

1 System.Attribute Class

```
2 [ILAsm]  
3 .class public abstract serializable Attribute extends System.Object  
4 [C#]  
5 public abstract class Attribute
```

6 Assembly Info:

- 7 • *Name:* mscorlib
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Type Attributes:

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

14 Summary

15 Serves as the base class for custom attributes.

16 Inherits From: System.Object

17
18 **Library:** BCL

19
20 **Thread Safety:** This type is safe for multithreaded operations.

22 Description

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

29
30 [*Note:* An attribute is an annotation that can be placed on an element of source code
31 and used to store application-specific information at compile time. This information is
32 stored in the metadata and can be accessed either during application execution, through
33 a process known as reflection, or when another tool reads the metadata. Attributes
34 might change the behavior of the application during execution, provide transaction
35 information about an object, or convey organizational information to a designer.]

36
37
38
39 The CLI predefines some attribute types and uses them to control runtime behavior.

1 Some languages predefine attribute types to represent language features not directly
2 represented in the Common Language Specification (CLS). User-defined attribute
3 classes, inheriting from `System.Attribute`, can also be created. The definition of such a
4 class includes the name of the attribute, its default behavior, and the information to be
5 stored.

6 Example

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

```
9  
10 [C#]  
  
11 using System;  
12  
13 [AttributeUsage(AttributeTargets.Class |  
14                 AttributeTargets.Struct,  
15                 AllowMultiple=true)]  
16 public class Author: Attribute  
17 {  
18     string authorName;  
19     public double verSion;  
20  
21     public Author(string name)  
22     {  
23         authorName = name;  
24         verSion = 1.0;  
25     }  
26  
27     public string getName()  
28     {  
29         return authorName;  
30     }  
31 }  
32  
33 [Author("Some Author")]  
34 class FirstClass  
35 {  
36     /*...*/  
37 }  
38  
39 class SecondClass // no Author attribute  
40 {  
41     /*...*/  
42 }  
43  
44 [Author("Some Author"),  
45     Author("Some Other Author", verSion=1.1)]  
46 class ThirdClass  
47 {  
48     /*...*/  
49 }  
50  
51 class AuthorInfo  
52 {  
53     public static void Main()
```

```

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

```

25 The output is

```

26 Author information for FirstClass
27
28 Some Author, version 1.00
29
30
31 Author information for SecondClass
32
33
34 Author information for ThirdClass
35
36
37 Some Author, version 1.00
38
39
40 Some Other Author, version 1.10
41
42
43
44

```

45

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 `System.Object` represent the
8 same type and value.

9 Parameters

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

10

11 Return Value

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

15 Description

16 [Note: This method overrides `System.Object.Equals`.

17

18]

19

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

3 Attribute.GetCustomAttribute(System.Reflection. 4 on.Assembly, System.Type) Method

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

12 Summary

13 Returns an instance of a specified custom attribute if a single instance of the attribute is
14 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.

17 Return Value

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

20 Description

21 [Note: If multiple instances of *attributeType* can be applied to *element*, use
22 System.Attribute.GetCustomAttributes.]
23
24

25 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.
System.Reflection.AmbiguousMatchException	More than one instance of the specified custom attribute was found.

1

2

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

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

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

[C#]
public static Attribute GetCustomAttribute(Module 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 module.

Parameters

Parameter	Description
<i>element</i>	A <code>System.Reflection.Module</code> instance.
<i>attributeType</i>	The <code>System.Type</code> 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 <code>null</code> .

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

2

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 <code>System.Reflection.ParameterInfo</code> instance.
<i>attributeType</i>	The <code>System.Type</code> 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 <code>null</code> .

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

2

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 <code>System.Reflection.MemberInfo</code> that describes a type member.
<i>attributeType</i>	The <code>System.Type</code> 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 <code>null</code> .

System.ArgumentException	<i>attributeType</i> is not a type derived from System.Attribute.
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.

1

2

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

3 **Attribute.GetCustomAttributes(System.Reflection.Assembly) Method** 4

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

10 **Summary**

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

12 **Parameters**

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

14 **Return Value**

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

18 **Exceptions**

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

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

3 **Attribute.GetCustomAttributes(System.Reflection.Module) Method** 4

```
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 element)
```

10 **Summary**

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

12 **Parameters**

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

14 **Return Value**

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

18 **Exceptions**

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

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

3 **Attribute.GetCustomAttributes(System.Reflec** 4 **tion.ParameterInfo) Method**

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

10 **Summary**

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

12 **Parameters**

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

14 **Return Value**

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

18 **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.Reflection.Assembly, System.Type) Method** 4

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

12 **Summary**

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

17 **Return Value**

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

21 **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.

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

Attribute.GetCustomAttributes(System.Reflection.Module, System.Type) Method

```
[ILAsm]
.method public hidebysig static class System.Attribute[]
GetCustomAttributes(class System.Reflection.Module element, class
System.Type attributeType)

[C#]
public static Attribute[] GetCustomAttributes(Module element, Type
attributeType)
```

Summary

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

Parameters

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

Return Value

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

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 <code>System.Attribute</code> .

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

3 `Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)` Method 4

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

12 Summary

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

15 Parameters

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

17 Return Value

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

21 Description

22 If *element* represents a method parameter, the array returned by
23 `System.Attribute.GetCustomAttributes` includes any *attributeType* instances for the
24 parameter *element* in the base methods.

25 Exceptions

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

System.ArgumentException

attributeType is not a type derived from
System.Attribute.

1

2

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

3 **Attribute.GetCustomAttributes(System.Reflection.MemberInfo) Method** 4

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

10 **Summary**

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

12 **Parameters**

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

14 **Return Value**

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

18 **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.

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

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

```
[ILAsm]
.method public hidebysig static class System.Attribute[]
GetCustomAttributes(class System.Reflection.MemberInfo element, class
System.Type type)

[C#]
public static Attribute[] GetCustomAttributes(MemberInfo element, Type
type)
```

Summary

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

Parameters

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

Return Value

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

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 <code>System.Attribute</code> .
System.NotSupportedException	<i>element</i> does not represent a constructor, method, property, event, type, or field member.

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 A `System.Int32` containing the hash code for the current instance.

10 Description

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

12

13 [*Note:* This method overrides `System.Object.GetHashCode()`.]

14

15

16

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

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

```
[ILAsm]
.method public hidebysig static bool IsDefined(class
System.Reflection.MemberInfo element, class System.Type attributeType)

[C#]
public static bool IsDefined(MemberInfo element, Type attributeType)
```

Summary

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

Parameters

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

Return Value

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

Exceptions

Exception	Condition
System.ArgumentNullException	<i>element</i> or <i>attributeType</i> is null.
System.ArgumentException	<i>attributeType</i> is not derived from <code>System.Attribute</code> .
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 the**
2 **implementation.**

3 Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type) Method 4

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

10 Summary

11 Returns a `System.Boolean` value indicating whether a specified custom attribute is
12 present in the metadata for the specified parameter.

13 Parameters

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

15 Return Value

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

18 Exceptions

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

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

3 Attribute.IsDefined(System.Reflection.Module, 4 element, System.Type) Method

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

10 Summary

11 Returns a `System.Boolean` value indicating whether a specified custom attribute is
12 present in the metadata for the specified module.

13 Parameters

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

15 Return Value

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

17 Exceptions

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

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 attributeType)  
8 [C#]  
9 public static bool IsDefined(Assembly element, Type attributeType)
```

10 Summary

11 Returns a `System.Boolean` value indicating whether a specified custom attribute is
12 present in the metadata for the specified assembly.

13 Parameters

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

14

15 Return Value

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

17 Exceptions

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

18

19