

1 System.Collections.Comparer Class

2
3

```
4 [ILASM]  
5 .class public sealed serializable Comparer extends  
6 System.Object implements System.Collections.IComparer  
  
7 [C#]  
8 public sealed class Comparer: IComparer
```

9 Assembly Info:

- 10 • Name: mscorlib
- 11 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 12 • Version: 1.0.x.x
- 13 • Attributes:
 - 14 ○ CLSCompliantAttribute(true)

15 Implements:

- 16 • System.Collections.IComparer

17 Summary

18

19 Provides the default implementation of the
20 **System.Collections.IComparer** interface.

21 Inherits From: System.Object

22

23 **Library:** BCL

24

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

27

28

29

1 **Comparer.Default** Field

```
2 [ILASM]  
3 .field public static initOnly class  
4 System.Collections.Comparer Default  
  
5 [C#]  
6 public static readonly Comparer Default
```

7 **Summary**

8 Returns a new **System.Collections.Comparer** instance containing
9 the default implementation of the **System.Collections.IComparer**
10 interface.

11 **Description**

12 This field is read-only.

13

1 Comparer.Compare(System.Object, 2 System.Object) Method

```
3 [ILASM]  
4 .method public final hidebysig virtual int32 Compare(object  
5 a, object b)  
  
6 [C#]  
7 public int Compare(object a, object b)
```

8 Summary

9 Returns the sort order of two **System.Object** instances.

10 Parameters

11
12

Parameter	Description
<i>a</i>	The first System.Object to compare.
<i>b</i>	The second System.Object to compare.

13
14
15

Return Value

16 A **System.Int32** containing a value that reflects the sort order of *a* as
17 compared to *b*. The following table defines the conditions under which
18 the returned value is a negative number, zero, or a positive number.

Value	Condition
Any negative number	$a < b$.
Zero	$a == b$.
Any positive number	$a > b$.

19
20
21
22

[Note: A null reference is considered to compare less than any other non-null object, and equal to any other null reference, independent of the underlying **System.Type** of either object.]

23 Description

24

The behavior of this method is as follows:

25
26

- If *a* implements the **System.IComparable** interface, returns *a.CompareTo(b)*.

27
28

- If *a* does not implement the **System.IComparable** interface but *b* does, returns *b.CompareTo(a)*.

1
2
3
4
5
6

- If *a* and *b* both are not **null** and do not implement the **System.IComparable** interface, **System.ArgumentException** is thrown.

Exceptions

Exception	Condition
System.ArgumentException	Both <i>a</i> and <i>b</i> are not null and do not implement the System.IComparable interface. -or- Both <i>a</i> and <i>b</i> are not null and are not assignment-compatible types.

7
8