

System.IComparable Interface

```
[ILAsm]
.class interface public abstract IComparable

[C#]
public interface IComparable
```

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

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

Library: BCL

Description

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

IComparable.CompareTo(System.Object) Method

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

Summary

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

Parameters

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

Return Value

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

Behaviors

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

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

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

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.