

.NET String.Format() to add commas in thousands place for a number

[Ask Question](#)

I want to add a comma in the thousands place for a number.

String.Format() ?

734

c#

.net



123

edited Dec 18 '15 at 11:54



Tamir Vered

8,804 4 38 51

asked Sep 19 '08 at 21:26



Seibar

36k 34 76 98

21 Answers



C# 7.1 (perhaps earlier?) makes this as easy and nice-looking as it should be, with string interpolation:

1

```
var jackpot = 1000000;  
var niceNumberString = $"Jackpot is {jackpot:n}";
```

[Home](#)[PUBLIC](#)[Stack Overflow](#)[Tags](#)[Users](#)[Jobs](#)**Teams**

Q&A for work

[Learn More](#)

1053



```
var niceMoneyString = $"Jackpot is {jackpot:C}";
```

answered Feb 9 at 13:21

[Mark Zhukovsky](#)

143 1 11

```
String.Format("{0:n}", 1234); // Output: 1,234.00
String.Format("{0:n0}", 9876); // No digits after the decimal point
```

edited Nov 7 '18 at 0:30

[AustinWBryan](#)

2,380 2 14 32

answered Sep 19 '08 at 21:29

[Seibar](#)

36k 34 76 98

35 `String.Format("{0:##,###0}", 1234);` also works with no decimal places. – [Greg Bray](#) Nov 5 '10 at 3:50

7 How can I replicate the "N" specifier but with as many decimal digits as are present in the number? – [Stephen Drew](#) Nov 23 '12 at 10:33

54 @Justin: According to msdn.microsoft.com/en-us/library/0c899ak8.aspx, the ',' (and the '.') are replaced with the correct localized characters. – [Roger Lipscombe](#) Mar 4 '13 at 9:29

@RogerLipscombe I hadn't realised that – [Justin](#) Mar 4 '13 at 9:58

5 @VVVV - then you are probably passing in a *string* instead of a *number*. If you have a string, you need to first convert to float or double. Try `string.Format("{0:n0}", Double.Parse(yourValue));` – [ToolmakerSteve](#) Oct 26 '17 at 2:51



`String.Format("0,###.###");` also works with **decimal** places

4



answered Feb 13 '18 at 12:20



[Abolfazl Rastgou](#)

263 3 7



just simple as this:

21

```
float num = 23658; // for example
num = num.ToString("N0"); // Returns 23,658
```



more info is in [Here](#)

answered Aug 24 '17 at 5:08



[a_m_dev](#)

1,315 3 13 33



The following example displays several values that are formatted by using custom format strings that include zero placeholders.

20

```
String.Format("{0:N1}", 29255.0);
```



Or

```
29255.0.ToString("N1")
```

result "29,255.0"

```
String.Format("{0:N2}", 29255.0);
```

Or

```
29255.0.ToString("N2")
```

result "29,255.00"

edited Mar 7 '17 at 12:11

answered Feb 23 '17 at 15:00



[Yitzhak Weinberg](#)

836 8 19



Simpler, using string interpolation instead of String.Format

7

```
${12456:n0}"; // 12,456  
${12456:n2}"; // 12,456.00
```



or using yourVariable

```
double yourVariable = 12456.0;  
${yourVariable:n0}";  
${yourVariable:n2}";
```

answered Nov 2 '16 at 20:22



[brakeroo](#)

679 6 17

I found this to be the simplest way:

`myInteger.ToString("N0")`

322

edited May 18 '16 at 13:48



[Jon Schneider](#)

13.7k 9 91 128

answered Oct 19 '10 at 5:04



[alchemical](#)

7,744 21 74 108

1 You can also use it with `string.Format`, as in `string.Format("Here is some number with commas, and no decimals, {0:N0}", 123456789);` – [Dan Morphis](#) Oct 21 '11 at 2:52

14 shouldn't it be `myInteger.ToString("N0")` ... `string.ToString` i don't think would work. – [Taylor Brown](#) May 14 '14 at 22:05

1 I know it's been 5 years now, but thanks! It works for numbers > 4 characters, and < 4 characters. – [AskYous](#) May 18 '15 at 16:21

@AskYous - Well, it also works for 4 characters.. Might as well say it works for any length. – [Broots Waymb](#) Nov 17 '17 at 15:27

This is working but it's changing on regional settings – [saulyasar](#) Mar 27 '18 at 14:38

28

The most voted answer was great and has been helpful for about 7 years. With the introduction of C# 6.0 and specifically the String Interpolation there's a neater and, IMO safer, way to do what has been asked to add commas in thousands place for a number :

```
var i = 5222000;
var s = $"{i:n} is the number"; // results to > 5,222,000.00 is the number
s = $"{i:n0} has no decimal"; // results to > 5,222,000 has no decimal
```

Where the variable `i` is put in place of the placeholder (i.e. `{0}`).
 So there's no need to remember which object goes to which position.
 The formatting (i.e. `:n`) hasn't changed. For a complete feature of what's new, you may [go to this page](#).

answered Aug 12 '15 at 2:51



von v.

13.5k 3 47 72

This is the best format. Works in all of those cases:

37

```
String.Format( "{0:##,###.##}", 0 ); // 0
String.Format( "{0:##,###.##}", 0.5 ); // 0.5 - some of the formats are
String.Format( "{0:##,###.##}", 12314 ); // 12,314
String.Format( "{0:##,###.##}", 12314.23123 ); // 12,314.23
String.Format( "{0:##,###.##}", 12314.2 ); // 12,314.2
String.Format( "{0:##,###.##}", 1231412314.2 ); // 1,231,412,314.2
```

answered Jul 9 '15 at 23:43



Dennis

465 4 8

1 What if I want dots as thousand separator and comma as decimal delimiter? – [FrenkyB](#) Sep 12 '17 at 14:16

@FrenkyB "{0:##,###.##}" should work – [thefoxrocks](#) Oct 12 '17 at 17:54

upvoted because it doesn't display 12,314.0 (like the `n1` format) but 12,314 :) – [NDUF](#) Apr 26 '18 at 8:59



If you want culture specific, you might want to try this:

92

```
(19950000.0).ToString("N",new CultureInfo("en-US")) =  
19,950,000.00
```



```
(19950000.0).ToString("N",new CultureInfo("is-IS")) =  
19.950.000,00
```

Note: Some cultures use , to mean decimal rather than . so be careful.

edited Dec 23 '14 at 11:18



[CodesInChaos](#)

89.9k 14 173 229

answered Oct 26 '10 at 9:22



[prabir](#)

6,364 2 24 43



You can use a function such as this to format numbers and optionally pass in the desired decimal places. If decimal places are not specified it will use two decimal places.

4



```
public static string formatNumber(decimal valueIn=0, int decimalPl  
{  
    return string.Format("{0:n" + decimalPlaces.ToString() + "}",  
}
```

I use decimal but you can change the type to any other or use an anonymous object. You could also add error checking for negative decimal place values.

answered Dec 23 '14 at 11:13

[dunwan](#)



1,202 1 11 8

Standard formats, with their related outputs,

69

```

Console.WriteLine("Standard Numeric Format Specifiers");
String s = String.Format("(C) Currency: . . . . . {0:C}\n" +
    "(D) Decimal: . . . . . {0:D}\n" +
    "(E) Scientific: . . . . . {1:E}\n" +
    "(F) Fixed point: . . . . . {1:F}\n" +
    "(G) General: . . . . . {0:G}\n" +
    "    (default): . . . . . {0} (default = 'G'" +
    "(N) Number: . . . . . {0:N}\n" +
    "(P) Percent: . . . . . {1:P}\n" +
    "(R) Round-trip: . . . . . {1:R}\n" +
    "(X) Hexadecimal: . . . . . {0:X}\n",
    - 1234, -1234.565F);

Console.WriteLine(s);

```

Example output (en-us culture):

```

(C) Currency: . . . . . ($1,234.00)
(D) Decimal: . . . . . -1234
(E) Scientific: . . . . . -1.234565E+003
(F) Fixed point: . . . . . -1234.57
(G) General: . . . . . -1234
    (default): . . . . . -1234 (default = 'G')
(N) Number: . . . . . -1,234.00
(P) Percent: . . . . . -123,456.50 %
(R) Round-trip: . . . . . -1234.565
(X) Hexadecimal: . . . . . FFFFFB2E

```

edited Dec 12 '14 at 16:19

answered Mar 1 '13 at 23:44



CoderTao

3,191 1 16 17



- 1 This answer packed a lot of useful information. Learning by example I see now what the 0 and 1 mean in the string format. – [user420667](#) Nov 2 '16 at 16:58



`String.Format("{0:#,###,###.##}", MyNumber)`

32

That will give you commas at the relevant points.



edited May 11 '14 at 18:24



[Dan](#)

3,127 15 65 118


answered Sep 19 '08 at 21:29



[Stephen Wrighton](#)

27.6k 5 59 84

- 10 The ".n" method is better since it should respect the user's locale. – [Torlack](#) Sep 19 '08 at 21:30
- 11 This is true, but it's not guaranteed to give you commas at the thousand point because it respect the user's locale. – [Stephen Wrighton](#) Sep 19 '08 at 21:35
- 2 right back at you: that is true, but it's not guaranteed to respect the user's locale because it uses commas as thousands separator. (As an example, in Portugal the comma is instead the decimal separator.) – [ANeves](#) May 19 '10 at 17:46
- 1 If you want to enforce values after the . you need to replace the # with a 0. [msdn.microsoft.com/en-us/library/0c899ak8\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/0c899ak8(v=vs.110).aspx): Zero replaces the zero with the corresponding digit if one is present; otherwise, zero appears in the result string whereas the "#" symbol is replaced with the corresponding digit if one is present; otherwise, no digit appears in the result string. – [cbillowes](#) Feb 6 '14 at 9:29

this method worked ok for my requirement, the msdn page about the `Int32.ToString` method that would be a primary place it would be used msdn.microsoft.com/en-us/library/8wch342y.aspx isn't very helpful for this particular application either – [stackuser83](#) Mar 25 '14 at 0:24 



Below is a good solution in Java though!

-2



```
NumberFormat fmt = NumberFormat.getCurrencyInstance();
System.out.println(fmt.format(n));
```

or for a more robust way you may want to get the locale of a particular place, then use as below:

```
int n=9999999;
Locale locale = new Locale("en", "US");
NumberFormat fmt = NumberFormat.getCurrencyInstance(locale);
System.out.println(fmt.format(n));
```

US Locale OUTPUT: \$9,999,999.00

German Locale output

```
Locale locale = new Locale("de", "DE");
```

OUTPUT: 9.999.999,00 €

Indian Locale output

```
Locale locale = new Locale("de", "DE");
```

OUTPUT: Rs.9,999,999.00

Estonian Locale output

```
Locale locale = new Locale("et", "EE");
```

OUTPUT: 9 999 999 €

As you can see in different outputs you don't have to worry about the separator being a **comma** or **dot** or even **space** you can get the number formatted according to the i18n standards

edited Dec 22 '13 at 6:06

answered Oct 5 '13 at 8:40



Anirudh

1,358 2 23 51

1 where did you use the variable locale ?? – Smith Dec 21 '13 at 20:29

well correctly spotted, is it infact to be used while getting the currency instance like: NumberFormat fmt = NumberFormat.getCurrencyInstance(locale); corrected the code, thanks! – Anirudh Dec 22 '13 at 6:05

3 Have you checked the tags? The question is about C# not Java! – Salar Sep 21 '15 at 4:50



-3



If you want to show it in DataGridView , you should change its type , because default is String and since you change it to decimal it considers as Number with floating point

```
Dim dt As DataTable = New DataTable
dt.Columns.Add("col1", GetType(Decimal))
dt.Rows.Add(1)
dt.Rows.Add(10)
dt.Rows.Add(2)
```

```
DataGridView1.DataSource = dt
```

answered Nov 28 '13 at 3:36



Ali

146 3 8 17

21

If you wish to force a "," separator regardless of culture (for example in a trace or log message), the following code will work and has the added benefit of telling the next guy who stumbles across it exactly what you are doing.

```
int integerValue = 19400320;
string formatted = string.Format(CultureInfo.InvariantCulture, "{0:N1}
```

sets formatted to "19,400,320"

answered Mar 27 '13 at 19:55



Ravi Desai

386 3 9

9

For example `String.Format("{0:0,0}", 1);` returns 01, for me is not valid

This works for me

```
19950000.ToString("#,#", CultureInfo.InvariantCulture));
```

output 19,950,000

answered Oct 23 '12 at 21:19



cmujica

958 14 24



-2

The method I used to not worry anymore about cultures and potential formatting issues is that I formatted it as currency and took out the currency symbol afterwards.



```
if (decimal.TryParse(tblCell, out result))  
  
    {  
        formattedValue = result.ToString("C").Substring(1);  
    }
```

answered Jun 13 '12 at 9:55



8 John Volante

21

- 6 This code is not culture independent - it will use whatever default culture is set on the machine running the code. This could create undesired output where that culture places their currency symbols at the end of the number rather than the start (e.g. fr-FR), or uses more than one character to denote the currency (e.g. da-DK), or does not separate thousands using commas (e.g. most of mainland Europe). – [raveturned](#) Jun 13 '12 at 21:08



11

```
int num = 98765432;  
Console.WriteLine(string.Format("{0:#,#}", num));
```



answered May 20 '10 at 15:36



p.campbell



68.7k 56 221 300

- 3 Or `Console.WriteLine("{0:#,#}",num);` if you just want to print it. But `string.Format(...)` is more useful I guess. – [Indy9000](#) Aug 25 '11 at 14:42



136



```
int number = 1000000000;
string whatYouWant = number.ToString("#,##0");
//You get: 1,000,000,000
```

edited May 19 '10 at 17:26


[p.campbell](#)

68.7k 56 221 300

answered Nov 17 '08 at 15:15

Ope Adeoye

- 26 This solution is not good from an internationalisation point of view - other cultures use characters other than , as a thousands separator, for example a space or even . – [Justin](#) Jan 31 '12 at 17:04
- 5 Works thanks + 1. Have extended so shows up to 2 d.p.
`number.ToString("#,##0.##")` – [Crab Bucket](#) Mar 30 '12 at 10:38
- 7 @MacSigler It's actually not true, see Roger Lipscombe's comment on the answer above: `String.Format` will apply localization automatically. – [Dan Bechard](#) Oct 17 '13 at 19:34
- 10 @MacSigler That's the thing though, this code **does not a/ways** print out a comma. Only when the culture is one that expects commas (e.g. en-US or invariant). If the culture is one that expects another separator (e.g. .), .NET will automatically replace the comma with that separator. Again, I urge you to read the link posted by Roger if you still do not understand why this is. – [Dan Bechard](#) Oct 18 '13 at 18:33
- 4 @MacSigler Justin's comment is still correct in that if you don't explicitly force the culture to en-US it will inherit culture settings from the local

machine. My understanding is that compiling the code above, then running it on two machines with different cultures (with different number separators) would produce different results. If you want it to *always* produce a comma, you need to explicitly set a culture that uses the comma (e.g. invariant). – [Dan Bechard](#) Oct 22 '13 at 14:12 



8



Note that the value that you're formatting should be numeric. It doesn't look like it will take a string representation of a number and format is with commas.

answered Oct 12 '08 at 0:35

[gjc](#)

protected by [Travis J](#) Jul 10 '13 at 21:34

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?