

# 1 System.Text.StringBuilder Class

```
2 [ILAsm]  
3 .class public sealed serializable StringBuilder extends System.Object  
4 [C#]  
5 public sealed class StringBuilder
```

## 6 Assembly Info:

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

## 12 Type Attributes:

- 13 • DefaultMemberAttribute("Chars") [*Note*: This attribute requires the  
14 RuntimeInfrastructure library.]

## 15 Summary

16 Represents a mutable string of characters.

## 17 Inherits From: System.Object

18  
19 **Library**: BCL

20  
21 **Thread Safety**: All public static members of this type are safe for multithreaded operations.  
22 No instance members are guaranteed to be thread safe.

## 24 Description

25 This class represents string-like objects that are mutable. After a  
26 `System.Text.StringBuilder` object has been created, it can be directly modified by  
27 removing, replacing, or inserting characters. This contrasts the `System.String` class,  
28 which represents an immutable string of characters.

29  
30 The `System.Text.StringBuilder.Capacity` of an instance of the  
31 `System.Text.StringBuilder` class is the maximum number of characters it can hold in  
32 the currently allocated space. The `System.Text.StringBuilder` will dynamically  
33 allocate more space when it is required.

34  
35 Unlike most types in the Base Class Library, the arguments to  
36 `System.Text.StringBuilder` members are assumed to be passed as *in / out*  
37 arguments (passed by reference). [*Note*: Normally arguments are *in* arguments  
38 (passed by value) unless explicitly declared as *out* or *in / out*.]

39

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20

[*Note*: An instance of `System.String` is said to be "immutable" because its value cannot be modified once it has been created. Methods on `System.String` that appear to modify a `System.String` instance actually return a new instance containing the modification. The `System.Text.StringBuilder` class provides methods that actually modify the contents of a string-like object.

Relational operators only perform reference comparisons (unless overloaded by a particular language compiler). Despite this restriction, relational operators can be used to compare `System.String` objects that are assigned literal values. Their values are immutable and can't change, so a reference comparison is sufficient. Because `System.Text.StringBuilder` instances are mutable, they should not be compared with relational operators.

For performance reasons a `System.Text.StringBuilder` might allocate more memory than needed. The amount of memory allocated is implementation specific.

]

# 1 **StringBuilder()** Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor()  
4 [C#]  
5 public StringBuilder()
```

## 6 **Summary**

7 Constructs and initializes a new, empty instance of the `System.Text.StringBuilder`  
8 class.

## 9 **Description**

10 The new instance of `System.Text.StringBuilder` represents a string equal to  
11 `System.String.Empty`. The `System.Text.StringBuilder.Capacity` is set to the default  
12 capacity.

13  
14 [*Note:* The default value of the `System.Text.StringBuilder.Capacity` property is  
15 implementation dependent.]  
16  
17

18

# 1 **StringBuilder(System.String) Constructor**

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(string value)  
4 [C#]  
5 public StringBuilder(string value)
```

## 6 **Summary**

7 Constructs and initializes a new instance of the `System.Text.StringBuilder` class, with  
8 the specified `System.String` as its value.

## 9 **Parameters**

Parameter	Description
<i>value</i>	A <code>System.String</code> containing the string value of the new instance of <code>System.Text.StringBuilder</code> .

10

## 11 **Description**

12 [*Note:* The `System.Text.StringBuilder.Capacity` of the new instance is  
13 implementation defined.]

14

15

16

# 1 `StringBuilder(System.Int32)` Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(int32 capacity)  
4 [C#]  
5 public StringBuilder(int capacity)
```

## 6 Summary

7 Constructs and initializes a new, empty instance of the `System.Text.StringBuilder`  
8 class, with a specified `System.Text.StringBuilder.Capacity`.

## 9 Parameters

Parameter	Description
<i>capacity</i>	A <code>System.Int32</code> containing the starting number of characters allowed in the <code>System.Text.StringBuilder</code> .

## 10 11 Description

12 If the specified `capacity` is less than the default capacity, the  
13 `System.Text.StringBuilder.Capacity` of the new instance of  
14 `System.Text.StringBuilder` is set to the default value. The  
15 `System.Text.StringBuilder` will dynamically allocate more space when it is required.

16 The new `System.Text.StringBuilder` is initialized to represent an empty string.

17  
18 [Note: The default value of the `System.Text.StringBuilder.Capacity` property is  
19 implementation dependent.]  
20  
21  
22

## 23 Exceptions

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>capacity</i> is less than zero.

24

25

# 1 `StringBuilder.Append(System.Char,` 2 `System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(valuetype System.Char value, int32 repeatCount)  
  
6 [C#]  
7 public StringBuilder Append(char value, int repeatCount)
```

## 8 Summary

9 Appends multiple copies of a character to the end of the current  
10 `System.Text.StringBuilder`.

## 11 Parameters

Parameter	Description
<i>value</i>	The <code>System.Char</code> to be appended.
<i>repeatCount</i>	A <code>System.Int32</code> containing the number of times to append <i>value</i> .

12

## 13 Return Value

14 The current instance after the operation has occurred.

## 15 Description

16 This method appends *repeatCount* copies of the specified character to the current  
17 instance.

18

19 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
20 necessary.

## 21 Exceptions

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>repeatCount</i> is less than zero.

22

23

# 1 `StringBuilder.Append(System.Char[],` 2 `System.Int32, System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(char[] value, int32 startIndex, int32 charCount)  
  
6 [C#]  
7 public StringBuilder Append(char[] value, int startIndex, int charCount)
```

## 8 Summary

9 Appends the string representation of an array of Unicode characters to the end of the  
10 current instance.

## 11 Parameters

Parameter	Description
<i>value</i>	The <code>System.Char</code> array to be appended.
<i>startIndex</i>	A <code>System.Int32</code> containing the index in <i>value</i> at which the subarray starts.
<i>charCount</i>	A <code>System.Int32</code> containing the number of characters to copy from <i>value</i> .

## 13 Return Value

14 The current instance after the operation has occurred.

## 15 Description

16 This method appends the specified range of characters from the *value* array to the  
17 current instance. If *value* is a null reference, and both *startIndex* and *charCount* are  
18 zero, no changes are made.

19  
20 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
21 necessary.

## 22 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference, and <i>startIndex</i> and <i>charCount</i> are not both zero.

**System.ArgumentOutOfRangeException**

*charCount* or *startIndex* is less than zero.

-or-

The sum of *startIndex* and *charCount* is greater than the length of *value*.

1

2

# 1 `StringBuilder.Append(System.String)` Method

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(string value)  
5 [C#]  
6 public StringBuilder Append(string value)
```

## 7 **Summary**

8 Appends a copy of a string to the end of the current instance.

## 9 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.String</code> to be appended.

10

## 11 **Return Value**

12 The current instance after the operation has occurred.

## 13 **Description**

14 This method appends the *value* string to the current instance. If *value* is a null  
15 reference, no changes are made.

16

17 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
18 necessary.

19

# 1 `StringBuilder.Append(System.String, System.Int32, System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(string value, int32 startIndex, int32 count)  
  
6 [C#]  
7 public StringBuilder Append(string value, int startIndex, int count)
```

## 8 Summary

9 Appends a copy of an array of Unicode characters, specified by a starting index and  
10 length, of a specified `System.String` to the end of the current instance.

## 11 Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> from which the substring will be taken.
<i>startIndex</i>	A <code>System.Int32</code> containing the index in <i>value</i> from which to start copying.
<i>count</i>	A <code>System.Int32</code> containing the number of characters to copy from <i>value</i> .

12

## 13 Return Value

14 The current instance after the operation has occurred.

## 15 Description

16 This method appends the specified range of characters in the *value* string to the current  
17 instance. If *value* is a null reference and *startIndex* and *count* are both zero, no changes  
18 are made.

19

20 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
21 necessary.

## 22 Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>value</i> is a null reference, and <i>startIndex</i> and <i>count</i> are not both zero.

**System.ArgumentOutOfRangeException**

*charCount* or *startIndex* is less than zero.

-or-

The sum of *startIndex* and *charCount* is greater than the length of *value*.

1

2

# 1 **StringBuilder.Append(System.Boolean)**

## 2 **Method**

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(bool value)  
  
6 [C#]  
7 public StringBuilder Append(bool value)
```

### 8 **Summary**

9 Appends the string representation of a `System.Boolean` to the end of the current  
10 instance.

### 11 **Parameters**

Parameter	Description
<i>value</i>	A <code>System.Boolean</code> to be appended.

### 12

### 13 **Return Value**

14 The current instance after the operation has occurred.

### 15 **Description**

16 This method appends `value.ToString()` to the current instance.

17  
18 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
19 necessary.

20

# 1 **StringBuilder.Append(System.SByte) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(int8 value)  
  
5 [C#]  
6 public StringBuilder Append(sbyte value)
```

## 7 **Summary**

8 Appends the string representation of a `System.SByte` to the end of the current instance.

## 9 **Type Attributes:**

- 10 • `CLSCompliantAttribute(false)`

## 11 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.SByte</code> to be appended.

## 12 13 **Return Value**

14 The current instance after the operation has occurred.

## 15 **Description**

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 `System.Text.StringBuilder.Append(System.Int16)`.

18 This method appends `value.ToString()` to the current instance.

19 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
20 necessary.  
21  
22

23

# 1 **StringBuilder.Append(System.Byte) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(unsigned int8 value)  
  
5 [C#]  
6 public StringBuilder Append(byte value)
```

## 7 **Summary**

8 Appends the string representation of a `System.Byte` to the end of the current instance.

## 9 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Byte</code> to be appended.

10

## 11 **Return Value**

12 The current instance after the operation has occurred.

## 13 **Description**

14 This method appends `value.ToString()` to the current instance.

15

16 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
17 necessary.

18

# 1 **StringBuilder.Append(System.Char) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(valuetype System.Char value)  
  
5 [C#]  
6 public StringBuilder Append(char value)
```

## 7 **Summary**

8 Appends the string representation of a Unicode character to the end of the current  
9 instance.

## 10 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Char</code> to be appended.

## 11 12 **Return Value**

13 The current instance after the operation has occurred.

## 14 **Description**

15 This method appends the specified character to the current instance.  
16  
17 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
18 necessary.

19

# 1 **StringBuilder.Append(System.Int16) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(int16 value)  
  
5 [C#]  
6 public StringBuilder Append(short value)
```

## 7 **Summary**

8 Appends the string representation of a `System.Int16` to the end of the current instance.

## 9 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Int16</code> to be appended.

10

## 11 **Return Value**

12 The current instance after the operation has occurred.

## 13 **Description**

14 This method appends `value.ToString()` to the current instance.

15

16 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
17 necessary.

18

# 1 **StringBuilder.Append(System.Int32) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(int32 value)  
  
5 [C#]  
6 public StringBuilder Append(int value)
```

## 7 **Summary**

8 Appends the string representation of a `System.Int32` to the end of the current instance.

## 9 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Int32</code> to be appended.

10

## 11 **Return Value**

12 The current instance after the operation has occurred.

## 13 **Description**

14 This method appends `value.ToString()` to the current instance.

15

16 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
17 necessary.

18

# 1 **StringBuilder.Append(System.Int64) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(int64 value)  
  
5 [C#]  
6 public StringBuilder Append(long value)
```

## 7 **Summary**

8 Appends the string representation of a `System.Int64` to the end of the current instance.

## 9 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Int64</code> to be appended.

10

## 11 **Return Value**

12 The current instance after the operation has occurred.

## 13 **Description**

14 This method appends `value.ToString()` to the current instance.

15

16 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
17 necessary.

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 **StringBuilder.Append(System.Single) Method**

```
4 [ILAsm]  
5 .method public hidebysig instance class System.Text.StringBuilder  
6 Append(float32 value)  
  
7 [C#]  
8 public StringBuilder Append(float value)
```

### 9 **Summary**

10 Appends the string representation of a `System.Single` to the end of the current  
11 instance.

### 12 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Single</code> to be appended.

### 14 **Return Value**

15 The current instance after the operation has occurred.

### 16 **Description**

17 This method appends `value.ToString()` to the current instance.

18  
19 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
20 necessary.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 **StringBuilder.Append(System.Double)** 4 **Method**

```
5 [ILAsm]  
6 .method public hidebysig instance class System.Text.StringBuilder  
7 Append(float64 value)  
8 [C#]  
9 public StringBuilder Append(double value)
```

### 10 **Summary**

11 Appends the string representation of a `System.Double` to the end of the current  
12 instance.

### 13 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Double</code> to be appended.

### 14 **Return Value**

15 The current instance after the operation has occurred.

### 16 **Description**

17 This method appends `value.ToString()` to the current instance.

18 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
19 necessary.  
20  
21  
22

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 **StringBuilder.Append(System.Decimal)** 4 **Method**

```
5 [ILAsm]  
6 .method public hidebysig instance class System.Text.StringBuilder  
7 Append(decimal value)  
8 [C#]  
9 public StringBuilder Append(decimal value)
```

### 10 **Summary**

11 Appends the string representation of a `System.Decimal` to the end of the current  
12 instance.

### 13 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Decimal</code> to be appended.

### 14 **Return Value**

15 The current instance after the operation has occurred.

### 16 **Description**

17 This method appends `value.ToString()` to the current instance.

18 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
19 necessary.  
20  
21  
22

# 1 `StringBuilder.Append(System.UInt16)`

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(unsigned int16 value)  
  
6 [C#]  
7 public StringBuilder Append(ushort value)
```

### 8 Summary

9 Appends the string representation of a `System.UInt16` to the end of the current  
10 instance.

### 11 Type Attributes:

- 12 • `CLSCompliantAttribute(false)`

### 13 Parameters

Parameter	Description
<i>value</i>	The <code>System.UInt16</code> to be appended.

### 14 Return Value

15 The current instance after the operation has occurred.

### 17 Description

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Append(System.Int32)`.

20 This method appends `value.ToString()` to the current instance.

21 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
22 necessary.

25

# 1 **StringBuilder.Append(System.UInt32)**

## 2 **Method**

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(unsigned int32 value)  
  
6 [C#]  
7 public StringBuilder Append(uint value)
```

### 8 **Summary**

9 Appends the string representation of a `System.UInt32` to the end of the current  
10 instance.

### 11 **Type Attributes:**

- 12 • `CLSCompliantAttribute(false)`

### 13 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.UInt32</code> to be appended.

### 14 **Return Value**

15 The current instance after the operation has occurred.

### 17 **Description**

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Append(System.Int64)`.

20 This method appends `value.ToString()` to the current instance.

21 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
22 necessary.

25

# 1 `StringBuilder.Append(System.UInt64)`

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Append(unsigned int64 value)  
  
6 [C#]  
7 public StringBuilder Append(ulong value)
```

### 8 Summary

9 Appends the string representation of a `System.UInt64` to the end of the current  
10 instance.

### 11 Type Attributes:

- 12 • `CLSCompliantAttribute(false)`

### 13 Parameters

Parameter	Description
<i>value</i>	The <code>System.UInt64</code> to be appended.

14

### 15 Return Value

16 The current instance after the operation has occurred.

### 17 Description

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Append(System.Decimal)`.

20

21 This method appends `value.ToString()` to the current instance.

22

23 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
24 necessary.

25

# 1 **StringBuilder.Append(System.Object) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(object value)  
  
5 [C#]  
6 public StringBuilder Append(object value)
```

## 7 **Summary**

8 Appends the string representation of an object to the end of the current instance.

## 9 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Object</code> to be appended.

10

## 11 **Return Value**

12 The current instance after the operation has occurred.

## 13 **Description**

14 This method appends `value.ToString()` to the current instance. If `value` is a null  
15 reference, no changes are made.

16

17 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
18 necessary.

19

# 1 **StringBuilder.Append(System.Char[]) Method**

```
2 [ILAsm]  
3 .method public hidebysig instance class System.Text.StringBuilder  
4 Append(char[] value)  
  
5 [C#]  
6 public StringBuilder Append(char[] value)
```

## 7 **Summary**

8 Appends the string representation of all of the characters in a `System.Array` to the end  
9 of the current instance.

## 10 **Parameters**

Parameter	Description
<i>value</i>	The array of <code>System.Char</code> to be appended.

## 11 **Return Value**

13 The current instance after the operation has occurred.

## 14 **Description**

15 This method appends all of the characters in the specified array to the current instance  
16 in the same order as they appear in *value*. If *value* is a null reference no changes are  
17 made.

18  
19 The `System.Text.StringBuilder.Capacity` of the current instance is increased as  
20 necessary.

21

# 1 `StringBuilder.AppendFormat(System.String,` 2 `System.Object, System.Object,` 3 `System.Object)` Method

```
4 [ILAsm]  
5 .method public hidebysig instance class System.Text.StringBuilder  
6 AppendFormat(string format, object arg0, object arg1, object arg2)  
  
7 [C#]  
8 public StringBuilder AppendFormat(string format, object arg0, object arg1,  
9 object arg2)
```

## 10 Summary

11 Appends the specified string to the current instance, with the format specifications in  
12 that string being replaced with the appropriately formatted string values of the specified  
13 objects.

## 14 Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> containing zero or more format specifications.
<i>arg0</i>	The first <code>System.Object</code> to be formatted. Can be a null reference.
<i>arg1</i>	The second <code>System.Object</code> to be formatted. Can be a null reference.
<i>arg2</i>	The third <code>System.Object</code> to be formatted. Can be a null reference.

15

## 16 Return Value

17 The current instance after the operation has occurred.

## 18 Description

19 This method appends the formatted copy of the specified string to the current instance.  
20 If an object referenced in the format string is `null`, an empty string is used in its place.

21

22 [Note: This version of `System.Text.StringBuilder.AppendFormat` is equivalent to  
23 `System.Text.StringBuilder.AppendFormat( null, format, new Object [] { arg0, arg1,`  
24 `arg2} )`. For more information on the format specification, see the `System.String` class  
25 overview.]

26

27

## 1 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is a null reference.
<b>System.FormatException</b>	<i>format</i> is invalid.

2

## 3 Example

4 [C#]

```
5 using System;
6 using System.Text;
7
8 public class StringBuilderTest {
9     public static void Main() {
10
11         StringBuilder sb = new StringBuilder("The high ");
12         Console.WriteLine( sb.AppendFormat("temperature today was {0} {1} {2}.",
13 "very", "very", "high") );
14     }
15 }
```

16

17 The output is

18

19 The high temperature today was very very high.

20

# 1 2 **StringBuilder.AppendFormat(System.IFormat** 3 **Provider, System.String, System.Object[])** 4 **Method**

```
5 [ILAsm]  
6 .method public hidebysig instance class System.Text.StringBuilder  
7 AppendFormat(class System.IFormatProvider provider, string format,  
8 object[] args)  
  
9 [C#]  
10 public StringBuilder AppendFormat(IFormatProvider provider, string format,  
11 params object[] args)
```

## 12 **Summary**

13 Appends the specified string to the current instance, with the format specifications in  
14 that string being replaced with the string values of the specified array of objects,  
15 formatted in accordance with the formatting object returned by the specified  
16 System.IFormatProvider.

## 17 **Parameters**

Parameter	Description
<i>provider</i>	A System.IFormatProvider that supplies a formatting object that provides culture specific formatting information. Can be a null reference.
<i>format</i>	A System.String containing zero or more format specifications.
<i>args</i>	A System.Object array to be formatted.

## 18 19 **Return Value**

20 The current instance after the operation has occurred.

## 21 **Description**

22 This method appends the formatted copy of the specified string to the current instance.  
23 If an object referenced in the format string is null, an empty string is used in its place.

24  
25 The format parameter is embedded with zero or more format specifications of the form,  
26 {N[, M][: formatString]}, where N is a zero-based integer indicating the argument to  
27 be formatted, M is an optional integer indicating the width of the region to contain the  
28 formatted value, and formatString is an optional string of formatting codes. [Note: For  
29 more information on the format specification see the System.String class overview.]

1  
2

### 3 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> or <i>args</i> is a null reference.
<b>System.FormatException</b>	<i>format</i> is invalid.

4  
5

### Example

6

[C#]

```
7 using System;
8 using System.Text;
9
10 public class StringBuilderTest {
11     public static void Main() {
12
13         string a = "very";
14         string b = "very";
15         string c = "high";
16
17         StringBuilder sb = new StringBuilder("The high ");
18         Console.WriteLine(sb.AppendFormat(null, "temperature today was {0}, {1}
19 {2}.". , a, b, c) );
20     }
21 }
22
```

23 The output is

24

25 The high temperature today was very, very high.

26

# 1 `StringBuilder.AppendFormat(System.String, System.Object[])` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 AppendFormat(string format, object[] args)  
  
6 [C#]  
7 public StringBuilder AppendFormat(string format, params object[] args)
```

## 8 Summary

9 Appends the specified string to the current instance, with the format specifications in  
10 that string being replaced with the appropriately formatted string values of the elements  
11 in the specified array.

## 12 Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> containing zero or more format specifications.
<i>args</i>	A <code>System.Object</code> array to be formatted.

## 13 Return Value

15 The current instance after operation has occurred.

## 16 Description

17 This method appends the formatted copy of the specified string to the current instance.  
18 If an object referenced in the format string is `null`, an empty string is used in its place.

19  
20 [Note: This version of `System.Text.StringBuilder.AppendFormat` is equivalent to  
21 `System.Text.StringBuilder.AppendFormat( null, format, args )`. For more  
22 information on the format specification see the `System.String` class overview.]  
23  
24

## 25 Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>format</i> is a null reference.

**System.FormatException**

*format* is invalid.

1

## 2 **Example**

3 [C#]

```
4 using System;
5 using System.Text;
6
7 public class StringBuilderTest {
8     public static void Main() {
9
10         string [] strings = {"very", "very", null, "high"};
11
12         StringBuilder sb = new StringBuilder("The high ");
13         Console.WriteLine( sb.AppendFormat("temperature today was {0}, {1}
14 {2}{3}.", strings) );
15     }
16 }
```

17  
18 The output is

19

20 The high temperature today was very, very high.

21

# 1 `StringBuilder.AppendFormat(System.String, System.Object, System.Object)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 AppendFormat(string format, object arg0, object arg1)  
  
6 [C#]  
7 public StringBuilder AppendFormat(string format, object arg0, object arg1)
```

## 8 Summary

9 Appends the specified string to the current instance, with the format specifications in  
10 that string being replaced with the appropriately formatted string values of the specified  
11 objects.

## 12 Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> containing zero or more format specifications.
<i>arg0</i>	The first <code>System.Object</code> to be formatted. Can be a null reference.
<i>arg1</i>	The second <code>System.Object</code> to be formatted. Can be a null reference.

13

## 14 Return Value

15 The current instance after the operation has occurred.

## 16 Description

17 This method appends the formatted copy of the specified string to the current instance.  
18 If an object referenced in the format string is `null`, an empty string is used in its place.

19

20 [Note: This version of `System.Text.StringBuilder.AppendFormat` is equivalent to  
21 `System.Text.StringBuilder.AppendFormat( null, format, new Object [] { arg0, arg1 }`  
22  `)`. For more information on the format specification, see the `System.String` class  
23 overview.]

24

25

## 26 Exceptions

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

<b>System.ArgumentNullException</b>	<i>format</i> is a null reference.
<b>System.FormatException</b>	<i>format</i> is invalid.

1

## 2 Example

3 [C#]

```
4 using System;
5 using System.Text;
6
7 public class StringBuilderTest {
8     public static void Main() {
9
10         StringBuilder sb = new StringBuilder("The high ");
11         Console.WriteLine( sb.AppendFormat("temperature today was {0} {1}.",
12 "very", "high") );
13     }
14 }
```

15

16 The output is

17

18 The high temperature today was very high.

19

20

# 1 **StringBuilder.AppendFormat(System.String,** 2 **System.Object) Method**

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 AppendFormat(string format, object arg0)  
  
6 [C#]  
7 public StringBuilder AppendFormat(string format, object arg0)
```

## 8 **Summary**

9 Appends the specified string to the current instance, with the format specifications in  
10 that string being replaced with the appropriately formatted string value of the specified  
11 object.

## 12 **Parameters**

Parameter	Description
<i>format</i>	A System.String containing zero or more format specifications.
<i>arg0</i>	A System.Object to be formatted.

## 13 **Return Value**

14 The current instance after the operation has occurred.

## 15 **Description**

16 This method appends the formatted copy of the specified string to the current instance.

17 [Note: This version of System.Text.StringBuilder.AppendFormat is equivalent to  
18 System.Text.StringBuilder.AppendFormat( null, *format*, new Object [] { *arg0* } ).  
19 For more information on the format specification, see the System.String class  
20 overview.]  
21  
22  
23  
24

## 25 **Exceptions**

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is a null reference.

**System.FormatException**

*format* is invalid.

1

## 2 **Example**

3 [C#]

```
4 using System;
5 using System.Text;
6
7 public class StringBuilderTest {
8     public static void Main() {
9
10         StringBuilder sb = new StringBuilder("The high ");
11         Console.WriteLine( sb.AppendFormat("temperature today was {0, 6}.", 88)
12 );
13     }
14 }
```

15  
16 The output is

17 The high temperature today was 88.

18

19

# 1 **StringBuilder.EnsureCapacity(System.Int32)**

## 2 **Method**

```
3 [ILAsm]  
4 .method public hidebysig instance int32 EnsureCapacity(int32 capacity)  
5 [C#]  
6 public int EnsureCapacity(int capacity)
```

### 7 **Summary**

8 Ensures that the capacity of the current instance is at least a specified value.

### 9 **Parameters**

Parameter	Description
<i>capacity</i>	A <code>System.Int32</code> containing the minimum capacity to ensure.

10

### 11 **Return Value**

12 A `System.Int32` equal to the new `System.Text.StringBuilder.Capacity` of the  
13 current instance.

### 14 **Description**

15 If the specified value is less than the current `System.Text.StringBuilder.Capacity`,  
16 no changes are made and `System.Text.StringBuilder.Capacity` remains the same.

17

18 [*Note:* For performance reasons, the new `System.Text.StringBuilder.Capacity` might  
19 be larger than the specified value. The amount of memory allocated by this method is  
20 implementation specific.]

21

22

### 23 **Exceptions**

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>capacity</i> is less zero.

24

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

# StringBuilder.Equals(System.Text.StringBuilder) Method

```
[ILAsm]  
.method public hidebysig instance bool Equals(class  
System.Text.StringBuilder sb)  
  
[C#]  
public bool Equals(StringBuilder sb)
```

## Summary

Determines whether the current instance and a specified `System.Text.StringBuilder` have the same value.

## Parameters

Parameter	Description
<i>sb</i>	A <code>System.Text.StringBuilder</code> .

## Return Value

`true` if the current instance and *sb* have the same value; otherwise, `false`.

# 1 **StringBuilder.Insert(System.Int32,**

## 2 **System.String) Method**

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, string value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, string value)
```

### 8 **Summary**

9 Inserts the string representation of a string object into the current instance at a  
10 specified index.

### 11 **Parameters**

Parameter	Description
<i>index</i>	A System.Int32 containing the index at which to insert.
<i>value</i>	The System.String to be inserted.

### 12

### 13 **Return Value**

14 The current instance after insertion has occurred.

### 15 **Description**

16 This method inserts *value* into the current instance at the specified location. Existing  
17 characters are shifted to make room for the new text, and  
18 System.Text.StringBuilder.Capacity is adjusted as necessary.

19 If *value* is System.String.Empty or a null reference, the System.Text.StringBuilder  
20 is not changed.  
21

### 22 **Exceptions**

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or-

*index* is less than zero.

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Boolean)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, bool value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, bool value)
```

## 8 Summary

9 Inserts the string representation of a `System.Boolean` value into the current instance at  
10 a specified index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Boolean</code> value to be inserted.

## 12 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

## 19 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.String, System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, string value, int32 count)  
  
6 [C#]  
7 public StringBuilder Insert(int index, string value, int count)
```

## 8 Summary

9 Inserts multiple copies of a string into the current instance at a specified index.

## 10 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.String</code> to be inserted.
<i>count</i>	A <code>System.Int32</code> containing the number of times the string is to be inserted.

## 11 12 Return Value

13 The current instance after insertion has occurred.

## 14 Description

15 This method inserts the *value* string *count* times into the current instance at the  
16 specified location. Existing characters are shifted to make room for the new text, and  
17 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

18  
19 If *value* is `Empty` or a null reference, the `System.Text.StringBuilder` is not changed.

## 20 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance  -or-

*index* is less than zero

-or-

*count* is less than zero

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.SByte)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, int8 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, sbyte value)
```

## 8 Summary

9 Inserts the string representation of a `System.SByte` into the current instance at the  
10 specified index.

## 11 Type Attributes:

- 12 • `CLSCompliantAttribute(false)`

## 13 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.SByte</code> to be inserted.

## 14 Return Value

15 The current instance after insertion has occurred.

## 17 Description

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Insert(System.Int32, System.Int16)`.

20 This method inserts `value.ToString()` into the current instance at the specified location.  
21 Existing characters are shifted to make room for the new text, and  
22 `System.Text.StringBuilder.Capacity` is adjusted as necessary.  
23

## 24 Exceptions

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>index</i> is greater than the length of the current instance

	-or- <i>index</i> is less than zero
--	----------------------------------------

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Byte)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, unsigned int8 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, byte value)
```

## 8 Summary

9 Inserts the string representation of a `System.Byte` into the current instance at the  
10 specified index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Byte</code> to be inserted.

## 12 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

## 19 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero

20  
21

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Int16)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, int16 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, short value)
```

## 8 Summary

9 Inserts the string representation of a `System.Int16` into the current instance at a  
10 specified index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Int16</code> to be inserted.

## 12 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

## 19 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero.

20  
21

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Char)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, valuetype System.Char value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, char value)
```

## 8 Summary

9 Inserts the string representation of a Unicode character into the current instance at a  
10 specified index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Char</code> to be inserted.

## 12 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

## 19 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero.

20  
21

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Char[])` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, char[] value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, char[] value)
```

## 8 Summary

9 Inserts the string representation of an array of Unicode characters into the current  
10 instance at a specified index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Char</code> array to be inserted.

## 12 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts all of the characters in the specified array into the current instance  
17 in the same order as they appear in *value*. Existing characters are shifted to make room  
18 for the new text, and `System.Text.StringBuilder.Capacity` is adjusted as necessary.

19  
20 If *value* is empty or a null reference, the `System.Text.StringBuilder` is not changed.

## 21 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero.

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Char[], System.Int32, System.Int32)` 3 **Method**

```
4 [ILAsm]  
5 .method public hidebysig instance class System.Text.StringBuilder  
6 Insert(int32 index, char[] value, int32 startIndex, int32 charCount)  
  
7 [C#]  
8 public StringBuilder Insert(int index, char[] value, int startIndex, int  
9 charCount)
```

## 10 **Summary**

11 Inserts the string representation of a subarray of Unicode characters into the current  
12 instance at a specified index.

## 13 **Parameters**

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Char</code> array from which to get the characters to be inserted.
<i>startIndex</i>	A <code>System.Int32</code> containing the starting index within <i>value</i> .
<i>charCount</i>	A <code>System.Int32</code> containing the number of characters to insert from <i>value</i> .

## 14 15 **Return Value**

16 The current instance after insertion has occurred.

## 17 **Description**

18 This method inserts the specified range of characters from *value* array into the current  
19 instance in the same order as they appear in *value*. Existing characters are shifted to  
20 make room for the new text, and `System.Text.StringBuilder.Capacity` is adjusted as  
21 necessary.

22  
23 If *value* is an empty array or a null reference and *startIndex* and *charCount* are both  
24 zero, the `System.Text.StringBuilder` is not changed.

## 25 **Exceptions**

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference, and <i>startIndex</i> and <i>charCount</i> are not both zero.
<b>System.ArgumentOutOfRangeException</b>	<p><i>index</i> is greater than the length of the current instance or less than zero.</p> <p>-or-</p> <p><i>startIndex</i> or <i>charCount</i> is less than zero or their sum is greater than the length of <i>value</i>.</p>

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Object)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, object value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, object value)
```

## 8 Summary

9 Inserts the string representation of an object into the current instance at a specified  
10 index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Object</code> to be inserted.

12

## 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

19

20 If `value` is a null reference, the `System.Text.StringBuilder` is not changed.

## 21 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero.

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.UInt64) Method`

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, unsigned int64 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, ulong value)
```

## 8 **Summary**

9 Inserts the string representation of a `System.UInt64` into the current instance at a  
10 specified index.

## 11 **Type Attributes:**

- 12 • `CLSCompliantAttribute(false)`

## 13 **Parameters**

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.UInt64</code> to be inserted.

## 14 **Return Value**

15 The current instance after insertion has occurred.

## 17 **Description**

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Insert(System.Int32, System.Decimal)`.

20 This method inserts `value.ToString()` into the current instance at the specified location.  
21 Existing characters are shifted to make room for the new text, and  
22 `System.Text.StringBuilder.Capacity` is adjusted as necessary.  
23

## 24 **Exceptions**

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>index</i> is greater than the length of the current instance

	-or- <i>index</i> is less than zero.
--	-----------------------------------------

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.UInt32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, unsigned int32 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, uint value)
```

## 8 Summary

9 Inserts the string representation of a `System.UInt32` into the current instance at the  
10 specified index.

## 11 Type Attributes:

- 12 • `CLSCompliantAttribute(false)`

## 13 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.UInt32</code> to be inserted.

## 15 Return Value

16 The current instance after insertion has occurred.

## 17 Description

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Insert(System.Int32, System.Int64)`.

20  
21 This method inserts `value.ToString()` into the current instance at the specified location.  
22 Existing characters are shifted to make room for the new text, and  
23 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

## 24 Exceptions

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>index</i> is greater than the length of the current instance

	-or- <i>index</i> is less than zero.
--	-----------------------------------------

1

2

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.UInt16)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, unsigned int16 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, ushort value)
```

## 8 Summary

9 Inserts the string representation of a `System.UInt16` into the current instance at the  
10 specified index.

## 11 Type Attributes:

- 12 • `CLSCompliantAttribute(false)`

## 13 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.UInt16</code> to insert.

## 14 Return Value

15 The current instance after insertion has occurred.

## 17 Description

18 This member is not CLS-compliant. For a CLS-compliant alternative, use  
19 `System.Text.StringBuilder.Insert(System.Int32, System.Int32)`.

20 This method inserts `value.ToString()` into the current instance at the specified location.  
21 Existing characters are shifted to make room for the new text, and  
22 `System.Text.StringBuilder.Capacity` is adjusted as necessary.  
23

## 24 Exceptions

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>index</i> is greater than the length of the current instance

	-or- <i>index</i> is less than zero.
--	-----------------------------------------

1

2

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 **StringBuilder.Insert(System.Int32,** 4 **System.Decimal) Method**

```
5 [ILAsm]  
6 .method public hidebysig instance class System.Text.StringBuilder  
7 Insert(int32 index, decimal value)  
8 [C#]  
9 public StringBuilder Insert(int index, decimal value)
```

### 10 **Summary**

11 Inserts the string representation of a `System.Decimal` into the current instance at the  
12 specified index.

### 13 **Parameters**

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Decimal</code> to be inserted.

### 15 **Return Value**

16 The current instance after insertion has occurred.

### 17 **Description**

18 This method inserts `value.ToString()` into the current instance at the specified location.  
19 Existing characters are shifted to make room for the new text, and  
20 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

### 21 **Exceptions**

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or-

	<i>index</i> is less than zero
--	--------------------------------

1

2

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 **StringBuilder.Insert(System.Int32,** 4 **System.Double) Method**

```
5 [ILAsm]  
6 .method public hidebysig instance class System.Text.StringBuilder  
7 Insert(int32 index, float64 value)  
8 [C#]  
9 public StringBuilder Insert(int index, double value)
```

### 10 **Summary**

11 Inserts the string representation of a `System.Double` into the current instance at the  
12 specified index.

### 13 **Parameters**

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Double</code> to be inserted.

### 15 **Return Value**

16 The current instance after insertion has occurred.

### 17 **Description**

18 This method inserts `value.ToString()` into the current instance at the specified location.  
19 Existing characters are shifted to make room for the new text, and  
20 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

### 21 **Exceptions**

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or-

	<i>index</i> is less than zero
--	--------------------------------

1

2

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## StringBuilder.Insert(System.Int32, System.Single) Method

```
[ILAsm]  
.method public hidebysig instance class System.Text.StringBuilder  
Insert(int32 index, float32 value)  
  
[C#]  
public StringBuilder Insert(int index, float value)
```

### Summary

Inserts the string representation of a `System.Single` into the current instance at the specified index.

### Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Single</code> to be inserted.

### Return Value

The current instance after insertion has occurred.

### Description

This method inserts `value.ToString()` into the current instance at the specified location. Existing characters are shifted to make room for the new text, and `System.Text.StringBuilder.Capacity` is adjusted as necessary.

### Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance  -or-

	<i>index</i> is less than zero
--	--------------------------------

1

2

# 1 **StringBuilder.Insert(System.Int32,**

## 2 **System.Int64) Method**

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, int64 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, long value)
```

### 8 **Summary**

9 Inserts the string representation of a `System.Int64` into the current instance at the  
10 specified index.

### 11 **Parameters**

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Int64</code> to be inserted.

### 12

### 13 **Return Value**

14 The current instance after insertion has occurred.

### 15 **Description**

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

### 19 **Exceptions**

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero.

20  
21

# 1 `StringBuilder.Insert(System.Int32,` 2 `System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Insert(int32 index, int32 value)  
  
6 [C#]  
7 public StringBuilder Insert(int index, int value)
```

## 8 Summary

9 Inserts the string representation of a `System.Int32` into the current instance at the  
10 specified index.

## 11 Parameters

Parameter	Description
<i>index</i>	A <code>System.Int32</code> containing the index at which to insert.
<i>value</i>	The <code>System.Int32</code> to be inserted.

12

## 13 Return Value

14 The current instance after insertion has occurred.

## 15 Description

16 This method inserts `value.ToString()` into the current instance at the specified location.  
17 Existing characters are shifted to make room for the new text, and  
18 `System.Text.StringBuilder.Capacity` is adjusted as necessary.

## 19 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> is greater than the length of the current instance -or- <i>index</i> is less than zero

20

21

# 1 `StringBuilder.Remove(System.Int32,` 2 `System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Remove(int32 startIndex, int32 length)  
  
6 [C#]  
7 public StringBuilder Remove(int startIndex, int length)
```

## 8 Summary

9 Removes a specified range of characters from the current instance.

## 10 Parameters

Parameter	Description
<i>startIndex</i>	A <code>System.Int32</code> containing the index at which to begin removal.
<i>length</i>	A <code>System.Int32</code> containing the number of characters to be removed.

11

## 12 Return Value

13 The current instance after removal has occurred.

## 14 Description

15 This method removes the specified range of characters from the current instance. The  
16 characters at (*startIndex* + *length*) are moved to *startIndex*, and the string value of the  
17 current instance is shortened by *length*.

18

19 [Note: The `System.Text.StringBuilder.Replace` method can be used to remove all  
20 instances of a string from a `System.Text.StringBuilder`.]

21

22

## 23 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>startIndex</i> or <i>length</i> is less than zero
	-or- The sum of <i>startIndex</i> and <i>length</i> is greater

than the length of the current instance.

1

## 2 **Example**

3 [C#]

```
4 using System;
5 using System.Text;
6
7 public class StringBuilderTest {
8     public static void Main() {
9
10         StringBuilder sb = new StringBuilder("0123456789");
11         Console.WriteLine(sb);
12         sb.Remove(3, 4);
13         Console.WriteLine(sb);
14     }
15 }
```

16  
17 The output is

18

19 0123456789

20

21

22 012789

23

24

# 1 `StringBuilder.Replace(System.Char,` 2 `System.Char, System.Int32, System.Int32)` 3 **Method**

```
4 [ILAsm]  
5 .method public hidebysig instance class System.Text.StringBuilder  
6 Replace(valuetype System.Char oldChar, valuetype System.Char newChar,  
7 int32 startIndex, int32 count)  
  
8 [C#]  
9 public StringBuilder Replace(char oldChar, char newChar, int startIndex,  
10 int count)
```

## 11 **Summary**

12 Replaces all instances of a specified character in a specified range with another specified  
13 character.

## 14 **Parameters**

Parameter	Description
<i>oldChar</i>	The <code>System.Char</code> to replace.
<i>newChar</i>	The <code>System.Char</code> with which to replace <i>oldChar</i> .
<i>startIndex</i>	A <code>System.Int32</code> containing the index from which to start replacing <i>oldChar</i> .
<i>count</i>	A <code>System.Int32</code> containing the length of the range in which to replace <i>oldChar</i> .

## 15 16 **Return Value**

17 The current instance after substitution has occurred.

## 18 **Description**

19 This method substitutes each occurrence of *oldChar* in the specified range of the current  
20 instance with *newChar*.

21  
22 This method is case-sensitive.

## 23 **Exceptions**

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

**System.ArgumentOutOfRangeException**

The sum of *startIndex* and *count* is larger than the length of the current instance

-or-

*startIndex* or *count* is less than zero.

1

2

# 1 `StringBuilder.Replace(System.Char, System.Char)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Replace(valuetype System.Char oldChar, valuetype System.Char newChar)  
6 [C#]  
7 public StringBuilder Replace(char oldChar, char newChar)
```

## 8 Summary

9 Replaces all instances of a specified character in the current instance with another  
10 specified character.

## 11 Parameters

Parameter	Description
<i>oldChar</i>	The <code>System.Char</code> to replace.
<i>newChar</i>	The <code>System.Char</code> with which to replace <i>oldChar</i> .

12

## 13 Return Value

14 The current instance after substitution has occurred.

## 15 Description

16 This method substitutes each occurrence of *oldChar* in the current instance with  
17 *newChar*.

18 This method is case-sensitive.  
19

20

# 1 `StringBuilder.Replace(System.String, System.String, System.Int32, System.Int32)`

## 2 Method

```
4 [ILAsm]  
5 .method public hidebysig instance class System.Text.StringBuilder  
6 Replace(string oldValue, string newValue, int32 startIndex, int32 count)  
  
7 [C#]  
8 public StringBuilder Replace(string oldValue, string newValue, int  
9 startIndex, int count)
```

### 10 Summary

11 Replaces all instances of a specified string in a specified range with another specified  
12 string.

### 13 Parameters

Parameter	Description
<i>oldValue</i>	A <code>System.String</code> containing the string value to replace.
<i>newValue</i>	A <code>System.String</code> containing the string value with which to replace <i>oldValue</i> . Can be a null reference.
<i>startIndex</i>	A <code>System.Int32</code> containing the location from which to start replacing <i>oldValue</i> .
<i>count</i>	A <code>System.Int32</code> containing the length of the range in which to replace <i>oldValue</i> .

### 14 Return Value

15 The current instance after substitution has occurred.

### 17 Description

18 This method substitutes each occurrence of *oldValue* in the specified range of the  
19 current instance with *newValue*. [Note: If *newValue* is null, instances of *oldValue* are  
20 removed.]

21  
22  
23  
24 This method is case-sensitive.

### 25 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>oldValue</i> is a null reference.
<b>System.ArgumentOutOfRangeException</b>	<i>startIndex</i> or <i>count</i> is less than zero. -or- The sum of <i>startIndex</i> and <i>count</i> is greater than the length of the current instance.
<b>System.ArgumentException</b>	The length of <i>oldvalue</i> is zero.

1

2

# 1 `StringBuilder.Replace(System.String, System.String)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance class System.Text.StringBuilder  
5 Replace(string oldValue, string newValue)  
  
6 [C#]  
7 public StringBuilder Replace(string oldValue, string newValue)
```

## 8 Summary

9 Replaces all instances of a specified string with another specified string.

## 10 Parameters

Parameter	Description
<i>oldValue</i>	A <code>System.String</code> containing the string value to replace.
<i>newValue</i>	A <code>System.String</code> containing the string value with which to replace <i>oldValue</i> . Can be a null reference.

11

## 12 Return Value

13 The current instance after substitution has occurred.

## 14 Description

15 This method substitutes each occurrence of *oldValue* in the current instance with  
16 *newValue*. [Note: If *newValue* is null, instances of *oldValue* are removed.]

17

18

19

20 This method is case-sensitive.

## 21 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>oldValue</i> is a null reference.
<b>System.ArgumentException</b>	The length of <i>oldvalue</i> is zero.

22

23

# 1 **StringBuilder.ToString(System.Int32,**

## 2 **System.Int32) Method**

```
3 [ILAsm]  
4 .method public hidebysig instance string ToString(int32 startIndex, int32  
5 length)  
6 [C#]  
7 public string ToString(int startIndex, int length)
```

### 8 **Summary**

9 Returns a `System.String` representation of a substring of the current instance.

### 10 **Parameters**

Parameter	Description
<i>startIndex</i>	A <code>System.Int32</code> containing the index at which the substring begins.
<i>length</i>	A <code>System.Int32</code> containing the length of the substring.

11

### 12 **Return Value**

13 A new `System.String` representing the characters in the specified range.

### 14 **Exceptions**

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>startIndex</i> or <i>length</i> is less than zero.  -or-  The sum of <i>startIndex</i> and <i>length</i> is greater than the length of the current instance.

15

16

# 1 **StringBuilder.ToString() Method**

```
2 [ILAsm]  
3 .method public hidebysig virtual string ToString()  
4 [C#]  
5 public override string ToString()
```

## 6 **Summary**

7 Returns a `System.String` representation of the value of the current instance.

## 8 **Return Value**

9 A `System.String` representing the current instance.

## 10 **Description**

11 This method overrides `System.Object.ToString`.

12

# 1 StringBuilder.Capacity Property

```
2 [ILAsm]
3 .property int32 Capacity { public hidebysig specialname instance int32
4 get_Capacity() public hidebysig specialname instance void
5 set_Capacity(int32 value) }
6
7 [C#]
8 public int Capacity { get; set; }
```

## 8 Summary

9 Gets or sets the maximum number of characters that can be contained in the memory  
10 allocated by the current instance.

## 11 Property Value

12 A `System.Int32` containing the maximum number of characters that can be contained in  
13 the memory allocated by the current instance.

## 14 Description

15 The `System.Text.StringBuilder.Capacity` property does not affect the string value of  
16 the current instance. The `System.Text.StringBuilder` will dynamically increase the  
17 `System.Text.StringBuilder.Capacity` and allocate more space when it is required.

18  
19 [*Note:* For performance reasons a `System.Text.StringBuilder` might allocate more  
20 memory than needed. The amount of memory allocated is implementation specific.]  
21  
22

## 23 Exceptions

Exception	Condition
<code>System.ArgumentException</code>	The value specified for a set operation is less than <code>System.Text.StringBuilder.Length</code> .

24

25

# 1 StringBuilder.Chars Property

```
2 [ILAsm]  
3 .property valuetype System.Char Chars[int32 index] { public hidebysig  
4 specialname instance valuetype System.Char get_Chars(int32 index) public  
5 hidebysig specialname instance void set_Chars(int32 index, valuetype  
6 System.Char value) }  
7 [C#]  
8 public char this[int index] { get; set; }
```

## 9 Summary

10 Gets or sets the character at a specified position in the current instance.

## 11 Property Value

12 A `System.Char` containing the Unicode character at location *index* in the current  
13 instance.

## 14 Description

15 *index* is the position of a character within the `System.Text.StringBuilder`. The first  
16 character in the string is at index 0. The length of a string is the number of characters it  
17 contains. The last accessible character of a `System.Text.StringBuilder` instance is at  
18 the index `System.Text.StringBuilder.Length - 1`.

## 19 Exceptions

Exception	Condition
<b>System.IndexOutOfRangeException</b>	<i>index</i> is greater than or equal to the length of the current instance. -or- <i>index</i> is less than zero.

20

21

# 1 **StringBuilder.Length Property**

```
2 [ILAsm]  
3 .property int32 Length { public hidebysig specialname instance int32  
4 get_Length() public hidebysig specialname instance void set_Length(int32  
5 value) }  
6 [C#]  
7 public int Length { get; set; }
```

## 8 **Summary**

9 Gets or sets the length of the current instance.

## 10 **Property Value**

11 A System.Int32 containing the length of the current instance.

## 12 **Description**

13 If the specified length is less than the current length, the System.Text.StringBuilder  
14 is truncated to the specified length. If the specified length is greater than the current  
15 length, the end of the string value of the System.Text.StringBuilder is padded with  
16 spaces.

17  
18 If the specified length is greater than the current  
19 System.Text.StringBuilder.Capacity, System.Text.StringBuilder.Capacity is set  
20 to the specified length.

21  
22 [*Note:* A space in Unicode format is defined as the hexadecimal value 0x20.]  
23  
24

## 25 **Exceptions**

<b>Exception</b>	<b>Condition</b>
<b>System.ArgumentOutOfRangeException</b>	The value specified for a set operation is less than 0.

26  
27