

1 System.CharEnumerator Class

```
2 [ILAsm]  
3 .class public sealed serializable CharEnumerator extends System.Object  
4 implements System.ICloneable, System.Collections.IEnumerator  
  
5 [C#]  
6 public sealed class CharEnumerator: ICloneable, IEnumerator
```

7 Assembly Info:

- 8 • *Name:* mscorlib
- 9 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 10 • *Version:* 2.0.x.x
- 11 • *Attributes:*
 - 12 ○ CLSCompliantAttribute(true)

13 Implements:

- 14 • **System.Collections.IEnumerator**
- 15 • **System.ICloneable**

16 Summary

17 Supports iteration over and provides read-only access to the individual characters in a
18 `System.String`.

19 Inherits From: System.Object

21 **Library:** BCL

23 **Thread Safety:** All public static members of this type are safe for multithreaded operations.
24 No instance members are guaranteed to be thread safe.

26 Description

27 [*Note:* `System.CharEnumerator` is used to support the `foreach` statement of the C#
28 programming language.

30 Several independent instances of `System.CharEnumerator` across one or more threads
31 can have access to a single instance of `System.String`.

33 For more information regarding the use of an enumerator, see
34 `System.Collections.IEnumerator`.

36]

37

1 CharEnumerator.Clone() Method

```
2 [ILAsm]  
3 .method public final hidebysig virtual object Clone()  
4 [C#]  
5 public object Clone()
```

6 Summary

7 Creates a copy of the current instance.

8 Return Value

9 A `System.Object` with the same state as the current instance.

10 Description

11 The return value is a `System.Object` containing a copy of the state of the current
12 instance when this method is invoked. This `System.Object` must be cast explicitly to a
13 `System.CharEnumerator` to be made usable as one.

14
15 [*Note:* This method is useful for saving the state of a `System.CharEnumerator` object
16 while iterating through the characters of a `System.String`.

17
18 This method is implemented to support the `System.ICloneable` interface.

19
20]

21

1 CharEnumerator.MoveNext() Method

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

6 Summary

7 Advances the current instance to the next element of the `System.String` if and only if
8 the current instance is not already placed beyond the final element of the
9 `System.String`.

10 Return Value

11 `true` if the enumerator was successfully advanced to the next element; otherwise,
12 `false`.

13 Description

14 If the enumerator is positioned after the final element of the `System.String` when this
15 method is called, the current instance remains in its current position and `false` is
16 returned.

17

1 CharEnumerator.Reset() Method

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

6 Summary

7 Repositions the current instance to its initial position, immediately before the first
8 character in the `System.String`.

9

1 CharEnumerator.Current Property

```
2 [ILAsm]  
3 .property valuetype System.Char Current { public hidebysig specialname  
4 instance valuetype System.Char get_Current() }  
5 [C#]  
6 public char Current { get; }
```

7 Summary

8 Gets the element in the `System.String` over which the current instance is positioned.

9 Property Value

10 The `System.Char` in the `System.String` over which the current instance is positioned.

11 Exceptions

Exception	Condition
System.InvalidOperationException	The current instance is positioned before the first element or after the last element of the <code>System.String</code> .

12

13

1
2
3
4
5
6
7
8
9
10
11
12

CharEnumerator.System.Collections.IEnumerator.Current Property

```
[ILAsm]  
.property object IEnumerator.Current { public hidebysig virtual abstract  
specialname object get_IEnumerator.Current() }  
  
[C#]  
object IEnumerator.Current { get; }
```

Summary

Implemented to support the System.Collections.IEnumerator interface. [Note: For more information, see System.Collections.IEnumerator.Current.]