

برای تبدیل تاریخ میلادی به شمسی راه‌های زیادی وجود دارد که خوب این هم یک نوع از آنها هست. برای این کار با استفاده از [Extension Method](#) برای کلاس DateTime تاریخ رو تبدیل می‌کنیم. برای فرمت‌های نمایشی تاریخ هم می‌بایست به صورت متن به Method بدهید تا تاریخ رو به اون صورتی که می‌خواهید نمایش بدهد، برای نوشتن فرمت‌ها هم می‌توانید از [این لینک](#) استفاده کنید. خوب برای این کار ابتدا یک کلاس با نام PersianDateExtensionMethods بسازید و تمام کد زیر رو جایگزین کدهای کلاس کنید:

```
using System;
using System.Globalization;
using System.Reflection;

namespace System
{
    public static class PersianDateExtensionMethods
    {
        private static CultureInfo _Culture;
        public static CultureInfo GetPersianCulture()
        {
            if (_Culture == null)
            {
                _Culture = new CultureInfo("fa-IR");
                DateTimeFormatInfo formatInfo = _Culture.DateTimeFormat;
                formatInfo.AbbreviatedDayNames = new[] { "ش", "ج", "پ", "چ", "س", "د", "ی" };
                formatInfo.DayNames = new[] { "یکشنبه", "دوشنبه", "سه شنبه", "چهارشنبه", "پنجشنبه", "جمعه", "شنبه" };
                var monthNames = new[]
                {
                    "فروردین", "اردیبهشت", "خرداد", "تیر", "مرداد", "شهریور", "مهر", "آبان", "آذر",
                    "دی", "بهمن", "اسفند", ""
                };
                formatInfo.AbbreviatedMonthNames =
                    formatInfo.MonthNames =
                    formatInfo.MonthGenitiveNames = formatInfo.AbbreviatedMonthGenitiveNames =
                    monthNames;
                formatInfo.AMDesignator = "ق.ظ";
                formatInfo.PMDesignator = "ب.ظ";
                formatInfo.ShortDatePattern = "yyyy/MM/dd";
                formatInfo.LongDatePattern = "dddd, dd MMMM, yyyy";
                formatInfo.FirstDayOfWeek = DayOfWeek.Saturday;
                System.Globalization.Calendar cal = new PersianCalendar();

                FieldInfo fieldInfo = _Culture.GetType().GetField("calendar", BindingFlags.NonPublic |
                    BindingFlags.Instance);
                if (fieldInfo != null)
                    fieldInfo.SetValue(_Culture, cal);

                FieldInfo info = formatInfo.GetType().GetField("calendar", BindingFlags.NonPublic |
                    BindingFlags.Instance);
                if (info != null)
                    info.SetValue(formatInfo, cal);

                _Culture.NumberFormat.NumberDecimalSeparator = "/";
                _Culture.NumberFormat.DigitSubstitution = DigitShapes.NativeNational;
                _Culture.NumberFormat.NumberNegativePattern = 0;
            }
            return _Culture;
        }

        public static string ToPeString(this DateTime date, string format = "yyyy/MM/dd")
        {
            return date.ToString(format, GetPersianCulture());
        }
    }
}
```

در کد بالا ما با استفاده از متد `GetPersianCulture` یک کالچر برای تاریخ شمسی و فرمت‌های اون می‌سازیم و در متد `ToPeString` از اون استفاده و به متد `ToString` از کلاس `DateTime` پاس می‌دهیم. چند مثال هم برای استفاده از متد `ToPeString`:

```
var date1 = DateTime.Now.ToPeString("yyyy/MM/dd");
var date2 = DateTime.Now.ToPeString("dddd, dd MMMM,yyyy");

//Output:
//1391/12/13
//1391, دسامبر 13, یکشنبه
```

نکته: با استفاده از `Culture` ای که در کلاس بالا در متد `GetPersianCulture` ساخته می‌شود امکانش هست که خود کلاس `DateTime` رو به شمسی تبدیل کرد، برای این کار باید به صورت زیر عمل کرد:

```
Thread.CurrentThread.CurrentCulture = Thread.CurrentThread.CurrentUICulture =
PersianDateExtensionMethods.GetPersianCulture();

var d1 = DateTime.Now.ToString();
//Output : 1391/12/13 11:25:44 ب.ظ
```

## نظرات خوانندگان

نویسنده: امیرحسین مرجانی  
تاریخ: ۱۳۹۱/۱۲/۱۴ ۰:۱۷

مطلب خوبی بود، ممنونم  
اگر احتیاجی به استفاده دوم نباشه، بهتر GetPersianCulture() به صورت Private تعریف بشه.

نویسنده: پدram جباری  
تاریخ: ۱۳۹۱/۱۲/۱۴ ۰:۲۳

مرسی،  
درسته، ولی برای نکته آخر مطلب به صورت public تعریف کردم

نویسنده: محمدی مهدی  
تاریخ: ۱۳۹۱/۱۲/۱۵ ۱۳:۵۶

ممکنه درباره این خط کد توضیح بدین:

```
FieldInfo fieldInfo = _Culture.GetType().GetField("calendar", BindingFlags.NonPublic | BindingFlags.Instance);
```

سپاس فراوان

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۱۵ ۱۴:۳۸

علت دسترسی به فیلد غیرعمومی (شده توسط مایکروسافت) و مقدار دهی آن است.

نویسنده: پدram جباری  
تاریخ: ۱۳۹۱/۱۲/۱۵ ۱۵:۱۹

با استفاده از این خط کد فیلد calendar در آبجکت Culture\_ رو میگیره و در دو خط پایین مقدار دهی می‌کنه،  
علت هم این هست که چون اون فیلد public نیست با استفاده از reflection اون رو مقدار دهی می‌کنیم.  
در مورد متد GetField هم در پارامتر دوم هم اون یک حالت فیلتر کردن هست که می‌گه فیلدی که public نیست و شامل مدیفایر static نباشد، برای دیدن تمام BindingFlags ها و توضیحتشون [این لینک](#) رو مشاهده کنید.

نویسنده: محمدی مهدی  
تاریخ: ۱۳۹۱/۱۲/۱۵ ۱۹:۱۳

ممنون از توضیحات شما.

نویسنده: omid  
تاریخ: ۱۳۹۱/۱۲/۱۶ ۵:۵۵

سلام  
آیا با تغییر culture به فارسی، timeZone هم تغییر می‌کنه یا بایستی به صورت جدا timezone رو هم تغییر بدیم؟

نویسنده: پدram جباری  
تاریخ: ۱۳۹۱/۱۲/۱۶ ۱۲:۴۶

سلام

نه Culture با Time Zone فرق داره، برای تغییر Time Zone باید جداگانه تغییرات رو انجام بدید. یکی از دلیل هایی که این دو رو با هم ترکیب نکردن این هست که زبان یک برنامه یا تقویم اون برنامه به منطقه ای که اون نرم افزار استفاده میشه مرتبط نیست و ممکنه تو هر منطقه زمانی باشه، که باید به اون منطقه زمانی ست بشه.

نویسنده: sh-a

تاریخ: ۱۳۹۲/۰۱/۱۵ ۱:۳۰

با سلام

آقای جباری با این روش راهی برای فارسی دیده شدن کنترهای datetimepicker یا monthcalendar ویژوال استدیو وجود داره ؟

ممنون میشم توضیح بدین

نویسنده: مهدی غفاری

تاریخ: ۱۳۹۲/۰۱/۱۵ ۱۱:۵۷

با سلام و تشکر از مطلب خوبتون  
میخواستم بدونم آیا این روش با سال کبیسه مشکلی نداره؟ یعنی مورد سال کبیسه هم محاسبه شده تو این روش؟  
ممنون

نویسنده: پدram جباری

تاریخ: ۱۳۹۲/۰۱/۱۵ ۱۲:۳۰

با سلام

راستش برای کنترلرهای مربوط به تاریخ نه این امکان وجود نداره، البته اگر منظورتون از فارسی کردن شمسی کردن تاریخ کنترلر هست ولی اگه منظورتون ترجمه اسم ماهها و غیره هست آره میشه از همین روش استفاده کرد فقط باید تو قسمت آخر رفلکشن رو حذف کنید.

قبلا در Winform و WPF تست کردم و جواب نداد، که البته مشکل از اونجایی هست که موقع نوشتن این کنترلرها این امکان که بشه فرمت تاریخ رو تغییر داد ( شمسی کردن ) قرار ندادن.  
البته قبلا در WPF یکی از دوستان کنترلر مربوط به تاریخ رو دوباره بازنویسی کرده بود که میتونستید با تغییر Culture فرمت تاریخ رو تغییر بدید، اگر پیدا کردم لینک رو اینجا قرار میدم.

نویسنده: پدram جباری

تاریخ: ۱۳۹۲/۰۱/۱۵ ۱۲:۵۱

با سلام

اگر PersianCalendar با کبیسه مشکلی نداشته باشه نه این هم مشکلی نداره، البته من خودم تست کردم که مشکلی ندیدم.

نویسنده: sh-a

تاریخ: ۱۳۹۲/۰۱/۱۵ ۱۷:۵۴

ممنون از جواب دوستان . تا اینجا فکر کنم برای کنترلرهای نمایش تاریخ فارسی ، از کارهای آقای هادی اسکندری استفاده کنم

باتشکر

نویسنده: debugger

تاریخ:

۹:۳ ۱۳۹۲/۰۳/۱۵

سلام

در قسمتی که آرایه مربوط به مقدار دهی اولیه ماهها را تعریف کردید تعداد آنها 13 می باشد که آیتم آخری برابر با رشته متنی خالی می باشد . دلیل خاصی دارد ؟

ممنون

نویسنده:

سید امیر سجادی

تاریخ:

۱۱:۱۰ ۱۳۹۲/۰۳/۱۵

به نظر شما این روش بهتره (تغییر culture) یا استفاده مستقیم از یک کلاس برای تبدیل تاریخ (با استفاده از PersianCalendar) ؟ کدوم یکی از نظر بهینه بودن برنامه و منطق بهتره ؟

نویسنده:

پدرام جباری

تاریخ:

۱۴:۵۹ ۱۳۹۲/۰۳/۱۵

سلام

خود Culture اصلی هم برای ماهها 13 تا تعریف کرده که البته من علتش رو نمی دونم ، داخل [این لینک](#) می تونید ببینید

نویسنده:

پدرام جباری

تاریخ:

۱۵:۳ ۱۳۹۲/۰۳/۱۵

سلام

من اول اشتباه منظورتون رو فهمیدم، بستگی به خودتون داره که از کدوم روش می خواهید استفاده کنید، خوب اگر Culture در Thread برنامه رو تغییر بدید باید از کلاس واسطه به تاریخ میلادی دسترسی داشته باشید، در هر صورت تفاوتی نداره چون در هر دو مورد یک بار روند ساخت Culture انجام میشه

نویسنده:

Imo0

تاریخ:

۱۳:۱۱ ۱۳۹۲/۰۶/۰۴

سلام. ممنون به خاطر این اطلاعات عالی.

یه مشکل . من از این کلاس استفاده کردم تویه پروژه وب که با MVC نوشتم. برای تغییر کلی این تاریخ هم اون رو توی App\_Start داخل فایل Global.asax گذاشتم. همه جا عالی کار میکنه و به صورت خودکار تاریخ فارسی میده. فقط یه جا نمیده. من برای بروزرسانی دیتابیس از CodeFirst Migration استفاده میکنم. برای مقدار دهی اولیه به مدلم تویه تابع Seed یه فیلدی دارم که تاریخه. تو اونجا تاریخو داره همون میلادی میگیره بازم؟ چرا؟

breakPoint هم که گذاشتم. با اینکه این تابع App\_Start از همه زودتر اجرا میشه و تنظیمات رو اوکی میکنه ولی بازم تو اون تابع Seed همون میلادیه!

نمیدونم کار خاصی باید کرد یا مثلا کدو تو یه جای دیگه غیره App\_Start بزارم و یا....

ممنون میشم جواب بدین .

نویسنده:

محسن خان

تاریخ:

۱۴:۰ ۱۳۹۲/۰۶/۰۴

جایی که Thread.CurrentThread.CurrentCulture رو مقدار دهی کردی، بررسی کن Thread.CurrentThread.ManagedThread Id چه مقداری داره و آیا با مقدار Id ManagedThread موجود در متد Seed یکی هست یا خیر.

نویسنده:

پدرام جباری

تاریخ:

۱۵:۱۹ ۱۳۹۲/۰۶/۰۴

سلام

آقای محسن خان درست میگن امکانش هست thread فرق کنه البته این رو هم در نظر بگیر که داخل دیتابیس به میلادی ذخیره میشه و موقع نمایش تبدیل میکنه و به شمسی به شما نشون میده

نویسنده: Imo0

تاریخ: ۱۶:۵۵ ۱۳۹۲/۰۶/۰۴

ممنون بابت جواب. اون یکی رو با راهنمایی شما حل کردم. ممنون. یه مشکل دیگه. من در این پروژه از Telerik Extension استفاده کردم. خودش این امکانو داره که بتونیش Globalization کنی. من این CultureInfo رو که شما درست کردینو بهش میدم اما نصفه نیمه شده. مثلاً نوشته 27 مرداد 2013؟ قسمت بالاشم که اسم ماه و سال هست، نوشته مثلاً مرداد 2013. در کل سال رو به همون صورت میلادی نشون میده و بقیه رو به شمسی. این CultureInfo شما چیزی دیگه‌ای کم نداره که اونم رو هم مقدار دهی کنیم، تکمیل بشه؟ من احساس میکنم همه توابع و خصوصیت‌هاش مقدار دهی نشده. در ضمن با اینکه تقویمشو Persian Calender دادیم پس چرا هنوز داره با همون میلادی روزها و بقیه رو حساب میکنه. تازه خلاصه اسم روزها رو هم نشون نمیده. جالب اینجاست که وقتی من تاریخ سیستم خودمو فارسی میکنم مثلاً این چیزاشو نمایش میده. چیزایی مثله خلاصه اسم روز. اگه بشه اینو به جورایی فارسی کرد خیلی خوب میشه. هم کلی تمپلیت داره. هم کلی افکت خوب داره. هم خود تلریک یه نوع Project template برای MVC4 درست کرده که وقتی باهاش پروژتونو ایجاد میکنید دیگه خودش اتوماتیک هر جا فیلدتونو تاریخ تعریف کنید خیلی باحال همه چیز رو میاره. ببینین میتونین یه کاریش بکنین. باز ممنون.



نویسنده: پدram جباری

تاریخ: ۱۷:۴۳ ۱۳۹۲/۰۶/۰۴

سلام

من خودم با اکستنشن telerik تو mvc کار نکردم ولی از صحبتی که شما کردید میشه جوابتون رو داد دقت کنید شما سمت سرور که زبان C# هست تاریخ رو به شمسی تغییر دادید، ولی سمت کلاینت که زبانش از جاوا اسکریپت هست تاریخ به میلادی هندل میشه که اگر بخواهید از خود telerik استفاده کنید باید فایل‌های جاوا اسکریپت قسمت datetimepicker (اگه اشتباه نگم اسمش رو) باید تغییر بدید ولی اگر هم بخواهید از کامپوننت‌های آماده استفاده کنید که زیاد

هست تو اینترنت سرچ کنید حتما پیدا می‌کنید

نویسنده: پوریا

تاریخ: ۱۵:۲۴ ۱۳۹۲/۰۹/۱۸

سلام خیلی خوبه منم از کلاس زیر استفاده می‌کنم  
اما تو سیلورلایت چطور استفاده کنم؟

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Reflection;
using System.Text;

public class PersianCulture : CultureInfo
{
    private readonly Calendar cal;
    private readonly Calendar[] optionals;

    public PersianCulture(): this("FA-IR", true)
    {
    }

    public PersianCulture(string cultureName, bool useUserOverride): base(cultureName, useUserOverride)
    {
        //Temporary Value for cal.
        cal = base.OptionalCalendars[0];

        //populating new list of optional calendars.
        var optionalCalendars = new List<Calendar>();
        optionalCalendars.AddRange(base.OptionalCalendars);
        optionalCalendars.Insert(0, new PersianCalendar());

        Type formatType = typeof(DateTimeFormatInfo);
        Type calendarType = typeof(Calendar);

        PropertyInfo idProperty = calendarType.GetProperty("ID", BindingFlags.Instance |
BindingFlags.NonPublic);
        FieldInfo optionalCalendarfield = formatType.GetField("optionalCalendars",
BindingFlags.Instance |
BindingFlags.NonPublic);

        //populating new list of optional calendar ids
        var newOptionalCalendarIDs = new Int32[optionalCalendars.Count];
        for (int i = 0; i < newOptionalCalendarIDs.Length; i++)
            newOptionalCalendarIDs[i] = (Int32)idProperty.GetValue(optionalCalendars[i], null);

        optionalCalendarfield.SetValue(DateTimeFormat, newOptionalCalendarIDs);

        optionals = optionalCalendars.ToArray();
        cal = optionals[0];
        DateTimeFormat.Calendar = optionals[0];

        DateTimeFormat.MonthNames = new[] { "شهریور", "مرداد", "تیر", "خرداد", "اردیبهشت", "فروردین", "اسفند", "بهمن", "دی", "آذر", "آبان", "مهر", string.Empty };
        DateTimeFormat.MonthGenitiveNames = new[] { "مرداد", "تیر", "خرداد", "اردیبهشت", "فروردین", "اسفند", "بهمن", "دی", "آذر", "آبان", "مهر", "شهریور", string.Empty };
        DateTimeFormat.AbbreviatedMonthNames = new[] { "مرداد", "تیر", "خرداد", "اردیبهشت", "فروردین", "اسفند", "بهمن", "دی", "آذر", "آبان", "مهر", "شهریور", string.Empty };
        DateTimeFormat.AbbreviatedMonthGenitiveNames = new[] { "مرداد", "تیر", "خرداد", "اردیبهشت", "فروردین", "اسفند", "بهمن", "دی", "آذر", "آبان", "مهر", "شهریور", string.Empty };

        DateTimeFormat.AbbreviatedDayNames = new string[] { "ی", "د", "س", "چ", "پ", "ج", "ش" };
        DateTimeFormat.ShortestDayNames = new string[] { "ی", "د", "س", "چ", "پ", "ج", "ش" };
        DateTimeFormat.DayNames = new string[] { "یکشنبه", "دوشنبه", "سه‌شنبه", "چهارشنبه", "پنجشنبه", "جمعه", "شنبه" };

        DateTimeFormat.AMDesignator = "ق.ظ";
        DateTimeFormat.PMDesignator = "ب.ظ";
    }
}
```

```

DateTimeFormat.ShortDatePattern = "yyyy/MM/dd";
DateTimeFormat.LongDatePattern = "yyyy/MM/dd";

DateTimeFormat.SetAllDateTimePatterns(new[] { "yyyy/MM/dd", 'd' });
//DateTimeFormat.SetAllDateTimePatterns(new[] { "dddd, dd MMMM yyyy", 'D' });
//DateTimeFormat.SetAllDateTimePatterns(new[] { "yyyy MMMM", 'y' });
//DateTimeFormat.SetAllDateTimePatterns(new[] { "yyyy MMMM", 'Y' });

}

public override Calendar Calendar
{
    get { return cal; }
}

public override Calendar[] OptionalCalendars
{
    get { return optionals; }
}
}

```

نویسنده: حسین اکبرزاده  
تاریخ: ۱۵:۱۳ ۱۳۹۲/۱۱/۱۸

سلام . من وقتی تاریخ رو به فرمت خورشیدی با همین روش و در global اعمال میکنم ، در kendoui در grid وقتی حالت ajax () رو فعال میکنم این Error رو میگیرم .

Specified time is not supported in this calendar. It should be between 03/21/0622 00:00:00 (Gregorian date) and 12/31/9999 23:59:59 (Gregorian date), inclusive.  
Parameter name: time

اما در حالت server () مشکلی نداره . اشکال از کجاست و چه جوری میشه حل کرد ؟

نویسنده: سوران  
تاریخ: ۱۵:۲۰ ۱۳۹۳/۰۱/۰۲

سلام  
من از ASP.NET MVC5 استفاده میکنم.  
می‌خواستم بدونم دقیقاً این کلاس رو باید در کجا قرار داد و چه تنظیماتی در global.asax انجام داد تا همه چی به صورت خودکار اجرا بشه.  
ممنون میشم راهنمایی کنید.



عنوان:	تأثیر فرهنگ جاری سیستم بر روی اعداد در دات نت
نویسنده:	وحید نصیری
تاریخ:	۸:۵ ۱۳۹۳/۰۹/۰۵
آدرس:	<a href="http://www.dotnettips.info">www.dotnettips.info</a>
گروه‌ها:	C#, Globalization, Persian, Culture

در ویندوز 8، میکروسافت سعی کرده‌است تا تنظیمات بومی مرتبط با ایران، با واقعیت انطباق بیشتری داشته باشد. برای مثال در فرهنگ فارسی سیستم، علامت ممیز آن / است؛ بجای . معمول.

**Customize Format**

Numbers | Currency | Time | Date

**Example**

Positive: 123,456,789/00      Negative: 123,456,789/00-

Positive: ۱۲۳۴۵۶۷۸۹/۰۰      Negative: -۱۲۳۴۵۶۷۸۹/۰۰

Decimal symbol: /

No. of digits after decimal: 2

Digit grouping symbol: ,

Digit grouping: 123,456,789

Negative sign symbol: -

Negative number format: 1/1-

Display leading zeros: 0/7

List separator: :

Measurement system: Metric

Standard digits: •۱۲۳۴۵۶۷۸۹

Use native digits: Context

Click Reset to restore the system default settings for numbers, currency, time, and date.

Reset

OK Cancel Apply

برای آزمایش آن، سعی کنید چنین برنامه‌ای را در ویندوز 8 اجرا کنید:

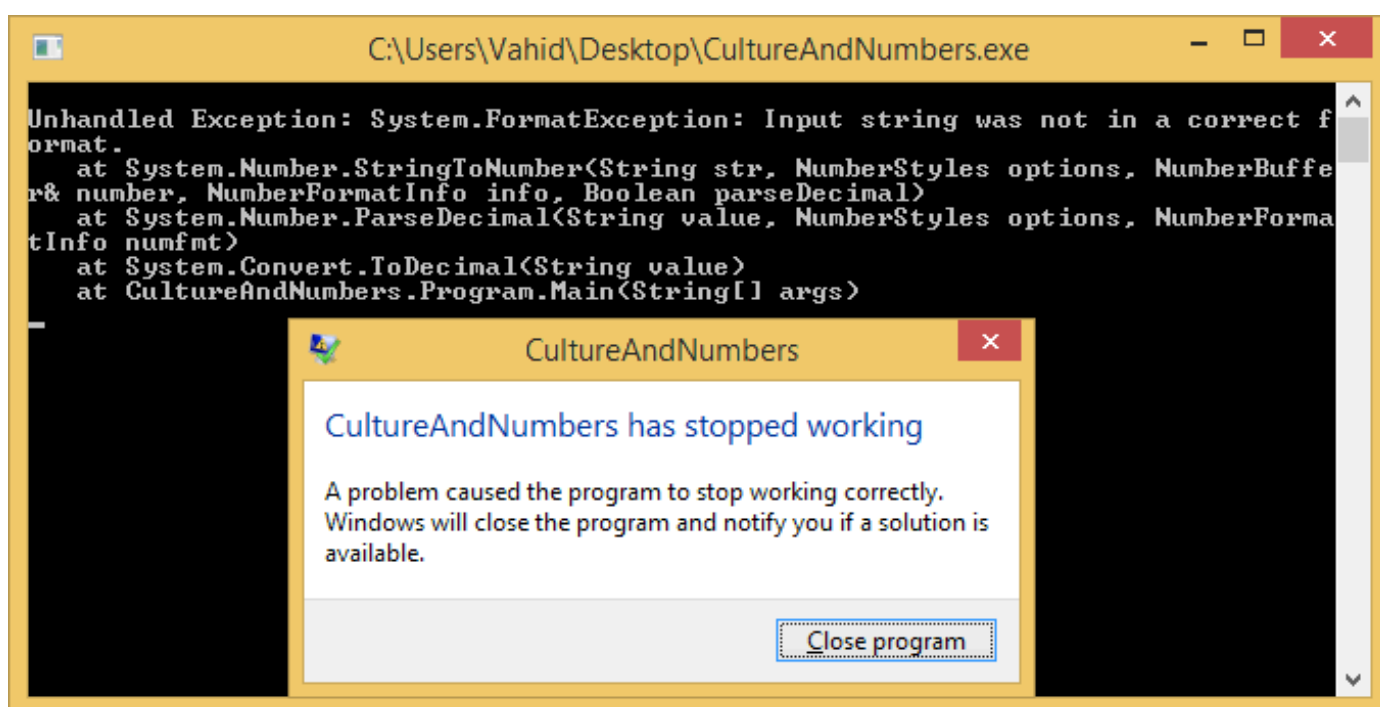
```
using System;
namespace CultureAndNumbers
```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            var number = Convert.ToDecimal("12.32");
            Console.WriteLine(number);
        }
    }
}

```

در اینجا سعی شده است یک عدد دسیمال رشته‌ای به معادل عددی آن تبدیل شود. خروجی آن به نحو ذیل است:



بله! چون در فرهنگ جاری سیستم، علامت ممیز دیگر نیست، رشته‌ی 12.32 نیز بی‌معنا است و قابل تبدیل به یک عدد دسیمال نخواهد بود.

همچنین باید دقت داشت تأثیر فرهنگ جاری سیستم بر روی متدهای `Convert.ToDecimal` و `decimal.Parse` یکسان است.

### روشی برای آزمایش موقت فرهنگ‌های مختلف

برای اینکه بتوان فرهنگ‌های مختلف را به سادگی مورد آزمایش قرار داد، نیاز است خاصیت `CurrentCulture` ترد جاری برنامه را تغییر داد و پس از پایان کار، مجدداً این ترد را به فرهنگ پیش از آزمایش تنظیم کرد. برای این منظور می‌توان از پیاده‌سازی الگوی `IDisposable` کمک گرفت:

```

public class CultureScope : IDisposable
{
    private readonly CultureInfo _originalCulture;

    public CultureScope(string culture)
    {
        _originalCulture = Thread.CurrentThread.CurrentCulture;
        Thread.CurrentThread.CurrentCulture = new CultureInfo(culture);
    }
}

```

```
public void Dispose()
{
    Thread.CurrentThread.CurrentCulture = _originalCulture;
}
}
```

در این حالت برای آزمایش فرهنگ فارسی نصب شده در سیستم می‌توان به صورت ذیل عمل کرد. این فرهنگ تنها در چارچوب قطعه کد `using`، تنظیم می‌شود و پس از آن، مجدداً برنامه با فرهنگ اصلی پیش از اجرای این قطعه کد به کار خود ادامه خواهد داد:

```
using (var cultureScope = new CultureScope("fa-IR"))
{
    Console.WriteLine(Thread.CurrentThread.CurrentCulture.NumberFormat.NumberDecimalSeparator);
    var number = decimal.Parse("12.32");
    Console.WriteLine(number);
}
```

### تعیین صریح فرهنگ مورد استفاده

یک راه حل برای رفع این مشکل، قید صریح فرهنگ مورد استفاده است. برای مثال اگر اعداد در بانک اطلاعاتی به صورت 12.32 ثبت شده‌اند، می‌توان نوشت:

```
using (var cultureScope = new CultureScope("fa-IR"))
{
    Console.WriteLine(Thread.CurrentThread.CurrentCulture.NumberFormat.NumberDecimalSeparator);
    var number = decimal.Parse("12.32", new CultureInfo("en"));
    Console.WriteLine(number);
}
```

در اینجا فرهنگ انگلیسی به صورت صریح ذکر شده‌است و دیگر فرهنگ تنظیم شده‌ی `fa-IR` مورد استفاده قرار نخواهد گرفت. اما این روش هم قابل اطمینان نیست. زیرا کاربر می‌تواند در کنترل پنل سیستم، به سادگی علامت ممیز را مثلاً به `#` تغییر دهد و در این حالت باز هم برنامه کرش خواهد کرد. راه حلی که برای این مساله در دات نت وجود دارد، فرهنگی است به نام `Invariant` که یک کپی فقط خواندنی از فرهنگ انگلیسی را به همراه دارد و در این حالت تنظیمات اختصاصی کاربر در کنترل پنل، ندید گرفته خواهند شد:

```
using (var cultureScope = new CultureScope("fa-IR"))
{
    Console.WriteLine(Thread.CurrentThread.CurrentCulture.NumberFormat.NumberDecimalSeparator);
    var number = decimal.Parse("12.32", CultureInfo.InvariantCulture);
    Console.WriteLine(number);
}
```

اینبار هر چند فرهنگ ترد جاری به `fa-IR` تنظیم شده‌است اما چون فرهنگ مورد استفاده `CultureInfo.InvariantCulture` است، از یک فرهنگ انگلیسی فقط خواندنی که تنظیمات محلی کاربر بر روی آن بی‌تأثیر است، استفاده خواهد شد. یک چنین کدی در تمام سیستم‌ها بدون مشکل کار می‌کند.

## نظرات خوانندگان

نویسنده: سالار خلیل زاده  
تاریخ: ۱۷:۲۷ ۱۳۹۳/۰۹/۰۵

با تشکر، نکته مهمی بود.  
و اینجاست که برنامه نویس‌ها می‌گن در سیستم من که درست کار می‌کنه! (: