

System.Char Structure

```
[ILASM]
.class public sequential sealed serializable Char extends
System.ValueType implements System.IComparable

[C#]
public struct Char: IComparable
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Implements:

- **System.IComparable**

Summary

Represents a Unicode character.

Inherits From: System.ValueType

Library: BCL

Thread Safety: This type is safe for multithreaded operations.

Description

The **System.Char** value type represents Unicode characters, with code points ranging from 0 to 65,535.

[*Note:* The *code point* of a Unicode character is that character's 2-byte, encoded value.]

[*Note:* The **System.Globalization.UnicodeCategory** enumeration describes the categories that a Unicode character can be mapped to. For information on mapping specific Unicode characters to Unicode categories, see the UnicodeData.txt file in the Unicode Character Database at <http://www.unicode.org/Public/UNIDATA/UnicodeCharacterDatabase.html>. The UnicodeData.txt file format is described at <http://www.unicode.org/Public/3.1-Update/UnicodeData-3.1.0.html>.]

1 Char.MaxValue Field

```
2 [ILASM]  
3 .field public static literal valuetype System.Char MaxValue  
4 = (char)0xFFFF  
5 [C#]  
6 public const char MaxValue = (char)0xFFFF
```

7 Summary

8 Contains the maximum code point for the **System.Char** type.

9 Description

10 The numeric value of this constant is 65,535.

11

1 Char.MinValue Field

```
2 [ILASM]  
3 .field public static literal valuetype System.Char MinValue  
4 = (char)0x0  
5 [C#]  
6 public const char MinValue = (char)0x0
```

7 Summary

8 Contains the minimum code point for the **System.Char** type.

9 Description

10 The numeric value of this constant is 0.

11

1 Char.CompareTo(System.Object) Method

```
2 [ILASM]  
3 .method public final hidebysig virtual int32  
4 CompareTo(object value)  
  
5 [C#]  
6 public int CompareTo(object value)
```

7 Summary

8 Returns the sort order of the current instance compared to the
9 specified **System.Object**.

10 Parameters

Parameter	Description
<i>value</i>	The System.Object to compare to the current instance.

14 Return Value

16 A **System.Int32** containing a value that reflects the sort order of the
17 current instance as compared to *value*. The following table defines the
18 conditions under which the returned value is a negative number, zero,
19 or a positive number.

Return Value	Description
Any negative number	Current instance < <i>value</i> .
Zero	Current instance == <i>value</i> .
Any positive number	Current instance > <i>value</i> , or <i>value</i> is a null reference.

21 Description

22 The comparison performed by this method is based on the code points
23 of the current instance and *value*, not necessarily their lexicographical
24 characteristics.

26 [Note: This method is implemented to support the
27 **System.IComparable** interface.]

28 Exceptions

1
2
3

Exception	Condition
System.ArgumentException	<i>value</i> is not a System.Char and is not a null reference.

1 Char.Equals(System.Object) Method

```
2 [ILASM]  
3 .method public hidebysig virtual bool Equals(object obj)  
4 [C#]  
5 public override bool Equals(object obj)
```

6 Summary

7 Determines whether the current instance and the specified
8 **System.Object** represent the same type and value.

9 Parameters

10
11

Parameter	Description
<i>obj</i>	The System.Object to compare to the current instance.

12
13
14

Return Value

15 **true** if *obj* represents the same type and value as the current
16 instance. If *obj* is a null reference or is not an instance of
17 **System.Char**, returns **false**.

18 Description

19 The comparison performed by this method is based on the code points
20 of the current instance and *obj*, not necessarily their lexicographical
21 characteristics.

22
23
24

[Note: This method overrides **System.Object.Equals**.]

1 Char.GetHashCode() Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4 [C#]  
5 public override int GetHashCode()
```

6 Summary

7 Generates a hash code for the current instance.

8 Return Value

9

10 A **System.Int32** value containing a hash code for the current
11 instance.

12 Description

13 The algorithm used to generate the hash code is unspecified.

14

15 [*Note:* This method overrides **System.Object.GetHashCode.**]

16

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

3 Char.GetNumericValue(System.Char) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static float64  
7 GetNumericValue(valuetype System.Char c)  
  
8 [C#]  
9 public static double GetNumericValue(char c)
```

10 Summary

11 Returns the numeric value associated with the specified Unicode
12 character.

13 Parameters

Parameter	Description
c	A Unicode character.

17 Return Value

19 A **System.Double** representing the numeric value associated with *c* if
20 and only if *c* has an associated numeric value; otherwise, -1.0.

21 Description

22 A character has an associated numeric value if and only if it is a
23 member of one of the following categories in
24 **System.Globalization.UnicodeCategory**:
25 **System.Globalization.UnicodeCategory.DecimalDigitNumber**,
26 **System.Globalization.UnicodeCategory.LetterNumber**, or
27 **System.Globalization.UnicodeCategory.OtherNumber**.

28 Example

30 The following example demonstrates the
31 **System.Char.GetNumericValue** method.

```
32 [C#]  
33  
34 using System;  
35 public class GetNumericValueExample {
```

```
1 public static void Main() {
2     Char[] cAry = {'8', 'V', Convert.ToChar(0X00BC)};
3     //Unicode U+00BC is the code point for the character
4     //representation of 1/4
5     foreach(Char c in cAry) {
6         Console.Write("Numeric value of Unicode " +
7             "character {0} ", c);
8         Console.WriteLine(" is {0}",
9             Char.GetNumericValue(c));
10    }
11 }
12 }
```

13 The output is

14
15 Numeric value of Unicode character 8 is 8

16
17
18 Numeric value of Unicode character V is -1

19
20
21 Numeric value of Unicode character ¼ is 0.25

22

23

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

3 Char.GetNumericValue(System.String, 4 System.Int32) Method

```
5 [ILASM]  
6 .method public hidebysig static float64  
7 GetNumericValue(string s, int32 index)  
  
8 [C#]  
9 public static double GetNumericValue(string s, int index)
```

10 Summary

11 Returns the numeric value associated with the Unicode character at
12 the specified position in the specified **System.String**.

13 Parameters

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies the position of a character in <i>s</i> .

17 Return Value

19 A **System.Double** representing the numeric value associated with the
20 **System.Char** at position *index* in *s* if and only if that **System.Char**
21 has an associated numeric value; otherwise, -1.0.

22 Description

23 A character has an associated numeric value if and only if it is a
24 member of one of the following categories in
25 **System.Globalization.UnicodeCategory**:
26 **System.Globalization.UnicodeCategory.DecimalDigitNumber**,
27 **System.Globalization.UnicodeCategory.LetterNumber**, or
28 **System.Globalization.UnicodeCategory.OtherNumber**.

29 Exceptions

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero. or

- 1
- 2
- 3

	greater than or equal to the length of s.
--	---

1 Char.GetUnicodeCategory(System.Char)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static valuetype  
5 System.Globalization.UnicodeCategory  
6 GetUnicodeCategory(valuetype System.Char c)  
  
7 [C#]  
8 public static UnicodeCategory GetUnicodeCategory(char c)
```

9 Summary

10 Determines the **System.Globalization.UnicodeCategory** of the
11 specified Unicode character.

12 Parameters

13
14

Parameter	Description
c	A Unicode character.

15
16
17

16 Return Value

18 The **System.Globalization.UnicodeCategory** of *c*.

19 Description

20 [Note: For more information regarding Unicode categories, see
21 **System.Globalization.UnicodeCategory**.]

22

1 Char.GetUnicodeCategory(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static valuetype  
5 System.Globalization.UnicodeCategory  
6 GetUnicodeCategory(string s, int32 index)  
  
7 [C#]  
8 public static UnicodeCategory GetUnicodeCategory(string s,  
9 int index)
```

10 Summary

11 Determines the **System.Globalization.UnicodeCategory** of the
12 character at the specified position in the specified **System.String**.

13 Parameters

14
15

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies the position of a character in <i>s</i> .

16
17
18

Return Value

19 The **System.Globalization.UnicodeCategory** of the **System.Char**
20 at position *index* in *s*.

21 Description

22 [Note: For more information regarding Unicode categories, see
23 **System.Globalization.UnicodeCategory**.]

24 Exceptions

25
26

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

27
28
29

1 Char.IsControl(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsControl(valuetype  
4 System.Char c)  
5 [C#]  
6 public static bool IsControl(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a control
9 character.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 true if *c* is a member of the following category in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.Control**; otherwise, **false**.

19

1 Char.IsControl(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsControl(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsControl(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a control character.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of the following
18 category in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.Control**; otherwise, **false**.

20 Exceptions

21
22

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

23
24
25

1 Char.IsDigit(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsDigit(valuetype  
4 System.Char c)  
5 [C#]  
6 public static bool IsDigit(char c)
```

7 Summary

8 Determines whether a Unicode character is a decimal digit.

9 Parameters

10
11

Parameter	Description
c	A Unicode character.

12
13
14

Return Value

15 **true** if *c* is a member of the following category in
16 **System.Globalization.UnicodeCategory**:
17 **System.Globalization.UnicodeCategory.DecimalDigitNumber**;
18 otherwise, **false**.

19 Description

20 [Note: **System.Char.IsDigit** determines if a Char is a radix-10 digit.
21 This contrasts with **System.Char.IsNumber**, which determines if a
22 **System.Char** is of any numeric Unicode category.]

23

1 Char.IsDigit(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsDigit(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsDigit(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a decimal digit.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

15 Return Value

17 **true** if the character at position *index* in *s* is a member of the following
18 category in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.DecimalDigitNumber**;
20 otherwise, **false**.

21 Description

22 [Note: **System.Char.IsDigit** determines if a **System.Char** is a radix-
23 10 digit. This contrasts with **System.Char.IsNumber**, which
24 determines if a **System.Char** is of any numeric Unicode category.]

25 Exceptions

26
27

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

28
29
30

1 Char.IsLetter(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsLetter(valuetype  
4 System.Char c)  
  
5 [C#]  
6 public static bool IsLetter(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a letter.

9 Parameters

10
11

Parameter	Description
c	A Unicode character.

12
13
14

Return Value

15 **true** if *c* is a member of one of the following categories in
16 **System.Globalization.UnicodeCategory**:
17 **System.Globalization.UnicodeCategory.UppercaseLetter**,
18 **System.Globalization.UnicodeCategory.LowercaseLetter**,
19 **System.Globalization.UnicodeCategory.TitlecaseLetter**,
20 **System.Globalization.UnicodeCategory.ModifierLetter**, or
21 **System.Globalization.UnicodeCategory.OtherLetter**; otherwise,
22 **false**.

23

1 Char.IsLetter(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsLetter(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsLetter(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a letter.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of one of the
18 following categories in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.UppercaseLetter**,
20 **System.Globalization.UnicodeCategory.LowercaseLetter**,
21 **System.Globalization.UnicodeCategory.TitlecaseLetter**,
22 **System.Globalization.UnicodeCategory.ModifierLetter**, or
23 **System.Globalization.UnicodeCategory.OtherLetter**; otherwise,
24 **false**.

25 Exceptions

26
27

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

28
29
30

1 Char.IsLetterOrDigit(System.Char)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static bool  
5 IsLetterOrDigit(valuetype System.Char c)  
  
6 [C#]  
7 public static bool IsLetterOrDigit(char c)
```

8 Summary

9 Determines whether the specified Unicode character is either a letter
10 or a decimal digit.

11 Parameters

12
13

Parameter	Description
c	A Unicode character.

14
15
16

Return Value

17 **true** if *c* is a member of one of the following categories in
18 **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.UppercaseLetter**,
20 **System.Globalization.UnicodeCategory.LowercaseLetter**,
21 **System.Globalization.UnicodeCategory.TitlecaseLetter**,
22 **System.Globalization.UnicodeCategory.ModifierLetter**,
23 **System.Globalization.UnicodeCategory.OtherLetter**, or
24 **System.Globalization.UnicodeCategory.DecimalDigitNumber**;
25 otherwise, **false**.

26

1 Char.IsLetterOrDigit(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsLetterOrDigit(string  
5 s, int32 index)  
  
6 [C#]  
7 public static bool IsLetterOrDigit(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is either a letter or a decimal digit.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of one of the
18 following categories in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.UppercaseLetter**,
20 **System.Globalization.UnicodeCategory.LowercaseLetter**,
21 **System.Globalization.UnicodeCategory.TitlecaseLetter**,
22 **System.Globalization.UnicodeCategory.ModifierLetter**,
23 **System.Globalization.UnicodeCategory.OtherLetter**, or
24 **System.Globalization.UnicodeCategory.DecimalDigitNumber**;
25 otherwise, **false**.
26
27

28 Exceptions

29
30

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

31
32
33

1 Char.IsLower(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsLower(valuetype  
4 System.Char c)  
5 [C#]  
6 public static bool IsLower(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a lowercase
9 letter.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 **true** if *c* is a member of the following category in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.LowercaseLetter**;
19 otherwise, **false**.

20

1 Char.IsLower(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsLower(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsLower(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a lowercase letter.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

15 Return Value

17 **true** if the character at position *index* in *s* is a member of the following
18 category in
19 **System.Globalization.UnicodeCategory: System.Globalization.Un**
20 **icodeCategory.LowercaseLetter**; otherwise, **false**.

21 Exceptions

22
23

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

24
25
26

1 Char.IsNumber(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsNumber(valuetype  
4 System.Char c)  
5 [C#]  
6 public static bool IsNumber(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a number.

9 Parameters

10
11

Parameter	Description
c	A Unicode character.

12
13
14

Return Value

15 **true** if *c* is a member of one of the following categories in
16 **System.Globalization.UnicodeCategory**:
17 **System.Globalization.UnicodeCategory.DecimalDigitNumber**,
18 **System.Globalization.UnicodeCategory.LetterNumber**, or
19 **System.Globalization.UnicodeCategory.OtherNumber**; otherwise,
20 **false**.

21 Description

22 [Note: **System.Char.IsNumber** determines if a **System.Char** is of
23 any numeric Unicode category. This contrasts with
24 **System.Char.IsDigit**, which determines if a **System.Char** is a radix-
25 10 digit.]

26

1 Char.IsNumber(System.String, 2 System.Int32) Method

```
3 [ILASM  
4 .method public hidebysig static bool IsNumber(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsNumber(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a number.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of one of the
18 following categories in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.DecimalDigitNumber**,
20 **System.Globalization.UnicodeCategory.LetterNumber**, or
21 **System.Globalization.UnicodeCategory.OtherNumber**; otherwise,
22 **false**.

23 Description

24 [Note: **System.Char.IsNumber** determines if a **System.Char** is of
25 any numeric Unicode category. This contrasts with
26 **System.Char.IsDigit**, which determines if a **System.Char** is a radix-
27 10 digit.]

28 Exceptions

29
30

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

1
2
3

1 Char.IsPunctuation(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool  
4 IsPunctuation(valuetype System.Char c)  
5  
6 [C#]  
7 public static bool IsPunctuation(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a punctuation
9 mark.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 **true** if *c* is a member of one of the following categories in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.ConnectorPunctuation**,
19 **System.Globalization.UnicodeCategory.DashPunctuation**,
20 **System.Globalization.UnicodeCategory.OpenPunctuation**,
21 **System.Globalization.UnicodeCategory.ClosePunctuation**,
22 **System.Globalization.UnicodeCategory.InitialQuotePunctuation**,
23
24 **System.Globalization.UnicodeCategory.FinalQuotePunctuation**,
25 or **System.Globalization.UnicodeCategory.OtherPunctuation**;
26 otherwise, **false**.

27

1 Char.IsPunctuation(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsPunctuation(string  
5 s, int32 index)  
  
6 [C#]  
7 public static bool IsPunctuation(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a punctuation mark.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of one of the
18 following categories in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.ConnectorPunctuation**,
20 **System.Globalization.UnicodeCategory.DashPunctuation**,
21 **System.Globalization.UnicodeCategory.OpenPunctuation**,
22 **System.Globalization.UnicodeCategory.ClosePunctuation**,
23 **System.Globalization.UnicodeCategory.InitialQuotePunctuation**
24 ,
25 **System.Globalization.UnicodeCategory.FinalQuotePunctuation**,
26 or **System.Globalization.UnicodeCategory.OtherPunctuation**;
27 otherwise, **false**.

28 Exceptions

29
30

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

31
32
33

1 Char.IsSeparator(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsSeparator(valuetype  
4 System.Char c)  
  
5 [C#]  
6 public static bool IsSeparator(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a separator
9 character.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 **true** if *c* is a member of one of the following categories in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.SpaceSeparator**,
19 **System.Globalization.UnicodeCategory.LineSeparator**, or
20 **System.Globalization.UnicodeCategory.ParagraphSeparator**;
21 otherwise, **false**.

22

1 Char.IsSeparator(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsSeparator(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsSeparator(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a separator character.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of one of the
18 following categories in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.SpaceSeparator**,
20 **System.Globalization.UnicodeCategory.LineSeparator**, or
21 **System.Globalization.UnicodeCategory.ParagraphSeparator**;
22 otherwise, **false**.

23 Exceptions

24
25

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

26
27
28

1 Char.IsSurrogate(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsSurrogate(valuetype  
4 System.Char c)  
  
5 [C#]  
6 public static bool IsSurrogate(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a surrogate
9 character.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 **true** if *c* is a member of the following category in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.Surrogate**; otherwise,
19 **false**.

20

1 Char.IsSurrogate(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsSurrogate(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsSurrogate(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a surrogate character.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

15 Return Value

17 **true** if the character at position *index* in *s* is a member of the following
18 category in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.Surrogate**; otherwise,
20 **false**.

21 Exceptions

22
23

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

24
25
26

1 Char.IsSymbol(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsSymbol(valuetype  
4 System.Char c)  
5 [C#]  
6 public static bool IsSymbol(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a symbol
9 character.

10 Parameters

Parameter	Description
c	A Unicode character.

14 Return Value

16 **true** if *c* is a member of one of the following categories in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.MathSymbol**,
19 **System.Globalization.UnicodeCategory.CurrencySymbol**,
20 **System.Globalization.UnicodeCategory.ModifierSymbol**, or
21 **System.Globalization.UnicodeCategory.OtherSymbol**; otherwise,
22 **false**.

23

1 Char.IsSymbol(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsSymbol(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsSymbol(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a symbol character.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* is a member of one of the
18 following categories in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.MathSymbol**,
20 **System.Globalization.UnicodeCategory.CurrencySymbol**,
21 **System.Globalization.UnicodeCategory.ModifierSymbol**, or
22 **System.Globalization.UnicodeCategory.OtherSymbol**; otherwise,
23 **false**.

24 Exceptions

25
26

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

27
28
29

1 Char.IsUpper(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsUpper(valuetype  
4 System.Char c)  
5 [C#]  
6 public static bool IsUpper(char c)
```

7 Summary

8 Determines whether the specified Unicode character is an uppercase
9 letter.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 **true** if *c* is a member of the following category in
17 **System.Globalization.UnicodeCategory**:
18 **System.Globalization.UnicodeCategory.UppercaseLetter**;
19 otherwise, **false**.

20

1 Char.IsUpper(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsUpper(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsUpper(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is an uppercase letter.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

15 Return Value

17 **true** if the character at position *index* in *s* is a member of the following
18 category in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.UppercaseLetter**;
20 otherwise, **false**.

21 Exceptions

22
23

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

24
25
26

1 Char.IsWhiteSpace(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static bool IsWhiteSpace(valuetype  
4 System.Char c)  
  
5 [C#]  
6 public static bool IsWhiteSpace(char c)
```

7 Summary

8 Determines whether the specified Unicode character is a whitespace
9 character.

10 Parameters

11
12

Parameter	Description
c	A Unicode character.

13
14
15

Return Value

16 **true** if *c* either has a code point of 0x0009, 0x000a, 0x000b, 0x000c,
17 0x000d, 0x0085, 0x2028, or 0x2029; or is a member of the following
18 category in **System.Globalization.UnicodeCategory**:
19 **System.Globalization.UnicodeCategory.SpaceSeparator**;
20 otherwise, **false**.

21

1 Char.IsWhiteSpace(System.String, 2 System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig static bool IsWhiteSpace(string s,  
5 int32 index)  
  
6 [C#]  
7 public static bool IsWhiteSpace(string s, int index)
```

8 Summary

9 Determines whether the character at the specified position in the
10 specified **System.String** is a whitespace character.

11 Parameters

12
13

Parameter	Description
<i>s</i>	A System.String .
<i>index</i>	A System.Int32 that specifies a character position in <i>s</i> .

14
15
16

Return Value

17 **true** if the character at position *index* in *s* either has a code point of
18 0x0009, 0x000a, 0x000b, 0x000c, 0x000d, 0x0085, 0x2028, or
19 0x2029; or is a member of the following category in
20 **System.Globalization.UnicodeCategory**:
21 **System.Globalization.UnicodeCategory.SpaceSeparator**;
22 otherwise, **false**.

23 Exceptions

24
25

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.ArgumentOutOfRangeException	The value of <i>index</i> is less than zero, or greater than or equal to the length of <i>s</i> .

26
27
28

1 Char.Parse(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static valuetype System.Char  
4 Parse(string s)  
5  
6 [C#]  
7 public static char Parse(string s)
```

7 Summary

8 Returns the specified **System.String** converted to a **System.Char**
9 value.

10 Parameters

Parameter	Description
s	A System.String containing a single Unicode character.

13

14 Return Value

15

16 The **System.Char** value obtained from s.

17 Exceptions

18

19

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s does not contain exactly one character.

20

21

22

1 Char.ToLower(System.Char) Method

```
2 [ILASM]
3 .method public hidebysig static valuetype System.Char
4 ToLower(valuetype System.Char c)
5
6 [C#]
7 public static char ToLower(char c)
```

7 Summary

8 Converts a **System.Char** to its lowercase equivalent.

9 Parameters

10
11

Parameter	Description
c	A Unicode character.

12
13
14

13 Return Value

15 The lowercase equivalent of *c*, or the value of *c* if and only if *c* is
16 already lowercase or does not have a lowercase equivalent.

17 Example

18

19 The following example demonstrates the **System.Char.ToLower**
20 method.

21
22

```
23 [C#]
24 using System;
25 public class CharToLower {
26     public static void Main() {
27         Char[] cAry = {'A', 'c', '*'};
28         foreach (Char c in cAry) {
29             Console.WriteLine("Char '{0}' ToLower is ", c);
30             Console.WriteLine("{0}", Char.ToLower(c));
31         }
32     }
33 }
```

33 The output is

34
35

Char 'A' ToLower is a

1
2
3
4
5
6
7
8

Char 'c' ToLower is c

Char '*' ToLower is *

1 Char.ToString(System.IFormatProvider)

2 Method

```
3 [ILASM]  
4 .method public final hidebysig virtual string  
5 ToString(class System.IFormatProvider provider)  
  
6 [C#]  
7 public string ToString(IFormatProvider provider)
```

8 Summary

9 Converts the value of this instance to its equivalent **String**
10 representation using the specified culture-specific format information.

11 Parameters

12
13

Parameter	Description
<i>provider</i>	(Reserved) An System.IFormatProvider interface implementation that supplies culture-specific formatting information.

14
15
16

15 Return Value

17 The **System.String** representation of the value of this instance as
18 specified by *provider*.

19 Description

20 *provider* is ignored; it does not participate in this operation.

21

1 Char.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** representation of the current instance.

12 Description

13 [*Note:* This method overrides **System.Object.ToString.**]
14

1 Char.ToUpper(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static valuetype System.Char  
4 ToUpper(valuetype System.Char c)  
5 [C#]  
6 public static char ToUpper(char c)
```

7 Summary

8 Converts a **System.Char** to its uppercase equivalent.

9 Parameters

10
11

Parameter	Description
c	A Unicode character.

12

13 Return Value

14

15 The uppercase equivalent of *c*, or the value of *c* if and only if *c* is
16 already uppercase or does not have an uppercase equivalent.

17 Example

18

19 The following example demonstrates the **System.Char.ToUpper**
20 method.

21
22

```
[C#]  
23 using System;  
24 public class CharToUpper {  
25     public static void Main() {  
26         Char[] cAry = {'A', 'c', '*'};  
27         foreach (Char c in cAry) {  
28             Console.WriteLine("Char '{0}' ToUpper is {1}",  
29                 c, Char.ToUpper(c));  
30             Console.WriteLine();  
31         }  
32     }  
33 }
```

34 The output is

35

36 Char 'A' ToUpper is A

1
2
3
4
5
6
7
8

Char 'c' ToUpper is C

Char '*' ToUpper is *