# 2016-10-26 MvvmCross app – Humanizer plug-in

## Goal

* Short talk (10 minutes) to show Humanizer plug-in installation and basic usage
* No slides, only show code

## Online help

* Humanizer main page: <http://humanizr.net/>
* Post by Scott Hanselman: <http://www.hanselman.com/blog/NuGetPackageOfTheWeekHumanizerMakesNETDataTypesMoreHuman.aspx>

## Introduction

* Humanizer manipulates and displays strings, enums, dates, times, timespans, numbers and quantities.
* It is open-source and handles multiple languages
* [Features](http://humanizr.net/#features)
  + String: [Humanize](http://humanizr.net/#humanize-string), [Dehumanize](http://humanizr.net/#dehumanize-string), [Transform](http://humanizr.net/#transform-string), [Truncate](http://humanizr.net/#truncate-string), [Format](http://humanizr.net/#format-string)
  + Enum: [Humanize](http://humanizr.net/#humanize-enums), [Dehumanize](http://humanizr.net/#dehumanize-enums)
  + Date and time: [Humanize DateTime](http://humanizr.net/#humanize-datetime), [Humanize TimeSpan](http://humanizr.net/#humanize-timespan), [Fluent date](http://humanizr.net/#fluent-date)
  + Collections: [Humanize](http://humanizr.net/#humanize-collections)
  + [Inflector methods](http://humanizr.net/#inflector-methods) : [Pluralize](http://humanizr.net/#pluralize), [Singularize](http://humanizr.net/#singularize), [Adding Words](http://humanizr.net/#adding-words), [ToQuantity](http://humanizr.net/#toquantity), [Ordinalize](http://humanizr.net/#ordinalize), [Titleize](http://humanizr.net/#titleize), [Pascalize](http://humanizr.net/#pascalize), [Camelize](http://humanizr.net/#camelize), [Underscore](http://humanizr.net/#underscore), [Dasherize & Hyphenate](http://humanizr.net/#dasherize--hyphenate)
  + Number: [Number to Numbers](http://humanizr.net/#number-to-numbers), [Number to words](http://humanizr.net/#number-to-words), [Number to ordinal words](http://humanizr.net/#number-to-ordinal-words), [Roman numerals](http://humanizr.net/#roman-numerals), [Metric numerals](http://humanizr.net/#metric-numerals)
  + Computer size: [ByteSize](http://humanizr.net/#bytesize)

## Published on GitHub

Code of the presentation of Humanizer Nuget package, made at X.NET on October 26, 2016. <http://www.meetup.com/Silicon-Valley-Cross-Platform-Net-Developers-Group/events/234537335/> <http://humanizr.net/>

## Steps

* **Create a new project**: UWP, “Humanizer test”
* **Install Humanizer as a Nuget package**:
  + Right click on Solution. Manage Nuget for Solution…, Humanizer.core
* Create a button. Center it. Add code behind.
* **MainPage.xaml**:

<Page

x:Class="Humanizer\_test\_1.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Humanizer\_test\_1"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<Button Content="Start"

HorizontalAlignment="Center" VerticalAlignment="Center"

Click="BtnStart\_OnClick"/>

</Grid>

</Page>

* **MainPage.xaml.cs**:

using Humanizer;

using Humanizer.Localisation;

using System;

using Windows.UI.Popups;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

namespace Humanizer\_test\_1

{

public sealed partial class MainPage : Page

{

public MainPage()

{

this.InitializeComponent();

}

private async void BtnStart\_OnClick(object sender, RoutedEventArgs e)

{

MessageDialog messageDialog;

// Date

DateTime date = DateTime.UtcNow;

messageDialog = new MessageDialog(date.Humanize());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(date.AddHours(-30).Humanize());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(date.AddHours(4).Humanize());

await messageDialog.ShowAsync();

// Timespan

TimeSpan timeSpan = TimeSpan.FromSeconds(100000); // Also try 100

messageDialog = new MessageDialog(timeSpan.Humanize());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(timeSpan.Humanize(precision: 4));

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(timeSpan.Humanize(precision: 2, maxUnit: TimeUnit.Hour, minUnit: TimeUnit.Second));

await messageDialog.ShowAsync();

// Plural & quantities

var text = "process";

messageDialog = new MessageDialog($"{text.ToQuantity(0)}, {text.ToQuantity(1)}, {text.ToQuantity(3)}");

await messageDialog.ShowAsync();

text = "men";

messageDialog = new MessageDialog(text.ToQuantity(1, ShowQuantityAs.Words));

await messageDialog.ShowAsync();

text = "case";

messageDialog = new MessageDialog(text.ToQuantity(0), "text.ToQuantity(0)");

await messageDialog.ShowAsync();

// Number to words

messageDialog = new MessageDialog(1234.ToWords());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(1234456789.ToWords());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(1234.ToOrdinalWords());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(1234.Ordinalize());

await messageDialog.ShowAsync();

// Computer sizes

messageDialog = new MessageDialog(1024.Kilobytes().ToString());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog(532000.Kilobytes().ToString());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog( 1024.Kilobytes().AddKilobytes(512).Bytes.ToString());

await messageDialog.ShowAsync();

messageDialog = new MessageDialog( 1024.Kilobytes().AddKilobytes(512).Megabytes.ToString());

await messageDialog.ShowAsync();

}

}

}