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

[ILAsm]
```
.class sequential ansi serializable sealed nested public beforefieldinit
LinkedList<T>.Enumerator extends System.ValueType implements
System.Collections.Generic.IEnumerator`1<!0>, System.IDisposable,
System.Collections.IEnumerator
```

[C#]
```
public struct LinkedList<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 00]
- *Version:* 4.0.0.0
- *Attributes:*
  - CLSCompliantAttribute(true)

Implements:

- **System.Collections.Generic.IEnumerator<T>**
- **System.Runtime.Serialization.IDeserializationCallback**
- **System.Runtime.Serialization.ISerializable**

Summary

Enumerates the elements of a ``System.Collections.Generic.LinkedList`1<T>``.

Inherits From: System.ValueType

Library: BCL

Usage

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

LinkedList<T>.Enumerator.Dispose() Method

[ILAsm]
```
.method public hidebysig newslot virtual final instance void Dispose() cil managed
```

[C#]
```
public void Dispose ()
```

Summary

Releases all resources used by the `` System.Collections.Generic.LinkedList`1<T>.Enumerator ``.

LinkedList<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 ``System.Collections.Generic.LinkedList`1<T>``.

Return Value

`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.LinkedList`1<T>.Enumerator.MoveNext`` advances the enumerator to the first element of the collection.

If ``System.Collections.Generic.LinkedList`1<T>.Enumerator.MoveNext`` passes the end of the collection, the enumerator is positioned after the last element in the collection and ``System.Collections.Generic.LinkedList`1<T>.Enumerator.MoveNext`` returns `false`. When the enumerator is at this position, subsequent calls to ``System.Collections.Generic.LinkedList`1<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. Subsequent calls throw an `System.InvalidOperationException`.

Exceptions

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

LinkedList<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. This class cannot be inherited.

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.

LinkedList<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

The element in the `System.Collections.Generic.LinkedList`1<T>` at the current position of the enumerator.

Description

`System.Collections.Generic.LinkedList`1<T>.Enumerator.Current` is undefined under any of the following conditions:

- The enumerator is positioned before the first element in the collection, immediately after the enumerator is created. `System.Collections.Generic.LinkedList`1<T>.Enumerator.MoveNext` must be called to advance the enumerator to the first element of the collection before reading the value of `System.Collections.Generic.LinkedList`1<T>.Enumerator.Current`.

- The last call to `System.Collections.Generic.LinkedList`1<T>.Enumerator.MoveNext` returned `false`, which indicates the end of the collection.

`System.Collections.Generic.LinkedList`1<T>.Enumerator.Current` returns the same object until `System.Collections.Generic.LinkedList`1<T>.Enumerator.MoveNext` is called. `System.Collections.Generic.LinkedList`1<T>.Enumerator.MoveNext` sets `System.Collections.Generic.LinkedList`1<T>.Enumerator.Current` to the next element. 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 though the enumerator is invalidated.

[*Note:* For better performance, this property does not throw an exception if the enumerator is positioned before the first element or after the last element; the value of the property is undefined.

]

LinkedList<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.