Composite Formatting

Podemos usar composite formatting en los siguientes metodos

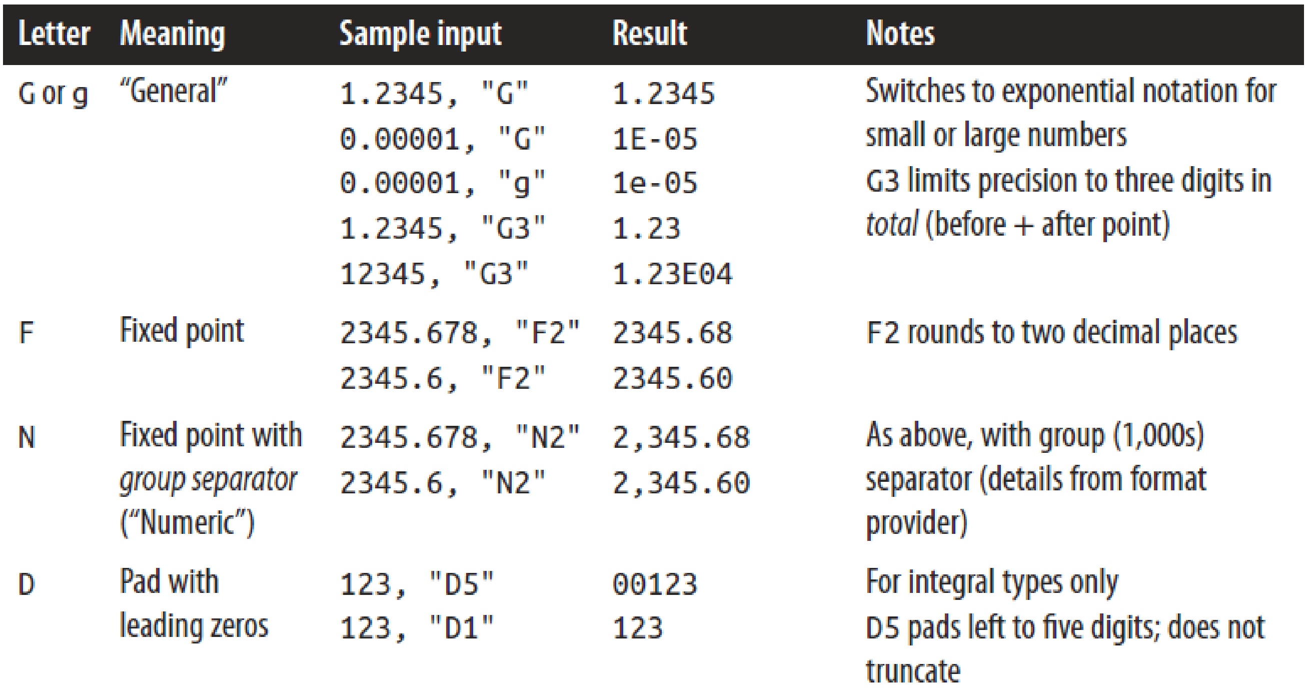
- Console.Writeline
- String.Format
- StringBuilder.AppendFormat
- TextWriter.WriteLine
- Debug.WriteLine
- Trace (varios métodos)

Cada item del formato esta conformado como sigue:

```
{index[,alignment][:formatString]}
```

Los unicos elementos obligatorios son las llaves y el index, el resto es opcional

Formatos Numericos Standard



Letter	Meaning	Sample input	Result	Notes

G or g	"General"	`1.2345, "G"`
		`0.00001, "G"`
		`0.00001, "g"`
		`1.2345, "G3"`
		`12345, "G3"`
			`1.2345`
			`1E-05`
			`1e-05`
			`1.23`
			`1.23E04`
				Switches to exponential notation for small or large numbers
				G3 limits precision to three digits in *total* (before + after point)

F	Fixed point	`2345.678, "F2"`
		`2345.6, "F2"`
			`2345.68`
			`2345.60`
				F2 rounds to two decimal places

N	Fixed point with *group separator* ("Numeric")	`2345.678, "N2"`
		`2345.6, "N2"`
			`2,345.68`
			`2,345.60`
				As above, with group (1,000s) separator (details from format provider)

D	Pad with leading zeros	`123, "D5"`
		`123, "D1"`
			`00123`
			`123`
				For integral types only
				D5 pads left to five digits; does not truncate

Letter	Meaning	Sample input	Result	Notes

E or e	Force exponential notation	`56789, "E"`
		`56789, "e"`
		`56789, "E2"`
			`5.678900E+004`
			`5.678900e+004`
			`5.68E+004`
				Six-digit default precision

C	Currency	`1.2, "C"`
		`1.2, "C4"`
			`$1.20`
			`$1.2000`
				C with no digit uses default number of D.P. from format provider

P	Percent	`.503, "P"`
		`.503, "P0"`
			`50.30 %`
			`50 %`
				Uses symbol and layout from format provider
				Decimal places can optionally be overridden

X or x	Hexadecimal	`47, "X"`
		`47, "x"`
		`47, "X4"`
			`2F`
			`2f`
			`002F`
				X for uppercase hex digits; x for lowercase hex digits
				Integrals only

R	Round-trip	`1f / 3f, "R"`	`0.333333343`	For the `float` and `double` types, R or `G17` squeeze out all digits to ensure exact round-tripping

Formatos Numericos Custom

Specifier	Meaning	Sample input	Result	Notes

#	Digit placeholder	`12.345, ".##"`
		`12.345, ".####"`
			`12.35`
			`12.345`
				Limits digits after D.P.

0	Zero placeholder	`12.345, ".00"`
		`12.345, ".0000"`
		`99, "000.00"`
			`12.35`
			`12.3450`
			`099.00`
				As above, but also pads with zeros before and after D.P.

.	Decimal point			Indicates D.P. Actual symbol comes from `NumberFormatInfo`

,	Group separator	`1234, "#,###,###"`
		`1234, "0,000,000"`
			`1,234`
			`0,001,234`
				Symbol comes from `NumberFormatInfo`

, (as above)	Multiplier	`1000000, "#,"`
		`1000000, "#,,`
			`1000`
			`1`
				If comma is at end or before D.P., it acts as a multiplier—dividing result by 1,000, 1,000,000, etc.

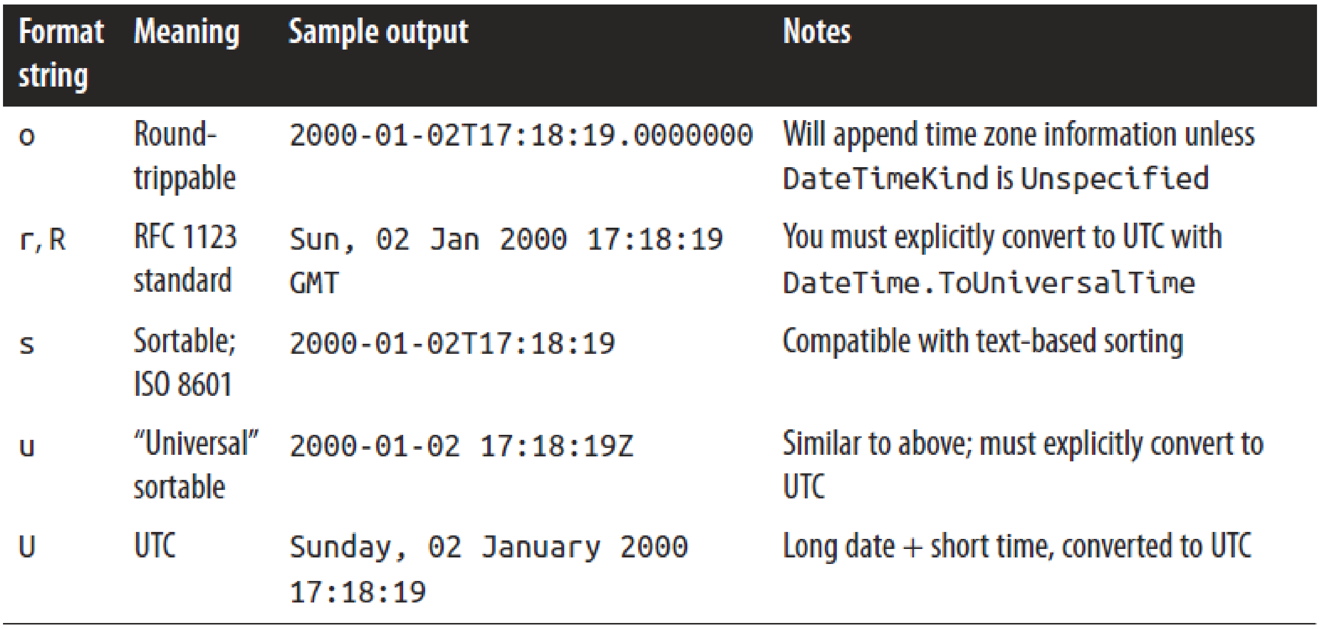
%	Percent notation	`0.6, "00%"`	`60%`	First multiplies by 100 and then substitutes percent symbol obtained from `NumberFormatInfo`

E0, e0, E+0, e+0 E-0, e-0	Exponent notation	1234, "0E0" 1234, "0E+0" 1234, "0.00E00" 1234, "0.00e00"	1E3 1E+3 1.23E03 1.23e03	
\	Literal character quote	50, @"\#0"	#50	Use in conjunction with an @ prefix on the string—or use \\
'xx''xx'	Literal string quote	50, "0 '...'"	50 ...	
;	Section separator	15, "#;(#);zero"	15	(If positive)
		-5, "#;(#);zero"	(5)	(If negative)
		0, "#;(#);zero"	zero	(If zero)
Any other char	Literal	35.2, "$0 . 00c"	$35 . 20c	

Formatos de Fecha y Hora Standard (dependientes de la cultura)

Format string	Meaning	Sample output
d	Short date	01/02/2000
D	Long date	Sunday, 02 January 2000
t	Short time	17:18
T	Long time	17:18:19
f	Long date + short time	Sunday, 02 January 2000 17:18
F	Long date + long time	Sunday, 02 January 2000 17:18:19
g	Short date + short time	01/02/2000 17:18
G (default)	Short date + long time	01/02/2000 17:18:19
m, M	Month and day	02 January
y, Y	Year and month	January 2000

Formatos de Fecha y Hora Standard (independientes de la cultura)



Format string	Meaning	Sample output	Notes
o	Round-trippable	2000-01-02T17:18:19.0000000	Will append time zone information unless DateTimeKind is Unspecified
r, R	RFC 1123 standard	Sun, 02 Jan 2000 17:18:19 GMT	You must explicitly convert to UTC with DateTime.ToUniversalTime
s	Sortable; ISO 8601	2000-01-02T17:18:19	Compatible with text-based sorting
u	"Universal" sortable	2000-01-02 17:18:19Z	Similar to above; must explicitly convert to UTC
U	UTC	Sunday, 02 January 2000 17:18:19	Long date + short time, converted to UTC

Formatos de Fecha y Hora Custom

Format specifier	Description	Examples
"d"	The day of the month, from 1 through 31.	2009-06-01T13:45:30 -> 1 2009-06-15T13:45:30 -> 15
"dd"	The day of the month, from 01 through 31.	2009-06-01T13:45:30 -> 01 2009-06-15T13:45:30 -> 15
"ddd"	The abbreviated name of the day of the week.	2009-06-15T13:45:30 -> Mon (en-US) 2009-06-15T13:45:30 -> Пн (ru-RU) 2009-06-15T13:45:30 -> lun. (fr-FR)
"dddd"	The full name of the day of the week.	2009-06-15T13:45:30 -> Monday (en-US) 2009-06-15T13:45:30 -> понедельник (ru-RU) 2009-06-15T13:45:30 -> lundi (fr-FR)
"f"	The tenths of a second in a date and time value.	2009-06-15T13:45:30.6170000 -> 6 2009-06-15T13:45:30.05 -> 0
"ff"	The hundredths of a second in a date and time value.	2009-06-15T13:45:30.6170000 -> 61 2009-06-15T13:45:30.0050000 -> 00
"fff"	The milliseconds in a date and time value.	6/15/2009 13:45:30.617 -> 617 6/15/2009 13:45:30.0005 -> 000
"ffff"	The ten thousandths of a second in a date and time value.	2009-06-15T13:45:30.6175000 -> 6175 2009-06-15T13:45:30.0000500 -> 0000
"fffff"	The hundred thousandths of a second in a date and time value.	2009-06-15T13:45:30.6175400 -> 61754 6/15/2009 13:45:30.000005 -> 00000
"ffffff"	The millionths of a second in a date and time value.	2009-06-15T13:45:30.6175420 -> 617542

Format specifier	Description	Examples
		2009-06-15T13:45:30.0000005 -> 000000
"fffffff"	The ten millionths of a second in a date and time value.	2009-06-15T13:45:30.6175425 -> 6175425 2009-06-15T13:45:30.0001150 -> 0001150
"F"	If non-zero, the tenths of a second in a date and time value.	2009-06-15T13:45:30.6170000 -> 6 2009-06-15T13:45:30.0500000 -> (no output)
"FF"	If non-zero, the hundredths of a second in a date and time value.	2009-06-15T13:45:30.6170000 -> 61 2009-06-15T13:45:30.0050000 -> (no output)
"FFF"	If non-zero, the milliseconds in a date and time value.	2009-06-15T13:45:30.6170000 -> 617 2009-06-15T13:45:30.0005000 -> (no output)
"FFFF"	If non-zero, the ten thousandths of a second in a date and time value.	2009-06-15T13:45:30.5275000 -> 5275 2009-06-15T13:45:30.0000500 -> (no output)
"FFFFF"	If non-zero, the hundred thousandths of a second in a date and time value.	2009-06-15T13:45:30.6175400 -> 61754 2009-06-15T13:45:30.0000050 -> (no output)
"FFFFFF"	If non-zero, the millionths of a second in a date and time value.	2009-06-15T13:45:30.6175420 -> 617542 2009-06-15T13:45:30.0000005 -> (no output)
"FFFFFFF"	If non-zero, the ten millionths of a second in a date and time value.	2009-06-15T13:45:30.6175425 -> 6175425 2009-06-15T13:45:30.0001150 -> 000115
"g", "gg"	The period or era.	2009-06-15T13:45:30.6170000 -> A.D.
"h"	The hour, using a 12-hour clock from 1 to 12.	2009-06-15T01:45:30 -> 1 2009-06-15T13:45:30 -> 1
"hh"	The hour, using a 12-hour clock from 01 to 12.	2009-06-15T01:45:30 -> 01 2009-06-15T13:45:30 -> 01
"H"	The hour, using a 24-hour clock from 0 to 23.	2009-06-15T01:45:30 -> 1 2009-06-15T13:45:30 -> 13
"HH"	The hour, using a 24-hour clock from 00 to 23.	2009-06-15T01:45:30 -> 01 2009-06-15T13:45:30 -> 13
"K"	Time zone information.	With DateTime values: 2009-06-15T13:45:30, Kind Unspecified -> 2009-06-15T13:45:30, Kind Utc -> Z 2009-06-15T13:45:30, Kind Local -> -07:00 (depends on local computer settings) With DateTimeOffset values: 2009-06-15T01:45:30-07:00 --> -07:00 2009-06-15T08:45:30+00:00 --> +00:00
"m"	The minute, from 0 through 59.	2009-06-15T01:09:30 -> 9 2009-06-15T13:29:30 -> 29
"mm"	The minute, from 00 through 59.	2009-06-15T01:09:30 -> 09

Format specifier	Description	Examples
		2009-06-15T01:45:30 -> 45
"M"	The month, from 1 through 12.	2009-06-15T13:45:30 -> 6
"MM"	The month, from 01 through 12.	2009-06-15T13:45:30 -> 06
"MMM"	The abbreviated name of the month.	2009-06-15T13:45:30 -> Jun (en-US) 2009-06-15T13:45:30 -> juin (fr-FR) 2009-06-15T13:45:30 -> Jun (zu-ZA)
"MMMM"	The full name of the month.	2009-06-15T13:45:30 -> June (en-US) 2009-06-15T13:45:30 -> juni (da-DK) 2009-06-15T13:45:30 -> uJuni (zu-ZA)
"s"	The second, from 0 through 59.	2009-06-15T13:45:09 -> 9
"ss"	The second, from 00 through 59.	2009-06-15T13:45:09 -> 09
"t"	The first character of the AM/PM designator.	2009-06-15T13:45:30 -> P (en-US) 2009-06-15T13:45:30 -> 午 (ja-JP) 2009-06-15T13:45:30 -> (fr-FR)
"tt"	The AM/PM designator.	2009-06-15T13:45:30 -> PM (en-US) 2009-06-15T13:45:30 -> 午後 (ja-JP) 2009-06-15T13:45:30 -> (fr-FR)
"y"	The year, from 0 to 99.	0001-01-01T00:00:00 -> 1 0900-01-01T00:00:00 -> 0 1900-01-01T00:00:00 -> 0 2009-06-15T13:45:30 -> 9 2019-06-15T13:45:30 -> 19
"yy"	The year, from 00 to 99.	0001-01-01T00:00:00 -> 01 0900-01-01T00:00:00 -> 00 1900-01-01T00:00:00 -> 00 2019-06-15T13:45:30 -> 19
"yyy"	The year, with a minimum of three digits.	0001-01-01T00:00:00 -> 001 0900-01-01T00:00:00 -> 900 1900-01-01T00:00:00 -> 1900 2009-06-15T13:45:30 -> 2009
"yyyy"	The year as a four-digit number.	0001-01-01T00:00:00 -> 0001 0900-01-01T00:00:00 -> 0900 1900-01-01T00:00:00 -> 1900 2009-06-15T13:45:30 -> 2009
"yyyyy"	The year as a five-digit number.	0001-01-01T00:00:00 -> 00001 2009-06-15T13:45:30 -> 02009

Format specifier	Description	Examples
"z"	Hours offset from UTC, with no leading zeros.	2009-06-15T13:45:30-07:00 -> -7
"zz"	Hours offset from UTC, with a leading zero for a single-digit value.	2009-06-15T13:45:30-07:00 -> -07
"zzz"	Hours and minutes offset from UTC.	2009-06-15T13:45:30-07:00 -> -07:00
":"	The time separator.	2009-06-15T13:45:30 -> : (en-US) 2009-06-15T13:45:30 -> . (it-IT) 2009-06-15T13:45:30 -> : (ja-JP)
"/"	The date separator.	2009-06-15T13:45:30 -> / (en-US) 2009-06-15T13:45:30 -> - (ar-DZ) 2009-06-15T13:45:30 -> . (tr-TR)
"*string*" **'*string*'**	Literal string delimiter.	2009-06-15T13:45:30 ("arr:" h:m t) -> arr: 1:45 P 2009-06-15T13:45:30 ('arr:' h:m t) -> arr: 1:45 P
%	Defines the following character as a custom format specifier.	2009-06-15T13:45:30 (%h) -> 1
\	The escape character.	2009-06-15T13:45:30 (h \h) -> 1 h
Any other character	The character is copied to the result string unchanged.	2009-06-15T01:45:30 (arr hh:mm t) -> arr 01:45 A