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

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

```
[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#]
```

Standard DateTime Formatting

In <u>DateTimeFormatInfo</u> there are defined standard patterns for the current culture. For example property <u>ShortTimePattern</u> 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 <u>DateTimeFormatInfo</u> and their values for en-US culture. First column contains format specifiers for the <u>String.Format</u> method.

Specifier DateTimeFormatInfo property Pattern value (for en-US culture)

```
t ShortTimePattern h:mm tt
d ShortDatePattern M/d/yyyy

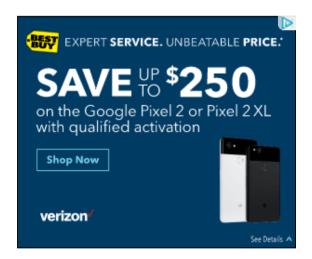
T LongTimePattern h:mm:ss tt
D LongDatePattern dddd, MMMM dd, yyyy

f (combination of D and t) dddd, MMMM dd, yyyy h:mm tt
```

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