

# 1 System.Collections.Generic.EqualityComparer 2 <T> Class

```
3 [ILAsm]  
4 .class public abstract serializable beforefieldinit  
5 System.Collections.Generic.EqualityComparer`1<T> extends System.Object  
6 implements System.Collections.IEqualityComparer, class  
7 System.Collections.Generic.IEqualityComparer`1<!0>  
  
8 [C#]  
9 public abstract class EqualityComparer<T>:  
10 System.Collections.Generic.IEqualityComparer<T>,  
11 System.Collections.IEqualityComparer
```

## 12 Assembly Info:

- 13 • *Name:* mscorlib
- 14 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 15 • *Version:* 4.0.0.0
- 16 • *Attributes:*
  - 17 ○ CLSCompliantAttribute(true)

## 18 Implements:

- 19 • System.Collections.Generic.IEqualityComparer<T>
- 20 • System.Collections.IEqualityComparer

## 21 Summary

22 Provides a base class for implementations of the  
23 System.Collections.Generic.IEqualityComparer`1<T> generic interface.

## 24 Inherits From: System.Object

25

26 **Library:** BCL

27

## 28 Description

29 Derive from this class to provide a custom implementation of the  
30 System.Collections.Generic.IEqualityComparer`1<T> generic interface for use with  
31 collection classes such as the System.Collections.Generic.Dictionary`2<T1,T2>  
32 generic class, or with methods such as System.Collections.Generic.List`1<T>.Sort.  
33

34 The System.Collections.Generic.EqualityComparer`1<T>.Default property checks  
35 whether type *T* implements the System.IEquatable`1<T> generic interface and, if so,  
36 returns an System.Collections.Generic.EqualityComparer`1<T> that contains the  
37 implementation of the System.IEquatable`1<T>.Equals method. Otherwise, it returns  
38 an System.Collections.Generic.EqualityComparer`1<T>, as provided by *T*.  
39

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

9

# 1 EqualityComparer<T>() Constructor

```
2 [ILAsm]  
3 .method family hidebysig specialname rtspecialname instance void .ctor()  
4 cil managed  
  
5 [C#]  
6 protected EqualityComparer ()
```

## 7 Summary

8 Initializes a new instance of the  
9 System.Collections.Generic.EqualityComparer`1<T> class.

10

# 1 EqualityComparer<T>.Equals(T, T) Method

```
2 [ILAsm]  
3 .method public hidebysig newslot abstract virtual instance bool Equals(!0  
4 x,!0 y) cil managed  
  
5 [C#]  
6 public abstract bool Equals (T x, T y)
```

## 7 Summary

8 When overridden in a derived class, determines whether two objects of type *T* are equal.

## 9 Parameters

Parameter	Description
<i>x</i>	The first object to compare.
<i>y</i>	The second object to compare.

10

## 11 Return Value

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

## 13 Description

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

## 18 How and When to Override

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

24

# 1 EqualityComparer<T>.GetHashCode(T)

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig newslot abstract virtual instance int32  
5 GetHashCode(!0 obj) cil managed  
  
6 [C#]  
7 public abstract int GetHashCode (T obj)
```

### 8 Summary

9 When overridden in a derived class, serves as a hash function for the specified object for  
10 hashing algorithms and data structures, such as a hash table.

### 11 Parameters

Parameter	Description
<i>obj</i>	The object for which to get a hash code.

### 12 Return Value

14 A hash code for the specified object.

### 15 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	The type of <i>obj</i> is a reference type and <i>obj</i> is null.

16  
17

# EqualityComparer<T>.System.Collections.IEqualityComparer.Equals(System.Object, System.Object) Method

```
[ILAsm]
.method private hidebysig newslot virtual final instance bool
System.Collections.IEqualityComparer.Equals(object x, object y) cil
managed

[C#]
bool IEqualityComparer.Equals (object x, object y)
```

## Summary

Determines whether the specified objects are equal.

## Parameters

Parameter	Description
<i>x</i>	The first object to compare.
<i>y</i>	The second object to compare.

## Return Value

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

## Description

This method is a wrapper for the `System.Collections.Generic.EqualityComparer<T>.Equals` method, so *obj* must be cast to the type specified by the generic argument *T* of the current instance. If it cannot be cast to *T*, an `System.ArgumentException` is thrown.

Comparing `null` is allowed and does not generate an exception.

## Exceptions

Exception	Condition
<code>System.ArgumentException</code>	<i>x</i> or <i>y</i> is of a type that cannot be cast to type <i>T</i> .



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

# EqualityComparer<T>.System.Collections.IEqualityComparer.GetHashCode(System.Object) Method

```
[ILAsm]  
.method private hidebysig newslot virtual final instance int32  
System.Collections.IEqualityComparer.GetHashCode(object obj) cil managed  
  
[C#]  
int IEqualityComparer.GetHashCode (object obj)
```

## Summary

Returns a hash code for the specified object.

## Parameters

Parameter	Description
<i>obj</i>	The <i>System.Object</i> for which a hash code is to be returned.

## Return Value

A hash code for the specified object.

## Description

This method is a wrapper for the *System.Collections.Generic.EqualityComparer`1<T>.GetHashCode* method, so *obj* must be a type that can be cast to the type specified by the generic type argument *T* of the current instance.

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	The type of <i>obj</i> is a reference type and <i>obj</i> is null.  -or-  <i>obj</i> is of a type that cannot be cast to type <i>T</i> .

# 1 EqualityComparer<T>.Default Property

```
2 [ILAsm]
3 .property class System.Collections.Generic.EqualityComparer`1<!0>
4 Default() { .get class System.Collections.Generic.EqualityComparer`1<!0>
5 System.Collections.Generic.EqualityComparer`1::get_Default() }
6
7 [C#]
8 public static System.Collections.Generic.EqualityComparer<T> Default {
9 get; }
```

## 9 Summary

10 Returns a default equality comparer for the type specified by the generic argument.

## 11 Property Value

12 The default instance of the `System.Collections.Generic.EqualityComparer`1<T>`  
13 class for type *T*.

## 14 Description

15 The `System.Collections.Generic.EqualityComparer`1<T>.Default` property checks  
16 whether type *T* implements the `System.IEquatable`1<T>` interface and, if so, returns  
17 an `System.Collections.Generic.EqualityComparer`1<T>` that uses that  
18 implementation. Otherwise, it returns an  
19 `System.Collections.Generic.EqualityComparer`1<T>` that uses the overrides of  
20 `System.Object.Equals` and `System.Object.GetHashCode` provided by *T*.

21