

1 System.Runtime.CompilerServices.CompilationRelaxations Enum

```
3 [ILAsm]
4 .class public sealed serializable CompilationRelaxations extends
5 System.Enum { .custom instance void System.FlagsAttribute::.ctor() = ( 01
6 00 00 00 ) }
7
8 [C#]
9 [Flags]public enum CompilationRelaxations
```

9 Assembly Info:

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

15 Type Attributes:

- 16 • FlagsAttribute

17 Summary

18 Indicates whether instruction checking is strictly ordered or relaxed, and whether strings
19 are interned. The flags come in complementary pairs. Setting neither flag of a pair
20 indicates that the corresponding characteristic should be left unchanged. Setting both
21 bits is an error that is detected by the constructor for
22 `System.Runtime.CompilerServices.CompilationRelaxationsAttribute`.

23 Inherits From: System.Enum

24

25 **Library:** RuntimeInfrastructure

26

1 CompilationRelaxations.NoStringInterning 2 Field

```
3 [ILAsm]  
4 .field public static literal valuetype  
5 System.Runtime.CompilerServices.CompilationRelaxations NoStringInterning =  
6 0x8  
  
7 [C#]  
8 NoStringInterning = 0x8
```

9 Summary

10 Indicates that literal strings are not interned; currently only noticed when set for
11 Assemblies.

12

1
2 **CompilationRelaxations.RelaxedArrayExceptions**
3 **Field**

```
4 [ILAsm]  
5 .field public static literal valuetype  
6 System.Runtime.CompilerServices.CompilationRelaxations  
7 RelaxedArrayExceptions = 0x200  
  
8 [C#]  
9 RelaxedArrayExceptions = 0x200
```

10 **Summary**

11 Indicates that instruction checking for `IndexOutOfRangeException`, `RankException`,
12 and `ArrayTypeMismatchException` is not strictly ordered (that is, it is relaxed).

13

1

2 **CompilationRelaxations.RelaxedInvalidCastEx** 3 **ception Field**

4

```
5 .field public static literal valuetype  
6 System.Runtime.CompilerServices.CompilationRelaxations  
7 RelaxedInvalidCastException = 0x80
```

8

```
9 [C#]  
RelaxedInvalidCastException = 0x80
```

10 **Summary**

11 Indicates that instruction checking for `InvalidCastException` is not strictly ordered
12 (that is, it is relaxed).

13

1

2 **CompilationRelaxations.RelaxedNullReferenceException Field**

4

```
[ILAsm]  
.field public static literal valuetype  
System.Runtime.CompilerServices.CompilationRelaxations  
RelaxedNullReferenceException = 0x20
```

8

```
[C#]  
RelaxedNullReferenceException = 0x20
```

9

10 **Summary**

11 Indicates that instruction checking for `NullReferenceException` and access violations is
12 not strictly ordered (that is, it is relaxed).

13

1
2 **CompilationRelaxations.RelaxedOverflowExceptions Field**
3

```
4 [ILAsm]  
5 .field public static literal valuetype  
6 System.Runtime.CompilerServices.CompilationRelaxations  
7 RelaxedOverflowExceptions = 0x800  
  
8 [C#]  
9 RelaxedOverflowExceptions = 0x800
```

10 **Summary**

11 Indicates that instruction checking for `OverflowException` and `DivideByZeroException`
12 is not strictly ordered (that is, it is relaxed).

13

1
2 **CompilationRelaxations.StrictArrayExceptions**
3 **Field**

```
4 [ILAsm]  
5 .field public static literal valuetype  
6 System.Runtime.CompilerServices.CompilationRelaxations  
7 StrictArrayExceptions = 0x100  
  
8 [C#]  
9 StrictArrayExceptions = 0x100
```

10 **Summary**

11 Indicates that instruction checking for `IndexOutOfRangeException`, `RankException`,
12 and `ArrayTypeMismatchException` is strictly ordered.

13

1
2 **CompilationRelaxations.StrictInvalidCastException**
3 **Field**

```
4 [ILAsm]  
5 .field public static literal valuetype  
6 System.Runtime.CompilerServices.CompilationRelaxations  
7 StrictInvalidCastException = 0x40  
  
8 [C#]  
9 StrictInvalidCastException = 0x40
```

10 **Summary**

11 Indicates that instruction checking for `InvalidCastException` is strictly ordered.

12

1

2 **CompilationRelaxations.StrictNullReferenceEx** 3 **ception Field**

4

```
[ILAsm]  
.field public static literal valuetype  
System.Runtime.CompilerServices.CompilationRelaxations  
StrictNullReferenceException = 0x10
```

7

8

```
[C#]  
StrictNullReferenceException = 0x10
```

9

10 **Summary**

11 Indicates that instruction checking for `NullReferenceException` and access violations is
12 strictly ordered.

13

1
2 **CompilationRelaxations.StrictOverflowExceptions**
3 **Field**

```
4 [ILAsm]  
5 .field public static literal valuetype  
6 System.Runtime.CompilerServices.CompilationRelaxations  
7 StrictOverflowExceptions = 0x400  
  
8 [C#]  
9 StrictOverflowExceptions = 0x400
```

10 **Summary**

11 Indicates that instruction checking for `OverflowException` and `DivideByZeroException`
12 is strictly ordered.

13

1 CompilationRelaxations.StringInterning Field

```
2 [ILAsm]  
3 .field public static literal valuetype  
4 System.Runtime.CompilerServices.CompilationRelaxations StringInterning =  
5 0x4  
  
6 [C#]  
7 StringInterning = 0x4
```

8 Summary

9 Indicates that literal strings are interned.

10