of the
23 `System.Delegate` class.

24

25 In addition to the methods that delegate types inherit from
26 `System.MulticastDelegate`, the common language infrastructure provides two special
27 methods: `BeginInvoke` and `EndInvoke`.

28

29 A `System.MulticastDelegate` has a linked list of delegates, called an invocation list,
30 consisting of one or more elements. When a multicast delegate is invoked, the delegates
31 in the invocation list are called synchronously in the order in which they appear. If an
32 error occurs during execution of the list then an exception is thrown.

33

1 MulticastDelegate(System.Object, 2 System.String) Constructor

```
3 [ILAsm]  
4 .method family hidebysig specialname rtspecialname instance void  
5 .ctor(object target, string 'method') cil managed  
  
6 [C#]  
7 protected MulticastDelegate (object target, string method)
```

8 Summary

9 Initializes a new instance of the System.MulticastDelegate class.

10 Parameters

Parameter	Description
<i>target</i>	The object on which <i>method</i> is defined.
<i>method</i>	The name of the method for which a delegate is created.

11 12 Description

13 This constructor cannot be used in application code. To create a delegate by specifying
14 the name of an instance method, use an overload of the
15 System.Delegate.CreateDelegate method that specifies a method name and a target
16 object. For example, the System.Delegate.CreateDelegate method overload creates a
17 delegate for an instance method with a specified name.

18 Exceptions

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

19 20 Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as System.Type.InvokeMember. Associated enumeration: System.Security.Permissions.

	ReflectionPermissionFlag.MemberAccess.
--	--

1

2

1 MulticastDelegate(System.Type, 2 System.String) Constructor

```
3 [ILAsm]  
4 .method family hidebysig specialname rtspecialname instance void  
5 .ctor(class System.Type target, string 'method') cil managed  
  
6 [C#]  
7 protected MulticastDelegate (Type target, string method)
```

8 Summary

9 Initializes a new instance of the System.MulticastDelegate class.

10 Parameters

Parameter	Description
<i>target</i>	The type of object on which <i>method</i> is defined.
<i>method</i>	The name of the static method for which a delegate is created.

11 12 Description

13 This constructor cannot be used in application code. To create a delegate by specifying
14 the name of a static method, use an overload of the System.Delegate.CreateDelegate
15 method that specifies a method name but does not specify a target object. For example,
16 the System.Delegate.CreateDelegate method overload creates a static delegate for a
17 method with a specified name.

18 Exceptions

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

19 20 Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as System.Type.InvokeMember. Associated enumeration: System.Security.Permissions.

	ReflectionPermissionFlag.MemberAccess
--	---------------------------------------

1

2

1
2 **MulticastDelegate.CombineImpl(System.Delegate)** Method
3

```
4 [ILAsm]  
5 .method family hidebysig virtual final instance class System.Delegate  
6 CombineImpl(class System.Delegate follow) cil managed  
  
7 [C#]  
8 protected override sealed Delegate CombineImpl (Delegate follow)
```

9 **Summary**

10 Combines this System.Delegate with the specified System.Delegate to form a new
11 delegate.

12 **Parameters**

Parameter	Description
<i>follow</i>	The delegate to combine with this delegate.

13
14 **Return Value**

15 A delegate that is the new root of the System.MulticastDelegate invocation list.

16 **Exceptions**

Exception	Condition
System.ArgumentException	<i>follow</i> does not have the same type as this instance.
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

17
18 **Permissions**

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as System.Type.InvokeMember. Associated enumeration: System.Security.Permissions.ReflectionPermissionFlag.MemberAccess

1

2

1 MulticastDelegate.Equals(System.Object)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual final instance bool Equals(object obj)  
5 cil managed  
  
6 [C#]  
7 public override sealed bool Equals (object obj)
```

8 Summary

9 Determines whether this multicast delegate and the specified object are equal.

10 Parameters

Parameter	Description
<i>obj</i>	The object to compare with this instance.

11 Return Value

13 true if *obj* and this instance have the same invocation lists; otherwise, false.

14 Description

15 Two delegates are equal if they are not null and are of exactly the same type, their
16 invocation lists contain the same number of elements, and every element in the
17 invocation list of the first delegate is equal to the corresponding element in the
18 invocation list of the second delegate.

19
20 Two invocation list elements are equal if they invoke the same instance method on the
21 same target instance, or they invoke the same static method.

22 Exceptions

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

23 Permissions

Permission	Description
------------	-------------

**System.Security.Permissions.
ReflectionPermission**

when invoked late-bound through mechanisms such as
System.Type.InvokeMember. Associated enumeration:
System.Security.Permissions.
ReflectionPermissionFlag.MemberAccess

1

2

1 MulticastDelegate.GetHashCode() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual final instance int32 GetHashCode() cil  
4 managed  
5 [C#]  
6 public override sealed int GetHashCode ()
```

7 Summary

8 Returns the hash code for this instance.

9 Return Value

10 A 32-bit signed integer hash code.

11 Exceptions

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

12

13 Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as <code>System.Type.InvokeMember</code> . Associated enumeration: <code>System.Security.Permissions.ReflectionPermissionFlag.MemberAccess</code>

14

15

1 MulticastDelegate.GetInvocationList() 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual final instance class System.Delegate[]  
5 GetInvocationList() cil managed  
  
6 [C#]  
7 public override sealed Delegate[] GetInvocationList ()
```

8 Summary

9 Returns the invocation list of this multicast delegate, in invocation order.

10 Return Value

11 An array of delegates whose invocation lists collectively match the invocation list of this
12 instance.

13 Description

14 The array contains a set of delegates, each with an invocation list of one entry. Invoking
15 these delegates sequentially, in the order they appear in the array, produces the same
16 results as invoking the current instance.

17 Exceptions

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

18 19 Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as System.Type.InvokeMember. Associated enumeration: System.Security.Permissions.ReflectionPermissionFlag.MemberAccess

20
21

1 MulticastDelegate.GetMethodImpl() Method

```
2 [ILAsm]  
3 .method family hidebysig virtual instance class  
4 System.ReflectionMethodInfo GetMethodImpl() cil managed
```

```
5 [C#]  
6 protected override System.Reflection.MethodInfo GetMethodImpl ()
```

7 Summary

8 Returns a static method represented by the current `System.MulticastDelegate`.

9 Return Value

10 A static method represented by the current `System.MulticastDelegate`.

11 Description

12 If the `System.MulticastDelegate` represents multiple methods, only one method is
13 returned. It is implementation-specific which method is returned.

14

1 2 MulticastDelegate.op_Equality(System.MulticastDelegate, System.MulticastDelegate) 3 4 Method

```
5 [ILAsm]  
6 .method public hidebysig specialname static bool op_Equality(class  
7 System.MulticastDelegate d1, class System.MulticastDelegate d2) cil  
8 managed  
  
9 [C#]  
10 public static bool op_Equality (MulticastDelegate d1, MulticastDelegate  
11 d2)
```

12 Summary

13 Determines whether two `System.MulticastDelegate` objects are equal.

14 Parameters

Parameter	Description
<i>d1</i>	The left operand.
<i>d2</i>	The right operand.

15 16 Return Value

17 `true` if *d1* and *d2* have the same invocation lists; otherwise, `false`.

18 Description

19 Two delegates are equal if they are not `null` and are of exactly the same type, their
20 invocation lists contain the same number of elements, and every element in the
21 invocation list of the first delegate is equal to the corresponding element in the
22 invocation list of the second delegate.

23
24 Two invocation list elements are equal if they invoke the same instance method on the
25 same target instance, or they invoke the same static method.

26 Exceptions

Exception	Condition
<code>System.MemberAccessException</code>	Cannot create an instance of an abstract class, or this

	member was invoked with a late-binding mechanism.
--	---

1

2 Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as <code>System.Type.InvokeMember</code> . Associated enumeration: <code>System.Security.Permissions.ReflectionPermissionFlag.MemberAccess</code>

3

4

1
2 **MulticastDelegate.op_Inequality(System.Mult**
3 **icastDelegate, System.MulticastDelegate)**
4 **Method**

```
5 [ILAsm]  
6 .method public hidebysig specialname static bool op_Inequality(class  
7 System.MulticastDelegate d1, class System.MulticastDelegate d2) cil  
8 managed  
  
9 [C#]  
10 public static bool op_Inequality (MulticastDelegate d1, MulticastDelegate  
11 d2)
```

12 **Summary**

13 Determines whether two System.MulticastDelegate objects are not equal.

14 **Parameters**

Parameter	Description
<i>d1</i>	The left operand.
<i>d2</i>	The right operand.

15
16 **Return Value**

17 true if *d1* and *d2* do not have the same invocation lists; otherwise, false.

18 **Description**

19 Two delegates are equal if they are not null and are of exactly the same type, their
20 invocation lists contain the same number of elements, and every element in the
21 invocation list of the first delegate is equal to the corresponding element in the
22 invocation list of the second delegate.

23
24 Two invocation list elements are equal if they invoke the same instance method on the
25 same target instance, or they invoke the same static method.

26 **Exceptions**

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this

	member was invoked with a late-binding mechanism.
--	---

1

2 **Permissions**

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as <code>System.Type.InvokeMember</code> . Associated enumeration: <code>System.Security.Permissions.ReflectionPermissionFlag.MemberAccess</code>

3

4

1
2 **MulticastDelegate.RemoveImpl(System.Delegate)**
3 **Method**

```
4 [ILAsm]  
5 .method family hidebysig virtual final instance class System.Delegate  
6 RemoveImpl(class System.Delegate 'value') cil managed  
  
7 [C#]  
8 protected override sealed Delegate RemoveImpl (Delegate value)
```

9 **Summary**

10 Removes an element from the invocation list of this System.MulticastDelegate that is
11 equal to the specified delegate.

12 **Parameters**

Parameter	Description
<i>value</i>	The delegate to search for in the invocation list.

13
14 **Return Value**

15 If *value* is found in the invocation list for this instance, then a new System.Delegate
16 without *value* in its invocation list; otherwise, this instance with its original invocation
17 list.

18 **Exceptions**

Exception	Condition
System.MemberAccessException	Cannot create an instance of an abstract class, or this member was invoked with a late-binding mechanism.

19
20 **Permissions**

Permission	Description
System.Security.Permissions.ReflectionPermission	when invoked late-bound through mechanisms such as System.Type.InvokeMember. Associated enumeration: System.Security.Permissions.ReflectionPermissionFlag.MemberAccess

1
2