

[Home](#)[PUBLIC](#)[Stack Overflow](#)[Tags](#)[Users](#)[Jobs](#)**Teams**
Q&A for work[Learn More](#)

Set Default DateTime Format c#

[Ask Question](#)

32

Is there a way of setting or overriding the default DateTime format for an entire application. I am writing an app in C# .Net MVC 1.0 and use alot of generics and reflection. Would be much simpler if I could override the default DateTime.ToString() format to be "dd-MMM-yyyy". I do not want this format to change when the site is run on a different machine.



7

Edit - Just to clarify I mean specifically calling the ToString, not some other extension function, this is because of the reflection / generated code. Would be easier to just change the ToString output.

[c#](#)[asp.net-mvc](#)[datetime](#)[format](#)[edited Oct 5 '14 at 12:32](#)[tereško](#)**52.8k** 21 78 136[asked Sep 7 '09 at 12:30](#)[Matthew Hood](#)**471** 2 11 22

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



The "default format" of a datetime is:

48

`ShortDatePattern + ' ' + LongTimePattern`



at least in the current [mono implementation](#). This is particularly painful in case you want to display something like 2001-02-03T04:05:06Z i.e. the date and time combined as specified in [ISO 8606](#), but not a big problem in your case:

```
using System;
using System.Globalization;
using System.Threading;

namespace test {
    public static class Program {
        public static void Main() {
            CultureInfo culture = (CultureInfo)CultureInfo.CurrentCulture;
            culture.DateTimeFormat.ShortDatePattern = "dd-";
            culture.DateTimeFormat.LongTimePattern = "";
            Thread.CurrentThread.CurrentCulture = culture;
            Console.WriteLine(DateTime.Now);
        }
    }
}
```

This will set the default behavior of ToString on datetimes to return the format you expect.

edited Apr 2 '12 at 5:27



Community ♦

1 1

answered Sep 7 '09 at 12:48



Paolo Tedesco


36.5k 25 118 168

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

1 What do you mean by "in the view"? And what did you do in the global.asax exactly? – [Paolo Tedesco](#) Sep 7 '09 at 13:28

1 If I put `<% CultureInfo culture = (CultureInfo)CultureInfo.CurrentCulture.Clone(); culture.DateTimeFormat.ShortDatePattern = "dd-MMM-yyyy"; culture.DateTimeFormat.LongTimePattern = ""; Thread.CurrentThread.CurrentCulture = culture; %>` at the top of the View it formats all the DateTime's on the page correctly. However when I put that code in the `Application_Start` of the global.asax it does not work. I presume this has got to do with the multiple threading of the requests in IIS. Where is the best place to put this code to affect my whole site. – [Matthew Hood](#) Sep 7 '09 at 14:08

2 I would try `Application_BeginRequest`, but I think you'd better ask a new question... – [Paolo Tedesco](#) Sep 7 '09 at 14:52

This is a good answer. The only problem is that a space is appended at the end. Is there any way how to remove that space? `ShortDatePattern + ' ' + LongTimePattern` – [czlatea](#) Dec 18 '14 at 10:19 



It is dependent on your application's localization-settings. [Change that accordingly](#) to get correct format.

8



Otherwise have a helper-class or an extension-method which always handles your DateTime.

```
public static string ToMyDateTime(this DateTime dateTime)
{
    return dateTime.ToString("dd-MMM-yyyy HH:mm:ss");
}
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

answered Sep 7 '09 at 12:38



Seb Nilsson

14.2k 29 85 115



1



[DateTime.ToString\(\)](#) combines the custom format strings returned by the ShortDatePattern and LongTimePattern properties of the [DateTimeFormatInfo](#). You can specify these patterns in [DateTimeFormatInfo.CurrentInfo](#).

I've never tried this my self.

answered Sep 7 '09 at 12:40



Tadas Šukys

3,357 3 21 30



1



If you want to be sure that your culture stays the same, just set it yourself to avoid troubles.

```
System.Globalization.CultureInfo ci = new System.Globaliza
System.Threading.Thread.CurrentThread.CurrentCulture = ci;
System.Threading.Thread.CurrentThread.CurrentUICulture = c;
```

The above example sets the culture of the thread to Belgian-Dutch.

CurrentCulture does all the date and time handling and CurrentUICulture handles UI localization like resources.

answered Sep 7 '09 at 12:51



Bjorn Bailleul

1,079 9 20



You can write an ExtensionMethod like this:

0

```
public static string ToString(this DateTime dateTime)
{
    return dateTime.ToString("needed format");
}
```



answered Sep 7 '09 at 12:35



ema

4,342 20 30



I'm not sure if this would work for a web app, but you could try to set the `DateTimeFormat` property for the current culture.

0



Check [this](#) and specially [this](#).

answered Sep 7 '09 at 12:49



Fernando

3,689 1 21 34