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:* 1.0.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**
obj	The **System.Object** to compare to the current instance.

Return Value

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

Returned Value	**Description**
Any negative value	The current instance is < *obj*.
Zero	The current instance is == *obj*.
Any positive value	The current instance is > than *obj* or *obj* 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) is required to return zero.

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

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

The exact behavior of this method is unspecified. The intent of this 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.

Usage

Use the **System.IComparable.CompareTo** method to determine the ordering of instances of a class.