Attribute Class

Reference

Definition

Namespace: System

Assembly: System.Runtime.dll

Represents the base class for custom attributes.

C#

```
[System.AttributeUsage(System.AttributeTargets.All,
AllowMultiple=false, Inherited=true)]
public abstract class Attribute
```

Inheritance Object → Attribute

Derived Microsoft.Extensions.Compliance.Classification.DataClassificationAttribute
Microsoft.Extensions.Configuration.ConfigurationKeyNameAttribute
Microsoft.Extensions.Configuration.UserSecrets.UserSecretsIdAttribute
Microsoft.Extensions.DependencyInjection.ActivatorUtilitiesConstructorAttribute
Microsoft.Extensions.DependencyInjection.FromKeyedServicesAttribute
More...

Attributes AttributeUsageAttribute

Examples

The following code example demonstrates the usage of Attribute.

C#

```
using System;
using System.Reflection;

// An enumeration of animals. Start at 1 (0 = uninitialized).
public enum Animal {
    // Pets.
    Dog = 1,
    Cat,
    Bird,
}

// A custom attribute to allow a target to have a pet.
public class AnimalTypeAttribute : Attribute {
    // The constructor is called when the attribute is set.
    public AnimalTypeAttribute(Animal pet) {
        thePet = pet;
    }

    // Keep a variable internally ...
    protected Animal thePet;

    // .. and show a copy to the outside world.
    public Animal Pet {
        get { return thePet; }
        set { thePet = value; }
    }
}

// A test class where each method has its own pet.
class AnimalTypeTestClass {
    [AnimalType(Animal.Dog)]
    public void DogMethod() {}

    [AnimalType(Animal.Cat)]
    public void CatMethod() {}

    [AnimalType(Animal.Bird)]
    public void BirdMethod() {}
}

class DemoClass {
    static void Main(string[] args) {
        AnimalTypeTestClass testClass = new AnimalTypeTestClass();
        Type type = testClass.GetType();
        // Iterate through all the methods of the class.
        foreach(MethodInfo mInfo in type.GetMethods()) {
            // Iterate through all the Attributes for each method.
            foreach (Attribute attr in
                Attribute.GetCustomAttributes(mInfo)) {
                // Check for the AnimalType attribute.
                if (attr.GetType() == typeof(AnimalTypeAttribute))
                    Console.WriteLine(
                        "Method {0} has a pet {1} attribute.",
                        mInfo.Name, ((AnimalTypeAttribute)attr).Pet);
            }
        }
    }
}
/*
 * Output:
 * Method DogMethod has a pet Dog attribute.
 * Method CatMethod has a pet Cat attribute.
 * Method BirdMethod has a pet Bird attribute.
 */
```

Remarks

The Attribute class associates predefined system information or user-defined custom information with a target element. A target element can be an assembly, class, constructor, delegate, enum, event, field, interface, method, portable executable file module, parameter, property, return value, struct, or another attribute.

Information provided by an attribute is also known as metadata. Metadata can be examined at run time by your application to control how your program processes data, or before run time by external tools to control how your application itself is processed or maintained. For example, .NET predefines and uses attribute types to control run-time behavior, and some programming languages use attribute types to represent language features not directly supported by the .NET common type system.

All attribute types derive directly or indirectly from the Attribute class. Attributes can be applied to any target element; multiple attributes can be applied to the same target element; and attributes can be inherited by an element derived from a target element. Use the AttributeTargets class to specify the target element to which the attribute is applied.

The Attribute class provides convenient methods to retrieve and test custom attributes. For more information about using attributes, see Applying Attributes and Attributes.

Constructors

Attribute()	Initializes a new instance of the Attribute class.

Properties

TypeId	When implemented in a derived class, gets a unique identifier for this Attribute.

Methods

Equals(Object)	Returns a value that indicates whether this instance is equal to a specified object.

GetCustom Attribute(Assembly, Type)	Retrieves a custom attribute applied to a specified assembly. Parameters specify the assembly and the type of the custom attribute to search for.
GetCustom Attribute(Assembly, Type, Boolean)	Retrieves a custom attribute applied to an assembly. Parameters specify the assembly, the type of the custom attribute to search for, and an ignored search option.
GetCustomAttribute(Member Info, Type)	Retrieves a custom attribute applied to a member of a type. Parameters specify the member, and the type of the custom attribute to search for.
GetCustomAttribute(Member Info, Type, Boolean)	Retrieves a custom attribute applied to a member of a type. Parameters specify the member, the type of the custom attribute to search for, and whether to search ancestors of the member.
GetCustomAttribute(Module, Type)	Retrieves a custom attribute applied to a module. Parameters specify the module, and the type of the custom attribute to search for.
GetCustomAttribute(Module, Type, Boolean)	Retrieves a custom attribute applied to a module. Parameters specify the module, the type of the custom attribute to search for, and an ignored search option.
GetCustom Attribute(ParameterInfo, Type)	Retrieves a custom attribute applied to a method parameter. Parameters specify the method parameter, and the type of the custom attribute to search for.
GetCustom Attribute(ParameterInfo, Type, Boolean)	Retrieves a custom attribute applied to a method parameter. Parameters specify the method parameter, the type of the custom attribute to search for, and whether to search ancestors of the method parameter.
GetCustom Attributes(Assembly)	Retrieves an array of the custom attributes applied to an assembly. A parameter specifies the assembly.
GetCustom Attributes(Assembly, Boolean)	Retrieves an array of the custom attributes applied to an assembly. Parameters specify the assembly, and an ignored search option.
GetCustom Attributes(Assembly, Type)	Retrieves an array of the custom attributes applied to an assembly. Parameters specify the assembly, and the type of the custom attribute to search for.
GetCustom Attributes(Assembly, Type, Boolean)	Retrieves an array of the custom attributes applied to an assembly. Parameters specify the assembly, the type of the custom attribute to search for, and an ignored search option.
GetCustomAttributes(Member Info)	Retrieves an array of the custom attributes applied to a member of a type. A parameter specifies the member.

GetCustomAttributes(MemberInfo, Boolean)	Retrieves an array of the custom attributes applied to a member of a type. Parameters specify the member, the type of the custom attribute to search for, and whether to search ancestors of the member.
GetCustomAttributes(MemberInfo, Type)	Retrieves an array of the custom attributes applied to a member of a type. Parameters specify the member, and the type of the custom attribute to search for.
GetCustomAttributes(MemberInfo, Type, Boolean)	Retrieves an array of the custom attributes applied to a member of a type. Parameters specify the member, the type of the custom attribute to search for, and whether to search ancestors of the member.
GetCustomAttributes(Module)	Retrieves an array of the custom attributes applied to a module. A parameter specifies the module.
GetCustomAttributes(Module, Boolean)	Retrieves an array of the custom attributes applied to a module. Parameters specify the module, and an ignored search option.
GetCustomAttributes(Module, Type)	Retrieves an array of the custom attributes applied to a module. Parameters specify the module, and the type of the custom attribute to search for.
GetCustomAttributes(Module, Type, Boolean)	Retrieves an array of the custom attributes applied to a module. Parameters specify the module, the type of the custom attribute to search for, and an ignored search option.
GetCustomAttributes(ParameterInfo)	Retrieves an array of the custom attributes applied to a method parameter. A parameter specifies the method parameter.
GetCustomAttributes(ParameterInfo, Boolean)	Retrieves an array of the custom attributes applied to a method parameter. Parameters specify the method parameter, and whether to search ancestors of the method parameter.
GetCustomAttributes(ParameterInfo, Type)	Retrieves an array of the custom attributes applied to a method parameter. Parameters specify the method parameter, and the type of the custom attribute to search for.
GetCustomAttributes(ParameterInfo, Type, Boolean)	Retrieves an array of the custom attributes applied to a method parameter. Parameters specify the method parameter, the type of the custom attribute to search for, and whether to search ancestors of the method parameter.
GetHashCode()	Returns the hash code for this instance.
GetType()	Gets the Type of the current instance. (Inherited from Object)
IsDefaultAttribute()	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.

IsDefined(Assembly, Type)	Determines whether any custom attributes are applied to an assembly. Parameters specify the assembly, and the type of the custom attribute to search for.
IsDefined(Assembly, Type, Boolean)	Determines whether any custom attributes are applied to an assembly. Parameters specify the assembly, the type of the custom attribute to search for, and an ignored search option.
IsDefined(MemberInfo, Type)	Determines whether any custom attributes are applied to a member of a type. Parameters specify the member, and the type of the custom attribute to search for.
IsDefined(MemberInfo, Type, Boolean)	Determines whether any custom attributes are applied to a member of a type. Parameters specify the member, the type of the custom attribute to search for, and whether to search ancestors of the member.
IsDefined(Module, Type)	Determines whether any custom attributes of a specified type are applied to a module. Parameters specify the module, and the type of the custom attribute to search for.
IsDefined(Module, Type, Boolean)	Determines whether any custom attributes are applied to a module. Parameters specify the module, the type of the custom attribute to search for, and an ignored search option.
IsDefined(ParameterInfo, Type)	Determines whether any custom attributes are applied to a method parameter. Parameters specify the method parameter, and the type of the custom attribute to search for.
IsDefined(ParameterInfo, Type, Boolean)	Determines whether any custom attributes are applied to a method parameter. Parameters specify the method parameter, the type of the custom attribute to search for, and whether to search ancestors of the method parameter.
Match(Object)	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.
MemberwiseClone()	Creates a shallow copy of the current Object. (Inherited from Object)
ToString()	Returns a string that represents the current object. (Inherited from Object)

Applies to

Product	Versions
.NET	Core 1.0, Core 1.1, Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7, 8

Product	Versions
.NET Framework	1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8, 4.8.1
.NET Standard	1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 2.0, 2.1
UWP	10.0
Xamarin.iOS	10.8
Xamarin.Mac	3.0

Thread Safety

This type is thread safe.

See also

- Applying Attributes
- Extending Metadata Using Attributes