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 00]
- *Version:* 2.0.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 `System.Text.StringBuilder` members are assumed to be passed as `in` / `out` arguments (passed by reference). [*Note:* Normally arguments are `in` arguments (passed by value) unless explicitly declared as `out` or `in` / `out`.]

[*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.

]

StringBuilder() Constructor

[ILAsm]
```
public rtspecialname specialname instance void .ctor()
```

[C#]
```
public StringBuilder()
```

Summary

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

Description

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

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

StringBuilder(System.String) Constructor

[ILAsm]
```
public rtspecialname specialname instance void .ctor(string value)
```

[C#]
```
public StringBuilder(string value)
```

Summary

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

Parameters

Parameter	Description
value	A `System.String` containing the string value of the new instance of `System.Text.StringBuilder`.

Description

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

StringBuilder(System.Int32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int32 capacity)

[C#]
public StringBuilder(int capacity)
```

Summary

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

Parameters

Parameter	Description
capacity	A `System.Int32` containing the starting number of characters allowed in the `System.Text.StringBuilder`.

Description

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

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

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*capacity* is less than zero.

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(valuetype System.Char value, int32 repeatCount)

[C#]
public StringBuilder Append(char value, int repeatCount)
```

Summary

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

Parameters

Parameter	**Description**
value	The `System.Char` to be appended.
repeatCount	A `System.Int32` containing the number of times to append *value*.

Return Value

The current instance after the operation has occurred.

Description

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

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

Exceptions

Exception	**Condition**
System.ArgumentOutOfRangeException	*repeatCount* is less than zero.

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(char[] value, int32 startIndex, int32 charCount)

[C#]
public StringBuilder Append(char[] value, int startIndex, int
charCount)
```

Summary

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

Parameters

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

Return Value

The current instance after the operation has occurred.

Description

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

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

Exceptions

Exception	**Condition**
System.ArgumentNullException	*value* is a null reference, and *startIndex* and *charCount* 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*.

StringBuilder.Append(System.String) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(string value)

[C#]
public StringBuilder Append(string value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.String` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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**
value	The `System.String` from which the substring will be taken.
startIndex	A `System.Int32` containing the index in *value* from which to start copying.
count	A `System.Int32` containing the number of characters to copy from *value*.

Return Value

The current instance after the operation 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	*value* is a null reference, and *startIndex* and *count* 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*.

StringBuilder.Append(System.Boolean) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(bool value)

[C#]
public StringBuilder Append(bool value)
```

Summary

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

Parameters

Parameter	Description
value	A `System.Boolean` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

StringBuilder.Append(System.SByte) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(int8 value)

[C#]
public StringBuilder Append(sbyte value)
```

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
value	The `System.SByte` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.Byte) Method

[ILAsm]

```
.method public hidebysig instance class System.Text.StringBuilder
Append(unsigned int8 value)
```

[C#]

```
public StringBuilder Append(byte value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Byte` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

StringBuilder.Append(System.Char) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(valuetype System.Char value)

[C#]
public StringBuilder Append(char value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Char` to be appended.

Return Value

The current instance after the operation has occurred.

Description

This method appends the specified character to the current instance.

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

StringBuilder.Append(System.Int16) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(int16 value)

[C#]
public StringBuilder Append(short value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Int16` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

StringBuilder.Append(System.Int32) Method

[ILAsm]
```
.method public hidebysig instance class System.Text.StringBuilder
Append(int32 value)
```

[C#]
```
public StringBuilder Append(int value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Int32` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

StringBuilder.Append(System.Int64) Method

[ILAsm]
```
.method public hidebysig instance class System.Text.StringBuilder
Append(int64 value)
```

[C#]
```
public StringBuilder Append(long value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Int64` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.Single) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(float32 value)

[C#]
public StringBuilder Append(float value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Single` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.Double) Method

[ILAsm]
```
.method public hidebysig instance class System.Text.StringBuilder
Append(float64 value)
```

[C#]
```
public StringBuilder Append(double value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Double` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.Decimal) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(decimal value)

[C#]
public StringBuilder Append(decimal value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Decimal` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

StringBuilder.Append(System.UInt16) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(unsigned int16 value)

[C#]
public StringBuilder Append(ushort value)
```

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
value	The `System.UInt16` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.UInt32) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(unsigned int32 value)

[C#]
public StringBuilder Append(uint value)
```

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
value	The `System.UInt32` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.UInt64) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(unsigned int64 value)

[C#]
public StringBuilder Append(ulong value)
```

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
value	The `System.UInt64` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

StringBuilder.Append(System.Object) Method

[ILAsm]
```
.method public hidebysig instance class System.Text.StringBuilder
Append(object value)
```

[C#]
```
public StringBuilder Append(object value)
```

Summary

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

Parameters

Parameter	Description
value	The `System.Object` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

StringBuilder.Append(System.Char[]) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Append(char[] value)

[C#]
public StringBuilder Append(char[] value)
```

Summary

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

Parameters

Parameter	Description
value	The array of `System.Char` to be appended.

Return Value

The current instance after the operation has occurred.

Description

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

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

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
AppendFormat(string format, object arg0, object arg1, object arg2)

[C#]
public StringBuilder AppendFormat(string format, object arg0, object
arg1, object arg2)
```

Summary

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

Parameters

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

Return Value

The current instance after the operation has occurred.

Description

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

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

Exceptions

Exception	Condition
System.ArgumentNullException	*format* is a null reference.
System.FormatException	*format* is invalid.

Example

[C#]

```
using System;
using System.Text;

public class StringBuilderTest {
  public static void Main() {

     StringBuilder sb = new StringBuilder("The high ");
     Console.WriteLine( sb.AppendFormat("temperature today was {0} {1}
{2}.", "very", "very", "high") );
  }
}
```

The output is

```
The high temperature today was very very high.
```

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
AppendFormat(class System.IFormatProvider provider, string format,
object[] args)

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

Summary

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

Parameters

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

Return Value

The current instance after the operation has occurred.

Description

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

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

Exceptions

Exception	**Condition**
System.ArgumentNullException	*format* or *args* is a null reference.
System.FormatException	*format* is invalid.

Example

[C#]

```
using System;
using System.Text;

public class StringBuilderTest {
  public static void Main() {

     string a = "very";
     string b = "very";
     string c = "high";

     StringBuilder sb = new StringBuilder("The high ");
     Console.WriteLine(sb.AppendFormat(null, "temperature today was
{0}, {1} {2}.", a, b, c) );
  }
}
```

The output is

```
The high temperature today was very, very high.
```

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
AppendFormat(string format, object[] args)

[C#]
public StringBuilder AppendFormat(string format, params object[]
args)
```

Summary

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

Parameters

Parameter	Description
format	A `System.String` containing zero or more format specifications.
args	A `System.Object` array to be formatted.

Return Value

The current instance after operation has occurred.

Description

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

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

Exceptions

Exception	Condition
System.ArgumentNullException	*format* is a null reference.

System.FormatException	*format* is invalid.

Example

[C#]

```
using System;
using System.Text;

public class StringBuilderTest {
  public static void Main() {

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

     StringBuilder sb = new StringBuilder("The high ");
     Console.WriteLine( sb.AppendFormat("temperature today was {0}, {1}
{2}{3}.", strings) );
  }
}
```

The output is

```
The high temperature today was very, very high.
```

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

[ILAsm]

```
.method public hidebysig instance class System.Text.StringBuilder
AppendFormat(string format, object arg0, object arg1)
```

[C#]

```
public StringBuilder AppendFormat(string format, object arg0, object
arg1)
```

Summary

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

Parameters

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

Return Value

The current instance after the operation has occurred.

Description

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

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

Exceptions

Exception	**Condition**

System.ArgumentNullException	*format* is a null reference.
System.FormatException	*format* is invalid.

Example

[C#]

```
using System;
using System.Text;

public class StringBuilderTest {
  public static void Main() {

     StringBuilder sb = new StringBuilder("The high ");
     Console.WriteLine( sb.AppendFormat("temperature today was {0}
{1}.", "very", "high") );
  }
}
```

The output is

```
The high temperature today was very high.
```

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
AppendFormat(string format, object arg0)

[C#]
public StringBuilder AppendFormat(string format, object arg0)
```

Summary

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

Parameters

Parameter	Description
format	A `System.String` containing zero or more format specifications.
arg0	A `System.Object` to be formatted.

Return Value

The current instance after the operation has occurred.

Description

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

[*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.]

Exceptions

Exception	Condition
System.ArgumentNullException	*format* is a null reference.
System.FormatException	*format* is invalid.

Example

[C#]

```
using System;
using System.Text;

public class StringBuilderTest {
  public static void Main() {

      StringBuilder sb = new StringBuilder("The high ");
      Console.WriteLine( sb.AppendFormat("temperature today was {0,
6}.", 88) );
  }
}
```

The output is

```
The high temperature today was     88.
```

StringBuilder.EnsureCapacity(System.Int32) Method

```
[ILAsm]
.method public hidebysig instance int32 EnsureCapacity(int32
capacity)

[C#]
public int EnsureCapacity(int capacity)
```

Summary

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

Parameters

Parameter	Description
capacity	A `System.Int32` containing the minimum capacity to ensure.

Return Value

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

Description

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

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*capacity* is less zero.

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
sb	A `System.Text.StringBuilder`.

Return Value

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

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

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

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.String` to be inserted.

Return Value

The current instance after insertion has occurred.

Description

This method inserts *value* 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.

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*index* is greater than the length of the current instance -or- *index* is less than zero.

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

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

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Boolean` value 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	*index* is greater than the length of the current instance -or- *index* is less than zero

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, string value, int32 count)

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

Summary

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

Parameters

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

Return Value

The current instance after insertion has occurred.

Description

This method inserts the *value* string *count* times 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.

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*index* is greater than the length of the current instance -or-

	index is less than zero -or- *count* is less than zero

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, int8 value)

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

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.SByte` to be inserted.

Return Value

The current instance after insertion has occurred.

Description

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

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	*index* is greater than the length of the current instance

	-or- *index* is less than zero

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, unsigned int8 value)

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Byte` 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	*index* is greater than the length of the current instance -or- *index* is less than zero

StringBuilder.Insert(System.Int32, System.Int16) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, int16 value)

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Int16` 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	*index* is greater than the length of the current instance -or- *index* is less than zero.

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, valuetype System.Char value)

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Char` 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	*index* is greater than the length of the current instance -or- *index* is less than zero.

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

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

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

Summary

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

Parameters

Parameter	**Description**
index	A `System.Int32` containing the index at which to insert.
value	The `System.Char` array to be inserted.

Return Value

The current instance after insertion has occurred.

Description

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

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

Exceptions

Exception	**Condition**
System.ArgumentOutOfRangeException	*index* is greater than the length of the current instance -or-

	index is less than zero.

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, char[] value, int32 startIndex, int32 charCount)

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

Summary

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

Parameters

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

Return Value

The current instance after insertion has occurred.

Description

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

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

Exceptions

Exception	Condition

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

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

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

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Object` 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.

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*index* is greater than the length of the current instance -or- *index* is less than zero.

StringBuilder.Insert(System.Int32, System.UInt64) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, unsigned int64 value)

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

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	**Description**
index	A `System.Int32` containing the index at which to insert.
value	The `System.UInt64` to be inserted.

Return Value

The current instance after insertion has occurred.

Description

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

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	*index* is greater than the length of the current instance

	-or- *index* is less than zero.

StringBuilder.Insert(System.Int32, System.UInt32) Method

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

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

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	**Description**
index	A `System.Int32` containing the index at which to insert.
value	The `System.UInt32` to be inserted.

Return Value

The current instance after insertion has occurred.

Description

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

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	*index* is greater than the length of the current instance

	-or- *index* is less than zero.

StringBuilder.Insert(System.Int32, System.UInt16) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, unsigned int16 value)

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

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.UInt16` to insert.

Return Value

The current instance after insertion has occurred.

Description

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

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	*index* is greater than the length of the current instance

	-or- *index* 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**
index	A `System.Int32` containing the index at which to insert.
value	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	*index* is greater than the length of the current instance -or- *index* 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
index	A `System.Int32` containing the index at which to insert.
value	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	*index* is greater than the length of the current instance -or- *index* is less than zero

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
index	A `System.Int32` containing the index at which to insert.
value	The `System.Single` 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	*index* is greater than the length of the current instance -or- *index* is less than zero

StringBuilder.Insert(System.Int32, System.Int64) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Insert(int32 index, int64 value)

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Int64` 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	*index* is greater than the length of the current instance -or- *index* is less than zero.

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

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

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

Summary

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

Parameters

Parameter	Description
index	A `System.Int32` containing the index at which to insert.
value	The `System.Int32` 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	*index* is greater than the length of the current instance -or- *index* is less than zero

StringBuilder.Remove(System.Int32, System.Int32) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Remove(int32 startIndex, int32 length)

[C#]
public StringBuilder Remove(int startIndex, int length)
```

Summary

Removes a specified range of characters from the current instance.

Parameters

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

Return Value

The current instance after removal has occurred.

Description

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

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*startIndex* or *length* is less than zero -or- The sum of *startIndex* and *length* is greater than the length of the current

	instance.

Example

[C#]

```
using System;
using System.Text;

public class StringBuilderTest  {
   public static void Main()  {

      StringBuilder sb = new StringBuilder("0123456789");
      Console.WriteLine(sb);
      sb.Remove(3, 4);
      Console.WriteLine(sb);
   }
}
```

The output is

```
0123456789


012789
```

StringBuilder.Replace(System.Char, System.Char, System.Int32, System.Int32) Method

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Replace(valuetype System.Char oldChar, valuetype System.Char
newChar, int32 startIndex, int32 count)

[C#]
public StringBuilder Replace(char oldChar, char newChar, int
startIndex, int count)
```

Summary

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

Parameters

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

Return Value

The current instance after substitution has occurred.

Description

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

This method is case-sensitive.

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.

StringBuilder.Replace(System.Char, System.Char) Method

[ILAsm]
```
.method public hidebysig instance class System.Text.StringBuilder
Replace(valuetype System.Char oldChar, valuetype System.Char
newChar)
```

[C#]
```
public StringBuilder Replace(char oldChar, char newChar)
```

Summary

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

Parameters

Parameter	Description
oldChar	The `System.Char` to replace.
newChar	The `System.Char` with which to replace *oldChar*.

Return Value

The current instance after substitution has occurred.

Description

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

This method is case-sensitive.

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Replace(string oldValue, string newValue, int32 startIndex, int32
count)

[C#]
public StringBuilder Replace(string oldValue, string newValue, int
startIndex, int count)
```

Summary

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

Parameters

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

Return Value

The current instance after substitution has occurred.

Description

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

This method is case-sensitive.

Exceptions

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

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

```
[ILAsm]
.method public hidebysig instance class System.Text.StringBuilder
Replace(string oldValue, string newValue)

[C#]
public StringBuilder Replace(string oldValue, string newValue)
```

Summary

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

Parameters

Parameter	Description
oldValue	A `System.String` containing the string value to replace.
newValue	A `System.String` containing the string value with which to replace *oldValue*. Can be a null reference.

Return Value

The current instance after substitution has occurred.

Description

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

This method is case-sensitive.

Exceptions

Exception	Condition
System.ArgumentNullException	*oldValue* is a null reference.
System.ArgumentException	The length of *oldvalue* is zero.

StringBuilder.ToString(System.Int32, System.Int32) Method

```
[ILAsm]
.method public hidebysig instance string ToString(int32 startIndex,
int32 length)

[C#]
public string ToString(int startIndex, int length)
```

Summary

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

Parameters

Parameter	Description
startIndex	A `System.Int32` containing the index at which the substring begins.
length	A `System.Int32` containing the length of the substring.

Return Value

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*startIndex* or *length* is less than zero. -or- The sum of *startIndex* and *length* is greater than the length of the current instance.

StringBuilder.ToString() Method

[ILAsm]
```
.method public hidebysig virtual string ToString()
```

[C#]
```
public override string ToString()
```

Summary

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

Return Value

A `System.String` representing the current instance.

Description

This method overrides `System.Object.ToString`.

StringBuilder.Capacity Property

[ILAsm]

```
.property int32 Capacity { public hidebysig specialname instance
int32 get_Capacity() public hidebysig specialname instance void
set_Capacity(int32 value) }
```

[C#]

```
public int Capacity { get; set; }
```

Summary

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

Property Value

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

Description

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

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

Exceptions

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

StringBuilder.Chars Property

[ILAsm]

```
.property valuetype System.Char Chars[int32 index] { public
hidebysig specialname instance valuetype System.Char get_Chars(int32
index) public hidebysig specialname instance void set_Chars(int32
index, valuetype System.Char value) }
```

[C#]

```
public char this[int index] { get; set; }
```

Summary

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

Property Value

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

Description

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

Exceptions

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

StringBuilder.Length Property

[ILAsm]

```
.property int32 Length { public hidebysig specialname instance int32
get_Length() public hidebysig specialname instance void
set_Length(int32 value) }
```

[C#]

```
public int Length { get; set; }
```

Summary

Gets or sets the length of the current instance.

Property Value

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

Description

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

If the specified length is greater than the current `System.Text.StringBuilder.Capacity`, `System.Text.StringBuilder.Capacity` is set to the specified length.

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

Exceptions

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