

System.Attribute Class

```
[ILASM]
.class public abstract serializable Attribute extends
System.Object

[C#]
public abstract class Attribute
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Type Attributes:

- AttributeUsageAttribute(AttributeTargets.All, AllowMultiple=false, Inherited=true)

Summary

Serves as the base class for custom attributes.

Inherits From: System.Object

Library: BCL

Thread Safety: This type is safe for multithreaded operations.

Description

All attributes, whether built-in or user-defined, derive directly or indirectly from **System.Attribute**. Attributes inherit certain default behaviors: the attribute may be associated with any target element (see **System.AttributeTargets**); may or may not be inherited by a derived element; and multiple instances may or may not be allowed on the same target element. These behaviors are specified using **System.AttributeUsageAttribute**.

[*Note:* An attribute is an annotation that may be placed on an element of source code and used to store application-specific information at compile time. This information is stored in the metadata and can be accessed either during application execution, through a process known as reflection, or when another tool reads the metadata. Attributes might change the behavior of the application during execution, provide

1 transaction information about an object, or convey organizational
2 information to a designer.]

3
4 The CLI predefines some attribute types and uses them to control
5 runtime behavior. Some languages predefine attribute types to
6 represent language features not directly represented in the Common
7 Language Specification (CLS). User-defined attribute classes, inheriting
8 from **System.Attribute**, may also be created. The definition of such a
9 class includes the name of the attribute, its default behavior, and the
10 information to be stored.

11 Example

12

13 The following example creates and assigns multiple custom attributes
14 to a class. The attribute contains the name of the programmer and the
15 version number of the class.

16

17

```
18 using System;
19
20 [AttributeUsage(AttributeTargets.Class|
21                 AttributeTargets.Struct,
22                 AllowMultiple=true)]
23 public class Author: Attribute
24 {
25     string authorName;
26     public double verSion;
27
28     public Author(string name)
29     {
30         authorName = name;
31         verSion = 1.0;
32     }
33
34     public string getName()
35     {
36         return authorName;
37     }
38 }
39
40 [Author("Some Author")]
41 class FirstClass
42 {
43     /*...*/
44 }
45
46 class SecondClass // no Author attribute
47 {
48     /*...*/
49 }
50
51 [Author("Some Author"),
52     Author("Some Other Author", verSion=1.1)]
```

```

1      class ThirdClass
2      {
3          /*...*/
4      }
5
6      class AuthorInfo
7      {
8          public static void Main()
9          {
10             PrintAuthorInfo(typeof(FirstClass));
11             PrintAuthorInfo(typeof(SecondClass));
12             PrintAuthorInfo(typeof(ThirdClass));
13         }
14         public static void PrintAuthorInfo(Type type)
15         {
16             Console.WriteLine("Author information for {0}",
17                               type);
18             Attribute[] attributeArray =
19                 Attribute.GetCustomAttributes(type);
20             foreach(Attribute attrib in attributeArray)
21             {
22                 if (attrib is Author)
23                 {
24                     Author author = (Author)attrib;
25                     Console.WriteLine("    {0}, version {1:f}",
26                                       author.getName(),
27                                       author.verSion);
28                 }
29             }
30             Console.WriteLine();
31         }
32     }
33

```

```

34     The output is
35
36     Author information for FirstClass
37
38
39     Some Author, version 1.00
40
41
42     Author information for SecondClass
43

```

1
2
3
4
5
6
7
8
9

Author information for ThirdClass

Some Author, version 1.00

Some Other Author, version 1.10

10

1 Attribute() Constructor

```
2 [ILASM]  
3 family rtspecialname specialname instance void .ctor()  
4 [C#]  
5 protected Attribute()
```

6 Summary

7 Constructs a new instance of the **System.Attribute** class.

8

1 Attribute.Equals(System.Object) Method

```
2 [ILASM]  
3 .method public hidebysig virtual bool Equals(object obj)  
4 [C#]  
5 public override bool Equals(object obj)
```

6 Summary

7 Determines whether the current instance and the specified
8 **System.Object** represent the same type and value.

9 Parameters

10
11

Parameter	Description
<i>obj</i>	The System.Object to compare to the current instance.

12
13
14

Return Value

15 A **System.Boolean** where **true** indicates *obj* represents the same
16 type and value as the current instance. If *obj* is a null reference or is
17 not an instance of **System.Attribute**, returns **false**.

18 Description

19 [Note: This method overrides **System.Object.Equals**.]
20

The following member must be implemented if the Reflection library is present in the implementation.

Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type) Method

```
[ILASM]  
.method public hidebysig static class System.Attribute  
GetCustomAttribute(class System.Reflection.MemberInfo  
element, class System.Type attributeType)
```

```
[C#]  
public static Attribute GetCustomAttribute(MemberInfo  
element, Type attributeType)
```

Summary

Returns an instance of a specified custom attribute if a single instance of the attribute is in the metadata for the specified member.

Parameters

Parameter	Description
<i>element</i>	An instance of a type derived from System.Reflection.MemberInfo that describes a type member.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

Return Value

The single instance of **System.Attribute** of type *attributeType* that is applied to *element*. Returns **null** if the specified attribute was not found.

Description

[Note: If multiple instances of *attributeType* can be applied to *element*, use **System.Attribute.GetCustomAttributes**.]

Exceptions

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .

1
2
3

System.NotSupportedException	<i>element</i> does not represent a constructor, method, property, event, type, or field member.
System.Reflection.AmbiguousMatchException	More than one instance of the specified custom attribute was found.

The following member must be implemented if the Reflection library is present in the implementation.

Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type) Method

```
[ILASM]
.method public hidebysig static class System.Attribute
GetCustomAttribute(class System.Reflection.ParameterInfo
element, class System.Type attributeType)

[C#]
public static Attribute GetCustomAttribute(ParameterInfo
element, Type attributeType)
```

Summary

Returns an instance of a specified custom attribute if a single instance of the attribute is in the metadata for the specified parameter.

Parameters

Parameter	Description
<i>element</i>	A System.Reflection.ParameterInfo instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

Return Value

The single instance of **System.Attribute** of type *attributeType* that is applied to *element*. Returns **null** if the specified attribute was not found.

Description

[Note: If multiple instances of *attributeType* can be applied to *element*, use **System.Attribute.GetCustomAttributes.**]

Exceptions

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .

1
2
3

System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .
System.Reflection.AmbiguousMatchException	More than one instance of the specified custom attribute was found.

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttribute(System.Reflection.Module, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute  
7 GetCustomAttribute(class System.Reflection.Module element,  
8 class System.Type attributeType)  
9  
10 [C#]  
11 public static Attribute GetCustomAttribute(Module element,  
Type attributeType)
```

12 **Summary**

13 Returns an instance of a specified custom attribute if a single instance
14 of the attribute is in the metadata for the specified module.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Module instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

19 **Return Value**

21 The single instance of **System.Attribute** of type *attributeType* that is
22 applied to *element*. Returns **null** if the specified attribute was not
23 found.

24 **Description**

25 [Note: If multiple instances of *attributeType* can be applied to
26 *element*, use **System.Attribute.GetCustomAttributes.**]

27 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .

1
2
3

System.Reflection. AmbiguousMatchException	More than one instance of the specified custom attribute was found.
---	---

1 **The following member must be implemented if the RuntimeInfrastructure library is**
2 **present in the implementation.**

3 **Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute  
7 GetCustomAttribute(class System.Reflection.Assembly  
8 element, class System.Type attributeType)
```

```
9 [C#]  
10 public static Attribute GetCustomAttribute(Assembly  
11 element, Type attributeType)
```

12 **Summary**

13 Returns an instance of a specified custom attribute if a single instance
14 of the attribute is in the metadata for the specified assembly.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Assembly instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

19 **Return Value**

21 The single instance of **System.Attribute** of type *attributeType* that is
22 applied to *element*. Returns **null** if the specified attribute was not
23 found.

24 **Description**

25 [Note: If multiple instances of *attributeType* can be applied to
26 *element*, use **System.Attribute.GetCustomAttributes.**]

27 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .

1
2
3

System.Reflection. AmbiguousMatchException	More than one instance of the specified custom attribute was found.
---	---

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.MemberInfo, System.Type)** 5 **Method**

```
6 [ILASM]  
7 .method public hidebysig static class System.Attribute[]  
8 GetCustomAttributes(class System.Reflection.MemberInfo  
9 element, class System.Type type)  
  
10 [C#]  
11 public static Attribute[] GetCustomAttributes(MemberInfo  
12 element, Type type)
```

13 **Summary**

14 Returns an array of the instances of a specified custom attribute if the
15 attribute is in the metadata for the specified member.

16 **Parameters**

Parameter	Description
<i>element</i>	An instance of a type derived from System.Reflection.MemberInfo that describes a type member.
<i>type</i>	The System.Type of the custom attribute for which to check.

19 **Return Value**

20 An array of type *type* containing the instances that are applied to
21 *element*. The array includes instances of *type* that are inherited by
22 *element*, if any. Returns an empty array if the specified attribute was
23 not found.
24
25

26 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>type</i> is null .
System.ArgumentException	<i>type</i> is not a type derived from System.Attribute .
System.NotSupportedException	<i>element</i> does not represent a constructor.

1
2
3

	method, property, event, type, or field member.
--	---

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.MemberInfo) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute[]  
7 GetCustomAttributes(class System.Reflection.MemberInfo  
8 element)  
9 [C#]  
10 public static Attribute[] GetCustomAttributes(MemberInfo  
11 element)
```

12 **Summary**

13 Returns an array of all custom attributes in the metadata for the
14 specified member.

15 **Parameters**

Parameter	Description
<i>element</i>	An instance of a type derived from System.Reflection.MemberInfo that describes a type member.

19 **Return Value**

21 A **System.Attribute** array containing all custom attributes that are
22 applied to *element*. The array includes custom attributes that are
23 inherited by *element*, if any. Returns an empty array if no custom
24 attributes were found in the metadata for *element*.

25 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> is null .
System.NotSupportedException	<i>element</i> does not represent a constructor, method, property, event, type, or field member.

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.ParameterInfo, System.Type)** 5 **Method**

```
6 [ILASM]  
7 .method public hidebysig static class System.Attribute[]  
8 GetCustomAttributes(class System.Reflection.ParameterInfo  
9 element, class System.Type attributeType)
```

```
10 [C#]  
11 public static Attribute[] GetCustomAttributes(ParameterInfo  
12 element, Type attributeType)
```

13 **Summary**

14 Returns an array of the instances of a specified custom attribute if the
15 attribute is in the metadata for the specified parameter.

16 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.ParameterInfo instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

19 **Return Value**

20 An array of type *attributeType* containing the instances that are
21 applied to *element*. The array includes any inherited instances of
22 *attributeType*. Returns an empty array if the specified attribute was
23 not found.
24
25

26 **Description**

27 If *element* represents a method parameter, the array returned by
28 **System.Attribute.GetCustomAttributes** includes any *attributeType*
29 instances for the parameter *element* in the base methods.

30 **Exceptions**

Exception	Condition
-----------	-----------

1
2
3

System.ArgumentNullException	<i>element</i> or <i>type</i> is null .
System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.Module, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute[]  
7 GetCustomAttributes(class System.Reflection.Module element,  
8 class System.Type attributeType)  
9  
10 [C#]  
11 public static Attribute[] GetCustomAttributes(Module  
element, Type attributeType)
```

12 **Summary**

13 Returns an array of the instances of a specified custom attribute if the
14 attribute is in the metadata for the specified module.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Module instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

19 **Return Value**

21 An array of type *attributeType* containing the instances that are
22 applied to *element*. The array includes any inherited instances of
23 *attributeType*. Returns an empty array if the specified attribute was
24 not found.

25 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>type</i> is null .
System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .

1 **The following member must be implemented if the RuntimeInfrastructure library is**
2 **present in the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.Assembly, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute[]  
7 GetCustomAttributes(class System.Reflection.Assembly  
8 element, class System.Type attributeType)  
9  
10 [C#]  
11 public static Attribute[] GetCustomAttributes(Assembly  
element, Type attributeType)
```

12 **Summary**

13 Returns an array of the instances of a specified custom attribute if the
14 attribute is in the metadata for the specified assembly.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Assembly instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

18 **Return Value**

19 An array of type *attributeType* containing the instances that are
20 applied to *element*. The array includes any inherited instances of
21 *attributeType*. Returns an empty array if the specified attribute was
22 not found.
23
24

25 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>type</i> is null .
System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute .

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.ParameterInfo) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute[]  
7 GetCustomAttributes(class System.Reflection.ParameterInfo  
8 element)  
9 [C#]  
10 public static Attribute[] GetCustomAttributes(ParameterInfo  
11 element)
```

12 **Summary**

13 Returns an array of all custom attributes in the metadata for the
14 specified parameter.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.ParameterInfo instance.

19 **Return Value**

21 A **System.Attribute** array containing all custom attributes that are
22 applied to *element*. The array includes any inherited custom attributes.
23 Returns an empty array if no custom attributes were found in the
24 metadata for *element*.

25 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> is null .

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.Module) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute[]  
7 GetCustomAttributes(class System.Reflection.Module element)  
  
8 [C#]  
9 public static Attribute[] GetCustomAttributes(Module  
10 element)
```

11 **Summary**

12 Returns an array of all custom attributes in the metadata for the
13 specified module.

14 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Module instance.

18 **Return Value**

20 A **System.Attribute** array containing all custom attributes that are
21 applied to *element*. The array includes any inherited custom attributes.
22 Returns an empty array if no custom attributes were found in the
23 metadata for *element*.

24 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> is null .

1 **The following member must be implemented if the RuntimeInfrastructure library is**
2 **present in the implementation.**

3 **Attribute.GetCustomAttributes(System.Re** 4 **flexion.Assembly) Method**

```
5 [ILASM]  
6 .method public hidebysig static class System.Attribute[]  
7 GetCustomAttributes(class System.Reflection.Assembly  
8 element)  
9 [C#]  
10 public static Attribute[] GetCustomAttributes(Assembly  
11 element)
```

12 **Summary**

13 Returns an array of all custom attributes in the metadata for the
14 specified assembly.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Assembly instance.

19 **Return Value**

21 A **System.Attribute** array containing all custom attributes that are
22 applied to *element*. The array includes any inherited custom attributes.
23 Returns an empty array if no custom attributes were found in the
24 metadata for *element*.

25 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> is null .

1 Attribute.GetHashCode() Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4 [C#]  
5 public override int GetHashCode()
```

6 Summary

7 Generates a hash code for the current instance.

8 Return Value

9

10 A **System.Int32** containing the hash code for the current instance.

11 Description

12 The algorithm used to generate the hash code is unspecified.

13

14 [*Note:* This method overrides **System.Object.GetHashCode.**]

15

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.IsDefined(System.Reflection.Me** 4 **mberInfo, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static bool IsDefined(class  
7 System.Reflection.MemberInfo element, class System.Type  
8 attributeType)  
9  
10 [C#]  
11 public static bool IsDefined(MemberInfo element, Type  
attributeType)
```

12 **Summary**

13 Returns a **System.Boolean** value indicating whether a specified
14 custom attribute is present in the metadata for the specified member.

15 **Parameters**

Parameter	Description
<i>element</i>	An instance of a type derived from System.Reflection.MemberInfo that describes a type member.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

18 **Return Value**

19 **true** if a custom attribute of type *attributeType* is applied to *element*
20 either directly or through inheritance; otherwise, **false**.

21 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not derived from System.Attribute .
System.NotSupportedException	<i>element</i> is not a constructor, method, property, event, type, or field.

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.IsDefined(System.Reflection.Par** 4 **ameterInfo, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static bool IsDefined(class  
7 System.Reflection.ParameterInfo element, class System.Type  
8 attributeType)  
9  
10 [C#]  
11 public static bool IsDefined(ParameterInfo element, Type  
attributeType)
```

12 **Summary**

13 Returns a **System.Boolean** value indicating whether a specified
14 custom attribute is present in the metadata for the specified
15 parameter.

16 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.ParameterInfo instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

20 **Return Value**

22 **true** if a custom attribute of type *attributeType* is applied to *element*
23 either directly or through inheritance; otherwise, **false**.

24 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not derived from System.Attribute .

1 **The following member must be implemented if the Reflection library is present in**
2 **the implementation.**

3 **Attribute.IsDefined(System.Reflection.Module, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static bool IsDefined(class  
7 System.Reflection.Module element, class System.Type  
8 attributeType)  
9  
10 [C#]  
11 public static bool IsDefined(Module element, Type  
attributeType)
```

12 **Summary**

13 Returns a **System.Boolean** value indicating whether a specified
14 custom attribute is present in the metadata for the specified module.

15 **Parameters**

16
17

Parameter	Description
<i>element</i>	A System.Reflection.Module instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

18

19 **Return Value**

20

21 **true** if a custom attribute of type *attributeType* is applied to *element*;
22 otherwise, **false**.

23 **Exceptions**

24

25

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not derived from System.Attribute .

26

27

28

1 **The following member must be implemented if the RuntimeInfrastructure library is**
2 **present in the implementation.**

3 **Attribute.IsDefined(System.Reflection.Assembly, System.Type) Method**

```
5 [ILASM]  
6 .method public hidebysig static bool IsDefined(class  
7 System.Reflection.Assembly element, class System.Type  
8 attributeType)  
9  
10 [C#]  
11 public static bool IsDefined(Assembly element, Type  
attributeType)
```

12 **Summary**

13 Returns a **System.Boolean** value indicating whether a specified
14 custom attribute is present in the metadata for the specified assembly.

15 **Parameters**

Parameter	Description
<i>element</i>	A System.Reflection.Assembly instance.
<i>attributeType</i>	The System.Type of the custom attribute for which to check.

18 **Return Value**

19 **true** if a custom attribute of type *attributeType* is applied to *element*;
20 otherwise, **false**.

21 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null .
System.ArgumentException	<i>attributeType</i> is not derived from System.Attribute .