

FORMÁTOVÁNÍ – POKRAČOVÁNÍ

Standardní formátovací řetězec pro datum a čas

Standardní formátovací řetězec pro formátovaný parametr typu `System.DateTime` obsahuje pouze jeden znak formátovacího specifikátoru.

Standardní formát vychází z formátu data a času třídy `System.Globalization.DateTimeFormatInfo` pro aktuální kulturu, implicitně podle nastavení Windows.

Přehled formátovacích specifikátorů je uveden v následující tabulce. Uváděná jména vlastností jsou součástí třídy `DateTimeFormatInfo`.

Formátovací specifikátor	Význam
d	Zkrácený formát data podle vzoru vlastnosti <code>ShortDatePattern</code> .
D	Dlouhý formát data podle vzoru vlastnosti <code>LongDatePattern</code> .
f	Kombinace specifikátoru D a t oddělená mezerou.
F	Dlouhý formát data a času podle vzoru vlastnosti <code>FullDateTimePattern</code> .
g	Kombinace specifikátoru d a t oddělená mezerou.
G	Kombinace specifikátoru d a T oddělená mezerou.
M nebo m	Formát data podle vzoru vlastnosti <code>MonthDayPattern</code> .
o	Tvar „cesta tam a zpět“ (angl. „round-trip“). Tento tvar garantuje, že datum a čas zkonvertovaný na řetězec bude možné zkonvertovat zpět na stejnou hodnotu. Má stejný tvar bez ohledu na aktuální kulturu.
R nebo r	Formát data a času podle vzoru vlastnosti <code>RFC1123Pattern</code> daného specifikací RFC (Request for Comments) 1123. Uvedená vlastnost je pouze pro čtení, tedy má stejný tvar bez ohledu na aktuální kulturu.
s	Formát data a času podle vzoru vlastnosti <code>SortableDateTimePattern</code> daného normou ISO 8601 umožňující třídění. Uvedená vlastnost je pouze pro čtení, tedy má stejný tvar bez ohledu na aktuální kulturu.
t	Zkrácený formát času podle vzoru vlastnosti <code>ShortTimePattern</code> .
T	Dlouhý formát času podle vzoru vlastnosti <code>LongTimePattern</code> .
u	Univerzální formát data a času umožňující třídění podle vzoru vlastnosti <code>UniversalSortableDateTimePattern</code> . Uvedená vlastnost je pouze pro čtení, tedy má stejný tvar bez ohledu na aktuální kulturu. Není prováděna konverze na jiné časové pásmo.
U	Univerzální formát data a času umožňující třídění. Odpovídá specifikátoru F s tím rozdílem, že čas je převeden na pásmo UTC (Coordinated Universal Time)..
Y nebo y	Formát data podle vzoru vlastnosti <code>YearMonthPattern</code> .

Příklady pro českou kulturu a instanci data a času v nespecifikovaném čase

```
new DateTime(2007, 1, 20, 23, 5, 10, 458)
```

jsou uvedeny v následující tabulce.

Formátovací specifikátor	Výstup
d	20.1.2007
D	20. ledna 2007
f	20. ledna 2007 23:05
F	20. ledna 2007 23:05:10
g	20.1.2007 23:05
G	20.1.2007 23:05:10
M nebo m	20 ledna
o	2007-01-20T23:05:10.4580000
R nebo r	Sat, 20 Jan 2007 23:05:10 GMT
s	2007-01-20T23:05:10
t	23:05
T	23:05:10
u	2007-01-20 23:05:10Z
U	20. ledna 2007 22:05:10
Y nebo y	leden 2007

Uživatелеm definovaný formátovací řetězec pro datum a čas

Uživatелеm definovaný formátovací řetězec pro datum a čas se skládá z jednoho nebo více formátovacích specifikátorů, které jsou uvedeny v následující tabulce. Uváděná jména vlastností jsou součástí třídy `DateTimeFormatInfo`.

Formátovací specifikátor	Význam
d	Den v měsíci v rozsahu 1 až 31 bez nuly na začátku.
dd	Den v měsíci v rozsahu 01 až 31 s nulou na začátku.
ddd	Zkrácený název dne v týdnu daný vlastností <code>AbbreviatedDayNames</code> .
dddd	Úplný název dne v týdnu daný vlastností <code>DayNames</code> .
f, ff atd. až ffffffff	Desetiny (f), setiny (ff), tisíciny (fff) atd. sekund se zobrazením koncových nevýznamných nul.
F, FF atd. až FFFFFFFF	Dtto jako f ale bez zobrazení koncových nevýznamných nul.
g nebo gg	Zkratka éry. Pro český kalendář (gregoriánský) se jedná o zkratku „n. l.“ (našeho letopočtu).
h	Hodiny v rozsahu 1 až 12 bez nuly na začátku.
hh	Hodiny v rozsahu 01 až 12 s nulou na začátku.
H	Hodiny v rozsahu 0 až 23 bez nuly na začátku.
HH	Hodiny v rozsahu 0 až 23 s nulou na začátku.
K	Text, který informuje o druhu časového pásma. Pro pásmo UTC se zobrazí z, pro pásmo „local“ se zobrazí čas ve tvaru +hh:mm resp. -hh:mm s uvedením o kolik hodin je rozdíl mezi místním časovým pásmem a pásmem GMT. Pro nespécifikované pásmo se nezobrazí nic.

Formátovací specifikátor	Význam
m	Minuty v rozsahu 0 až 59 bez nuly na začátku.
mm	Minuty v rozsahu 00 až 59 s nulou na začátku.
M	Měsíc v rozsahu 1 až 12 bez nuly na začátku.
MM	Měsíc v rozsahu 01 až 12 bez nuly na začátku.
MMM	Zkrácený název měsíce daný vlastností <code>AbbreviatedMonthNames</code> .
MMMM	Úplný název měsíce daný vlastností <code>MonthNames</code> .
s	Sekundy v rozsahu 0 až 59 bez nuly na začátku.
ss	Sekundy v rozsahu 00 až 59 s nulou na začátku.
t	Reprezentuje první znak zkratky A.M./P.M. (v aktuální kultuře) daný vlastností <code>AMDesignator</code> a <code>PMDesignator</code> .
tt	Reprezentuje zkratku A.M/P.M. (v aktuální kultuře) daný vlastností <code>AMDesignator</code> a <code>PMDesignator</code> .
y	Dvouciferný rok bez nuly na začátku.
yy	Dvouciferný rok s nulou na začátku.
yyy	Tříciferný rok s nulami na začátku. Pokud je rok větší než 999, zobrazí se celý.
yyyy a více y	Čtyřciferný a víceciferný rok s nulami na začátku.
z	Posun časového pásma od GMT v hodinách bez nuly na začátku, např. +1.
zz	Posun časového pásma od GMT v hodinách s nulou na začátku, např. +01.
zzz	Posun časového pásma od GMT v hodinách a minutách, např. +01:00.
:	Oddělovač času daný vlastností <code>TimeSeparator</code> .
/	Oddělovač data daný vlastností <code>DateSeparator</code> .
'ABC' nebo "ABC"	Znaky v apostrofěch nebo uvozovkách se zobrazí ve výsledném textu ve stejném tvaru jako ve formátovacím řetězci.
%c	Znak c reprezentuje jeden z formátovacích specifikátorů, který je ve formátovacím řetězci uveden samostatně. Např. formátovací řetězec <code>{0:d}</code> zobrazí datum ve zkráceném tvaru, zatímco řetězec <code>{0:%d}</code> zobrazí den v měsíci.
\c	Ve výsledném textu se zobrazí znak c.
jiný znak	Jakýkoli jiný znak se zobrazí ve výsledném textu ve stejném tvaru jako ve formátovacím řetězci.

Příklady pro českou kulturu a instanci data a času z předchozí kapitoly jsou uvedeny v následující tabulce.

Formátovací položka	Výstup
<code>{0:d. M. yy}</code>	20. 1. 07
<code>{0:dd/MM/yyyy}</code>	20.01.2007
<code>{0:ddd dd.MMM.yyyy}</code>	so 20.I.2007
<code>{0:dddd dd.MMMM yyyy}</code>	sobota 20.ledna 2007

Formátovací položka	Výstup
{0:d.m.yy g}	20.5.07 n. 1.
{0:hh:mm:ss,ffff}	11:05:10,4580
{0:HH:mm:ss,FFFF}	23:05:10,458
{0:HH:mm:ss t}	23:05:10 o
{0:HH:mm:ss tt}	23:05:10 odp.
{0:HH:mm:ss z}	23:05:10 +1
{0:HH:mm:ss K}	23:05:10
{0:%d}	20

Formátovací řetězec pro hodnoty výčtových typů

Formátovací řetězec pro hodnoty výčtových typů obsahuje pouze jeden znak formátovacího specifikátoru. Přehled formátovacích specifikátorů je uveden v následující tabulce.

Formátovací specifikátor	Význam
G nebo g	Hodnota se převede na řetězec. Není-li to možné, převede se na celé číslo. Je-li výčtový typ deklarován s atributem <code>Flags</code> , pokusí se převést hodnotu na textovou reprezentaci kombinace příznaků oddělených čárkou.
F nebo f	Dtto G nebo g s tím rozdílem, že se pokusí převést hodnotu na textovou reprezentaci kombinace výčtových konstant, i když výčtový typ není deklarován s atributem <code>Flags</code> .
D nebo d	Hodnota se převede na celé číslo v desítkové soustavě.
X nebo x	Hodnota se převede na celé číslo v hexadecimálním tvaru délky minimálně 8 číslic doplněný zleva nulami.

Jsou deklarovány následující výčtové typy:

```
enum Den { Pondělí = 0x01, Úterý = 0x02, Středa = 0x04, Čtvrtek = 0x08,
          Pátek = 0x10, Sobota = 0x20, Neděle = 0x40 }
```

[Flags]

```
enum Dny { Pondělí = 0x01, Úterý = 0x02, Středa = 0x04, Čtvrtek = 0x08,
          Pátek = 0x10, Sobota = 0x20, Neděle = 0x40 }
```

Příklady pro hodnoty výčtových typů `Den` a `Dny` jsou uvedeny v následující tabulce.

Formátovací položka	Hodnota	Výstup
{0:G}	Den.Středa	Středa
{0:G}	Den.Středa+1	5
{0:G}	Dny.Pondělí Dny.Středa	Pondělí, Středa
{0:G}	Den.Pondělí Den.Středa	5
{0:F}	Dny.Pondělí Dny.Středa	Pondělí, Středa
{0:F}	Den.Pondělí Den.Středa	Pondělí, Středa
{0:D}	Den.Pátek	16
{0:X}	Den.Pátek	0000010

Použití kultury

Kultura je reprezentována třídou `System.Globalization.CultureInfo`.

Jeden z konstruktorů této třídy má parametr typu `string` reprezentující název kultury a podkultury. Např. `cs-CZ` pro českou kulturu – podkulturu Česká republika, `de-AT` pro německou kulturu – podkulturu Rakousko apod.

Druhý z konstruktorů této třídy má parametr typu `int` reprezentující číslo kultury.

Třída obsahuje různé vlastnosti popisující kulturu, mj. také:

- `DateTimeFormat` typu `DateTimeFormatInfo`,
- `NumberFormat` typu `NumberFormatInfo`,

které jsou využívány pro formátování.

Statická vlastnost `CurrentCulture` poskytuje (nikoli nastavuje) instanci kultury pro aktuální podproces.

Pro každý podproces aplikace lze nastavit samostatnou kulturu.

Aktuální kulturu pro daný podproces poskytuje nebo nastavuje vlastnost `CurrentCulture` třídy `System.Threading.Thread`.

Aktuální podproces aplikace poskytuje statická vlastnost `CurrentThread` třídy `Thread`.

Aktuální kulturu aktuálního podprocesu lze tedy nastavit nebo získat pomocí výrazu

```
System.Threading.Thread.CurrentThread.CurrentCulture
```

nebo získat pomocí výrazu

```
System.Globalization.CultureInfo.CurrentCulture.
```

Příklad

V následujícím příkladu se vytvoří instance české kultury. V ní se změní oddělovač tisíců na tečku. Instance kultury se nastaví jako aktuální kultura aplikace a vypíše se celé číslo pomocí formátovacího řetězce `N`.

```
CultureInfo ci = new CultureInfo("cs-CZ"); // Czech
ci.NumberFormat.NumberGroupSeparator = "."; // oddělovač tisíců
// v číselném formátu

Thread.CurrentThread.CurrentCulture = ci;
Console.WriteLine("{0:N}", 1234);
```

Výstup programu bude následující:

```
1.234,00
```

Rozhaní IFormatProvider

Třída `string` nabízí ještě další metodu `Format`, mající tvar

```
string Format(IFormatProvider provider, string format,
    params object[] args)
```

Parametr `provider` reprezentuje poskytovatele formátu. Rozhraní `IFormatProvider` implementují třídy `NumberFormatInfo`, `DateTimeFormatInfo` a `CultureInfo`.

V metodě `Format` bez parametru typu `IFormatProvider`

```
string Format(string format, params object[] args)
```

se použije jako poskytovatel formátu aktuální kultura aktuálního podprocesu.

Příklad

Uvedenou metodu `Format` lze např. použít v následujících příkazech

```
DateTime d = new DateTime(2007, 2, 28, 10, 23, 5);
Console.WriteLine(string.Format(new CultureInfo("de-De"), "{0:D}", d));
Console.WriteLine(string.Format(NumberFormatInfo.InvariantInfo,
    "{0:0.00}", 123.458));
```

Druhý příkaz vypíše dlouhý formát data pro německou kulturu. Třetí příkaz vypíše číslo v neutrální kultuře.

Výstup programu bude následující:

```
Mittwoch, 28. Februar 2007
123.46
```

Metody ToString

Základní datové typy, výčtové typy a typ `DateTime` obsahují kromě předdefinované virtuální metody `ToString()` třídy `object` další verze této metody:

```
string ToString (IFormatProvider provider)
string ToString (string format)
string ToString (string format, IFormatProvider provider)
```

První verze převádí hodnotu na řetězec s použitím všeobecného formátu ("G") a informací o kultuře.

Druhá verze převádí hodnotu na řetězec s použitím zadaného formátovacího řetězce pro aktuální kulturu.

Třetí verze převádí hodnotu na řetězec s použitím zadaného formátovacího řetězce a informací o kultuře.

Příklad

```
class Program
{
    static void Main(string[] args)
    {
        double a = 123.456;
        DateTime d = new DateTime(2007, 2, 28);
        Console.WriteLine(d.ToString(new CultureInfo("en-US")));
        Console.WriteLine(a.ToString("0.00"));
        Console.WriteLine(a.ToString("0.00", new CultureInfo("en-US")));
        Console.ReadKey();
    }
}
```

Výstup programu bude následující

```
2/28/2007 12:00:00 AM
123,46
123.46
```