

System.CharEnumerator Class

```
[ILASM]
.class public sealed serializable CharEnumerator extends
System.Object implements System.ICloneable,
System.Collections.IEnumerator

[C#]
public sealed class CharEnumerator: ICloneable, 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)

Implements:

- **System.Collections.IEnumerator**
- **System.ICloneable**

Summary

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

Inherits From: System.Object

Library: BCL

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

Description

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

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

For more information regarding the use of an enumerator, see **System.Collections.IEnumerator**.]

1 CharEnumerator.Clone() Method

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

6 Summary

7 Creates a copy of the current instance.

8 Return Value

9

10 A **System.Object** with the same state as the current instance.

11 Description

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

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

20
21 This method is implemented to support the **System.ICloneable**
22 interface.]

23

1 CharEnumerator.MoveNext() Method

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

6 Summary

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

10 Return Value

11

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

14 Description

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

18

1 CharEnumerator.Reset() Method

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

6 Summary

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

9

1 CharEnumerator.Current Property

```
2 [ILASM]
3 .property valuetype System.Char Current { public hideby sig
4 specialname 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
9 instance is positioned.

10 Property Value

11

12 The **System.Char** in the **System.String** over which the current
13 instance is positioned.

14 Exceptions

15

16

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

17

18

19

1 CharEnumerator.System.Collections.IEnumerator.Current Property

```
3 [ILASM]
4 .property object IEnumerator.Current { public hidebysig
5 virtual abstract specialname object
6 get_IEnumerator.Current() }

7 [C#]
8 object IEnumerator.Current { get; }
```

9 Summary

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

13