System.Collections.IList Interface

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

[C#]
public interface IList: 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)

Type Attributes:

- DefaultMemberAttribute("Item") [*Note:* This attribute requires the RuntimeInfrastructure library.]

Implements:

- **System.Collections.ICollection**
- **System.Collections.IEnumerable**

Summary

Implemented by classes that support a collection of objects that can be individually indexed.

Library: BCL

Description

[*Note:* **System.Collections.IList** implementations fall into three categories: read-only, fixed-size, variable-size. A read-only list cannot be modified. A fixed-size list allows the modification of existing elements, but does not allow the addition or removal of elements. A variable-size list allows the modification, addition, and removal of elements.]

IList.Add(System.Object) Method

```
[ILASM]
.method public hidebysig virtual abstract int32 Add(object
value)

[C#]
int Add(object value)
```

Summary

Adds an item to the current instance.

Parameters

Parameter	Description
value	The **System.Object** to add to the current instance.

Return Value

A **System.Int32** containing the index of the current instance into which the new element was inserted.

Behaviors

As described above.

Usage

Use the **System.Collections.IList.Add** method to add another element to the current instance. The index into which that element is added is implementation-dependent.

Exceptions

Exception	Condition
System.NotSupportedException	The current instance is read-only or has a fixed size.

IList.Clear() Method

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

[C#]
void Clear()
```

Summary

Removes all items from the current instance.

Behaviors

As described above.

How and When to Override

Implementations of this method can vary in how a call to this method affects the capacity of a list. Typically, the count is set to zero. The capacity can be set to zero, some default, or remain unchanged.

Usage

Use this method to delete all values from the current instance.

Exceptions

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

IList.Contains(System.Object) Method

```
[ILASM]
.method public hidebysig virtual abstract bool
Contains(object value)

[C#]
bool Contains(object value)
```

Summary

Determines whether the current instance contains a specific value.

Parameters

Parameter	Description
value	The **System.Object** to locate in the current instance.

Return Value

true if the **System.Object** is found in the current instance; otherwise, **false**.

Behaviors

As described above.

Usage

Use the **System.Collections.IList.Contains** method to determine if a particular **System.Object** is an element of the current instance.

IList.IndexOf(System.Object) Method

```
[ILASM]
.method public hidebysig virtual abstract int32
IndexOf(object value)

[C#]
int IndexOf(object value)
```

Summary

Determines the index of a specific item in the current instance.

Parameters

Parameter	Description
value	The **System.Object** to locate in the current instance.

Return Value

The index of *value* if found in the current instance; otherwise, -1.

Behaviors

As described above.

How and When to Override

The default implementations of this method use **System.Object.Equals** to search for value in the current instance.

Usage

Use **System.Collections.IList.IndexOf** to determine if a **System.Object** is contained in the current instance and, if it is contained, its index in the current instance.

IList.Insert(System.Int32, System.Object) Method

```
[ILASM]
.method public hidebysig virtual abstract void Insert(int32
index, object value)

[C#]
void Insert(int index, object value)
```

Summary

Inserts an item to the current instance at the specified position.

Parameters

Parameter	Description
index	A **System.Int32** that specifies the zero-based index at which *value* is inserted.
value	The **System.Object** to insert into the current instance.

Behaviors

If *index* equals the number of items in the **System.Collections.IList**, then *value* is required to be appended to the end of the current instance.

Usage

Use **System.Collections.IList.Insert** to place a new element into a specific position in the current instance.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*index* is not a valid index in the current instance (i.e. is greater than the number of elements in the current instance).
System.NotSupportedException	The current instance is read-only or has a fixed size.

1
2
3

IList.Remove(System.Object) Method

```
[ILASM]
.method public hidebysig virtual abstract void
Remove(object value)

[C#]
void Remove(object value)
```

Summary

Removes the first occurrence of a specified **System.Object** from the current instance.

Parameters

Parameter	Description
value	The **System.Object** to remove from the current instance.

Behaviors

As described above.

In addition, if *value* is **null** or is not found in the current instance, it is required that no exception be thrown and the current instance remain unchanged.

How and When to Override

The default implementations of this method use **System.Object.Equals** to search for value in the current instance.

Usage

Use **System.Collections.IList.Remove** to delete a specified **System.Object** from the current instance.

Exceptions

Exception	Condition
System.NotSupportedException	The current instance is read-only or has a fixed size.

1
2
3

IList.RemoveAt(System.Int32) Method

```
[ILASM]
.method public hidebysig virtual abstract void
RemoveAt(int32 index)

[C#]
void RemoveAt(int index)
```

Summary

Removes the item at the specified index of the current instance.

Parameters

Parameter	Description
index	A **System.Int32** that specifies the zero-based index of the item to remove.

Behaviors

As described above.

Usage

Use **System.Collections.IList.RemoveAt** to delete a specified **System.Object** from the current instance.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*index* is not a valid index in current instance.
System.NotSupportedException	The current instance is read-only or has a fixed size.

IList.IsFixedSize Property

```
[ILASM]
.property bool IsFixedSize { public hidebysig virtual
abstract specialname bool get_IsFixedSize() }

[C#]
bool IsFixedSize { get; }
```

Summary

Gets a **System.Boolean** value indicating whether the current instance has a fixed size.

Property Value

true if the current instance has a fixed size; otherwise, **false**.

Description

This property is read-only.

[*Note:* A collection with a fixed size does not allow the addition or removal of elements, but it allows the modification of existing elements.]

Behaviors

Any method that adds or removes an element of a collection is required to check the value of this property for the particular collection before adding or removing elements. If the value of this property is **false**, any attempt to add or remove an element of the current instance is required to throw a **System.NotSupportedException**.

Default

The default of this property is **false**.

How and When to Override

Override this property, setting the value to **true**, in order to prevent the addition or removal of elements in the current instance.

Usage

Use **System.Collections.IList.IsFixedSize** to secure the current instance from modification from methods, such as **System.Collections.IList.Add** and

System.Collections.IList.Remove, which add or remove elements from a list.

IList.IsReadOnly Property

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

[C#]
bool IsReadOnly { get; }
```

Summary

Gets a value indicating whether the current instance is read-only.

Property Value

true if the current instance is read-only; otherwise, **false**.

Description

This property is read-only.

[*Note:* A collection that is read-only does not allow the modification, addition, or removal of elements.]

Behaviors

Any method that modifies, adds, or removes an element of a collection is required to check the value of this property for the particular collection before executing. If the value of this property is **false**, any attempt to modify, add, or remove an element of the current instance is required to throw a **System.NotSupportedException**.

Default

The default of this property is **false**.

How and When to Override

Override this property, setting the value to **true**, in order to prevent the modification, addition, or removal of elements in the current instance.

Usage

Use **System.Collections.IList.IsReadOnly** to secure the current instance from modification from methods, such as **System.Collections.IList.Add** and **System.Collections.IList.Remove**, which modify, add, or remove elements from a list.

1

IList.Item Property

```
[ILASM]
.property object Item[int32 index] { public hidebysig
virtual abstract specialname object get_Item(int32 index)
public hidebysig virtual abstract specialname void
set_Item(int32 index, object value) }

[C#]
object this[int index] { get; set; }
```

Summary

Gets or sets the element at the specified index in the current instance.

Parameters

Parameter	Description
index	A **System.Int32** that specifies the zero-based index of the element to get or set.

Property Value

The element at the specified index in the current instance.

Behaviors

As described above.

Usage

Use this property for subscript indexing for the current instance in the following form: `myCollection[index]`.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*index* is not a valid index in the current instance.
System.NotSupportedException	The property is being set and the current instance is read-only.