

System.Collections.IComparer Interface

```
[ILAsm]  
.class interface public abstract IComparer  
  
[C#]  
public interface IComparer
```

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)

Summary

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

Library: BCL

Description

The default implementation of this interface is `System.Collections.Comparer`.

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

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

```
[ILAsm]  
.method public hidebysig virtual abstract int32 Compare(object x, object  
y)  
  
[C#]  
int Compare(object x, object y)
```

Summary

Returns the sort order of two `System.Object` instances.

Parameters

Parameter	Description
<code>x</code>	First <code>System.Object</code> to compare.
<code>y</code>	Second <code>System.Object</code> to compare.

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of `x` as compared to `y`. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Value	Condition
A negative number	$x < y$.
Zero	$x == y$.
A positive number	$x > y$.

Description

Behaviors

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

2
3 `System.Collections.IComparer.Compare (A, A)` is required to return zero.

4
5 If `System.Collections.IComparer.Compare(A, B)` returns zero, then
6 `System.Collections.IComparer.Compare (B, A)` is required to return zero.

7
8 If `System.Collections.IComparer.Compare(A, B)` returns zero and
9 `System.Collections.IComparer.Compare(B, C)` returns zero then
10 `System.Collections.IComparer.Compare (A, C)` is required to return zero.

11
12 If `System.Collections.IComparer.Compare(A, B)` returns a value other than zero, then
13 `System.Collections.IComparer.Compare (B, A)` is required to return a value of the
14 opposite sign.

15
16 If `System.Collections.IComparer.Compare(A, B)` returns a value x not equal to zero,
17 and `System.Collections.IComparer.Compare(B, C)` returns a value y of the same sign
18 as x, then `System.Collections.IComparer.Compare (A, C)` is required to return a value
19 of the same sign as x and y.

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

24
25]

26 Usage

27 This interface is used in conjunction with the `System.Array.Sort` and
28 `System.Array.BinarySearch` methods.