There are following custom format specifiers `y` (year), `M` (month), `d` (day), `h` (hour 12), `H` (hour 24), `m` (minute), `s` (second), `f` (second fraction), `F` (second fraction, trailing zeroes are trimmed), `t` (P.M or A.M) and `z` (time zone).

Following examples demonstrate how are the format specifiers rewritten to the output.

[C#]

```
// create date time 2008-03-09 16:05:07.123
DateTime dt = new DateTime(2008, 3, 9, 16, 5, 7, 123);

String.Format("{0:y yy yyy yyyy}", dt);  // "8 08 008 2008"   year
String.Format("{0:M MM MMM MMMM}", dt);  // "3 03 Mar March"  month
String.Format("{0:d dd ddd dddd}", dt);  // "9 09 Sun Sunday" day
String.Format("{0:h hh H HH}",     dt);  // "4 04 16 16"      hour 12/24
String.Format("{0:m mm}",          dt);  // "5 05"            minute
String.Format("{0:s ss}",          dt);  // "7 07"            second
String.Format("{0:f ff fff ffff}", dt);  // "1 12 123 1230"   sec.fraction
String.Format("{0:F FF FFF FFFF}", dt);  // "1 12 123 123"    without zeroes
String.Format("{0:t tt}",          dt);  // "P PM"            A.M. or P.M.
String.Format("{0:z zz zzz}",      dt);  // "-6 -06 -06:00"   time zone
```

You can use also **date separator** `/` (slash) and **time sepatator** `:` (colon). These characters will be rewritten to characters defined in the current DateTimeFormatInfo.DateSeparator and DateTimeFormatInfo.TimeSeparator.

[C#]

```
// date separator in german culture is "." (so "/" changes to ".")
String.Format("{0:d/M/yyyy HH:mm:ss}", dt); // "9/3/2008 16:05:07" - english (en-US)
String.Format("{0:d/M/yyyy HH:mm:ss}", dt); // "9.3.2008 16:05:07" - german (de-DE)
```

Here are some examples of custom date and time formatting:

[C#]

```
// month/day numbers without/with leading zeroes
String.Format("{0:M/d/yyyy}", dt);            // "3/9/2008"
String.Format("{0:MM/dd/yyyy}", dt);          // "03/09/2008"

// day/month names
String.Format("{0:ddd, MMM d, yyyy}", dt);    // "Sun, Mar 9, 2008"
String.Format("{0:dddd, MMMM d, yyyy}", dt);  // "Sunday, March 9, 2008"

// two/four digit year
String.Format("{0:MM/dd/yy}", dt);            // "03/09/08"
String.Format("{0:MM/dd/yyyy}", dt);          // "03/09/2008"
```

Standard DateTime Formatting

In DateTimeFormatInfo there are defined standard patterns for the current culture. For example property ShortTimePattern is string that contains value `h:mm tt` for **en-US** culture and value `HH:mm` for **de-DE** culture.

Following table shows patterns defined in DateTimeFormatInfo and their values for en-US culture. First column contains format specifiers for the String.Format method.

Specifier	DateTimeFormatInfo property	Pattern value (for en-US culture)
`t`	ShortTimePattern	`h:mm tt`
`d`	ShortDatePattern	`M/d/yyyy`
`T`	LongTimePattern	`h:mm:ss tt`
`D`	LongDatePattern	`dddd, MMMM dd, yyyy`
`f`	*(combination of `D` and `t`)*	`dddd, MMMM dd, yyyy h:mm tt`

`F`	FullDateTimePattern	`dddd, MMMM dd, yyyy h:mm:ss tt`
`g`	*(combination of `d` and `t`)*	`M/d/yyyy h:mm tt`
`G`	*(combination of `d` and `T`)*	`M/d/yyyy h:mm:ss tt`
`m, M`	MonthDayPattern	`MMMM dd`
`y, Y`	YearMonthPattern	`MMMM, yyyy`
`r, R`	RFC1123Pattern	`ddd, dd MMM yyyy HH':'mm':'ss 'GMT'` *(*)*
`s`	SortableDateTimePattern	`yyyy'-'MM'-'dd'T'HH':'mm':'ss` *(*)*
`u`	UniversalSortableDateTimePattern	`yyyy'-'MM'-'dd HH':'mm':'ss'Z'` *(*)*
		(*) = culture independent

Following examples show usage of **standard format specifiers** in String.Format method and the resulting output.

[C#]

```
String.Format("{0:t}", dt);  // "4:05 PM"                         ShortTime
String.Format("{0:d}", dt);  // "3/9/2008"                        ShortDate
String.Format("{0:T}", dt);  // "4:05:07 PM"                      LongTime
String.Format("{0:D}", dt);  // "Sunday, March 09, 2008"          LongDate
String.Format("{0:f}", dt);  // "Sunday, March 09, 2008 4:05 PM"  LongDate+ShortTime
String.Format("{0:F}", dt);  // "Sunday, March 09, 2008 4:05:07 PM" FullDateTime
String.Format("{0:g}", dt);  // "3/9/2008 4:05 PM"                ShortDate+ShortTime
String.Format("{0:G}", dt);  // "3/9/2008 4:05:07 PM"             ShortDate+LongTime
String.Format("{0:m}", dt);  // "March 09"                        MonthDay
String.Format("{0:y}", dt);  // "March, 2008"                     YearMonth
String.Format("{0:r}", dt);  // "Sun, 09 Mar 2008 16:05:07 GMT"   RFC1123
String.Format("{0:s}", dt);  // "2008-03-09T16:05:07"             SortableDateTime
String.Format("{0:u}", dt);  // "2008-03-09 16:05:07Z"            UniversalSortableDateTime
```

See also

- [C#] String Format for Double – format float numbers
- [C#] String Format for Int – format (align) integer numbers
- [C#] IFormatProvider for Numbers – parse float numbers with IFormatProvider
- [C#] Custom IFormatProvider – string formatting with custom IFormatProvider
- [C#] Align String with Spaces – how to align text to the right or left
- [C#] Indent String with Spaces – how to indent text with repeated spaces

- Custom Date and Time Format Strings – MSDN – custom date-time formatting
- Standard Date and Time Format Strings – MSDN – standard date-time formatting