

1 System.Collections.IComparer Interface

2
3

```
4 [ILASM]  
5 .class interface public abstract IComparer  
6 [C#]  
7 public interface IComparer
```

8 Assembly Info:

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

14 Summary

15

16 Provides a mechanism to customize the sort ordering of a collection.

17 **Library:** BCL

18

19 Description

20 The default implementation of this interface is
21 **System.Collections.Comparer**.

22

23 [*Note:* **System.Collections.IComparer** contains the
24 **System.Collections.IComparer.Compare** method. The consumer of
25 an object should call this method when sorting members of a
26 collection.]

27

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

```
3 [ILASM]  
4 .method public hidebysig virtual abstract int32  
5 Compare(object x, object y)  
  
6 [C#]  
7 int Compare(object x, object y)
```

8 Summary

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

10 Parameters

11
12

Parameter	Description
x	First System.Object to compare.
y	Second System.Object to compare.

13
14
15

Return Value

16 A **System.Int32** containing a value that reflects the sort order of x as
17 compared to y. 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	$x < y$.
Zero	$x == y$.
Any positive number	$x > y$.

19

20 Description

21 Behaviors

22 For any objects A, B, and C, the following are required to be true:

23

24 **System.Collections.IComparer.Compare** (A, A) is required to
25 return zero.

26

27 If **System.Collections.IComparer.Compare**(A, B) returns zero, then
28 **System.Collections.IComparer.Compare** (B, A) is required to
29 return zero.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

If **System.Collections.IComparer.Compare**(A, B) returns zero and **System.Collections.IComparer.Compare**(B, C) returns zero then **System.Collections.IComparer.Compare** (A, C) is required to return zero.

If **System.Collections.IComparer.Compare**(A, B) returns a value other than zero, then **System.Collections.IComparer.Compare** (B, A) is required to return a value of the opposite sign.

If **System.Collections.IComparer.Compare**(A, B) returns a value x not equal to zero, and **System.Collections.IComparer.Compare**(B, C) returns a value y of the same sign as x, then **System.Collections.IComparer.Compare** (A, C) is required to return a value of the same sign as x and y.

[*Note:* The exact ordering of this method is unspecified. The intent of the method is to provide a mechanism that orders instances of a class in a manner that is consistent with the mathematical definitions of the relational operators (<, >, and ==), without regard for class-specific definitions of the operators.]

22 **Usage**

23 This interface is used in conjunction with the **System.Array.Sort** and
24 **System.Array.BinarySearch** methods.

25