System.Collections.ICollection Interface

```
[ILASM]
.class interface public abstract ICollection implements
System.Collections.IEnumerable

[C#]
public interface ICollection: IEnumerable
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Implements:

- **System.Collections.IEnumerable**

Summary

Serves as the base interface for implementing collection classes. Defines size, enumeration, and synchronization methods for all system collections.

Library: BCL

Description

[*Note:* **System.Collections.ICollection** contains the **System.Collections.ICollection.CopyTo** method. The consumer of a collection object that implements this interface should call this method when copying the elements of that object to a **System.Array**.]

ICollection.CopyTo(System.Array, System.Int32) Method

[ILASM]

```
.method public hidebysig virtual abstract void CopyTo(class
System.Array array, int32 index)
```

[C#]

```
void CopyTo(Array array, int index)
```

Summary

Copies the elements from the current instance to the specified **System.Array**, starting at the specified index in the array.

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.

Behaviors

As described above.

Usage

Use this method to copy from a collection to a **System.Array**.

Exceptions

Exception	Condition
System.ArgumentNullException	*array* is **null**.
System.ArgumentOutOfRangeException	*index* < 0.
System.ArgumentException	*array* has more than one dimension. *index* is greater than or equal to *array*.Length. The sum of *index* and the **System.Collections.ICollection.Count**

	of the current instance is greater than *array*.Length.
System.InvalidCastException	At least one element in the current instance is not assignment-compatible with the type of *array*.

ICollection.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.

Behaviors

The **System.Collections.ICollection.Count** property is required to return the total number of elements contained in the current instance.

ICollection.IsSynchronized Property

[ILASM]

```
.property bool IsSynchronized { public hidebysig virtual
abstract specialname bool get_IsSynchronized() }
```

[C#]

```
bool IsSynchronized { get; }
```

Summary

Gets a **System.Boolean** indicating whether access to the current instance is synchronized (thread-safe).

Property Value

true if access to the current instance is synchronized (thread-safe); otherwise, **false**.

Description

This property is read-only.

Behaviors

As described above.

Usage

To synchronize a collection, use **System.Collections.ICollection.SyncRoot** to obtain a **System.Object** with which to synchronize the collection.

ICollection.SyncRoot Property

[ILASM]

```
.property object SyncRoot { public hidebysig virtual
abstract specialname object get_SyncRoot() }
```

[C#]

```
object SyncRoot { get; }
```

Summary

Gets a **System.Object** that can be used for thread-safe synchronized access to the current instance.

Property Value

A **System.Object** that can be used for thread-safe synchronized access to the current instance.

Description

This property is read-only.

[*Note:* For collections with no publicly available underlying store, the expected implementation is to simply return the current instance. Note that this might not be sufficient for collections that wrap other collections; those should return the underlying collection's **System.Collections.ICollection.SyncRoot** property.]