System.Collections.Generic.IEqualityComparer<-T> Interface

```
[ILAsm]
.class interface public abstract serializable beforefieldinit
System.Collections.Generic.EqualityComparer`1<T> extends System.Object
implements System.Collections.IEqualityComparer, class
System.Collections.Generic.IEqualityComparer`1<!0>

[C#]
public interface IEqualityComparer<in T>
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 4.0.0.0
- *Attributes:*
    - CLSCompliantAttribute(true)

Summary

Defines methods to support the comparison of objects for equality.

Library: BCL

Description

This interface allows the implementation of customized equality comparison for collections. That is, you can create your own definition of equality for type *T*, and specify that this definition be used with a collection type that accepts the System.Collections.Generic.IEqualityComparer`1<T> generic interface. Constructors of the System.Collections.Generic.Dictionary`2<T1,T2> generic collection type accept this interface.

A default implementation of this interface is provided by the System.Collections.Generic.EqualityComparer`1<T>.Default property of the System.Collections.Generic.EqualityComparer`1<T> generic class. The System.StringComparer class implements System.Collections.Generic.IEqualityComparer`1<T> of type System.String.

This interface supports only equality comparisons. Customization of comparisons for sorting and ordering is provided by the System.Collections.Generic.IComparer`1<T> generic interface.

We recommend that you derive from the System.Collections.Generic.EqualityComparer`1<T> class instead of implementing the System.Collections.Generic.IEqualityComparer`1<T> interface, because the System.Collections.Generic.EqualityComparer`1<T> class tests for equality using the System.IEquatable`1<T>.Equals method instead of the System.Object.Equals method. This is consistent with the `Contains`, `IndexOf`, `LastIndexOf`, and `Remove` methods of the `System.Collections.Generic.Dictionary`2<T1,T2>` class and other generic collections.

IEqualityComparer<-T>.Equals(T, T) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool Equals(!0
x,!0 y) cil managed

[C#]
public bool Equals (T x, T y)
```

Summary

Determines whether the specified objects are equal.

Parameters

Parameter	Description
x	The first object of type *T* to compare.
y	The second object of type *T* to compare.

Return Value

`true` if the specified objects are equal; otherwise, `false`.

Description

Implement this method to provide a customized equality comparison for type *T*.

Behaviors

Implementations are required to ensure that if the System.Collections.Generic.IEqualityComparer`1<T>.Equals method returns `true` for two objects x and y, then the value returned by the System.Collections.Generic.IEqualityComparer`1<T>.GetHashCode method for x must equal the value returned for y.

The System.Collections.Generic.IEqualityComparer`1<T>.Equals method is reflexive, symmetric, and transitive. That is, it returns `true` if used to compare an object with itself; `true` for two objects x and y if it is `true` for y and x; and `true` for two objects x and z if it is `true` for x and y and also `true` for y and z.

IEqualityComparer<-T>.GetHashCode(T) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance int32
GetHashCode(!0 obj) cil managed

[C#]
public int GetHashCode (T obj)
```

Summary

Returns a hash code for the specified object.

Parameters

Parameter	Description
obj	The `System.Object` for which a hash code is to be returned.

Return Value

A hash code for the specified object.

Description

Implement this method to provide a customized hash code for type *T*, corresponding to the customized equality comparison provided by the ``System.Collections.Generic.IEqualityComparer`1<T>.Equals`` method.

Behaviors

Implementations are required to ensure that if the ``System.Collections.Generic.IEqualityComparer`1<T>.Equals`` method returns `true` for two objects x and y, then the value returned by the ``System.Collections.Generic.IEqualityComparer`1<T>.GetHashCode`` method for x must equal the value returned for y.

Exceptions

Exception	Condition
System.ArgumentNullException	The type of *obj* is a reference type and *obj* is `null`.