

System.Collections.Generic.EqualityComparer<T> Class

```
[ILAsm]
.class 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 abstract class EqualityComparer<T>:
System.Collections.Generic.IEqualityComparer<T>,
System.Collections.IEqualityComparer
```

Assembly Info:

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

Implements:

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

Summary

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

Inherits From: System.Object

Library: BCL

Description

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

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

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.

EqualityComparer<T>() Constructor

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

Summary

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

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

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

Summary

When overridden in a derived class, determines whether two objects of type *T* 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

The `System.Collections.Generic.EqualityComparer<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*.

How and When to Override

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

EqualityComparer<T>.GetHashCode(T)

Method

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

Summary

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

Parameters

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

Return Value

A hash code for the specified object.

Exceptions

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

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> .

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
System.ArgumentNullException	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> .

EqualityComparer<T>.Default Property

```
[ILAsm]
.property class System.Collections.Generic.EqualityComparer`1<!0>
Default() { .get class System.Collections.Generic.EqualityComparer`1<!0>
System.Collections.Generic.EqualityComparer`1::get_Default() }

[C#]
public static System.Collections.Generic.EqualityComparer<T> Default {
get; }
```

Summary

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

Property Value

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

Description

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