

# 1 System.IComparable Interface

```
2 [ILAsm]  
3 .class interface public abstract IComparable  
  
4 [C#]  
5 public interface IComparable
```

## 6 Assembly Info:

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

## 12 Summary

13 Implemented by classes that support an ordering of instances of the class.

14 **Library:** BCL

15

## 16 Description

17 [Note: System.IComparable contains the System.IComparable.CompareTo method. The  
18 consumer of an object should call this method when sorting instances of a class.]

19

20

21

# 1 IComparable.CompareTo(System.Object) 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract int32 CompareTo(object obj)  
5 [C#]  
6 int CompareTo(object obj)
```

## 7 Summary

8 Returns the sort order of the current instance compared to the specified object.

## 9 Parameters

Parameter	Description
<i>obj</i>	The <code>System.Object</code> to compare to the current instance.

10

## 11 Return Value

12 The return value is a negative number, zero, or a positive number reflecting the sort  
13 order of the current instance as compared to *obj*. For non-zero return values, the exact  
14 value returned by this method is unspecified. The following table defines the return  
15 value:

Returned Value	Description
A negative value	The current instance is $<$ <i>obj</i> .
Zero	The current instance is $==$ <i>obj</i> .
A positive value	The current instance is $>$ than <i>obj</i> or <i>obj</i> is a null reference.

16

## 17 Behaviors

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

19

20 A.CompareTo(A) is required to return zero.

21

22 If A.CompareTo(B) returns zero then B.CompareTo(A) is required to return zero.

23

24 If A.CompareTo(B) returns zero and B.CompareTo(C) returns zero then A.CompareTo(C)

1 is required to return zero.

2

3 If A.CompareTo(B) returns a value other than zero then B.CompareTo(A) is required to  
4 return a value of the opposite sign.

5

6 If A.CompareTo(B) returns a value  $x$  not equal to zero, and B.CompareTo(C) returns a  
7 value  $y$  of the same sign as  $x$ , then A.CompareTo(C) is required to a value of the same  
8 sign as  $x$  and  $y$ .

9

10 The exact behavior of this method is unspecified. The intent of this method is to provide  
11 a mechanism that orders instances of a class in a manner that is consistent with the  
12 mathematical definitions of the relational operators ( $<$ ,  $>$ , and  $=$ ), without regard for  
13 class-specific definitions of the operators.

#### 14 **Usage**

15 Use the `System.IComparable.CompareTo` method to determine the ordering of instances  
16 of a class.

17

18