

عنوان:	تأثیر فرهنگ جاری سیستم بر روی اعداد در دات نت
نویسنده:	وحید نصیری
تاریخ:	۸:۵ ۱۳۹۳/۰۹/۰۵
آدرس:	www.dotnettips.info
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

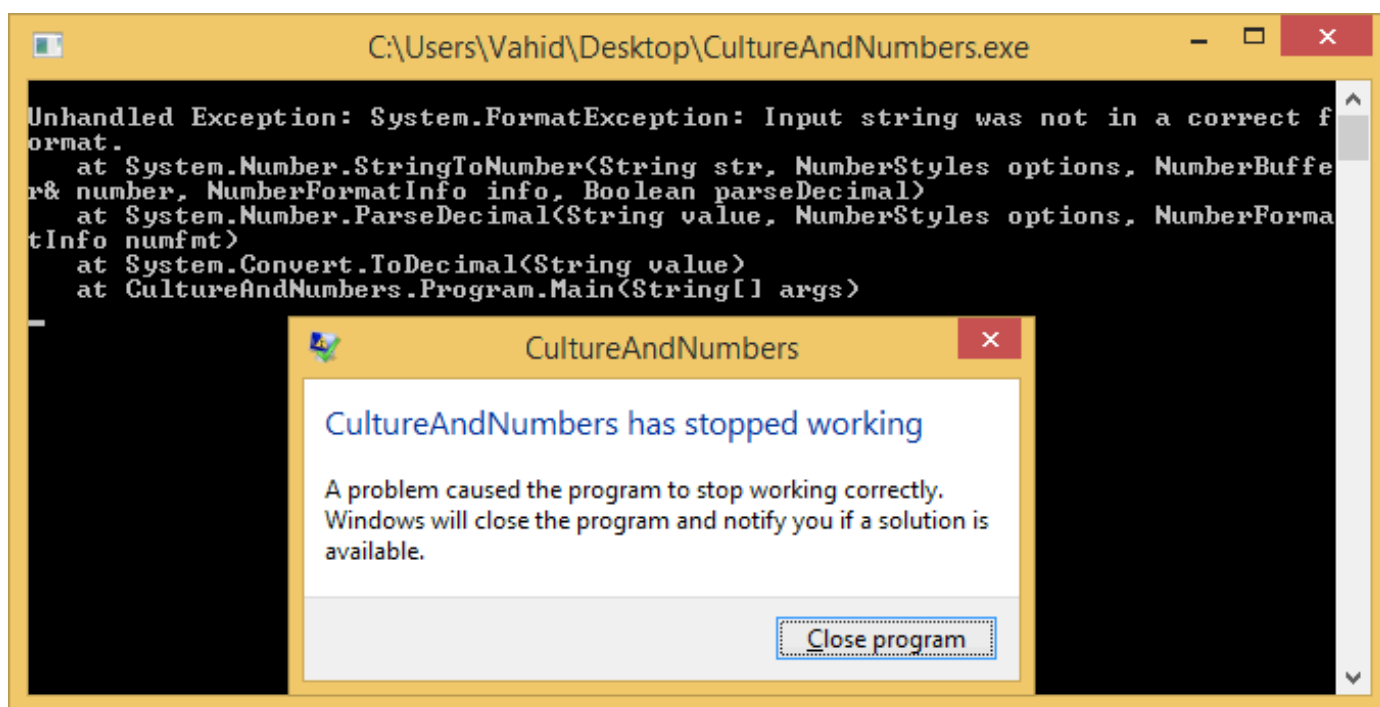
OK Cancel Apply

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! (: