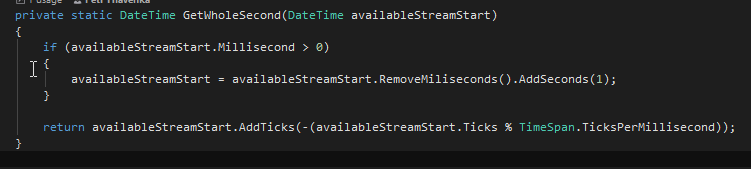


# jak dostat celou sekundu

The Millisecond property gives an integer between 0 and 999 (inclusive). So if the time of day before the operation was, say, 23:48:49.1234567, then that integer will be 123, and the time of day after the operation is 23:48:49.0004567. So it has not truncated to a whole number of seconds.

Abych dostal uplne celou sekundu beze zbytku potrebuju toto:



# Formaty datumu :

14.12.2017 => <TextBlock Text="{Binding DateFrom , StringFormat='dd.MM.yyyy'}" />

14.12.2017 14:28 <GridViewColumn DisplayMemberBinding="{Binding Path=LastModifiedStamp, StringFormat='dd.MM.yyyy HH:mm'}">

GuiResources nebo Localisation:

MessagingService?.PublicateMessage(String.Format(Localisation.LoadingMessages, loading, totalCount));

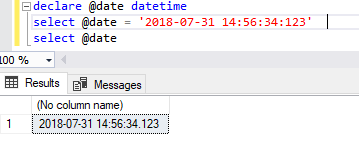


# Zapis datumu v kodu

//selectedDate = new DateTime(2100, 1, 1);

# Zapis datumu v management studiu

* @StartDate = '20150527', @EndDate = '20150528'
* Nebo:



<https://stackoverflow.com/questions/3025361/c-sharp-datetime-to-yyyymmddhhmmss-format>

// 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

// 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

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

/\*

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

\*/

**Update** using c# 6 string interpolation format

// create date time 2008-03-09 16:05:07.123

DateTime dt = new DateTime(2008, 3, 9, 16, 5, 7, 123);

$"{dt:y yy yyy yyyy}"; // "8 08 008 2008" year

$"{dt:M MM MMM MMMM}"; // "3 03 Mar March" month

$"{dt:d dd ddd dddd}"; // "9 09 Sun Sunday" day

$"{dt:h hh H HH}"; // "4 04 16 16" hour 12/24

$"{dt:m mm}"; // "5 05" minute

$"{dt:s ss}"; // "7 07" second

$"{dt:f ff fff ffff}"; // "1 12 123 1230" sec.fraction

$"{dt:F FF FFF FFFF}"; // "1 12 123 123" without zeroes

$"{dt:t tt}"; // "P PM" A.M. or P.M.

$"{dt:z zz zzz}"; // "-6 -06 -06:00" time zone

// month/day numbers without/with leading zeroes

$"{dt:M/d/yyyy}"; // "3/9/2008"

$"{dt:MM/dd/yyyy}"; // "03/09/2008"

// day/month names

$"{dt:ddd, MMM d, yyyy}"; // "Sun, Mar 9, 2008"

$"{dt:dddd, MMMM d, yyyy}"; // "Sunday, March 9, 2008"

// two/four digit year

$"{dt:MM/dd/yy}"; // "03/09/08"

$"{dt:MM/dd/yyyy}"; // "03/09/2008"

# ****Formatovani cisel****

Koukni sem:

<https://docs.microsoft.com/cs-cz/dotnet/standard/base-types/standard-numeric-format-strings>

<https://docs.microsoft.com/cs-cz/dotnet/standard/base-types/custom-numeric-format-strings>

Thousand separator :

<TextBlock Text="{Binding Path=Price, StringFormat={}{0:N0}}" />



Zobrazeni s menou na konci (cislo musí byt typu decimal)

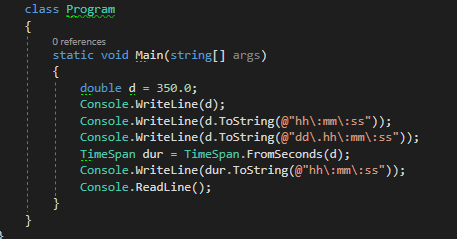
<TextBlock Text="{Binding Path=Price, StringFormat={}{0:c}}" />



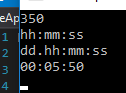
Zobrazeni poctu desetinnych mist

DataMemberBinding="{Binding PriceValue, StringFormat=N1}" // N2 = dve desetinna mista N3 = tri atd..

Pro zapis cisla ve formatu hh:mm:ss je potreba cislo nejprve prevest na TimeSpan.FromNeco:



Vystup:



StringBuilder

**Toto vezme selected item z listview, a kazdou jeho pripojovanou propertu odtabulatoruje. Na konci prida radek a pokracuje dalsim itemem:**

var sb = new StringBuilder();

foreach (var item in listView.SelectedItems)

{

if (item is MediaMessage mm)

{

sb.Append($"{mm.Id}\t");

sb.Append($"{mm.MediumReference.MediumVersion.Name}\t");

sb.Append($"{mm.AdvertisedFrom}\t");

sb.Append($"{mm.NormCreative.Motive.MotiveVersion.PrimaryMotivlet.Owner.Name}\t");

sb.Append($"{mm.NormCreative.Motive.MotiveVersion.Name}\t");

sb.Append($"{mm.CodingPlausibility.Name}\t");

sb.AppendLine();

}

}

Clipboard.SetText(sb.ToString());

}

**Vysledek:**

