System.Collections.Generic.ICollection<T> Interface

```
[ILAsm]
.class interface public abstract ICollection`1<T> implements
System.Collections.Generic.IEnumerable`1<T>

[C#]
public interface ICollection<T>: IEnumerable<T>
```

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)

Implements:

- **System.Collections.Generic.IEnumerable<T>**

Summary

Defines size and copying methods for all generic collections.

Library: BCL

Description

[*Note:* This interface is the base interface for classes in the `System.Collections.Generic` namespace.

This interface extends `System.Collections.Generic.IEnumerable<T>`; `System.Collections.Generic.IDictionary<T,U>` and `System.Collections.Generic.IList<T>` are more specialized interfaces that extend `System.Collections.Generic.ICollection<T>`.

Some collections that limit access to their elements, like the `System.Collections.Generic.Queue<T>`class and the `System.Collections.Generic.Stack<T>` class, directly implement the `System.Collections.Generic.ICollection<T>` interface.

]

ICollection<T>.Add(T) Method

[ILAsm]

```
.method public hidebysig virtual abstract void Add(!0 item)
```

[C#]

```
void Add(T item)
```

Summary

Adds an item to the current collection.

Parameters

Parameter	**Description**
item	The item to add to the current collection.

Exceptions

Exception	**Condition**
System.NotSupportedException	The current collection is read-only.

ICollection<T>.Clear() Method

[ILAsm]
```
.method public hidebysig virtual abstract void Clear()
```

[C#]
```
void Clear()
```

Summary

Removes all items from the current collection.

Description

`System.Collections.Generic.ICollection<T>.Count` is set to zero.

Exceptions

Exception	Condition
System.NotSupportedException	The current collection is read-only.

ICollection<T>.Contains(T) Method

[ILAsm]

```
.method public hidebysig virtual abstract bool Contains(!0 item)
```

[C#]

```
bool Contains(T item)
```

Summary

Determines whether the current collection contains a specific value.

Parameters

Parameter	Description
item	The object to locate in the current collection.

Return Value

`true`, if item is found in the current collection; otherwise, `false`.

Description

Implementations of this interface can vary in how they determine equality of objects; for example, some types use the default comparer, while others allow the user to specify the comparer to be used.

ICollection<T>.CopyTo(T[], System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual abstract void CopyTo(!0[] array, int32 index)

[C#]
void CopyTo(T[] array, int index)
```

Summary

Copies the elements of the current collection to a `System.Array`, starting at the specified index.

Parameters

Parameter	Description
array	A one-dimensional, zero-based `System.Array` that is the destination of the elements copied from the current instance.
index	A `System.Int32` that specifies the zero-based index in *array* at which copying begins.

Description

This operation overwrites the current contents of the array.

Exceptions

Exception	Condition
System.ArgumentNullException	*array* is `null`.
System.ArgumentOutOfRangeException	*index* < 0.
System.ArgumentException	*array* has more than one dimension. -or- *index* is greater than or equal to *array*.Length. -or-

	The sum of *index* and the `System.Collections.ICollection.Count` of the current instance is greater than *array*.Length. -or- Type `T` is not assignable to the element type of the destination array.

ICollection<T>.Remove(T) Method

[ILAsm]

```
.method public hidebysig virtual abstract bool Remove(!0 item)
```

[C#]

```
bool Remove(T item)
```

Summary

Removes the first occurrence of an item from the current collection.

Parameters

Parameter	Description
item	The item to remove from the current collection.

Return Value

`true`, if *item* was removed from the current collection; `false` if *item* was not found in the current collection.

Description

If *item* was found, but cannot be removed for some reason, some unspecified exception is thrown.

Implementations of this interface can vary in how they determine equality of objects; for example, some types use the default comparer, while others allow the user to specify the comparer to be used.

Exceptions

Exception	Condition
System.NotSupportedException	The current collection is read-only.

ICollection<T>.Count Property

```
[ILAsm]
.property int32 Count { public hidebysig virtual abstract specialname
int32 get_Count() }

[C#]
int Count { get; }
```

Summary

Gets the number of elements contained in the current instance.

Property Value

A `System.Int32` that indicates the number of elements contained in the current instance.

Description

This property is read-only.

ICollection<T>.IsReadOnly Property

[ILAsm]

```
.property bool IsReadOnly { public hidebysig virtual abstract specialname
bool get_IsReadOnly() }
```

[C#]

```
bool IsReadOnly { get; }
```

Summary

Indicates whether the current collection is read-only.

Property Value

`true`, if the current collection is read-only; otherwise, `false`.

Description

This property is read-only.

A collection that is read-only does not allow the addition, removal, or modification of elements after the collection is created.