

1 System.Globalization.NumberFormatInfo

2 Class

```
3 [ILAsm]  
4 .class public sealed serializable NumberFormatInfo extends System.Object  
5 implements System.ICloneable, System.IFormatProvider  
  
6 [C#]  
7 public sealed class NumberFormatInfo: ICloneable, IFormatProvider
```

8 Assembly Info:

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

14 Implements:

- 15 • **System.ICloneable**
- 16 • **System.IFormatProvider**

17 Summary

18 Supplies culture-specific formatting information for string representations of numeric
19 values.

20 Inherits From: System.Object

21
22 **Library:** BCL

23
24 **Thread Safety:** This type is safe for multithreaded operations.
25

26 Description

27 System.Globalization.NumberFormatInfo supplies symbols such as currency symbols
28 and decimal separators.

29
30 *[Note: A System.Globalization.NumberFormatInfo instance typically contains the set*
31 *of symbols for a specific language and culture. Instances of*
32 *System.Globalization.NumberFormatInfo can be created to provide customized*
33 *formatting information.]*
34
35

36

1 NumberFormatInfo() Constructor

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

6 Summary

7 Constructs and initializes a new instance of the
8 System.Globalization.NumberFormatInfo class.

9 Description

10 The new instance is not read-only, and is otherwise identical to the
11 System.Globalization.NumberFormatInfo instance returned by the
12 System.Globalization.NumberFormatInfo.InvariantInfo property.

13

1 NumberFormatInfo.Clone() Method

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

6 Summary

7 Creates a copy of the current instance.

8 Return Value

9 A `System.Object` that is a copy of the current instance.

10 Description

11 The `System.Globalization.NumberFormatInfo.Clone` method returns a new instance
12 of `System.Globalization.NumberFormatInfo` with property values that are equal to
13 the property values of the current instance except for the
14 `System.Globalization.DateTimeFormatInfo.IsReadOnly` property, which is always
15 `false`.

16
17 [*Note:* This method is implemented to support the `System.ICloneable` interface.]
18
19

20

1 NumberFormatInfo.GetFormat(System.Type) 2 Method

```
3 [ILAsm]  
4 .method public final hidebysig virtual object GetFormat(class System.Type  
5 formatType)  
  
6 [C#]  
7 public object GetFormat(Type formatType)
```

8 Summary

9 Returns an object of the specified type that provides formatting services.

10 Parameters

Parameter	Description
<i>formatType</i>	The <code>System.Type</code> of the formatting object to be returned.

11

12 Return Value

13 The current instance, if *formatType* is of type
14 `System.Globalization.NumberFormatInfo`; otherwise, a null reference.

15 Description

16 [Note: This method is implemented to support the `System.IFormatProvider` interface.]
17
18

19

1
2 **NumberFormatInfo.ReadOnly(System.Globalization.NumberFormatInfo) Method**
3

```
4 [ILAsm]  
5 .method public hidebysig static class  
6 System.Globalization.NumberFormatInfo ReadOnly(class  
7 System.Globalization.NumberFormatInfo nfi)  
  
8 [C#]  
9 public static NumberFormatInfo ReadOnly(NumberFormatInfo nfi)
```

10 **Summary**

11 Creates a read-only copy of the specified System.Globalization.NumberFormatInfo
12 instance.

13 **Parameters**

Parameter	Description
<i>nfi</i>	A System.Globalization.NumberFormatInfo object to copy.

14
15 **Return Value**

16 A System.Object that is a copy of the current instance, and cannot be altered.

17 **Exceptions**

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

18
19

1 NumberFormatInfo.CurrencyDecimalDigits 2 Property

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

9 Summary

10 Gets or sets the number of decimal places in currency values.

11 Property Value

12 A `System.Int32` containing the number of decimal places in currency values.

13 Description

14 The culture-invariant value for this property is 2.

15 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 99.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

16

17

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

NumberFormatInfo.CurrencyDecimalSeparator Property

```
[ILAsm]  
.property string CurrencyDecimalSeparator { public hidebysig specialname  
instance string get_CurrencyDecimalSeparator() public hidebysig  
specialname instance void set_CurrencyDecimalSeparator(string value) }  
  
[C#]  
public string CurrencyDecimalSeparator { get; set; }
```

Summary

Gets or sets the symbol used as the decimal separator in currency values.

Property Value

A `System.String` containing the decimal separator used in currency values.

Description

The culture-invariant value for this property is ".".

Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

1 NumberFormatInfo.CurrencyGroupSeparator 2 Property

```
3 [ILAsm]  
4 .property string CurrencyGroupSeparator { public hidebysig specialname  
5 instance string get_CurrencyGroupSeparator() public hidebysig specialname  
6 instance void set_CurrencyGroupSeparator(string value) }  
7 [C#]  
8 public string CurrencyGroupSeparator { get; set; }
```

9 Summary

10 Gets or sets the symbol used to separate groups of digits to the left of the decimal point
11 in currency values.

12 Property Value

13 A `System.String` containing the group separator used in currency values.

14 Description

15 The culture-invariant value for this property is ",".

16 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

17

18

1 NumberFormatInfo.CurrencyGroupSizes

2 Property

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

9 Summary

10 Gets or sets the number of digits in each group to the left of the decimal point in
11 currency values.

12 Property Value

13 A *System.Int32* array containing elements that define the number of digits in each
14 group in currency values.

15 Description

16 All elements of the array except the last are required to be between 1 and 9, inclusive.
17 The last element can be 0.

18
19 The first element of the array defines the number of elements in the first group of digits
20 located immediately to the left of the
21 *System.Globalization.NumberFormatInfo.CurrencyDecimalSeparator*. Each
22 subsequent element refers to the next group of digits located to the left of the previous
23 group. If the last element of the array is not zero, any remaining digits are grouped
24 based on the last element of the array. If the last element is zero, the remaining digits
25 are not grouped.

26
27 The culture-invariant value for this property is an array with a single element containing
28 the value 3.

29 Exceptions

Exception	Condition
System.ArgumentNullException	The array specified for a set operation is a null reference.
System.ArgumentException	One of the elements in the array specified for a set operation is not between 0 and 9. -or-

	The array contains an element, other than the last element, that is set to 0.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

1

2 Example

3 The following example demonstrates the effects of different
4 `System.Globalization.NumberFormatInfo.CurrencyGroupSizes` property values.

5 [C#]

```

7 using System;
8 using System.Globalization;
9 class Test {
10     public static void Main() {
11         NumberFormatInfo nfi = new NumberFormatInfo();
12
13         decimal myMoney = 9999999994444333221.00m;
14         nfi.CurrencyGroupSizes = new int[] {1,2,3,4,0};
15         Console.WriteLine("{0}",myMoney.ToString("C",nfi));
16
17         myMoney = 123456789123456.78m;
18         nfi.CurrencyGroupSizes = new int[] {3};
19         Console.WriteLine("{0}",myMoney.ToString("C",nfi));
20
21         nfi.CurrencyGroupSizes = new int[] {3,0};
22         Console.WriteLine("{0}",myMoney.ToString("C",nfi));
23     }
24 }
25
26

```

27 The output is

28 \$999999999,4444,333,22,1.00

29

30

31 \$123,456,789,123,456.78

32

33

34 \$123456789123,456.78

35

36

37

1 NumberFormatInfo.CurrencyNegativePattern 2 Property

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

9 Summary

10 Gets or sets the format of negative currency values.

11 Property Value

12 A System.Int32 between 0 and 15 inclusive, which specifies the format of negative
13 currency values.

14 Description

15 The following table describes the valid values for this property. "\$" is used as the value
16 for System.Globalization.NumberFormatInfo.CurrencySymbol, "-" is used as the
17 value for System.Globalization.NumberFormatInfo.NegativeSign, and 999
18 represents any numeric value.

Value	Pattern
0	(\$999)
1	-\$999
2	\$-999
3	\$999-
4	(999\$)
5	-999\$
6	999-\$
7	999\$-
8	-999 \$

9	-\$ 999
10	999 \$-
11	\$ 999-
12	\$ -999
13	999- \$
14	(\$ 999)
15	(999 \$)

1
2 The culture-invariant value for this property is 0.

3 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 15.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

4 5 Example

6 The following example demonstrates the effects of different
7 `System.Globalization.NumberFormatInfo.CurrencyNegativePattern` property
8 values.

```
9
10 [C#]
11 using System;
12 using System.Globalization;
13 class Test {
14     public static void Main() {
15         NumberFormatInfo nfi = new NumberFormatInfo();
16         decimal myMoney = -99999999999999.00m;
17         for (int i = 0; i<=15; i++) {
18             nfi.CurrencyNegativePattern = i;
19             Console.WriteLine("pattern # {0}: {1}", i, myMoney.ToString("C", nfi));
20         }
21     }
22 }
```

1
2 The output is
3
4 pattern # 0: (\$9,999,999,999,999.00)
5
6
7 pattern # 1: -\$9,999,999,999,999.00
8
9
10 pattern # 2: \$-9,999,999,999,999.00
11
12
13 pattern # 3: \$9,999,999,999,999.00-
14
15
16 pattern # 4: (9,999,999,999,999.00\$)
17
18
19 pattern # 5: -9,999,999,999,999.00\$
20
21
22 pattern # 6: 9,999,999,999,999.00-\$
23
24
25 pattern # 7: 9,999,999,999,999.00\$-
26
27
28 pattern # 8: -9,999,999,999,999.00 \$
29
30
31 pattern # 9: -\$ 9,999,999,999,999.00
32
33
34 pattern # 10: 9,999,999,999,999.00 \$-
35
36
37 pattern # 11: \$ 9,999,999,999,999.00-
38
39
40 pattern # 12: \$ -9,999,999,999,999.00
41
42
43 pattern # 13: 9,999,999,999,999.00- \$
44
45
46 pattern # 14: (\$ 9,999,999,999,999.00)
47
48

1 pattern # 15: (9,999,999,999,999.00 \$)

2

3

1 NumberFormatInfo.CurrencyPositivePattern 2 Property

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

9 Summary

10 Gets or sets the format of positive currency values.

11 Property Value

12 A System.Int32 between 0 and 3 inclusive, containing the format of positive currency
13 values.

14 Description

15 The following table describes the valid values for this property. "\$" is used as the value
16 for System.Globalization.NumberFormatInfo.CurrencySymbol, and 999 represents
17 any numeric value.

Value	Pattern
0	\$999
1	999\$
2	\$ 999
3	999 \$

18
19 The culture-invariant value for this property is 0.

20 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 3.

System.InvalidOperationException

The current instance is read-only and a set operation was attempted.

1

2 Example

3 The following example demonstrates the effects of different
4 System.Globalization.NumberFormatInfo.CurrencyPositivePattern property
5 values.

6

7 [C#]

```
8 using System;
9 using System.Globalization;
10 class Test {
11     public static void Main() {
12         NumberFormatInfo nfi = new NumberFormatInfo();
13         decimal myMoney = 999999999999.00m;
14         for (int i = 0; i<=3; i++) {
15             nfi.CurrencyPositivePattern = i;
16             Console.WriteLine("pattern # {0}: {1}", i, myMoney.ToString("C", nfi));
17         }
18     }
19 }
```

20
21 The output is

22

23 pattern # 0: \$9,999,999,999,999.00

24

25

26 pattern # 1: 9,999,999,999,999.00\$

27

28

29 pattern # 2: \$ 9,999,999,999,999.00

30

31

32 pattern # 3: 9,999,999,999,999.00 \$

33

34

1 NumberFormatInfo.CurrencySymbol Property

```
2 [ILAsm]  
3 .property string CurrencySymbol { public hidebysig specialname instance  
4 string get_CurrencySymbol() public hidebysig specialname instance void  
5 set_CurrencySymbol(string value) }  
  
6 [C#]  
7 public string CurrencySymbol { get; set; }
```

8 Summary

9 Gets or sets the currency symbol.

10 Property Value

11 A System.String containing the currency symbol.

12 Description

13 The culture-invariant value for this property is the Unicode currency symbol 0x00a4.

14 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

15

16

1 NumberFormatInfo.CurrentInfo Property

```
2 [ILAsm]  
3 .property class System.Globalization.NumberFormatInfo CurrentInfo { public  
4 hidebysig static specialname class System.Globalization.NumberFormatInfo  
5 get_CurrentInfo() }  
6 [C#]  
7 public static NumberFormatInfo CurrentInfo { get; }
```

8 Summary

9 Gets a System.Globalization.NumberFormatInfo instance containing formatting
10 information for the current system culture.

11 Property Value

12 A read-only System.Globalization.NumberFormatInfo containing the settings for the
13 current system culture.

14 Description

15 This property is read-only.

16

1 NumberFormatInfo.InvariantInfo Property

```
2 [ILAsm]
3 .property class System.Globalization.NumberFormatInfo InvariantInfo {
4 public hidebysig static specialname class
5 System.Globalization.NumberFormatInfo get_InvariantInfo() }
6
7 [C#]
8 public static NumberFormatInfo InvariantInfo { get; }
```

8 Summary

9 Gets a System.Globalization.NumberFormatInfo instance containing formatting
10 information that is culture-independent and does not change.

11 Property Value

12 A read-only System.Globalization.NumberFormatInfo with property values which are
13 universally supported. The property values of the returned
14 System.Globalization.NumberFormatInfo are not impacted by changes to the current
15 culture.

16 Description

17 This property is read-only.

18
19 The following table lists the property values of the
20 System.Globalization.NumberFormatInfo returned by this property.

Property	Default	Description
CurrencyDecimalDigits	2	The number of decimal places in currency values.
CurrencyDecimalSeparator	","	The string used as the decimal separator in currency values.
CurrencyGroupSeparator	","	The string used to separate groups of digits to the left of the decimal point in currency values.
CurrencyGroupSizes	3	The number of digits in each group to the left of the decimal point in currency values.
CurrencyNegativePattern	0	The format of negative currency values.
CurrencyPositivePattern	0	The format of positive currency values.
CurrencySymbol	0x00a4	The Unicode currency symbol.

NaNSymbol	"NaN"	The string used to represent undefined floating-point values.
NegativeInfinitySymbol	"-Infinity"	The string used to represent negative infinities.
NegativeSign	"-"	The string used to indicate negative values.
NumberDecimalDigits	2	The default number of decimal places.
NumberDecimalSeparator	","	The string used as the decimal separator.
NumberGroupSeparator	","	The string used to separate groups of digits to the left of the decimal point.
NumberGroupSizes	3	The number of digits in each group to the left of the decimal point.
NumberNegativePattern	1	The format of negative values.
PercentDecimalDigits	2	The default number of decimal places in percent values.
PercentDecimalSeparator	","	The string used as the decimal separator in percent values.
PercentGroupSeparator	","	The string used to separate groups of digits to the left of the decimal point in percent values.
PercentGroupSizes	3	The number of digits in each group to the left of the decimal in percent values.
PercentNegativePattern	0	The format of negative percent values.
PercentPositivePattern	0	The format of positive percent values.
PercentSymbol	"%"	The percent symbol.
PerMilleSymbol	"‰"	The per mille symbol.
PositiveInfinitySymbol	"Infinity"	The string used to represent positive infinities.
PositiveSign	"+"	The string used to indicate positive values.

1

2

1 NumberFormatInfo.IsReadOnly Property

```
2 [ILAsm]  
3 .property bool IsReadOnly { public hidebysig specialname instance bool  
4 get_IsReadOnly() }  
  
5 [C#]  
6 public bool IsReadOnly { get; }
```

7 Summary

8 Gets a value indicating whether the current instance is read-only.

9 Property Value

10 true if the current instance is read-only; otherwise false.

11 Description

12 This property is read-only.

13

14 [*Note:* Attempting to perform an assignment to a property of a read-only
15 System.Globalization.NumberFormatInfo causes a
16 System.InvalidOperationException.]

17

18

19

1 NumberFormatInfo.NaNSymbol Property

```
2 [ILAsm]  
3 .property string NaNSymbol { public hidebysig specialname instance string  
4 get_NaNSymbol() public hidebysig specialname instance void  
5 set_NaNSymbol(string value) }  
  
6 [C#]  
7 public string NaNSymbol { get; set; }
```

8 Summary

9 Gets or sets the symbol that represents NaN (Not-a-Number) floating-point values.

10 Property Value

11 A `System.String` containing the symbol for NaN values.

12 Description

13 The culture-invariant value for this property is "NaN".

14
15 [*Note:* For more information on NaN values, see `System.Double` or `System.Single`.]
16
17

18 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

19

20

1 NumberFormatInfo.NegativeInfinitySymbol

2 Property

```
3 [ILAsm]  
4 .property string NegativeInfinitySymbol { public hidebysig specialname  
5 instance string get_NegativeInfinitySymbol() public hidebysig specialname  
6 instance void set_NegativeInfinitySymbol(string value) }  
7 [C#]  
8 public string NegativeInfinitySymbol { get; set; }
```

9 Summary

10 Gets or sets the symbol that represents negative infinity.

11 Property Value

12 A System.String containing the symbol for negative infinity.

13 Description

14 The culture-invariant value for this property is "-Infinity".

15
16 [*Note:* For more information on negative infinity, see System.Double or
17 System.Single.]
18
19

20 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

21

22

1 NumberFormatInfo.NegativeSign Property

```
2 [ILAsm]
3 .property string NegativeSign { public hidebysig specialname instance
4 string get_NegativeSign() public hidebysig specialname instance void
5 set_NegativeSign(string value) }
6 [C#]
7 public string NegativeSign { get; set; }
```

8 Summary

9 Gets or sets the symbol used to represent negative values.

10 Property Value

11 A System.String containing the symbol that indicates a value is negative.

12 Description

13 The culture-invariant value for this property is "-".

14 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

15

16

1 NumberFormatInfo.NumberDecimalDigits 2 Property

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

9 Summary

10 Gets or sets the number of decimal places for numeric values.

11 Property Value

12 A `System.Int32` containing the number of decimal places for numeric values.

13 Description

14 The culture-invariant value for this property is 2.

15 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 99.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

16

17

1 NumberFormatInfo.NumberDecimalSeparator 2 Property

```
3 [ILAsm]  
4 .property string NumberDecimalSeparator { public hidebysig specialname  
5 instance string get_NumberDecimalSeparator() public hidebysig specialname  
6 instance void set_NumberDecimalSeparator(string value) }  
7 [C#]  
8 public string NumberDecimalSeparator { get; set; }
```

9 Summary

10 Gets or sets the symbol used as the decimal separator for numeric values.

11 Property Value

12 A System.String containing the decimal separator.

13 Description

14 The culture-invariant value for this property is ".".

15 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

16

17

1 NumberFormatInfo.NumberGroupSeparator 2 Property

```
3 [ILAsm]  
4 .property string NumberGroupSeparator { public hidebysig specialname  
5 instance string get_NumberGroupSeparator() public hidebysig specialname  
6 instance void set_NumberGroupSeparator(string value) }  
7  
8 [C#]  
9 public string NumberGroupSeparator { get; set; }
```

9 Summary

10 Gets or sets the symbol used to separate groups of digits to the left of the decimal point
11 for numeric values.

12 Property Value

13 A `System.String` containing the group separator.

14 Description

15 The culture-invariant value for this property is ",".

16 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

17

18

1 NumberFormatInfo.NumberGroupSizes

2 Property

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

9 Summary

10 Gets or sets the number of digits in each group to the left of the decimal point for
11 numeric values.

12 Property Value

13 A *System.Int32* array containing elements that define the number of digits in each
14 group in numeric values.

15 Description

16 All elements of the array except the last are required to be between 1 and 9, inclusive.
17 The last element can be 0.

18
19 The first element of the array defines the number of elements in the first group of digits
20 located immediately to the left of the
21 *System.Globalization.NumberFormatInfo.NumberDecimalSeparator*. Each
22 subsequent element refers to the next group of digits located to the left of the previous
23 group. If the last element of the array is not zero, any remaining digits are grouped
24 based on the last element of the array. If the last element is zero, the remaining digits
25 are not grouped.

26
27 The culture-invariant value for this property is an array with a single element containing
28 the value 3.

29 Exceptions

Exception	Condition
System.ArgumentNullException	The array specified for a set operation is a null reference.
System.ArgumentOutOfRangeException	One of the elements in the array specified for a set operation is not between 0 and 9.

System.InvalidOperationException

The current instance is read-only and a set operation was attempted.

1

2 Example

3 The following example demonstrates the effects of different
4 System.Globalization.NumberFormatInfo.NumberGroupSizes property values.

5

6 [C#]

```
7 using System;  
8 using System.Globalization;  
9 class Test {  
10     public static void Main() {  
11         NumberFormatInfo nfi = new NumberFormatInfo();  
12  
13         decimal data = 9999999994444333221.00m;  
14         nfi.NumberGroupSizes = new int[] {1,2,3,4,0};  
15         Console.WriteLine("{0}",data.ToString("N",nfi));  
16  
17         data = 123456789123456.78m;  
18         nfi.NumberGroupSizes = new int[] {3};  
19         Console.WriteLine("{0}",data.ToString("N",nfi));  
20  
21         nfi.NumberGroupSizes = new int[] {3,0};  
22         Console.WriteLine("{0}",data.ToString("N",nfi));  
23     }  
24 }  
25
```

26 The output is

27

28 999999999,4444,333,22,1.00

29

30

31 123,456,789,123,456.78

32

33

34 123456789123,456.78

35

36

1 NumberFormatInfo.NumberNegativePattern

2 Property

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

9 Summary

10 Gets or sets the format of negative values.

11 Property Value

12 A System.Int32 between 0 and 4 inclusive that specifies the format of negative values.

13 Description

14 The following table describes the valid values for this property. "-" is used as the value
15 for System.Globalization.NumberFormatInfo.NegativeSign, and 999 represents any
16 numeric value.

Value	Pattern
0	(999)
1	-999
2	- 999
3	999-
4	999 -

17
18 The culture-invariant value for this property is 1.

19 Exceptions

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

	than 0 or greater than 4.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

Example

The following example demonstrates the effects of different System.Globalization.NumberFormatInfo.NumberNegativePattern property values.

[C#]

```
using System;
using System.Globalization;
class Test {
    public static void Main() {
        NumberFormatInfo nfi = new NumberFormatInfo();
        Double data = -9999999999999.00;
        for (int i = 0; i<=4; i++) {
            nfi.NumberNegativePattern = i;
            Console.WriteLine("pattern # {0}: {1}",i,data.ToString("N",nfi));
        }
    }
}
```

The output is

```
pattern # 0: (9,999,999,999,999.00)
pattern # 1: -9,999,999,999,999.00
pattern # 2: - 9,999,999,999,999.00
pattern # 3: 9,999,999,999,999.00-
pattern # 4: 9,999,999,999,999.00 -
```

1 NumberFormatInfo.PercentDecimalDigits

2 Property

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

9 Summary

10 Gets or sets the number of decimal places in percent values.

11 Property Value

12 A `System.Int32` containing the number of decimal places in percent values.

13 Description

14 The culture-invariant value for this property is 2.

15 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 99.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

16

17

1 NumberFormatInfo.PercentDecimalSeparator 2 Property

```
3 [ILAsm]  
4 .property string PercentDecimalSeparator { public hidebysig specialname  
5 instance string get_PercentDecimalSeparator() public hidebysig specialname  
6 instance void set_PercentDecimalSeparator(string value) }  
7 [C#]  
8 public string PercentDecimalSeparator { get; set; }
```

9 Summary

10 Gets or sets the symbol used as the decimal separator in percent values.

11 Property Value

12 A `System.String` containing the decimal separator used in percent values.

13 Description

14 The culture-invariant value for this property is ".".

15 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

16

17

1 NumberFormatInfo.PercentGroupSeparator 2 Property

```
3 [ILAsm]  
4 .property string PercentGroupSeparator { public hidebysig specialname  
5 instance string get_PercentGroupSeparator() public hidebysig specialname  
6 instance void set_PercentGroupSeparator(string value) }  
7 [C#]  
8 public string PercentGroupSeparator { get; set; }
```

9 Summary

10 Gets or sets the symbol used to separate groups of digits to the left of the decimal point
11 in percent values.

12 Property Value

13 A `System.String` containing the group separator symbol used in percent values.

14 Description

15 The culture-invariant value for this property is ",".

16 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

17

18

1 NumberFormatInfo.PercentGroupSizes

2 Property

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

9 Summary

10 Gets or sets the number of digits in each group to the left of the decimal point in
11 percent values.

12 Property Value

13 A *System.Int32* array containing elements that define the number of digits in each
14 group in percent values.

15 Description

16 All elements of the array except the last are required to be between 1 and 9, inclusive.
17 The last element can be 0.

18
19 The first element of the array defines the number of elements in the first group of digits
20 located immediately to the left of the
21 *System.Globalization.NumberFormatInfo.PercentDecimalSeparator*. Each
22 subsequent element refers to the next group of digits located to the left of the previous
23 group. If the last element of the array is not zero, any remaining digits are grouped
24 based on the last element of the array. If the last element is zero, the remaining digits
25 are not grouped.

26
27 The culture-invariant value for this property is an array with a single element containing
28 the value 3.

29 Exceptions

Exception	Condition
System.ArgumentNullException	The array specified for a set operation is a null reference.
System.ArgumentException	One of the elements in the array specified for a set operation is not between 0 and 9. -or-

	The array contains an element, other than the last element, that is set to 0.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

1

2 Example

3 The following example demonstrates the effects of different
4 `System.Globalization.NumberFormatInfo.PercentGroupSizes` property values.

5
6 [C#]

```
7 using System;
8 using System.Globalization;
9 class Test {
10     public static void Main() {
11         NumberFormatInfo nfi = new NumberFormatInfo();
12
13         decimal data = 9999999994444333221.00m;
14         nfi.PercentGroupSizes = new int[] {1,2,3,4,0};
15         Console.WriteLine("{0}",data.ToString("P",nfi));
16
17         data = 123456789123456.78m;
18         nfi.PercentGroupSizes = new int[] {3};
19         Console.WriteLine("{0}",data.ToString("P",nfi));
20
21         nfi.PercentGroupSizes = new int[] {3,0};
22         Console.WriteLine("{0}",data.ToString("P",nfi));
23     }
24 }
25
```

26 The output is

27
28 99999999944,4433,322,10,0.00 %

29

30

31 12,345,678,912,345,678.00 %

32

33

34 12345678912345,678.00 %

35

36

1 NumberFormatInfo.PercentNegativePattern 2 Property

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

9 Summary

10 Gets or sets the format of negative percent values.

11 Property Value

12 A System.Int32 between 0 and 2 inclusive that specifies the format of negative percent
13 values.

14 Description

15 The following table describes the valid values for this property. "%" is used as the value
16 for System.Globalization.NumberFormatInfo.PercentSymbol, "-" is used as the value
17 for System.Globalization.NumberFormatInfo.NegativeSign, and 999 represents any
18 numeric value.

Value	Pattern
0	-999 %
1	-999%
2	-%999

19 The culture-invariant value for this property is 0.
20

21 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 2.

System.InvalidOperationException

The current instance is read-only and a set operation was attempted.

1

2 Example

3 The following example demonstrates the effects of different
4 System.Globalization.NumberFormatInfo.PercentNegativePattern property values.

5

6 [C#]

```
7 using System;  
8 using System.Globalization;  
9 class Test {  
10     public static void Main() {  
11         NumberFormatInfo nfi = new NumberFormatInfo();  
12         decimal data = -.9900m;  
13         for (int i = 0; i<=2; i++) {  
14             nfi.PercentNegativePattern = i;  
15             Console.WriteLine("pattern # {0}: {1}", i, data.ToString("P", nfi));  
16         }  
17     }  
18 }  
19
```

20 The output is

21

22 pattern # 0: -99.00 %

23

24

25 pattern # 1: -99.00%

26

27

28 pattern # 2: -%99.00

29

30

1 NumberFormatInfo.PercentPositivePattern 2 Property

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

9 Summary

10 Gets or sets the format of positive percent values.

11 Property Value

12 A System.Int32 between 0 and 2 inclusive that specifies the format of positive percent
13 values.

14 Description

15 The following table describes the valid values for this property. "%" is used as the value
16 for System.Globalization.NumberFormatInfo.PercentSymbol, and 999 represents a
17 numeric value.

Value	Pattern
0	999 %
1	999%
2	%999

18
19 The culture-invariant value for this property is 0.

20 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is less than 0 or greater than 2.
System.InvalidOperationException	The current instance is read-only and a set

operation was attempted.

1

2 **Example**

3 The following example demonstrates the effects of different
4 `System.Globalization.NumberFormatInfo.PercentPositivePattern` property values.

5

6 [C#]

```
7 using System;  
8 using System.Globalization;  
9 class Test {  
10     public static void Main() {  
11         NumberFormatInfo nfi = new NumberFormatInfo();  
12         decimal data = .9900m;  
13         for (int i = 0; i<=2; i++) {  
14             nfi.PercentPositivePattern = i;  
15             Console.WriteLine("pattern # {0}: {1}",i,data.ToString("P",nfi));  
16         }  
17     }  
18 }
```

19

20 The output is

21

22 pattern # 0: 99.00 %

23

24

25 pattern # 1: 99.00%

26

27

28 pattern # 2: %99.00

29

30

1 NumberFormatInfo.PercentSymbol Property

```
2 [ILAsm]  
3 .property string PercentSymbol { public hidebysig specialname instance  
4 string get_PercentSymbol() public hidebysig specialname instance void  
5 set_PercentSymbol(string value) }  
  
6 [C#]  
7 public string PercentSymbol { get; set; }
```

8 Summary

9 Gets or sets the symbol that represents percentage values.

10 Property Value

11 A System.String containing the percent symbol.

12 Description

13 The culture-invariant value for this property is "%".

14 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

15

16

1 NumberFormatInfo.PerMilleSymbol Property

```
2 [ILAsm]
3 .property string PerMilleSymbol { public hidebysig specialname instance
4 string get_PerMilleSymbol() public hidebysig specialname instance void
5 set_PerMilleSymbol(string value) }
6
7 [C#]
8 public string PerMilleSymbol { get; set; }
```

8 Summary

9 Gets or sets the per mille symbol.

10 Property Value

11 A System.String containing the per mille symbol.

12 Description

13 The culture-invariant value for this property is "%/100".

14 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

15

16

1 NumberFormatInfo.PositiveInfinitySymbol 2 Property

```
3 [ILAsm]  
4 .property string PositiveInfinitySymbol { public hidebysig specialname  
5 instance string get_PositiveInfinitySymbol() public hidebysig specialname  
6 instance void set_PositiveInfinitySymbol(string value) }  
  
7 [C#]  
8 public string PositiveInfinitySymbol { get; set; }
```

9 Summary

10 Gets or sets the symbol that represents positive infinity.

11 Property Value

12 A `System.String` containing the symbol for positive infinity.

13 Description

14 The culture-invariant value for this property is "Infinity".

15
16 [*Note:* For more information on positive infinity, see `System.Double` or `System.Single`.]
17
18

19 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference.
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

20

21

1 NumberFormatInfo.PositiveSign Property

```
2 [ILAsm]  
3 .property string PositiveSign { public hidebysig specialname instance  
4 string get_PositiveSign() public hidebysig specialname instance void  
5 set_PositiveSign(string value) }  
6 [C#]  
7 public string PositiveSign { get; set; }
```

8 Summary

9 Gets or sets the symbol used to represent positive values.

10 Property Value

11 A `System.String` containing the symbol that indicates the value is positive.

12 Description

13 The culture-invariant value for this property is "+".

14 Exceptions

Exception	Condition
System.ArgumentNullException	The value specified for a set operation is a null reference
System.InvalidOperationException	The current instance is read-only and a set operation was attempted.

15

16