

# System.Diagnostics.ConditionalAttribute Class

```
[ILAsm]
.class public sealed serializable ConditionalAttribute extends
System.Attribute

[C#]
public sealed class ConditionalAttribute: Attribute
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Type Attributes:

- AttributeUsageAttribute(AttributeTargets.Method, AllowMultiple=true, Inherited=true)

## Summary

Indicates to compilers that a method is callable if and only if a specified pre-processing identifier has been defined on the method.

## Inherits From: System.Attribute

## Library: BCL

**Thread Safety:** All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

## Description

[*Note:* A `System.Diagnostics.ConditionalAttribute`, which has an associated condition `System.Diagnostics.ConditionalAttribute.ConditionString`, can be attached to the definition of a method, creating a *conditional method*. Thereafter, when a compiler encounters a call to that method, it might choose to ignore the call unless a compilation variable is defined at the site of the call, with a value that matches in a case-sensitive manner the `System.Diagnostics.ConditionalAttribute.ConditionString` supplied to the `System.Diagnostics.ConditionalAttribute`.

Note that compilers might provide several techniques to define such compilation variables, such as:

- 1       • compiler command-line switches (for example, /define:DEBUG )
- 2       • environment variables in the operating system shell (for example, SET DEBUG=1)
- 3       • as pragmas in the source code (for example, #define DEBUG, to define the
- 4        compilation variable, or #undef DEBUG to undefine it)
- 5 CLS-Compliant compilers are permitted to ignore uses of the
- 6 System.Diagnostics.ConditionalAttribute.
- 7
- 8 ]

## 9   **Example**

10       The following example demonstrates the use of

11       System.Diagnostics.ConditionalAttribute with a particular compiler that supports

12       the use of this attribute. The

13       System.Diagnostics.ConditionalAttribute.ConditionString property of the current

14       attribute is initialized as "DEBUG".

```
15  
16     [C#]  
  
17     using System;  
18     using System.Diagnostics;  
19  
20     public class MyClass {  
21  
22         [ConditionalAttribute("DEBUG")]  
23         public static void Display() {  
24  
25             Console.WriteLine("Compiled with DEBUG");  
26         }  
27     }  
28  
29     public class TestCondition {  
30  
31         public static void Main() {  
32  
33             Console.WriteLine("How was this compiled?");  
34             MyClass.Display();  
35             Console.WriteLine("<eop>");  
36         }  
37     }  
38
```

39       When this code is compiled with the compilation-variable DEBUG defined at the callsite, the

40       output when run is

```
41  
42     How was this compiled?  
43  
44  
45     Compiled with DEBUG  
46
```

```
1
2 <eop>
3
4
5 When this code is compiled without the compilation-variable DEBUG defined at the callsite,
6 the output when run is
7
8 How was this compiled?
9
10
11 <eop>
12
13
```

# ConditionalAttribute(System.String)

## Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(string
conditionString)

[C#]
public ConditionalAttribute(string conditionString)
```

## Summary

Constructs and initializes a new instance of the `System.Diagnostics.ConditionalAttribute` class.

## Parameters

Parameter	Description
<i>conditionString</i>	A <code>System.String</code> that contains the pre-processing identifier that makes callable the target method of the current instance.

## Description

This constructor initializes the `System.Diagnostics.ConditionalAttribute.ConditionString` property of the current instance using *conditionString*.

# ConditionalAttribute.ConditionString Property

```
[ILAsm]  
.property string ConditionString { public hidebysig specialname instance  
string get_ConditionString() }  
  
[C#]  
public string ConditionString { get; }
```

## Summary

Gets the `System.String` that contains the pre-processing identifier that makes callable the target method of the current instance.

## Property Value

A `System.String` that contains the pre-processing identifier that makes callable the target method of the current instance.

## Description

This property is read-only.