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
x	First `System.Object` to compare.
y	Second `System.Object` 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

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

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

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

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.

]

Usage

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