

System.Collections.Generic.Stack<T>.Enumerator Structure

```
[ILAsm]
.class ansi serializable sealed nested public beforefieldinit
Stack<T>.Enumerator extends System.ValueType implements
System.Collections.Generic.IEnumerator`1<T>, System.IDisposable,
System.Collections.IEnumerator

[C#]
public struct Stack<T>.Enumerator:
System.Collections.Generic.IEnumerator<T>
```

Assembly Info:

- *Name:* System
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 4.0.0.0
- *Attributes:*
 - CLSCompliantAttribute(true)

Implements:

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

Summary

Enumerates the elements of a Stack<T>

Inherits From: System.ValueType

Library: BCL

Usage

For a detailed description regarding the use of an enumerator, see
System.Collections.Generic.IEnumerator<T>.

1 **Stack<T>.Enumerator.Dispose() Method**

```
2 [ILAsm]  
3 .method public hidebysig newslot virtual final instance void Dispose() cil  
4 managed  
  
5 [C#]  
6 public void Dispose ()
```

7 **Summary**

8 Releases all resources used by the Stack(Of T).Enumerator.

9

Stack<T>.Enumerator.MoveNext() Method

```
[ILAsm]  
.method public hidebysig newslot virtual final instance bool MoveNext()  
cil managed  
  
[C#]  
public bool MoveNext ()
```

Summary

Advances the enumerator to the next element of the Stack(Of T).

Return Value

A System.Boolean that is true if the enumerator was successfully advanced to the next element; false if the enumerator has passed the end of the collection.

Description

After an enumerator is created, the enumerator is positioned before the first element in the collection, and the first call to System.Collections.Generic.Stack<T>.Enumerator.MoveNext advances the enumerator to the first element of the collection.

If System.Collections.Generic.Stack<T>.Enumerator.MoveNext passes the end of the collection, the enumerator is positioned after the last element in the collection and System.Collections.Generic.Stack<T>.Enumerator.MoveNext returns false. When the enumerator is at this position, subsequent calls to System.Collections.Generic.Stack<T>.Enumerator.MoveNext also return false.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and its behavior is undefined.

Exceptions

Exception	Condition
System.InvalidOperationException	The collection was modified after the enumerator was created.

Stack<T>.Enumerator.System.Collections.IEnumerator.Reset() Method

```
[ILAsm]  
.method private hidebysig newslot virtual final instance void  
System.Collections.IEnumerator.Reset() cil managed  
  
[C#]  
void IEnumerator.Reset ()
```

Summary

Sets the enumerator to its initial position, which is before the first element in the collection.

Description

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and the next call to `System.Collections.IEnumerator.MoveNext` or `System.Collections.IEnumerator.Reset` throws an `System.InvalidOperationException`.

Exceptions

Exception	Condition
System.InvalidOperationException	The collection was modified after the enumerator was created.

Stack<T>.Enumerator.Current Property

```
[ILAsm]  
.property instance !0 Current  
  
[C#]  
public T Current { get; }
```

Summary

Gets the element at the current position of the enumerator.

Property Value

Type: T, the element in the Stack(Of T) at the current position of the enumerator.

Description

Current is undefined under any of the following conditions:

1. The enumerator is positioned before the first element in the collection, immediately after the enumerator is created. MoveNext must be called to advance the enumerator to the first element of the collection before reading the value of Current.

2. The last call to MoveNext returned false, which indicates the end of the collection.

Current returns the same object until MoveNext is called. MoveNext sets Current to the next element. If the collection is modified between MoveNext and Current, Current returns the element that it is set to, even if the enumerator is already invalidated.

Stack<T>.Enumerator.System.Collections.IEnumerator.Current Property

```
[ILAsm]  
.property instance object System.Collections.IEnumerator.Current  
  
[C#]  
object System.Collections.IEnumerator.Current { get; }
```

Summary

Gets the element at the current position of the enumerator.

Property Value

The element in the collection at the current position of the enumerator.

Description

After an enumerator is created or after a `System.Collections.IEnumerator.Reset` is called, `System.Collections.IEnumerator.MoveNext` must be called to advance the enumerator to the first element of the collection before reading the value of `System.Collections.IEnumerator.Current`; otherwise, `System.Collections.IEnumerator.Current` is undefined.

`System.Collections.IEnumerator.Current` also throws an exception if the last call to `System.Collections.IEnumerator.MoveNext` returned false, which indicates the end of the collection.

`System.Collections.IEnumerator.Current` does not move the position of the enumerator, and consecutive calls to `System.Collections.IEnumerator.Current` return the same object until either `System.Collections.IEnumerator.MoveNext` or `System.Collections.IEnumerator.Reset` is called.

An enumerator remains valid as long as the collection remains unchanged. If changes are made to the collection, such as adding, modifying, or deleting elements, the enumerator is irrecoverably invalidated and the next call to `System.Collections.IEnumerator.MoveNext` or `System.Collections.IEnumerator.Reset` throws an `System.InvalidOperationException`. If the collection is modified between `System.Collections.IEnumerator.MoveNext` and `System.Collections.IEnumerator.Current`, `System.Collections.IEnumerator.Current` returns the element that it is set to, even if the enumerator is already invalidated.

Exceptions

Exception	Condition
-----------	-----------

System.InvalidOperationException

The enumerator is positioned before the first element of the collection or after the last element.

1

2