

# 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 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

# 1 I Enumerator.MoveNext() Method

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

## 6 Summary

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

## 8 Return Value

9

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

## 13 Description

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

## 19 Behaviors

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

## 33 Usage

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

1 **Exceptions**

2

3

<b>Exception</b>	<b>Condition</b>
<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

# 1 I Enumerator.Current Property

```
2 [ILASM]
3 .property object Current { public hidebysig virtual
4 abstract specialname object get_Current() }
5
6 [C#]
7 object Current { get; }
```

## 7 Summary

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

## 10 Property Value

11

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

## 14 Description

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

## 19 Behaviors

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

29

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

36

37 This property is read-only.

## 38 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