

# System.Collections.IEnumerator Interface

```
[ILASM]
.class interface public abstract IEnumerator

[C#]
public interface IEnumerator
```

## Assembly Info:

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

## Summary

Implemented by classes that support a simple iteration over a collection.

**Library:** BCL

## Description

[Note: **System.Collections.IEnumerator** contains the **System.Collections.IEnumerator.MoveNext** and **System.Collections.IEnumerator.Reset** methods and the **System.Collections.IEnumerator.Current** property. The consumer of an object should call these methods or use this property when iterating over or reading the elements of a collection.

When an enumerator is instantiated or a call is made to **System.Collections.IEnumerator.Reset**, the enumerator is positioned immediately before the first element of the collection and a snapshot of the collection is taken. When the enumerator is in this position, a call to **System.Collections.IEnumerator.MoveNext** is necessary before reading **System.Collections.IEnumerator.Current** from the collection. If changes are made to the collection, such as adding, modifying or deleting elements, the snapshot gets out of sync and the enumerator throws a **System.InvalidOperationException**. Two enumerators instantiated from the same collection at the same time can have different snapshots of the collection.

Enumerators are intended to be used only to read data in the collection.

1 An enumerator does not have exclusive access to the collection for  
2 which it was instantiated.]

3

# IEnumerator.MoveNext() Method

```
[ILASM]
.method public hidebysig virtual abstract bool MoveNext()

[C#]
bool MoveNext()
```

## Summary

Advances the current instance to the next element of the collection.

## Return Value

**true** if the current instance was successfully advanced to the next element; **false** if the current instance has passed the end of the collection.

## Description

[Note: When the current instance is constructed or after **System.Collections.IEnumerator.Reset** is called, the current instance is positioned immediately before the first element of the collection. Use **System.Collections.IEnumerator.MoveNext** to position it over the first element of the collection.]

## Behaviors

A call to **System.Collections.IEnumerator.MoveNext** is required to position the current instance over the next element in the collection and return **true** if the current instance was not positioned beyond the last element of the collection when **System.Collections.IEnumerator.MoveNext** was called. If the current instance is already positioned immediately after the last element of the collection, a call to **System.Collections.IEnumerator.MoveNext** is required to return **false**, and the current instance is required to remain in the same position. If elements are added, removed, or repositioned in the collection after the current instance was instantiated, it is required that a call to **System.Collections.IEnumerator.MoveNext** throw **System.InvalidOperationException**.

## Usage

Use the **System.Collections.IEnumerator.MoveNext** method to check if the current instance is positioned immediately after the last element of the collection, and to position it over the next element if it is not already past the last element of the collection. This allows the use of a conditional loop to iterate over the entire collection.

1   **Exceptions**

2

3

Exception	Condition
<b>System.InvalidOperationException</b>	The collection was modified after the current instance was instantiated.

4

5

6

# 1 I Enumerator.Reset() Method

```
2 [ILASM]  
3 .method public hidebysig virtual abstract void Reset()  
  
4 [C#]  
5 void Reset()
```

## 6 Summary

7 Positions the enumerator immediately before the first element in the  
8 collection.

## 9 Description

10 [Note: When the current instance is constructed or after  
11 **System.Collections.IEnumerator.Reset** is called, the current  
12 instance is positioned immediately before the first element of the  
13 collection, use **System.Collections.IEnumerator.MoveNext** to  
14 position the current instance over the first element of the collection.]

## 15 Behaviors

16 A call to **System.Collections.IEnumerator.Reset** is required to  
17 position the current instance immediately before the first element of  
18 the collection. If elements are added, removed, or repositioned in the  
19 collection after the current instance was instantiated, it is required that  
20 a call to **System.Collections.IEnumerator.Reset** throw a  
21 **System.InvalidOperationException**.

## 22 How and When to Override

23 A call to **System.Collections.IEnumerator.Reset** can involve taking  
24 a new snapshot of the collection or simply moving to the beginning of  
25 the collection. The preferred implementation is to simply move the  
26 current instance to the beginning of the collection, before the first  
27 element. This invalidates the current instance if the collection has been  
28 modified since the current instance was constructed, which is  
29 consistent with **System.Collections.IEnumerator.MoveNext** and  
30 **System.Collections.IEnumerator.Current**.

## 31 Usage

32 Use the **System.Collections.IEnumerator.MoveNext** method to  
33 check if the current instance is positioned immediately past the last  
34 element of the collection, and to position it over the next element if it  
35 is not already past the last element of the collection.

1   **Exceptions**

2

3

Exception	Condition
<b>System.InvalidOperationException</b>	The collection was modified after the enumerator was instantiated.

4

5

6

# IEnumerator.Current Property

```
[ILASM]
.property object Current { public hidebysig virtual
abstract specialname object get_Current() }

[C#]
object Current { get; }
```

## Summary

Gets the element in the collection over which the current instance is positioned.

## Property Value

The element in the collection over which the current instance is positioned.

## Description

[*Note:* When the current instance is constructed or after **System.Collections.IEnumerator.Reset** is called, use **System.Collections.IEnumerator.MoveNext** to position the current instance over the first element of the collection.]

## Behaviors

It is required that **System.Collections.IEnumerator.Current** return the element in the collection over which the current instance is positioned unless it is positioned before the first or after the last element of the collection. If the current instance is positioned before the first element or after the last element of the collection, or elements were added, removed, or repositioned in the collection after the current instance was instantiated, **System.Collections.IEnumerator.Current** is required to throw **System.InvalidOperationException**.

It is also required that **System.Collections.IEnumerator.Current** not change the position of the current instance: consecutive calls to **System.Collections.IEnumerator.Current** are required to return the same object until either **System.Collections.IEnumerator.MoveNext** or **System.Collections.IEnumerator.Reset** is called.

This property is read-only.

## Usage

1 Use **System.Collections.IEnumerator.Current** to get the element in  
2 the collection over which the current instance is positioned, provided  
3 that the current instance is not positioned before the first element or  
4 after the last element of the collection.

5 **Exceptions**

6  
7

Exception	Condition
<b>System.InvalidOperationException</b>	The current instance is positioned before the first element or after the last element of the collection.
	The collection was modified after the current instance was instantiated.

8  
9