C# Format Specifiers—Numeric Format Specifiers

Jump to: navigation, search

Visual C# Tutorials

Visual C# .NET Tutorials

C# Format Specifiers

- **Numeric Format Specifiers**
- Picture Format Specifiers
- DateTime Format Specifiers

Numeric Format Specifiers

Table B-1 lists the numeric format specifiers supported by the Format method on the predefined numeric types.

Table B-1. Numeric Format Specifiers

Specifier	String result	Datatype
C[n]	$XX,XX.XX	Currency
	($XX,XXX.XX)	
D[n]	[-]XXXXXXX	Decimal
E[n] or e[n]	[-]X.XXXXXXE+xxx	Exponent
	[-]X.XXXXXXe+xxx	
	[-]X.XXXXXXE-xxx	
	[-]X.XXXXXXe-xxx	
F[n]	[-]XXXXXXX.XX	Fixed point
G[n]	General or scientific	General
N[n]	[-]XX,XXX.XX	Number
X[n] or x[n]	Hex representation	Hex

This example uses numeric format specifiers without precision specifiers:

```
using System;
class TestDefaultFormats {
  static void Main( ) {
    int i = 654321;
    Console.WriteLine("{0:C}", i); // $654,321.00
    Console.WriteLine("{0:D}", i); // 654321
    Console.WriteLine("{0:E}", i); // 6.543210E+005
    Console.WriteLine("{0:F}", i); // 654321.00
    Console.WriteLine("{0:G}", i); // 654321
    Console.WriteLine("{0:N}", i); // 654,321.00
    Console.WriteLine("{0:X}", i); // 9FBF1
    Console.WriteLine("{0:x}", i); // 9fbf1
  }
}
```

This example uses numeric format specifiers with precision specifiers on a variety of int values:

```
using System;
class TestIntegerFormats {
  static void Main( ) {
    int i = 123;
    Console.WriteLine("{0:C6}", i); // $123.000000
    Console.WriteLine("{0:D6}", i); // 000123
    Console.WriteLine("{0:E6}", i); // 1.230000E+002
    Console.WriteLine("{0:G6}", i); // 123
    Console.WriteLine("{0:N6}", i); // 123.000000
    Console.WriteLine("{0:X6}", i); // 00007B
    i = -123;
    Console.WriteLine("{0:C6}", i); // ($123.000000)
    Console.WriteLine("{0:D6}", i); // -000123
    Console.WriteLine("{0:E6}", i); // -1.230000E+002
    Console.WriteLine("{0:G6}", i); // -123
    Console.WriteLine("{0:N6}", i); // -123.000000
    Console.WriteLine("{0:X6}", i); // FFFF85
    i = 0;
    Console.WriteLine("{0:C6}", i); // $0.000000
    Console.WriteLine("{0:D6}", i); // 000000
    Console.WriteLine("{0:E6}", i); // 0.000000E+000
    Console.WriteLine("{0:G6}", i); // 0
    Console.WriteLine("{0:N6}", i); // 0.000000
    Console.WriteLine("{0:X6}", i); // 000000
  }
}
```

This example uses numeric format specifiers with precision specifiers on a variety of double values:

```
using System;
class TestDoubleFormats {
  static void Main( ) {
    double d = 1.23;
    Console.WriteLine("{0:C6}", d); // $1.230000
    Console.WriteLine("{0:E6}", d); // 1.230000E+000
    Console.WriteLine("{0:G6}", d); // 1.23
    Console.WriteLine("{0:N6}", d); // 1.230000
    d = -1.23;
    Console.WriteLine("{0:C6}", d); // ($1.230000)
    Console.WriteLine("{0:E6}", d); // -1.230000E+000
    Console.WriteLine("{0:G6}", d); // -1.23
    Console.WriteLine("{0:N6}", d); // -1.230000
    d = 0;
    Console.WriteLine("{0:C6}", d); // $0.000000
    Console.WriteLine("{0:E6}", d); // 0.000000E+000
    Console.WriteLine("{0:G6}", d); // 0
    Console.WriteLine("{0:N6}", d); // 0.000000
```