

System.Text.StringBuilder Class

```
[ILASM]
.class public sealed serializable StringBuilder extends
System.Object

[C#]
public sealed class StringBuilder
```

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)

Type Attributes:

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

Summary

Represents a mutable string of characters.

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

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

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

Unlike most types in the Base Class Library, the arguments to

1 **System.Text.StringBuilder** members are assumed to be passed as
2 **in / out** arguments (passed by reference). [*Note:* Normally arguments
3 are **in** arguments (passed by value) unless explicitly declared as **out**
4 or **in / out**.]

5
6 [*Note:* An instance of **System.String** is said to be "immutable"
7 because its value cannot be modified once it has been created.
8 Methods on **System.String** that appear to modify a **System.String**
9 instance actually return a new instance containing the modification.
10 The **System.Text.StringBuilder** class provides methods that actually
11 modify the contents of a string-like object.

12
13 Relational operators only perform reference comparisons (unless
14 overloaded by a particular language compiler). Despite this restriction,
15 relational operators can be used to compare **System.String** objects
16 that are assigned literal values. Their values are immutable and can't
17 change, so a reference comparison is sufficient. Because
18 **System.Text.StringBuilder** instances are mutable, they should not
19 be compared with relational operators.

20
21 For performance reasons a **System.Text.StringBuilder** might
22 allocate more memory than needed. The amount of memory allocated
23 is implementation specific.]

24

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
8 **System.Text.StringBuilder** class.

9 **Description**

10 The new instance of **System.Text.StringBuilder** represents a string
11 equal to **System.String.Empty**. The
12 **System.Text.StringBuilder.Capacity** is set to the default capacity.
13
14 [*Note:* The default value of the **System.Text.StringBuilder.Capacity**
15 property is implementation dependent.]

16

1 **StringBuilder(System.Int32) Constructor**

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

7 **Summary**

8 Constructs and initializes a new, empty instance of the
9 **System.Text.StringBuilder** class, with a specified
10 **System.Text.StringBuilder.Capacity**.

11 **Parameters**

12
13

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

14
15

15 **Description**

16 If the specified capacity is less than the default capacity, the
17 **System.Text.StringBuilder.Capacity** of the new instance of
18 **System.Text.StringBuilder** is set to the default value. The
19 **System.Text.StringBuilder** will dynamically allocate more space
20 when it is required.

21
22
23
24

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

25
26

[*Note:* The default value of the **System.Text.StringBuilder.Capacity** property is implementation dependent.]

27

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

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

7 **Summary**

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

11 **Parameters**

12
13

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

14
15

15 **Description**

16 [Note: The **System.Text.StringBuilder.Capacity** of the new instance
17 is implementation defined.]

18

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

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

9 **Summary**

10 Appends multiple copies of a character to the end of the current
11 **System.Text.StringBuilder**.

12 **Parameters**

13
14

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

15
16
17

16 **Return Value**

18 The current instance after peroration has occurred.

19 **Description**

20 This method appends *repeatCount* copies of the specified character to
21 the current instance.
22
23 The **System.Text.StringBuilder.Capacity** of the current instance is
24 increased as necessary.

25 **Exceptions**

26
27

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

28
29
30

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

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

10 Summary

11 Appends the string representation of an array of Unicode characters to
12 the end of the current instance.

13 Parameters

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

16 Return Value

17 The current instance after peroration has occurred.

20 Description

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

24 The **System.Text.StringBuilder.Capacity** of the current instance is
25 increased as necessary.

27 Exceptions

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

1
2
3

System.ArgumentOutOfRangeException	<i>charCount</i> or <i>startIndex</i> is less than zero. -or- The sum of <i>startIndex</i> and <i>charCount</i> is greater than the length of <i>value</i> .
---	--

1 **StringBuilder.Append(System.String)**

2 **Method**

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

8 **Summary**

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

10 **Parameters**

11
12

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

13
14
15

14 **Return Value**

16 The current instance after peroration has occurred.

17 **Description**

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

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

23

StringBuilder.Append(System.String, System.Int32, System.Int32) Method

```
[ILASM]
.method public hidebysig instance class
System.Text.StringBuilder Append(string value, int32
startIndex, int32 count)

[C#]
public StringBuilder Append(string value, int startIndex,
int count)
```

Summary

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

Parameters

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

Return Value

The current instance after peroration has occurred.

Description

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

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

Exceptions

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

1
2
3

System.ArgumentOutOfRangeException	<i>charCount</i> or <i>startIndex</i> is less than zero. -or- The sum of <i>startIndex</i> and <i>charCount</i> is greater than the length of <i>value</i> .
---	--

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

2 **Method**

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 the current instance.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 The current instance after peroration has occurred.

18 **Description**

19 This method appends *value.ToString()* to the current instance.

20
21
22

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

23

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

2 **Method**

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

8 **Summary**

9 Appends the string representation of a **System.SByte** to the end of
10 the current instance.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

21 This member is not CLS-compliant. For a CLS-compliant alternative,
22 use **System.Text.StringBuilder.Append(System.Int16)**.

23
24 This method appends *value*.ToString() to the current instance.

25
26 The **System.Text.StringBuilder.Capacity** of the current instance is
27 increased as necessary.

28

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

2 **Method**

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

8 **Summary**

9 Appends the string representation of a **System.Byte** to the end of the
10 current instance.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 The current instance after peroration has occurred.

18 **Description**

19 This method appends *value.ToString()* to the current instance.

20
21
22

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

23

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

2 **Method**

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

9 **Summary**

10 Appends the string representation of a Unicode character to the end of
11 the current instance.

12 **Parameters**

13
14

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

15
16
17

16 **Return Value**

18 The current instance after peroration has occurred.

19 **Description**

20 This method appends the specified character to the current instance.

21
22
23

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

24

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

2 **Method**

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

8 **Summary**

9 Appends the string representation of a **System.Int16** to the end of
10 the current instance.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 The current instance after peroration has occurred.

18 **Description**

19 This method appends *value.ToString()* to the current instance.

20
21
22

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

23

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

2 **Method**

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

8 **Summary**

9 Appends the string representation of a **System.Int32** to the end of
10 the current instance.

11 **Parameters**

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

14 **Return Value**

15 The current instance after peroration has occurred.

18 **Description**

19 This method appends *value.ToString()* to the current instance.

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

23

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

2 **Method**

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

8 **Summary**

9 Appends the string representation of a **System.Int64** to the end of
10 the current instance.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 The current instance after peroration has occurred.

18 **Description**

19 This method appends *value.ToString()* to the current instance.

20
21
22

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

23

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

3 **StringBuilder.Append(System.Single)** 4 **Method**

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

10 **Summary**

11 Appends the string representation of a **System.Single** to the end of
12 the current instance.

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

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

22
23
24

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

25

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

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

```
5 [ILASM]  
6 .method public hidebysig instance class  
7 System.Text.StringBuilder 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
12 the current instance.

13 **Parameters**

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

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

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

22
23 The **System.Text.StringBuilder.Capacity** of the current instance is
24 increased as necessary.

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

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

```
5 [ILASM]  
6 .method public hidebysig instance class  
7 System.Text.StringBuilder 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
12 the current instance.

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

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

22
23
24

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

25

1 **StringBuilder.Append(System.UInt16)**

2 **Method**

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 the current instance.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

21 This member is not CLS-compliant. For a CLS-compliant alternative,
22 use **System.Text.StringBuilder.Append(System.Int32)**.

23
24 This method appends *value.ToString()* to the current instance.

25
26 The **System.Text.StringBuilder.Capacity** of the current instance is
27 increased as necessary.

28

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

2 **Method**

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 the current instance.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

21 This member is not CLS-compliant. For a CLS-compliant alternative,
22 use **System.Text.StringBuilder.Append(System.Int64)**.

23
24 This method appends *value*.ToString() to the current instance.

25
26 The **System.Text.StringBuilder.Capacity** of the current instance is
27 increased as necessary.

28

1 **StringBuilder.Append(System.UInt64)**

2 **Method**

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 the current instance.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after peroration has occurred.

20 **Description**

21 This member is not CLS-compliant. For a CLS-compliant alternative,
22 use **System.Text.StringBuilder.Append(System.Decimal)**.

23
24 This method appends *value.ToString()* to the current instance.

25
26 The **System.Text.StringBuilder.Capacity** of the current instance is
27 increased as necessary.

28

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

2 **Method**

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

8 **Summary**

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

11 **Parameters**

12 Parameter	13 Description
14 <i>value</i>	The System.Object to be appended.

15 **Return Value**

16
17 The current instance after peroration has occurred.

18 **Description**

19 This method appends *value.ToString()* to the current instance. If *value*
20 is a null reference, no changes are made.

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

24

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

2 **Method**

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

8 **Summary**

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

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 The current instance after peroration has occurred.

18 **Description**

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

22
23
24

The **System.Text.StringBuilder.Capacity** of the current instance is increased as necessary.

25

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

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

10 **Summary**

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

14 **Parameters**

15
16

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

17
18
19

18 **Return Value**

20 The current instance after peroration has occurred.

21 **Description**

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

24
25
26
27
28

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

29 **Exceptions**

30
31

Exception	Condition
System.ArgumentNullException	<i>format</i> or <i>arg0</i> is a null reference.
System.FormatException	<i>format</i> is invalid.

1

2 **Example**

3

4

[C#]

5

using System;

6

using System.Text;

7

8

public class StringBuilderTest {

9

public static void Main() {

10

StringBuilder sb = new StringBuilder("The high ");

11

Console.WriteLine(sb.AppendFormat("temperature today

12

was {0, 6}.". ", 88));

13

14

}

15

}

16

17

The output is

18

The high temperature today was 88.

19

20

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

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

10 **Summary**

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

14 **Parameters**

15
16

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

17
18
19

18 **Return Value**

20 The current instance after peroration has occurred.

21 **Description**

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

25
26 [Note: This version of **System.Text.StringBuilder.AppendFormat** is
27 equivalent to **System.Text.StringBuilder.AppendFormat**(**null**,
28 *format*, **new Object[]** {*arg0*, *arg1*}). For more information on the
29 format specification, see the **System.String** class overview.]

30 **Exceptions**

31
32

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

1

2 **Example**

3

4

[C#]

5

using System;

6

using System.Text;

7

8

public class StringBuilderTest {

9

public static void Main() {

10

StringBuilder sb = new StringBuilder("The high ");

11

Console.WriteLine(sb.AppendFormat("temperature today

12

was {0} {1}.", "very", "high"));

13

}

14

}

15

16

17

The output is

18

19

The high temperature today was very high.

20

21

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

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

11 Summary

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

15 Parameters

16
17

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

18
19
20

Return Value

21 The current instance after peroration has occurred.

22 Description

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

26
27 [Note: This version of **System.Text.StringBuilder.AppendFormat** is
28 equivalent to **System.Text.StringBuilder.AppendFormat**(**null**,
29 *format*, **new Object[]** { *arg0*, *arg1*, *arg2*}). For more information on
30 the format specification, see the **System.String** class overview.]

31 Exceptions

32
33

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

1

2

Example

3

4

[C#]

5

```
using System;
```

6

```
using System.Text;
```

7

8

```
public class StringBuilderTest {
```

9

```
    public static void Main() {
```

10

```
        StringBuilder sb = new StringBuilder("The high ");
```

11

```
        Console.WriteLine(sb.AppendFormat("temperature today
```

12

13

```
was {0} {1} {2}.", "very", "very", "high"));
```

14

```
    }
```

15

```
}
```

16

17

The output is

18

19

The high temperature today was very very high.

20

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

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

10 Summary

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

14 Parameters

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

17 Return Value

18 The current instance after peroration has occurred.

21 Description

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

25
26 [Note: This version of **System.Text.StringBuilder.AppendFormat** is
27 equivalent to **System.Text.StringBuilder.AppendFormat**(**null**,
28 *format*, *args*). For more information on the format specification see the
29 **System.String** class overview.]

30 Exceptions

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

1

2 **Example**

3

4

[C#]

5

using System;

6

using System.Text;

7

8

public class StringBuilderTest {

9

public static void Main() {

10

string [] strings = {"very", "very", null, "high"};

11

StringBuilder sb = new StringBuilder("The high ");

12

13

Console.WriteLine(sb.AppendFormat("temperature today
was {0}, {1} {2}{3}.", strings));

14

15

}

16

}

17

18

19

The output is

20

21

The high temperature today was very, very high.

22

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

```
4 [ILASM]  
5 .method public hidebysig instance class  
6 System.Text.StringBuilder AppendFormat(class  
7 System.IFormatProvider provider, string format, class  
8 System.Object[] args)
```

```
9 [C#]  
10 public StringBuilder AppendFormat(IFormatProvider provider,  
11 string format, params object[] args)
```

12 Summary

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

21 Return Value

22
23 The current instance after peroration has occurred.

24 Description

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

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

1 **Exceptions**

2

3

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

4

5 **Example**

6

7

[C#]

8

using System;

9

using System.Text;

10

public class StringBuilderTest {

11

public static void Main() {

12

13

string a = "very";

14

string b = "very";

15

string c = "high";

16

StringBuilder sb = new StringBuilder("The high ");

17

Console.WriteLine(sb.AppendFormat(null, "temperature

18

today was {0}, {1} {2}.", a, b, c));

19

}

20

}

21

}

22

23

24

The output is

25

26

The high temperature today was very, very high.

27

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

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

8 **Summary**

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

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 A **System.Int32** equal to the new
18 **System.Text.StringBuilder.Capacity** of the current instance.

19 **Description**

20 If the specified value is less than the current
21 **System.Text.StringBuilder.Capacity**, no changes are made and
22 **System.Text.StringBuilder.Capacity** remains the same.

23
24 [Note: For performance reasons, the new
25 **System.Text.StringBuilder.Capacity** might be larger than the
26 specified value. The amount of memory allocated by this method is
27 implementation specific.]

28 **Exceptions**

29
30

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

31
32
33

1 `StringBuilder.Equals(System.Text.StringBuilder)` Method

```
3 [ILASM]  
4 .method public hidebysig instance bool Equals(class  
5 System.Text.StringBuilder sb)  
  
6 [C#]  
7 public bool Equals(StringBuilder sb)
```

8 **Summary**

9 Determines whether the current instance and a specified
10 **System.Text.StringBuilder** have the same value.

11 **Parameters**

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

14 **Return Value**

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

19

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

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

10 Summary

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

13 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.
<i>count</i>	A System.Int32 containing the number of times the string is to be inserted.

16 Return Value

17 The current instance after insertion has occurred.
18

20 Description

21 This method inserts the *value* string *count* times into the current
22 instance at the specified location. Existing characters are shifted to
23 make room for the new text, and
24 **System.Text.StringBuilder.Capacity** is adjusted as necessary.
25

26 If *value* is **Empty** or a null reference, the **System.Text.StringBuilder**
27 is not changed.

28 Exceptions

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

1
2
3

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

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

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 instance at a specified index.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 The current instance after insertion has occurred.

18 **Description**

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

23
24 If *value* is **Empty** or a null reference, the **System.Text.StringBuilder**
25 is not changed.

26 **Exceptions**

27
28

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

1
2
3

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

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

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 current instance at a specified index.

11 **Parameters**

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

14 **Return Value**

15 The current instance after insertion has occurred.

18 **Description**

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

23 **Exceptions**

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

1
2
3

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

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 instance at the specified index.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 The current instance after insertion has occurred.

20 **Description**

21 This member is not CLS-compliant. For a CLS-compliant alternative,
22 use **System.Text.StringBuilder.Insert(System.Int32,**
23 **System.Int16)**.

24
25 This method inserts *value.ToString()* into the current instance at the
26 specified location. Existing characters are shifted to make room for the
27 new text, and **System.Text.StringBuilder.Capacity** is adjusted as
28 necessary.

29 **Exceptions**

30
31

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

1
2
3

System.ArgumentOutOfRangeException	<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.Byte)` Method

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

9 Summary

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

12 Parameters

13
14

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

15
16
17

16 Return Value

18 The current instance after insertion has occurred.

19 Description

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

24 Exceptions

25
26

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

1
2
3

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

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 instance at a specified index.

11 **Parameters**

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

14 **Return Value**

15 The current instance after insertion has occurred.

18 **Description**

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

23 **Exceptions**

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

1
2
3

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

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

9 **Summary**

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

12 **Parameters**

13
14

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

15
16
17

16 **Return Value**

18 The current instance after insertion has occurred.

19 **Description**

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

24 **Exceptions**

25
26

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

1
2
3

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

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

9 **Summary**

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

12 **Parameters**

13
14

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

15
16
17

16 **Return Value**

18 The current instance after insertion has occurred.

19 **Description**

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

24
25
26

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

27 **Exceptions**

28
29

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

1
2
3

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

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

```
4 [ILASM]  
5 .method public hidebysig instance class  
6 System.Text.StringBuilder Insert(int32 index, class  
7 System.Char[] value, int32 startIndex, int32 charCount)
```

```
8 [C#]  
9 public StringBuilder Insert(int index, char[] value, int  
10 startIndex, int charCount)
```

11 **Summary**

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

14 **Parameters**

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

17

18 **Return Value**

19

20 The current instance after insertion has occurred.

21 **Description**

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

26

27 If *value* is an empty array or a null reference and *startIndex* and
28 *charCount* are both zero, the **System.Text.StringBuilder** is not
29 changed.

1 Exceptions
2
3

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

4
5
6

1 **StringBuilder.Insert(System.Int32,** 2 **System.Int32) Method**

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 instance at the specified index.

11 **Parameters**

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

14 **Return Value**

15 The current instance after insertion has occurred.

18 **Description**

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

23 **Exceptions**

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

1
2
3

1 **StringBuilder.Insert(System.Int32,** 2 **System.Int64) Method**

```
3 [ILASM]  
4 .method public hidebysig instance class  
5 System.Text.StringBuilder 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
10 instance at the specified index.

11 **Parameters**

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

14 **Return Value**

15 The current instance after insertion has occurred.

18 **Description**

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

23 **Exceptions**

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

1
2
3

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

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

```
5 [ILASM]  
6 .method public hidebysig instance class  
7 System.Text.StringBuilder Insert(int32 index, float32  
8 value)  
9 [C#]  
10 public StringBuilder Insert(int index, float value)
```

11 **Summary**

12 Inserts the string representation of a **System.Single** into the current
13 instance at the specified index.

14 **Parameters**

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

18 **Return Value**

20 The current instance after insertion has occurred.

21 **Description**

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

26 **Exceptions**

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

- 1
- 2
- 3

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

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

StringBuilder.Insert(System.Int32, System.Double) Method

```
[ILASM]
.method public hidebysig instance class
System.Text.StringBuilder Insert(int32 index, float64
value)

[C#]
public StringBuilder Insert(int index, double value)
```

Summary

Inserts the string representation of a **System.Double** into the current instance at the specified index.

Parameters

Parameter	Description
<i>index</i>	A System.Int32 containing the index at which to insert.
<i>value</i>	The System.Double 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
System.ArgumentOutOfRangeException	<i>index</i> is greater than the length of the current instance -or-

- 1
- 2
- 3

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

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

StringBuilder.Insert(System.Int32, System.Decimal) Method

```
[ILASM]
.method public hidebysig instance class
System.Text.StringBuilder Insert(int32 index, decimal
value)

[C#]
public StringBuilder Insert(int index, decimal value)
```

Summary

Inserts the string representation of a **System.Decimal** into the current instance at the specified index.

Parameters

Parameter	Description
<i>index</i>	A System.Int32 containing the index at which to insert.
<i>value</i>	The System.Decimal 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
System.ArgumentOutOfRangeException	<i>index</i> is greater than the length of the current instance -or-

- 1
- 2
- 3

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

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

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

9 **Summary**

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

12 **Type Attributes:**

- 13 • CLSCompliantAttribute(false)

14 **Parameters**

15
16

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

17
18
19

18 **Return Value**

20 The current instance after insertion has occurred.

21 **Description**

22 This member is not CLS-compliant. For a CLS-compliant alternative,
23 use **System.Text.StringBuilder.Insert(System.Int32,**
24 **System.Int32)**.

25
26 This method inserts *value.ToString()* into the current instance at the
27 specified location. Existing characters are shifted to make room for the
28 new text, and **System.Text.StringBuilder.Capacity** is adjusted as
29 necessary.

30 **Exceptions**

31
32

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

- 1
- 2
- 3

System.ArgumentOutOfRangeException	<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.UInt32) Method**

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

9 **Summary**

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

12 **Type Attributes:**

- 13 • CLSCompliantAttribute(false)

14 **Parameters**

15
16

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

17
18
19

18 **Return Value**

20 The current instance after insertion has occurred.

21 **Description**

22 This member is not CLS-compliant. For a CLS-compliant alternative,
23 use **System.Text.StringBuilder.Insert(System.Int32,**
24 **System.Int64)**.

25
26 This method inserts *value.ToString()* into the current instance at the
27 specified location. Existing characters are shifted to make room for the
28 new text, and **System.Text.StringBuilder.Capacity** is adjusted as
29 necessary.

30 **Exceptions**

31
32

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

1
2
3

System.ArgumentOutOfRangeException	<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.UInt64) Method**

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

9 **Summary**

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

12 **Type Attributes:**

- 13 • CLSCompliantAttribute(false)

14 **Parameters**

15
16

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

17
18
19

18 **Return Value**

20 The current instance after insertion has occurred.

21 **Description**

22 This member is not CLS-compliant. For a CLS-compliant alternative,
23 use **System.Text.StringBuilder.Insert(System.Int32,**
24 **System.Decimal)**.

25
26 This method inserts *value.ToString()* into the current instance at the
27 specified location. Existing characters are shifted to make room for the
28 new text, and **System.Text.StringBuilder.Capacity** is adjusted as
29 necessary.

30 **Exceptions**

31
32

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

- 1
- 2
- 3

System.ArgumentOutOfRangeException	<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.Object) Method**

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

8 **Summary**

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

11 **Parameters**

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

14 **Return Value**

15 The current instance after insertion has occurred.

18 **Description**

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

23
24 If *value* is a null reference, the **System.Text.StringBuilder** is not
25 changed.

26 **Exceptions**

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

1
2
3

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

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

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

9 Summary

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

11 Parameters

12
13

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

14
15
16

15 Return Value

17 The current instance after removal has occurred.

18 Description

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

23
24
25
26

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

27 Exceptions

28
29

Exception	Condition
System.ArgumentOutOfRangeException	<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

4

[C#]

5

using System;

6

using System.Text;

7

8

public class StringBuilderTest {

9

public static void Main() {

10

StringBuilder sb = new StringBuilder("0123456789");

11

Console.WriteLine(sb);

12

sb.Remove(3, 4);

13

Console.WriteLine(sb);

14

}

15

}

16

17

18

The output is

19

0123456789

20

21

22

012789

23

24

25

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

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

10 Summary

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

13 Parameters

14
15

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

16
17
18

17 Return Value

19 The current instance after substitution has occurred.

20 Description

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

24
25

This method is case-sensitive.

26 Exceptions

27
28

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

29
30
31

1 `StringBuilder.Replace(System.String,` 2 `System.String, System.Int32,` 3 `System.Int32) Method`

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

11 Summary

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

14 Parameters

15
16

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

17
18
19

18 Return Value

20 The current instance after substitution has occurred.

21 Description

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

25
26 This method is case-sensitive.

27 Exceptions

28
29

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

1
2
3

1 **StringBuilder.Replace(System.Char,** 2 **System.Char) Method**

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

9 **Summary**

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

12 **Parameters**

13
14

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

15
16
17

16 **Return Value**

18 The current instance after substitution has occurred.

19 **Description**

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

22 This method is case-sensitive.
23

24

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

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

12 Summary

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

15 Parameters

16
17

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

18
19
20

Return Value

21 The current instance after substitution has occurred.

22 Description

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

25
26 This method is case-sensitive.

27 Exceptions

28
29

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

1
2
3

System.ArgumentOutOfRangeException	The sum of <i>startIndex</i> and <i>count</i> is larger than the length of the current instance -or- <i>startIndex</i> or <i>count</i> is less than zero.
---	---

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
8 instance.

9 **Return Value**

10

11 A **System.String** representing the current instance.

12 **Description**

13 This method overrides **System.Object.ToString**.

14

1 **StringBuilder.ToString(System.Int32,** 2 **System.Int32) Method**

```
3 [ILASM]  
4 .method public hidebysig instance string ToString(int32  
5 startIndex, int32 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
10 instance.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 A new **System.String** representing the characters in the specified
18 range.

19 **Exceptions**

20
21

Exception	Condition
System.ArgumentOutOfRangeException	<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.

22
23
24

1 **StringBuilder.Capacity** Property

```
2 [ILASM]
3 .property int32 Capacity { public hidebysig specialname
4 instance int32 get_Capacity() public hidebysig specialname
5 instance void 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
10 in the memory allocated by the current instance.

11 **Property Value**

12

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

15 **Description**

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

21

22 [*Note:* For performance reasons a **System.Text.StringBuilder** might
23 allocate more memory than needed. The amount of memory allocated
24 is implementation specific.]

25 **Exceptions**

26

27

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

28

29

30

1 **StringBuilder.Chars** Property

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

10 **Summary**

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

13 **Property Value**

15 A **System.Char** containing the Unicode character at location *index* in
16 the current instance.

17 **Description**

18 *index* is the position of a character within the
19 **System.Text.StringBuilder**. The first character in the string is at
20 index 0. The length of a string is the number of characters it contains.
21 The last accessible character of a **System.Text.StringBuilder**
22 instance is at the index **System.Text.StringBuilder.Length** - 1.

23 **Exceptions**

24
25

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

26
27
28

1 **StringBuilder.Length** Property

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

8 **Summary**

9 Gets or sets the length of the current instance.

10 **Property Value**

11

12 A **System.Int32** containing the length of the current instance.

13 **Description**

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

19
20 If the specified length is greater than the current
21 **System.Text.StringBuilder.Capacity**,
22 **System.Text.StringBuilder.Capacity** is set to the specified length.

23
24 [*Note:* A space in Unicode format is defined as the hexadecimal value
25 0x20.]

26 **Exceptions**

27

28

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than System.Int32.MaxValue .

29

30