

# System.Runtime.CompilerServices.CompilationRelaxationsAttribute Class

```
[ILAsm]
.class public serializable beforefieldinit CompilationRelaxationsAttribute
extends System.Attribute

[C#]
public class CompilationRelaxationsAttribute: 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.Assembly | AttributeTargets.Class | AttributeTargets.Struct | AttributeTargets.Constructor | AttributeTargets.Method, AllowMultiple=false, Inherited=false)

## Summary

This attribute allows the user to specify whether the ordering of instruction checking is strict or relaxed. It also allows the user to specify whether or not strings are interned.

## Inherits From: System.Attribute

**Library:** RuntimeInfrastructure

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

## Description

Optimizers are granted additional latitude for relaxed exceptions. A method is relaxed for certain kinds of exceptions if the innermost custom attribute `System.Runtime.CompilerServices.CompilationRelaxationAttribute` pertaining to that kind of exception is present and specifies to relax those kinds of exceptions. (Here, "innermost" means inspecting the method, its class, and its assembly, in that order.)

[*Note:* For background and implementation information for relaxed exception handling, plus examples, see Annex F of Partition VI.]

1  
2 [Note: See the `System.Runtime.CompilerServices.CompilationRelaxations`  
3 enumeration for a description of the kinds of exceptions that can be relaxed or made  
4 strict.]  
5  
6  
7

8 The runtime can create one string object for each unique string literal, rather than  
9 making multiple copies. This is called string interning, which internally requires building  
10 auxiliary tables that consume memory resources. String interning can be enabled or  
11 disabled at the assembly level via this attribute.  
12

# CompilationRelaxationsAttribute(System.Runtime.CompilerServices.CompilationRelaxations) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(valuetype
System.Runtime.CompilerServices.CompilationRelaxations relaxations)

[C#]
public CompilationRelaxationsAttribute(CompilationRelaxations relaxations)
```

## Summary

Constructs and initializes a new instance of the `System.Runtime.CompilerServices.CompilationRelaxationsAttribute` class with the specified `System.Runtime.CompilerServices.CompilationRelaxations` value.

## Parameters

Parameter	Description
<i>relaxations</i>	A a bitwise OR combination of enumeration values from <code>System.Runtime.CompilerServices.CompilationRelaxations</code> that specifies the desired ordering of instruction checking and string interning.

## Description

The *relaxations* flags describe whether to change exception strictness, or string interning. For exceptions of kind xxx, setting `Strictxxx` makes those exceptions have strict ordering, and setting `Relaxedxxx` makes those exceptions have relaxed ordering. Setting neither flag for xxx causes no change for exceptions of kind xxx; i.e., the strictness/relaxation for xxx defaults to what it would be if the attribute were not present. See `System.Runtime.CompilerServices.CompilationRelaxations` for a list of the xxx choices.)

## Exceptions

Exception	Condition
<code>System.ArgumentException</code>	For some xxx, <i>relaxations</i> has both <code>Strictxxx</code> and <code>Relaxedxx</code> set, or has both <code>StringInterning</code> and <code>NoStringInterning</code> set.

# CompilationRelaxationsAttribute(System.Int32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int32 relaxations)

[C#]
public CompilationRelaxationsAttribute(int relaxations)
```

## Summary

Constructs and initializes a new instance of the `System.Runtime.CompilerServices.CompilationRelaxationsAttribute` class with the specified `System.Runtime.CompilerServices.CompilationRelaxations` value.

## Parameters

Parameter	Description
<i>relaxations</i>	A bitwise OR combination of enumeration values from <code>System.Runtime.CompilerServices.CompilationRelaxations</code> that specifies the desired ordering of instruction checking and string interning.

## Description

The *relaxations* flags describe whether to change exception strictness, or string interning. For exceptions of kind xxx, setting `Strictxxx` makes those exceptions have strict ordering, and setting `Relaxedxxx` makes those exceptions have relaxed ordering. Setting neither flag for xxx causes no change for exceptions of kind xxx; i.e., the strictness/relaxation for xxx defaults to what it would be if the attribute were not present. See `System.Runtime.CompilerServices.CompilationRelaxations` for a list of the xxx choices.)

## Exceptions

Exception	Condition
<b>System.ArgumentException</b>	For some xxx, <i>relaxations</i> has both <code>Strictxxx</code> and <code>Relaxedxxx</code> set, or has both <code>StringInterning</code> and <code>NoStringInterning</code> set.

# CompilationRelaxationsAttribute.CompilationRelaxations Property

```
[ILAsm]
.property int32 CompilationRelaxations { public hidebysig specialname
instance int32 get_CompilationRelaxations() }

[C#]
public int CompilationRelaxations { get; }
```

## Summary

Gets the set of values sent to the `System.Runtime.CompilerServices.CompilationRelaxationsAttribute` constructor that specifies the desired ordering of instruction checking and string interning.

## Property Value

The current set of values that specifies the desired ordering of instruction checking and string interning.

## Description

This property is read-only.