

عنوان: از متد DateTime.ToString بدون پارامتر استفاده نکنید!

نویسنده: وحید نصیری

تاریخ: ۲۱:۵ ۱۳۹۱/۰۸/۲۶

آدرس: www.dotnettips.info

برچسب‌ها: C#, DateTime

در حین تهیه کتابخانه Silverlight DatePicker فارسی، گاهی استفاده کنندگان گزارش می‌دادند که برنامه روی سیستم‌های مختلف کرش می‌کند یا تبدیل تاریخ درست انجام نمی‌شود. مشکل هم پس از بررسی طولانی به این ترتیب مشخص شد که استفاده از DateTime.ToString بدون ذکر پارامترهایی که در ادامه توضیح داده خواهند شد، اشتباه است.

متد ToString بر اساس تنظیمات محلی عمل می‌کند

خروجی فراخوانی ذیل

```
DateTime.Now.ToString()
```

در یک سیستم می‌تواند

```
01/11/2012 09:49:08 ق.ظ
```

و در سیستمی دیگر

```
11/1/2012 9:49:08 AM
```

باشد.

این مساله خصوصا برای ذخیره سازی و پردازش اطلاعات به صورت رشته بسیار مهم و مساله ساز است. فرض کنید در یک شبکه با تنظیمات محلی متفاوت، کاربران اطلاعات تاریخ متفاوتی را به بانک اطلاعاتی ارسال کنند. پردازش صحیح این تاریخ‌ها تقریباً غیرممکن است. حالت اول روز 11 ماه یک را نمایش می‌دهد و حالت دوم روز 1 ماه 11. در حالیکه هر دو تاریخ در یک روز ثبت شده‌اند اما تنظیمات محلی کاربران متناظر یکسان نبوده است. برای رفع این مشکل نیاز است ToString را مستقل از تنظیمات محلی کاربران کرد:

```
DateTime.Now.ToString(CultureInfo.InvariantCulture)
```

این مورد نکته‌ای است که اگر از FxCop برای آنالیز اسمبلی‌های برنامه خود استفاده کنید، حتما گوشزد خواهد شد. همچنین ReSharper نیز رعایت آن را در نگارش‌های اخیر خود گنجانده است.

مشکل دیگر مشابه در حین کار با Silverlight و WPF، استفاده و پردازش e.NewValue تغییرات خواص است. این مقدار به صورت object ارسال می‌شود و برای پردازش آن نباید e.NewValue.ToString فراخوانی شود. روش صحیح دریافت تاریخ از آن باید به صورت زیر باشد:

```
public static DateTime? DateTimeTryParse(object data)
{
    if (data == null)
        return null;

    if (data.GetType().Equals(typeof(DateTime)))
        return (DateTime)data;

    DateTime result;
    if (DateTime.TryParse((string)data, CultureInfo.InvariantCulture, DateTimeStyles.None, out result))
        return result;

    return null;
}
```

دو نکته در اینجا قابل توجه است:

- در متد `DateTime.TryParse` بجای `data.ToString`، به `string` تبدیل شده است. متد `DateTime.TryParse` نیز حالت ویژه‌ای پیدا کرده و `CultureInfo.InvariantCulture` در آن قید شده است.
- همچنین چون نوع `e.NewValue` تغییرات دریافتی از نوع `object` می‌باشد، بهتر است ابتدا بررسی شود که آیا `DateTime` است یا خیر. سپس سایر بررسی‌ها صورت گیرد.

نظرات خوانندگان

نویسنده: رضا نباتی
تاریخ: ۲۱:۲۳ ۱۳۹۱/۰۸/۲۶

به نظر بنده بهترین راه برای گرفتن خروجی دلخواه ارسال فرمت استرینگ DateTime برای نمایش آن در خروجی می باشد.

```
DateTime thisDate1 = new DateTime(2011, 6, 10);  
Console.WriteLine("Today is " + thisDate1.ToString("MMMM dd, yyyy") + ".");
```

نویسنده: بهروز راد
تاریخ: ۸:۲۷ ۱۳۹۱/۰۸/۲۷

البته ابزار Code Analysis موجود در Visual Studio Ultimate هم از FxCop در پشت صحنه استفاده می کنه و این موارد رو گوشزد می کنه.

نویسنده: میثم جوادی
تاریخ: ۱۲:۷ ۱۳۹۱/۰۸/۲۷

من هم یه بار به همچین مشکلی ، شبیه به این برخوردی بودم
[توضیحات](#)

نویسنده: صابر فتح الهی
تاریخ: ۱۴:۱۴ ۱۳۹۱/۰۸/۲۷

همونطوری که مهندس توی پست شون ذکر کردن نسخه جدید reSharper هم این عمل گوشزد می کنه

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 فرمت‌های‌های 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, اسفند, 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 تاریخ. 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

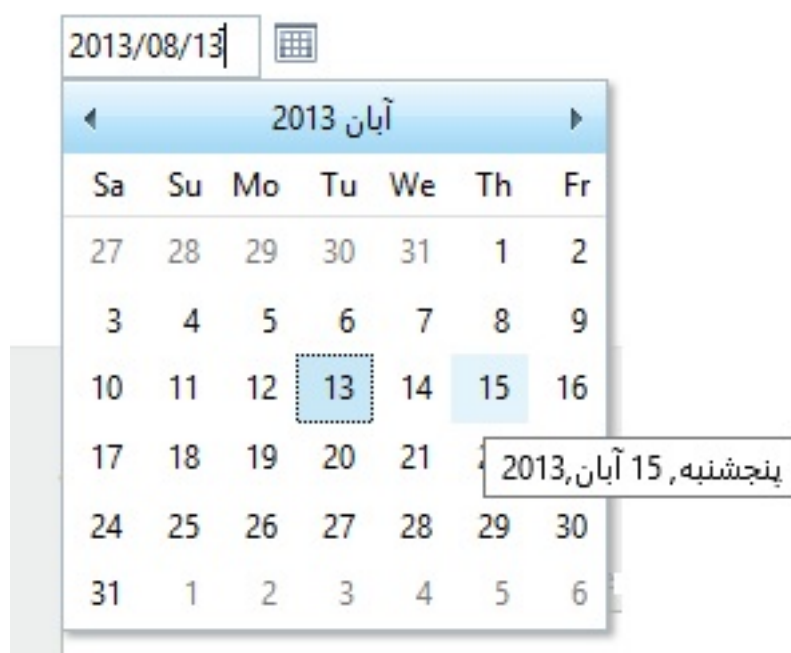
سلام

آقای محسن خان درست میگویند 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: ۱۳۹۲/۱۱/۱۸ ۱۵:۱۳

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: ۱۳۹۳/۰۱/۰۲ ۱۵:۲۰

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.

نویسنده: م علیزاده
تاریخ: ۱۳۹۴/۰۸/۰۷ ۰:۱۳

با سلام،
من هم با این مشکل رو به رو شده ام. وقتی مدلی که یک فیلد تاریخ Required دارد رو به ویو میفرستم این خطا رو میده .
میخواستم بدونم راه حلی برای این مشکل پیدا کردید؟ ممنون

نویسنده: محسن خان
تاریخ: ۱۳۹۴/۰۸/۰۷ ۰:۵۸

DateTime در دات نت یک value type هست و نال قبول نمیکنه؛ مگر اینکه ? DateTime تعریف بشه.

سلام فرقی نمی‌کنه کلاس را کجا قرار بدید. این کلاس ایستا حاوی یک extension method هست که با افزودنش به پروژه به طور خودکار نوع داده datetime آن را خواهد شناخت. توصیه میکنم مقالات مربوطه به اکستنشن متد را بخوانید تا با نحوه کار آن آشنا شوید.

فایل global.asax مهم نیست از هر جا به بعد که فرهنگ ترد جاری تغییر کند نوع خروجی datetime متفاوت خواهد شد ولی میگوییم global.asax چون توانایی اجرا در اولین فرصت را دارد یعنی زمانی که برنامه شروع به کار میکند. الزامی هم به پروژه وب نیست بلکه میتواند در Main یک اپلیکیشن ویندوزی هم اینکار را انجام داد.

همانطور که در توضیح پروژه [PersianDateTime](#) آمده است، کلاس PersianDateTime جایگزینی است برای System.DateTime برای استفاده در پروژه‌هایی که احتیاج به تاریخ شمسی و ساعت رسمی ایران یا سایر کشورهای فارسی‌زبان، مستقل از Time Zone سیستم و در نظر گرفتن Daylight Saving Time، دارند. این کلاس شامل اکثر متدها، پراپرتی‌ها و عملگرهای متداول System.DateTime است.

دسترسی به تاریخ و ساعت فعلی :

```
PersianDateTime now = PersianDateTime.Now;

string persianDateTime = now.ToString(); // 1392/03/09 23:37:57
string persianDate = now.ToString(PersianDateTimeFormat.Date); // 1392/03/09
string persianFullDateTime = now.ToString("dddd d MMMM yyyy ساعت hh:mm:ss tt"); // پنج شنبه 9 خرداد
1392 ساعت 11:37:57 ب.ظ

TimeSpan persianTime = now.TimeOfDay; // 23:37:57.4641984
```

نحوه محاسبه PersianDateTime.Now بر اساس فیلد استاتیک PersianDateTime.Mode است که مقدار پیش‌فرض آن PersianDateTimeMode.UtcOffset است.

PersianDateTime.Mode را می‌توان یکی از سه مقدار زیر قرار داد :

System : بر اساس تاریخ و زمان سیستم (Time Zone فعلی سیستم)

PersianTimeZoneInfo : بر اساس Time Zone تعیین شده در فیلد استاتیک PersianDateTime.PersianTimeZoneInfo (مستقل از Time Zone سیستم)

UtcOffset : بر اساس اختلاف ساعت با UTC، مشخص شده در فیلد استاتیک PersianDateTime.OffsetFromUtc (مستقل از Time Zone سیستم)

مقدار پیش‌فرض PersianDateTime.PersianTimeZoneInfo برابر Iran Standard Time Zone است. توجه داشته باشید که در این حالت از DaylightSavingTime تعیین شده در Time Zone استفاده خواهد شد که مثلاً برای ایران با زمان واقعی آن اختلاف دارد و [باید آنرا اصلاح کرد](#).

مقدار پیش‌فرض PersianDateTime.OffsetFromUtc برابر 3 ساعت و نیم است. در این حالت DaylightSavingTime با توجه به مقادیر سه فیلد استاتیک DaylightSavingTimeEnd، DaylightSavingTimeStart و DaylightSavingTime تعیین می‌شود که به صورت پیش‌فرض برابر 24 اول فروردین، ساعت 24 سی‌ام شهریور و یک ساعت است.

تمام فیلدهای استاتیک ذکر شده به صورت public تعریف شده‌اند تا برنامه‌نویسان سایر کشورهای فارسی‌زبان بتوانند به دلخواه خود آنرا تغییر دهند.

نحوه کار با مقادیر رشته‌ای تاریخ شمسی هم اینگونه است :

```
PersianDateTime persianDate1 = PersianDateTime.Parse("1392/03/02");
PersianDateTime persianDate2 = PersianDateTime.Parse("1392/03/02", "23:32:56");
```

چند سازنده هم وجود دارد برای کسانی که تاریخ را به صورت int و ساعت را int یا short (بدون ثانیه) ذخیره می‌کنند :

```
PersianDateTime persianDate1 = new PersianDateTime(13920310);
PersianDateTime persianDate2 = new PersianDateTime(13920310, 233256);
PersianDateTime persianDate3 = new PersianDateTime(13920310, (short)2332);
```

تبدیل تاریخ میلادی به شمسی :

```
DateTime miladiDate = new DateTime(2013, 5, 31);
PersianDateTime persianDate = new PersianDateTime(miladiDate);
```

تبدیل تاریخ شمسی به میلادی :

```
PersianDateTime persianDate = PersianDateTime.Parse("1392/03/02");
DateTime miladiDate = persianDate.ToDateTime();
```

علاوه بر متد ToString معمولی، دو overload دیگر از این متد برای نمایش فرمت‌های مختلف PersianDateTime وجود دارد :

```
public string ToString(PersianDateTimeFormat format);
public string ToString(string format);
```

که اولی برای فرمت‌های معمول و دومی برای هر نوع فرمت دلخواه قابل استفاده است که چند نمونه آنرا در قسمت تعیین تاریخ و ساعت فعلی دیدید.

برخی از خصوصیات کلاس PersianDateTime :

Year : سال

Month : ماه

Day : روز

TimeOfDay : زمان سپری شده از ابتدای روز

TimeOfWeek : زمان سپری شده از ابتدای هفته

TimeOfMonth : زمان سپری شده از ابتدای ماه

TimeOfYear : زمان سپری شده از ابتدای سال

DaysInYear : تعداد روز سال

DaysInMonth : تعداد روز ماه

DayOfWeek : چندمین روز هفته

DayOfYear : چندمین روز سال

DayName : نام روز

MonthName : نام ماه

Date : تاریخ بدون زمان

FirstDayOfWeek : اولین روز هفته

LastDayOfWeek : آخرین روز هفته

FirstDayOfMonth : اولین روز ماه

LastDayOfMonth : آخرین روز ماه

FirstDayOfYear : اولین روز سال

LastDayOfYear : آخرین روز سال

برخی دیگر از متدهای کلاس PersianDateTime :

```
public PersianDateTime Add(TimeSpan value);  
public PersianDateTime AddDays(double value);  
public PersianDateTime AddMonths(int months);  
public PersianDateTime AddYears(int value);  
public PersianDateTime AddHours(double value);  
public PersianDateTime AddMinutes(double value);  
public PersianDateTime AddSeconds(double value);
```

همچنین تمام عملگرهای +, -, <, >, <=, >=, ==, و != قابل استفاده هستند.

نظرات خوانندگان

نویسنده: م شرافتی
تاریخ: ۱۰:۳۱ ۱۳۹۲/۱۱/۲۱

با سلام .
اگه تاریخ شمسی را مثلا با این فرمت

```
string persianFullDateTime = now.ToString( "dddd d MMMM yyyy ساعت hh:mm:ss tt" ); //
```

در دیتابیس ذخیره کنیم. بعد برای sort کردن برحسب تاریخ به مشکل بر نمی‌خوریم؟

نویسنده: محسن خان
تاریخ: ۱۰:۴۵ ۱۳۹۲/۱۱/۲۱

این فرمت خاص کاربرد نمایشی داره بیشتر.

نویسنده: سارا محمدی
تاریخ: ۹:۲۷ ۱۳۹۲/۱۲/۱۳

با سلام یه فیلد تاریخ دارم وقتی با code first و seeed به این فیلد مقدار میدم پایگاه داده آپدیت نمی‌شود
و در متد up این فیلد تعریف نشده چطوری این فیلد را تعریف کنم

نویسنده: محسن خان
تاریخ: ۹:۳۲ ۱۳۹۲/۱۲/۱۳

از کتابخانه PersianDateTime استفاده کردید؟ [چه خطایی گرفتید](#)؟ PersianDateTime به عنوان نوع خاصیت تعریف شده؟ معادلی
نداره سمت بانک اطلاعاتی.

نویسنده: سارا محمدی
تاریخ: ۹:۴۰ ۱۳۹۲/۱۲/۱۳

ممنون

بعد از اجرای add-migration initial در اجرای دستور update-database خطای زیر را داد
An error occurred while preparing the command definition. See the inner exception for detail
و در کلاس initial در متد UP تعریفی برای این فیلد وجود ندارد

```
public PersianDateTime EnrollmentDate { get; set; }
```

نویسنده: محسن خان
تاریخ: ۹:۵۵ ۱۳۹۲/۱۲/۱۳

تعریفی نداره چون معادلی نداره در بانک اطلاعاتی. EF فیلد شما را باید بتونه به یک نوع خاص در بانک اطلاعاتی نگاشت کنه. چنین
نوعی در سمت بانک اطلاعاتی وجود خارجی نداره. البته در SQL Server میشه [نوع جدید تعریف کرد](#) . بعد از تعریف، این طرف با
Column typeName باید مشخص بشه.

نویسنده: مهدی
تاریخ: ۹:۴۷ ۱۳۹۳/۰۳/۳۰

```
using (var db = new DBTestEntities())  
{
```



```

dataGridView1.DataSource =(from t in db.test
                           select new
                           {
                               Id= t.Id,
                               time = new PersianDateTime(
DateTime.Parse(t.ResponseDate.ToString())).ToString(PersianDateTimeFormat.DateShortTime)
                           }).ToList();
}

```

برای ذخیره اطلاعات در بانک این کلاس خوب کار میکند و مشکلی نیست . ولی برای نمایش در برنامه ویندوزی کد بالا عمل نمیکند و هیچ چیزی نشان نمی‌دهد . یک جدول ساده داریم که شامل دو فیلد Id و datetime است .

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۲ ۱۳۹۳/۰۳/۳۰

قسمت‌های مختلف ورودی را باید مجزا دیباگ کنید و سپس نتیجه را بررسی کنید. خصوصا این نکته « [از متد DateTime.ToString بدون پارامتر استفاده نکنید!](#) »

نویسنده: مهدی
تاریخ: ۱۵:۳۶ ۱۳۹۳/۰۳/۳۰

این تمام کد برنامه است :

```

using System;
using System.Linq;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // این قسمت برای ورود اطلاعات به بانک است و با کمک کتابخانه پرتن دات نت تاریخ شمسی را به میلادی تبدیل و ذخیره میکنم
            using(var db = new h7Entities())
            {
                var t = new test
                {
                    ResponseDate = PersianDateTime.Parse(textBox1.Text).ToDateTime()
                };
                db.test.Add(t);
                db.SaveChanges();
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // این قسمت هم فقط اطلاعات واکنشی شده را در گرید نمایش میدهد . بدون هیچ شرطی ، یک سلکت ساده . . فقط از پرتن دانت نت برای تبدیل میلادی به شمسی کمک میگیرم
            using (var db = new h7Entities())
            {
                dataGridView1.DataSource =(from t in db.test
                                         select new
                                         {
                                             Id= t.Id,
                                             time = new
PersianDateTime(DateTime.Parse(t.ResponseDate.ToString())).ToString(PersianDateTimeFormat.DateShortTime
)
                                         }).ToList();
            }
        }
    }
}

```

کتابخانه PersianDateTime را از نیوگت دریافت کردم .
ولی چیزی در گرید نمایش نمیدهد .

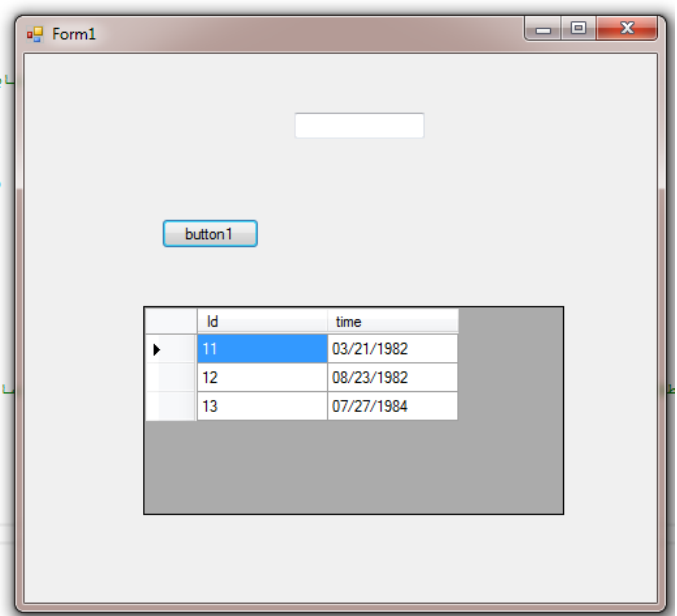
مدل برنامه هم :

```
public partial class test
{
    public int Id { get; set; }
    public Nullable<System.DateTime> ResponseDate { get; set; }
}
```

سوال دیگه اینکه وقتی تبدیلی انجام نمی‌شود ، خروجی زیر را دارم :

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    // پایگاه پرژن دات نت تاریخ شمسی را به میلادی تبدیل و ذخیره میکنم
    using (var db = new h7Entities())
    {
        var t = new test
        {
            ResponseDate = PersianDateTime.Parse(textBox1.Text).ToDateTime()
        };
        db.test.Add(t);
        db.SaveChanges();
    }
}

1 reference
private void Form1_Load(object sender, EventArgs e)
{
    // داده ... فقط از پرژن دات نت برای تبدیل میلادی به شمسی کمک میگیرم
    using (var db = new h7Entities())
    {
        dataGridView1.DataSource = (from t in db.test
                                    select new
                                    {
                                        Id = t.Id,
                                        time = t.ResponseDate
                                    }).ToList();
    }
}
```



حالا چطور از فیلدی که تاریخ را نمایش میدهد فقط آن را تبدیل به شمسی و نمایش دهد ؟ شبیه این 02/02/1365 ؟
تشکر

نویسنده: وحید نصیری
تاریخ: ۱۷:۷ ۱۳۹۳/۰۳/۳۰

کلاس [PersianDateTime](#) سازنده‌های متعددی دارد. از سازنده‌ای که نوع DateTime را مستقیماً دریافت می‌کند، استفاده کنید:

```
new PersianDateTime(t.ResponseDate.Value).ToString(PersianDateTimeFormat.DateShortTime)
```

یعنی با وجود این سازنده، نیازی به [ToString](#) و سپس [DateTime.Parse](#) آن نیست:

```
/// <summary>
/// Initializes a new instance of the PersianDateTime class to a specified dateTime.
/// </summary>
/// <param name="dateTime">A date and time in the Gregorian calendar.</param>
```

```
public PersianDateTime(DateTime dateTime)
{
    _dateTime = dateTime;
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۳۰ ۱۹:۲۷

یک مثال ساده بر مبنای کدهای شما: [WindowsFormsTests.zip](#)

نویسنده: mmb462
تاریخ: ۱۳۹۳/۰۴/۱۵ ۱۳:۱۴

با سلام و تشکر
چگونه یک تاریخ معتبر را بررسی کنیم؟

مقدمه

در حالت پیشرفته‌ی تزریق وابستگی‌ها در دات نت، با توجه به اینکه کار وهله سازی کلاس‌ها به یک کتابخانه جانبی به نام [IoC Container](#) واگذار می‌شود، امکان یک سری دخل و تصرف نیز در این میان فراهم می‌گردد. برای مثال الان که ما می‌توانیم یک کلاس را توسط IoC container به صورت خودکار وهله سازی کنیم، خوب، چرا اجرای متدهای آن‌را تحت نظر قرار ندهیم. مثلاً حاصل آن‌ها را بتوانیم پیش از اینکه به فراخوان بازگشت داده شود، کش کنیم یا کلاً تغییر دهیم. به این کار [AOP](#) یا Aspect oriented programming نیز گفته می‌شود.

واقعیت این است که یک چنین مفهومی از سال‌های دور به نام [Hooking](#) در برنامه‌های WIN32 API خالص نیز وجود داشته است. Hook ها یا قلاب‌ها دقیقاً کار Interception دنیای AOP را انجام می‌دهند. به این معنا که خودشان را بجای یک متد ثبت کرده و کار ردیابی یا حتی تغییر عملکرد آن متد خاص را می‌توانند انجام دهند. برای مثال اگر برای متد [gethostbyname](#) ویندوز یک Hook بنویسیم، برنامه استفاده کننده، تنها متد ما را بجای متد اصلی [gethostbyname](#) واقع در Kerne132 ویندوز، مشاهده می‌کند و درخواست‌های DNS خودش را به این متد ویژه ما ارسال خواهد کرد؛ بجای ارسال درخواست‌ها به متد اصلی. در این بین می‌توان درخواست‌های DNS را لاگ کرد و یا حتی تغییر جهت داد.

انجام Interception در دنیای دات نت با استفاده از امکانات Reflection و [Reflection.Emit](#) قابل انجام است و یا حتی بازنویسی اسمبلی‌ها و افزودن کدهای IL مورد نیاز به آن‌ها که به آن IL Weaving هم گفته می‌شود. اما در دنیای WIN32 انجام چنین کاری ساده نیست و ترکیبی است از زبان اسمبلی و کتابخانه‌های نوشته شده به زبان C.

برای ساده سازی نوشتن Hook های ویندوزی، کتابخانه‌ای به نام [easy hook](#) ارائه شده است که امکان تزریق Hook های دات نت را به درون پروسه برنامه‌های Native ویندوز دارد. این قلاب‌ها که در اینجا متدهای دات نت هستند، نهایتاً بجای توابع اصلی ویندوز جا زده خواهند شد. بنابراین می‌توانند عملیات آن‌ها را ردیابی کنند و یا حتی پارامترهای آن‌ها را دریافت و مقدار دیگری را بجای تابع اصلی، بازگشت دهند. در ادامه قصد داریم اصول و نکات کار با [easy hook](#) را در طی یک مثال بررسی کنیم.

صورت مساله

می‌خواهیم کلیه درخواست‌های تاریخ اکسپلورر ویندوز را ردیابی کرده و بجای ارائه تاریخ استاندارد میلادی، تاریخ شمسی را جایگزین آن کنیم.

از کجا شروع کنیم؟

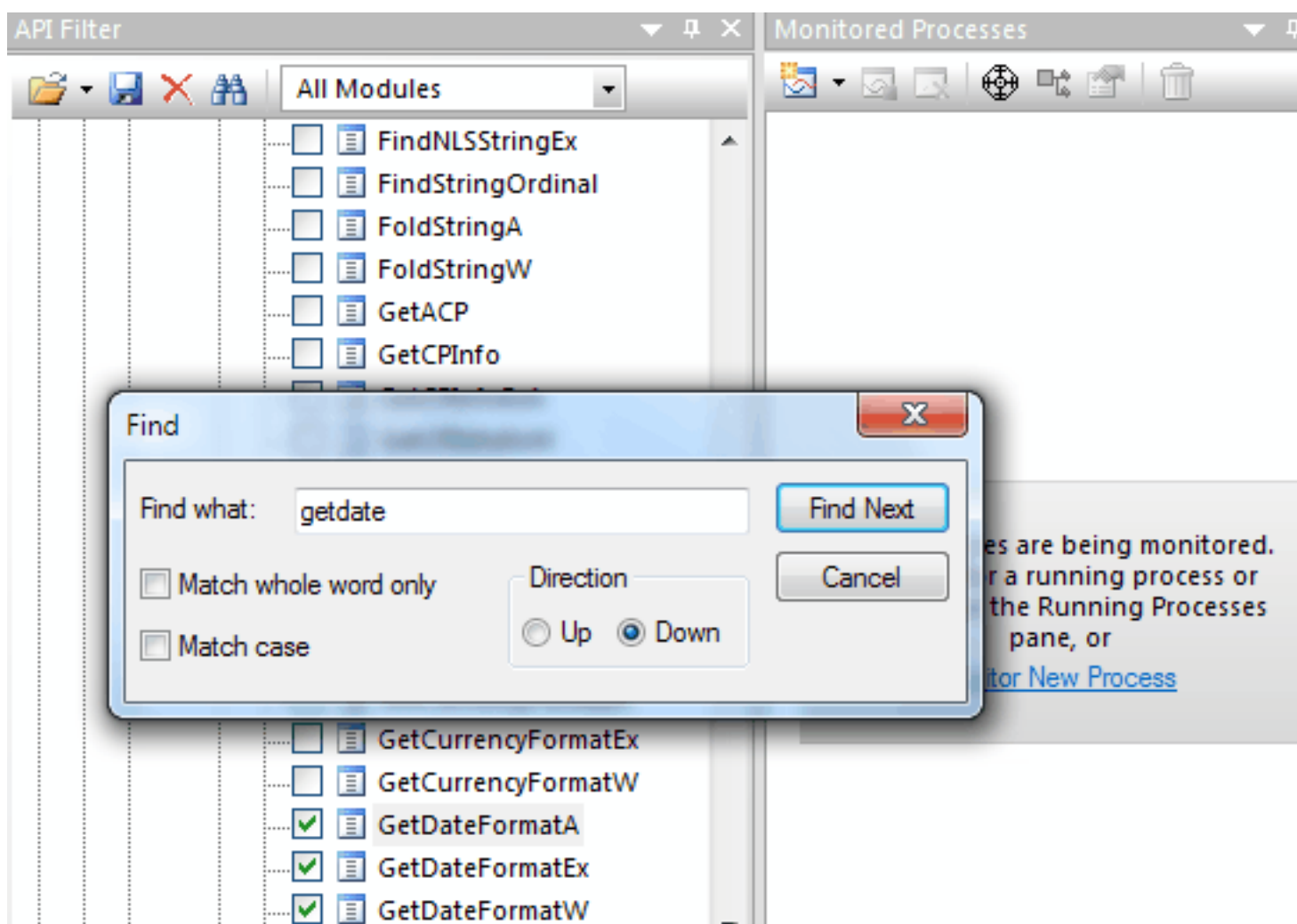
ابتدا باید دریابیم که اکسپلورر ویندوز از چه توابع API ای برای پردازش‌های درخواست‌های تاریخ و ساعت خودش استفاده می‌کند، تا بتوانیم برای آن‌ها Hook بنویسیم. برای این منظور می‌توان از برنامه‌ی بسیار مفیدی به نام API Monitor استفاده کرد. این برنامه‌ی رایگان را از آدرس ذیل می‌توانید دریافت کنید:

<http://www.rohitab.com/apimonitor>

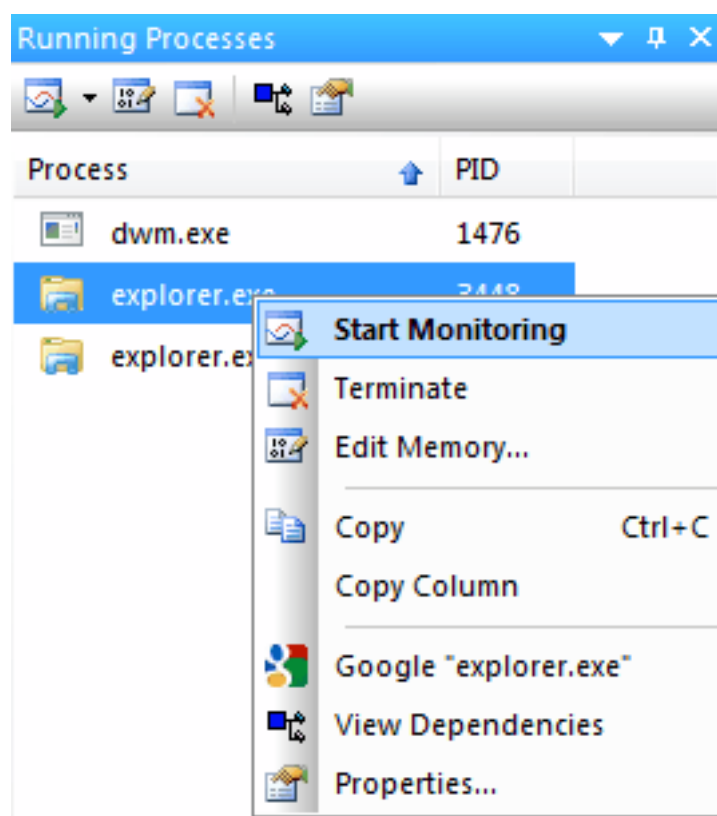
اگر علاقمند به ردیابی برنامه‌های 32 بیتی هستید باید [apimonitor-x86.exe](#) را اجرا کنید و اگر نیاز به ردیابی برنامه‌های 64 بیتی دارید باید [apimonitor-x64.exe](#) را اجرا نمایید. بنابراین اگر پس از اجرای این برنامه، برای مثال فایرفاکس را در لیست پروسه‌های آن مشاهده نکردید، یعنی [apimonitor-x64.exe](#) را اجرا کرده‌اید؛ از این جهت که فایرفاکس عمومی تا این تاریخ، نسخه 32 بیتی است و نه 64 بیتی.

پس از اجرای برنامه API Monitor، در قسمت API Filter آن باید مشخص کنیم که علاقمند به ردیابی کدامیک از توابع API ویندوز هستیم. در اینجا چون نمی‌دانیم دقیقاً کدام تابع کار ارائه تاریخ را به اکسپلورر ویندوز عهده دار است، شروع به جستجو می‌کنیم و هر تابعی را که نام date یا time در آن وجود داشت، تیک می‌زنیم تا در کار ردیابی لحاظ شود.

تغییر عملکرد و یا ردیابی توابع ویندوز با استفاده از Hook های دات نت



سپس نیاز است بر روی نام اکسپلورر در لیست پروسه‌های این برنامه کلیک راست کرده و گزینه Start monitoring را انتخاب کرد:



اندکی صبر کنید یا یک صفحه جدید اکسپلورر ویندوز را باز کنید تا کار ردیابی شروع شود:

تغییر عملکرد و یا ردیابی توابع ویندوز با استفاده از Hook های دات نت

Monitored Processes

C:\Windows\explorer.exe - PID: 3448

C:\Windows\explorer.exe - PID: 3592

Summary | 8 calls | 5 KB used | explorer.exe

#	Time of Day	Thread	Module	API
1	12:30:08.299 PM	1	timedate.cpl	GetDateFormatW
2	12:30:08.299 PM	1	timedate.cpl	GetDateFormatW
3	12:30:08.549 PM	1	timedate.cpl	GetDateFormatW
4	12:30:42.391 PM	1	timedate.cpl	GetDateFormatW
5	12:30:42.391 PM	1	timedate.cpl	GetDateFormatW
6	12:31:00.854 PM	1	explorer.exe	GetTimeFormatW

Parameters: GetDateFormatW (Kernel32.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	LCID	Locale	LOCALE_USER_DEFAULT	LOCALE_USER_DEFAULT
2	DWORD	dwFlags	DATE_LONGDATE 64	DATE_LONGDATE 64
3	const SYSTEMTI...	lpDate	NULL	NULL
4	LPCTSTR	lpFormat	NULL	NULL
5	LPTSTR	lpDateStr	0x00000000336e7f0	0x00000000336e7f0
	TCHAR		Not Available	Thursday, October 03, 2013
6	int	cchDate	256	256

int

Return

31

همانطور که مشاهده می‌کنید، ویندوز برای ردیابی تاریخ در اکسپلورر خودش از توابع GetDateFormatW و GetTimeFormatW استفاده می‌کند. ابتدا یک تاریخ را توسط آرگومان lpDate یا lpTime به یکی از توابع یاد شده ارسال کرده و سپس خروجی را از آرگومان lpDateStr یا lpTimeStr دریافت می‌کند. خوب؛ به نظر شما اگر این خروجی‌ها را با یک ساعت و تاریخ شمسی جایگزین کنیم بهتر نیست؟!

نوشتن Hook برای توابع GetTimeFormatW و GetDateFormatW ویندوز اکسپلورر

ابتدا کتابخانه easy hook را از مخزن کد CodePlex آن دریافت کنید:

<https://easyhook.codeplex.com>

سپس یک پروژه کنسول ساده را آغاز کنید. همچنین به این Solution، یک پروژه Class library جدید را نیز اضافه نمایید. پروژه کنسول، کار نصب Hook را انجام می‌دهد و پروژه کتابخانه‌ای اضافه شده، کار مدیریت هوک‌ها را انجام خواهد داد. سپس به هر دو پروژه، ارجاعی را به اسمبلی EasyHook.dll اضافه کنید.

الف) ساختار کلی کلاس Hook

کلاس Hook واقع در پروژه Class library باید یک چنین امضایی را داشته باشد:

```
namespace ExplorerPCal.Hooks
{
    public class GetDateTimeFormatInjection : IEntryPoint
    {
        public GetDateTimeFormatInjection(RemoteHooking.IContext context, string channelName)
        {
        }
    }
}
```

```
// connect to host...
_interface = RemoteHooking.IpcConnectClient<MessagesReceiverInterface>(channelName);
_interface.Ping();
}

public void Run(RemoteHooking.IContext context, string channelName)
{
}
}
```

یعنی باید اینترفیس IEntryPoint کتابخانه easy hook را پیاده سازی کند. این اینترفیس خالی است و صرفاً کار علامتگذاری کلاس Hook را انجام می‌دهد. همچنین این کلاس باید دارای سازنده‌ای با امضایی که ملاحظه می‌کنید و یک متد Run، دقیقاً با همین امضای فوق باشد.

ب) نوشتن توابع Hook

کار نوشتن قلاب برای توابع API ویندوز در متد Run انجام می‌شود. سپس باید توسط متد LocalHook.Create کار را شروع کرد. در اینجا مشخص می‌کنیم که نیاز است تابع GetDateFormatW واقع در kernel32.dll ردیابی شود.

```
public void Run(RemoteHooking.IContext context, string channelName)
{
    GetDateFormatHook = LocalHook.Create(
        InTargetProc: LocalHook.GetProcAddress("kernel32.dll",
"GetDateFormatW"),
        InNewProc: new GetDateFormatDelegate(getDateFormatInterceptor),
        InCallback: this);
}
```

در ادامه توسط یک delegate، کلیه فراخوانی‌های ویندوز را که قرار است به GetDateFormatW اصلی ارسال شوند، ردیابی کرده و تغییر می‌دهیم.

ج) نحوه مشخص سازی امضای delegateهای Hook

اگر امضای متد GetDateFormatW به نحو ذیل باشد:

```
[DllImport("kernel32.dll", CallingConvention = CallingConvention.StdCall, CharSet = CharSet.Auto,
SetLastError = true)]
public static extern int GetDateFormatW(
    uint locale,
    uint dwFlags, // NLS_DATE_FLAGS
    SystemTime lpDate,
    [MarshalAs(UnmanagedType.LPWStr)] string lpFormat,
    StringBuilder lpDateStr,
    int sbSize);
```

دقیقاً delegate متناظر با آن نیز باید ابتدا توسط ویژگی [UnmanagedFunctionPointer](#) مزین شده و آن نیز دارای امضایی همانند تابع API اصلی باشد:

```
[UnmanagedFunctionPointer(CallingConvention.StdCall, CharSet = CharSet.Auto, SetLastError = true)]
private delegate int GetDateFormatDelegate(
    uint locale,
    uint dwFlags,
    SystemTime lpDate,
    [MarshalAs(UnmanagedType.LPWStr)] string lpFormat,
    StringBuilder lpDateStr,
    int sbSize);
```

سپس callback نهایی که کار دریافت پیام‌های ویندوز را انجام خواهد داد نیز، همان امضاء را خواهد داشت:

```
private int getDateFormatInterceptor(
    uint locale,
    uint dwFlags,
    SystemTime lpDate,
```



```
string lpFormat,  
StringBuilder lpDateStr,  
int sbSize)  
{  
}
```

در اینجا برای تغییر فرمت تاریخ ویندوز تنها کافی است lpDateStr را مقدار دهی کنیم. ویندوز lpDate و سایر پارامترها را به این متد ارسال می‌کند؛ در اینجا فرصت خواهیم داشت تا بر اساس این اطلاعات، lpDateStr صحیحی را تولید و مقدار دهی کنیم.

د) نصب Hook نوشته شده

باید دقت داشت که هر دو برنامه نصاب Hook و همچنین کتابخانه Hook، باید دارای امضای دیجیتالی باشند. بنابراین به برگه signing خواص پروژه مراجعه کرده و یک فایل snk را به هر دو پروژه اضافه نمایید. سپس در برنامه نصاب، یک کلاس را با امضای ذیل تعریف کنید:

```
public class MessagesReceiverInterface : MarshalByRefObject  
{  
    public void Ping()  
    {  
    }  
}
```

این کلاس با استفاده از امکانات Remoting دات نت، پیام‌های دریافتی از هوک دات نتی تزریق شده به درون یک پروسه دیگر را دریافت می‌کند.

سپس در ابتدای برنامه نصاب، یک کانال Remoting باز شده (که آرگومان جنریک آن دقیقاً همین نام کلاس MessagesReceiverInterface فوق را دریافت می‌کند)

```
var channel = RemoteHooking.IpcCreateServer<MessagesReceiverInterface>(ref _channelName,  
WellKnownObjectMode.SingleCall);
```

و سپس توسط متد RemoteHooking.Inject کار تزریق ExplorerPCal.Hooks.dll به درون پروسه اکسپلورر ویندوز انجام می‌شود:

```
RemoteHooking.Inject(  
    explorer.Id,  
    InjectionOptions.Default | InjectionOptions.DoNotRequireStrongName,  
    "ExplorerPCal.Hooks.dll", // 32-bit version (the same, because of using AnyCPU)  
    "ExplorerPCal.Hooks.dll", // 64-bit version (the same, because of using AnyCPU)  
    _channelName  
);
```

پارامتر اول متد RemoteHooking.Inject، شماره PID یک پروسه است. این شماره را از طریق متد Process.GetProcesses می‌توان بدست آورد. سپس یک سری پیش فرض مشخص می‌شوند و در ادامه مسیر کامل دو DLL هوک دات نتی باید مشخص شوند. چون تنظیمات پروژه هوک را بر روی [Any CPU](#) قرار داده‌ایم، فقط کافی است یک نام DLL را برای هوک‌های 64 بیتی و 32 بیتی ذکر کنیم.

پارامتر و پارامترهای بعدی، اطلاعاتی هستند که به سازنده کلاس هوک ارسال می‌شوند. بنابراین این سازنده می‌تواند تعداد پارامترهای متغیری داشته باشد:

```
.ctor(IContext, %ArgumentList%)  
void Run(IContext, %ArgumentList%)
```

چند نکته تکمیلی مهم برای کار با کتابخانه Easy hook
- کتابخانه easy hook فعلاً با ویندوز 8 سازگار نیست.

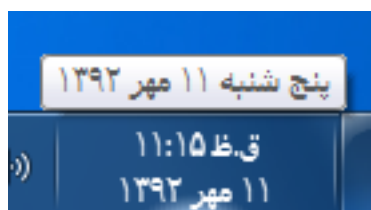
- برای توزیع هوک‌های خود باید تمام فایل‌های همراه کتابخانه easy hook را نیز توزیع کنید و فقط به چند DLL ابتدایی آن بسنده نباید کرد.
- اگر هوک شما بلافاصله سبب کرش پروسه هدف شد، یعنی امضای تابع API شما ایراد دارد و نیاز است چندین و سایت را جهت یافتن امضایی صحیح بررسی کنید. برای مثال در [امضای عمومی](#) متد `GetDateFormatW`، پارامتر `SystemTime` به صورت `struct` تعریف شده است؛ درحالیکه ویندوز ممکن است برای دریافت زمان جاری به این پارامتر نال ارسال کند. اما `struct` دات نت برخلاف `struct` زبان C یک `value type` است و نال پذیر نیست. به همین جهت کلیه امضاها عمومی که در مورد این متد در اینترنت یافت می‌شوند، در عمل غلط هستند و باید `SystemTime` را یک کلاس دات نت که `Refrence type` است، تعریف کرد تا نال پذیر شود و hook کرش نکرده یا اشتباه عمل نکند.
- زمانیکه یک هوک easy hook بر روی پروسه هدف نصب می‌شود، دیگر قابل `unload` کامل نیست و نیاز است برای کارهای برنامه نویسی و به روز رسانی فایل `dll` جدید، پروسه هدف را خاتمه داد.
- متد `Run` هوک باید همیشه در حال اجرا باشد تا توسط CLR بلافاصله خاتمه نیافته و هوک از حافظه خارج نشود. اینکار را توسط روش ذیل انجام دهید:

```
try
{
    while (true)
    {
        Thread.Sleep(500);
        _interface.Ping();
    }
}
catch
{
    _interface = null;
    // .NET Remoting will raise an exception if host is unreachable
}
```

- تا زمانیکه برنامه نصاب هوک که توسط Remoting دات نت، کانالی را به این هوک گشوده است، باز است، حلقه فوق اجرا می‌شود. با بسته شدن برنامه نصاب، متد `Ping` دیگر قابل دستیابی نبوده و بلافاصله این حلقه خاتمه می‌یابد.
- استفاده همزمان از `API Monitor` ذکر شده در ابتدای بحث و یک هوک نصب شده، سبب کرش برنامه هدف خواهد شد.

سورس کامل این پروژه را در اینجا می‌توانید دریافت کنید

[شمسی ساز تاریخ اکسپلورر ویندوز](#)



نظرات خوانندگان

نویسنده: میثم هوشمند
تاریخ: ۱۳۹۲/۰۷/۱۲ ۰:۴۳

ممکن هست که یک مثال برای لاگ کردن درخواستها ارسالی به یک متد مانند همان متد gethostbyname ارائه نمایید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۱۲ ۱:۰۰

سورس پروژه قابل دریافت هست و به عنوان یک قالب برای این نوع کارها میشه ازش استفاده کرد.
برای مثال در این قالب جایی که هوک تعریف میشه به این صورت تغییر خواهد کرد:

```
_getHostByNameHook = LocalHook.Create(  
    InTargetProc: LocalHook.GetProcAddress("ws2_32.dll",  
    "gethostbyname"),  
    InNewProc: new GetHostByNameDelegate(getHostByNameHooked),  
    InCallback: this);
```

delegate آن چنین تعریفی خواهد داشت (توضیحات آن در متن فوق هست؛ قسمت (ج) نوشتن هوک):

```
[UnmanagedFunctionPointer(CallingConvention.StdCall, CharSet = CharSet.Ansi, SetLastError = false)]  
private delegate IntPtr GetHostByNameDelegate(string name);
```

و جایی که نهایتا درخواستهای DNS دریافت می‌شوند به صورت زیر تعریف خواهد شد:

```
private static IntPtr getHostByNameHooked(string name)  
{  
    // redirect ...  
    //if (name.StartsWith("www.google.com"))  
    //{  
    //    return Native.GetHostByName("127.0.0.1");  
    //}  
    return Native.gethostbyname(name);  
}
```

در اینجا می‌شود درخواستهای DNS را تغییر جهت داد (مثلا گوگل را بلاک کرد)، یا همان تابع اصلی ویندوز را اجرا کرد و یا name دریافتی را در یک فایل مثلا لاگ کرد.

نویسنده: علی قمشلویی
تاریخ: ۱۳۹۲/۰۷/۱۲ ۱۰:۳۵

با سلام و تشکر
این آموزش یکی ناب‌ترین آموزش‌ها می‌باشد چون من مدتی با easyhook کار کردم و کمبود آموزش‌های فارسی در رابطه با هوک بسیار چشمگیر بود.
در رابطه با تغییر تاریخ در شیرپوینت شما چه پیشنهادی دارید؟ چون تحقیقات بنده در این رابطه به جایی نرسید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۱۲ ۱۱:۴۴

من روی تاریخ شیرپوینت وقت نگذاشتم ولی به نظر « [OnetUtil.dll](#) » رو باید مدنظر داشته باشید.

نویسنده: محبوه محمدی

تاریخ: ۱۳۹۲/۰۷/۱۳ ۱۰:۲۳

سلام. ممنون بابت مطلب خوبتون.

Hooking یکی از مفاهیم جالب و خیلی کاربردی هست. یکی از استفاده‌های خوبش زمانیه که یکی از دکمه‌های پر استفاده کیبوردتون خراب شده و امکان تعمیر سخت افزایش وجود نداره. به جای عوض کردن کیبورد مخصوصا وقتی لپ تاپ باشه، به راحتی می‌تونید کیبوردتون رو Hook کنید و یکی از کلیدهای کم مصرف‌تر رو جایگزین کلید خراب کنید!

نویسنده: شمس
تاریخ: ۱۳۹۲/۰۷/۱۴ ۷:۵۸

بسیار جالب بود.
خاطرات گذشته را زنده کرد.

نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۲/۰۷/۱۶ ۹:۱۹

سلام و تشکر
یادمه قبلا گفته بودین توی یکی از مطالبتون [اینجا](#) که من از این روش برای دستکاری تقویم استفاده نمی‌کنم و ایراد داره. خواستم بدونم که الان هم همون نظر رو دارین و این یه مثال هست برای کار با هوک، یا اینکه فکر میکنین این روش خوبی برای فارسی سازی تقویم هست و نظرتون عوض شده؟

نویسنده: محمدرضا دستوری
تاریخ: ۱۳۹۲/۰۷/۱۶ ۹:۲۲

تا به حال همچنین توضیح کاملی و خوبی به همراه یک مثال عالی هیچ جا ندیده بودم و میشه گفت روان‌ترین توضیح هوک رو ارائه دادید
من هم علاقه‌مند شدم یک سری از نرم افزارهای روزمره رو به این ترتیب بومی کنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۰:۰۰

بله. این یک مثال کلی هم هست.
+

- خوب؛ این کار فوق رایگان هست! (و همچنین سورس باز؛ حداقل می‌تونید بررسی کنید که آیا فقط تاریخ پروسه اکسپلورر را هوک کرده یا جاهای دیگری رو هم تحت نظر قرار داده)
- Global hook یا System wide hook نیست. کارهایی که قبلا انجام شدن، کل سیستم رو تحت نظر قرار می‌دن. کار فوق، فقط و فقط پروسه explorer ویندوز رو تحت نظر قرار می‌ده. به همین جهت تاثیر خاصی روی سایر پروسه‌های سیستم نداره. همچنین چون کل سیستم رو مونیتور نمی‌کنه، روی کارایی کلی آن هم تاثیر منفی نخواهد گذاشت. به علاوه روش فوق ساختارهای درونی تاریخ سیستم رو دستکاری نمی‌کنه. فقط رشته نهایی نمایشی (آرگومان‌های lpDateStr یا lpTimeStr) را شمس می‌کنه.

نویسنده: هیمن روحانی
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۲:۲

سلام ممنون از مطلب خوبتون. شما در مورد انجام این کار در دات نت این رو نوشتید (در دنیای دات نت با استفاده از امکانات Reflection قابل انجام است و یا حتی بازنویسی اسمبلی‌ها و افزودن کدهای IL مورد نیاز به آن‌ها که به آن IL Weaving هم گفته می‌شود.). من می‌خوام اسمبلی اصلی شیرپوینت رو در حین اجرا تغییر بدم ولی Reflection سربرار زیادی داره و نمی‌خوام اصل اسمبلی رو تغییر بدم. آیا کتابخانه یا فریم ورکی رو سراغ دارید که بشه کدهای دات نت رو در حین اجرا تغییر بده؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۳:۴۰

Cinject - Code Inject & Runtime Intelligence

Rewrite MSIL Code on the Fly with the .NET Framework Profiling API

نویسنده: هیمن روحانی
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۴:۲۳

Cinject یک اسبلی جدید براتون می‌سازه. ابزارهای دیگه‌ی هستند که این کارو انجام میدن مثل [Reflexil](#) که افزونه Reflector است و ساده‌تر میشه باهاش کار کرد. اکثرا این ابزارها از کتابخانه [Mono.Cecil](#) استفاده میکنن که کارش بازنویسی MSIL است.

برای استفاده از .NET Profiling API باید بتونید با ++c اینترفیس [ICorProfilerCallback2](#) رو پیاده سازی کنید و یک پروفایلر بسازید که من نتونستم انجامش بدم. این مورد واقعا عالی و مزایای زیادی داره ولی متاسفانه پیاده سازیش سخته. دنبال ابزار یا کتابخانه‌ی هستم که از .NET Profiling API استفاده کرده باشه و این روشو ساده کرده باشه.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۶:۳۲

برای یک برنامه وب روش‌های ساده‌تری هم برای نمایش یا دستکاری اطلاعات روی صفحه هست. مثلا مانند مطلب [تغییر تاریخ وبلاگ‌های بلاگر](#) در همین سایت. یا حتی میشه اینکار رو با [افزونه‌های مرورگرها](#) نیز انجام داد. اتفاقا خیلی هم [مرسوم است](#) این روش. این هم [یک مثال خاصش در مورد شیرپوینت](#).

نویسنده: هیمن روحانی
تاریخ: ۱۳۹۲/۰۷/۱۷ ۱۰:۰۶

همه این روش‌ها رو دیدم که بعضی هاشون از جاوااسکریپت هم استفاده میکنند. ولی این روشها مشکلات زیادی دارن. شیرپوینت فقط یک سایت ساده نیست، Object Model داره کلی سرویس داره، فرم اینوپس و... شیرپوینت میتونه همزمان از چند تا تقویم پشتیبانی بکنه. این روشها خیلی سرسری مشکلو حل میکنن درکل نمیشه با این روشها یک راه کار درست حسابی ارائه داد. اگه بشه از .NET Profiling API استفاده کنم میتونم تقویم Exchange Outlook Web App رو هم شمس‌ی کنم.

نویسنده: هیمن روحانی
تاریخ: ۱۳۹۲/۰۷/۱۸ ۱۰:۰۷

خسته نباشد. این API Monitor می‌تونه برنامه‌های دات نت رو مانیتور کنه؟ یا فقط برای کدهای مدیریت نشده کار می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۱۸ ۱۰:۲۸

با برنامه‌های دات نت هم کار می‌کنه. مثلا متد `new WebClient().DownloadData` را در برنامه دات نت خودتون فراخوانی کنید. بعد در برنامه API Monitor تیک قسمت مربوط به شبکه و اینترنت را قرار دهید (تصویر اول مقاله فوق). تمام فراخوانی‌های شبکه برنامه را مونیتور می‌کند. ضمنا برنامه API Monitor قابل بسط است. یعنی اگر به پوشه API آن مراجعه کنید یک سری فایل XML در آن قرار دارد که تعاریف توابع DLLهای مختلف در آن ارائه شده‌اند. اگر تعریفی یا DLL ای در آن نیست، قابل افزودن است. یا حتی اگر خودتان نمی‌توانید اینکار را انجام دهید، فایل هدر آنرا [در انجمن این برنامه](#) ارسال کنید تا به شما کمک کنند.

نویسنده: هیمن روحانی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۱۰:۴۸

در مقدمه این مطلب به تزریق [AOP](#) اشاره کردید و فرمودید Hookها یا قلابها دقیقاً کار Interception دنیای AOP را انجام می‌دهند ولی در AOP همه چیز از مدیریت وهله سازی شروع میشه و اگر امکان مدیریت وهله سازی نباشه همیشه AOP امکان پذیر نیست. در AOP انجام تغییر در کدها فقط در اسمبلی هایی که در پروسه جاری لود شده اند امکان پذیره. اما در Hooking شما پروسه‌های دیگر رو مانیتور و زمان احتیاج نتیجه برگشتی یک تابعو تغییر میدید. قبلاً در همین مطلب این سوالو پرسیدم: [آیا کتابخانه یا فریم ورکی رو سراغ دارید که بشه کدهای دات نت رو در حین اجرا تغییر بده؟](#) بهتره سوالمو اینجوری اصلاح کنم: [آیا کتابخانه یا فریم ورکی رو سراغ دارید که بتونه کدهای دات نت رو در پروسه‌ی دیگر تغییر بده؟](#) آیا با EasyHook میشه همین کارو برای برنامه‌های دات نت در حال اجرا انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۱۱ ۱۳۹۲/۰۷/۲۰

- با easy hook همیشه [JIT Compiler دات نت رو هوک کرد](#). در اینجا فرصت خواهید داشت تا بدنه متد رو با کدهای IL بازنویسی کنید. در مورد JIT Hooking اگر اطلاعات بیشتری خواستید، [اینجا](#)

- [CLR Injection: Runtime Method Replacer](#)

- [NET CLR Injection: Modify IL Code during Run-time.](#)

- [Hawkeye - The .Net Runtime Object Editor](#)

- کار خود مایکروسافت است: [Moles - Replace any .NET method with a delegate](#) البته مایکروسافت برای کارهای Native ویندوز هم یک کتابخانه به نام [detours](#) دارد. نسخه 32 بیتی آن عمومی است و با C سازگار است.

- [Modifying IL at runtime](#)

- [WPF Snoop](#)

نویسنده: reza
تاریخ: ۱۱:۰۰ ۱۳۹۲/۰۸/۲۶

سلام؛ می‌خواهم توی برنامه‌های مختلفی (تمام برنامه‌ها) که روی ویندوز نصب است وقتی یک کلید میانبری زدم مثل alt+shift+r متنی که در حال تایپ هستم یا متنی که داخل textbox ی تایپ کردم، روش تغییراتی انجام بدم چه کاری باید انجام بدم؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۲ ۱۳۹۲/۰۸/۲۶

- easyhook کار global hooking را انجام نمی‌دهد؛ به ازای یک یا چند پروسه مشخص کار می‌کند.

- برای تغییر یک متن در حال نمایش در ویندوز عموماً از هوک کردن متد [ExtTextOutW](#) استفاده می‌شود. خروجی این متد را در ویندوز XP راحت می‌شود تغییر داد. در ویندوز 7 به علت پیشرفته شدن خروجی‌های متنی و پشتیبانی از انواع و اقسام فرهنگ‌های مختلف، خروجی آن به صورت گلیف است و نه متن. یعنی به مقدار پارامتر fuOptions آن باید دقت داشت و اگر مساوی ETO_GLYPH_INDEX بود، تا جایی که [تحقیق کردم](#) قابل برگشت به متن یونیکد اصلی آن نیست (یا کار ساده‌ای نیست این پردازش). (قبل از اینکه از متد دریافت تاریخ استفاده کنم، سعی کردم از این روش کمک بگیرم و روی ویندوز XP جواب داد اما روی ویندوز 7 کار نکرد)

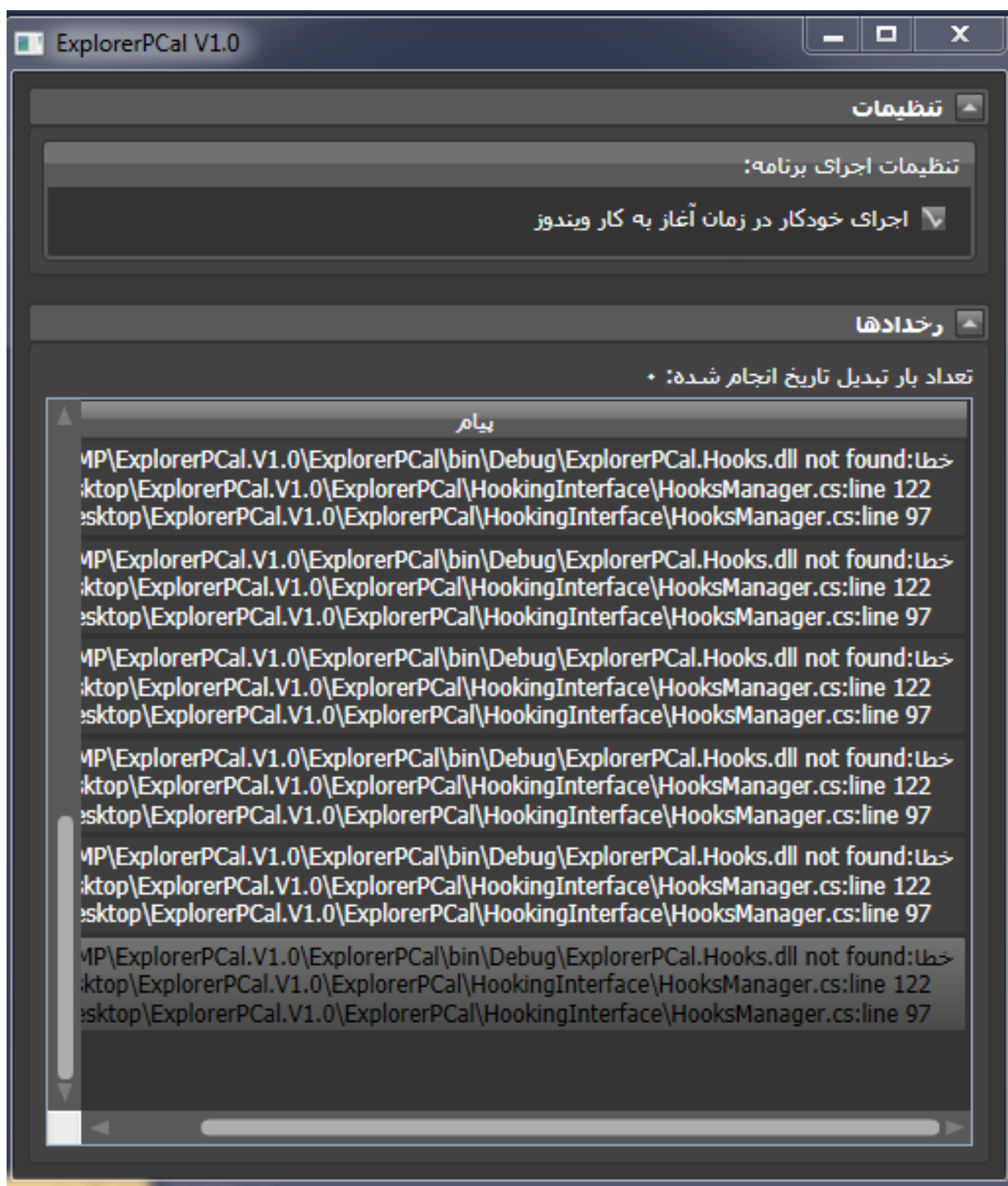
نویسنده: علیرضا صبوری
تاریخ: ۲:۲۲ ۱۳۹۲/۱۱/۱۳

سلام

با توجه به اینکه استفاده از easyhook در ویندوز 8 پشتیبانی نمیشه هنوز راهکاری جایگزین برای انجام این اعمال در ویندوزهای جدید معرفی نشده ؟

نویسنده: خیام
تاریخ: ۱۵:۱۶ ۱۳۹۲/۱۲/۲۲

وقتی برنامه رو اجرا میکنم با این خطا مواجه میشم (از ویندوز 7 استفاده میکنم)



- لطفا برای گزارش خطا از [قسمت مخصوص بازخوردهای پروژه آن در سایت](#) استفاده کنید.
همچنین در این حالت فایل `ErrosLog.Log` آن را هم فراموش نکنید (ریز خطاها در آن ثبت می شوند).

اگر پروژه را خودتان کامپایل کرده اید (که اینطور به نظر می رسد با توجه به پوشه debug)، برنامه اجرا نمی شود چون تمام فایل های exe و dll همراه easy hook را برای اجرا نیاز دارد و این ها باید کنار فایل اجرایی اصلی برنامه همانند بسته ای که برای دریافت در سایت قرار گرفته، کپی شوند.

این نکته در متن هم ذکر شده: قسمت «چند نکته تکمیلی مهم برای کار با کتابخانه Easy hook» انتهای بحث: «برای توزیع هوک های خود باید تمام فایل های همراه کتابخانه easy hook را نیز توزیع کنید و فقط به چند DLL ابتدایی آن بسنده نباید کرد»

نویسنده: دریا 69
تاریخ: ۱۳۹۳/۰۳/۲۷ ۸:۴۷

با سلام و تشکر بابت آموزش خوبتون
بخشید من به سوال در مورد خود api monitor دارم.
اینکه بعد از اجرای یک فایل و بدست آوردن فراخوانی ها، برای اینکه فراخوانی ها رو به بردار ویژگی تبدیل کنم و برای کلاس بندی ازشون استفاده کنم باید در فایل xml یا اکسل بریزم ولی وقتی میریزم ساختار سلسله مراتبیش رو دیگه نمایش نمیده. میخواسم ببینم چیکار باید بکنم که موقع کپی کردن در فایل متنی ساختار سلسله مراتبی و یا اینکه کدوم فراخوانی زیرمجموعه دیگری هست حفظ بشه؟
سوال دیگم اینه که چجوری با استفاده از این نرم افزار و بدون چک کردن تک تک فراخوانی ها و پارامترشون به صورت جداگانه، میتونیم بفهمیم بعد از اجرای یک فایل پارامتر کدام یک از فراخوانیامون دچار تغییر شدن؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۲۷ ۱۰:۱

اگر این برنامه ی ثالث قابلیت خاصی را ندارد، [در انجمن پشتیبانی آن](#) مطرح کنید تا اضافه شود.

نویسنده: دریا 69
تاریخ: ۱۳۹۳/۰۸/۰۹ ۹:۱۷

سلام
بخشید میشه راهنمایی بفرمایید اگه بخوایم به فایل در حال اجرا رو همه فراخوانی هایی که صدا میزنه رو به دام بندازیم چیکار باید بکنیم؟ یعنی بعد از هرفراخوانی متوقف بشه پارامترای داخل فراخوانی استخراج بشه و بعد اجرای برنامه ادامه پیدا کنه با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۹ ۹:۵۴

در قسمت API Filter آن، تمام موارد موجود را تیک بزنید. به این صورت تمام توابع WIN32 API تعریف شده در اینجا تحت نظر قرار می گیرند. زمانیکه فراخوانی شدند در قسمت لاگ صفحه، ریز جزئیات آن ها (تمام ورودی ها و خروجی های توابع) قابل بررسی و ذخیره خواهد بود.

نویسنده: دریا 69
تاریخ: ۱۳۹۳/۰۸/۰۹ ۱۴:۴۶

درسته توی apimonitor باید صبر کنیم فایل اجرایی کامل اجرا شه تا کامل لاگ رفتاریشو استخراج کنیم
ولی من میخوام در حین اجرای فایل با رسیدن به هر فراخوانی نرم افزار متوقف بشه پارامترای داخلشو بگیرم بعد فایل به اجرا ادامه بده. چجوری میتونم این کارو بکنم. نمیخوام هیچ تابعی رو تغییر بدم فقط میخوام مقادیر داخل api ها رو بگیرم
با easyhook در سی # نمیشه؟
من برای شناسایی بدافزار با استفاده از apimonitor لاگ رفتاری رو استخراج کردم از روش با استفاده از فراخوانیای سیستمی و

پارامترای داخلش مدل ساختیم و برای مرحله تشخیص میخوام یه فایل رو در حین اجرا تشخیص بدم بدافزار هست یا نه فقط شما بفرمایید چجوری با رسیدن به هر فراخوانی من به دامش بندازم و اطلاعات درونشو بگیرم ممنون

نویسنده: وحید نصیری
تاریخ: ۱۵:۲۲ ۱۳۹۳/۰۸/۰۹

- مراجعه کنید به راهنمای اصلی نرم افزار، قسمت [break points](#) آن (اواسط صفحه).
- با easy hook هم امکان پذیر است. فقط باید چند هزار تعریف API ایی را که در برنامه‌ی API Monitor حاضر و آماده وجود دارند، تبدیل کنید به کدهای سی‌شارپ و بعد استفاده کنید.
- ضمن اینکه برای آنالیز بدافزارها استفاده از API Monitor کافی نیست. باید مراجعه کنید به برنامه‌هایی مانند [OllyDBG](#).

نویسنده: دریا 69
تاریخ: ۱۷:۲۶ ۱۳۹۳/۰۸/۰۹

چون من میخوام به صورت اتوماتیک این کارو انجام بدم با breakpoint از API Monitor فقط به صورت دستی میتونم بعد از هر break پارامترها رو بگیرم

نویسنده: دریا 69
تاریخ: ۱۵:۸ ۱۳۹۳/۰۸/۱۱

سلام
ببخشید گزارشی که در ادرس زیر اومده این کاری که بنده گفتم رو با ابزار easyhook کرده ممنون میشم ببینید و بفرمایید آیا همون کاری هست که من در بالا گفتم. فقط بفرمایید همون کاری که من در بالا گفتم رو انجام میده؟
ایا این کارو با تزریق dll انجام میده؟

[4-API-DLLs.pdf](#)

با تشکر فراوان

نویسنده: وحید نصیری
تاریخ: ۱۵:۱۸ ۱۳۹۳/۰۸/۱۱

- اگر سورس easy hook را دریافت کنید، یکی از مثال‌های آن همین [file monitor](#) است که در گزارش فوق آمده‌است. در هر حالتی، تفاوتی نمی‌کند؛ یک سری توابع API را باید توسط آن مشخص کنید و بعد در این بین، یا اطلاعات رد و بدل شده را می‌توانید لاگ کنید و یا تغییر دهید. در مطلب جاری ریز جزئیات اینکار قدم به قدم بررسی شده.
- در کل اگر دقیقاً می‌دانید چه توابع API ایی را باید لاگ کنید، از روش ارائه شده در مطلب جاری استفاده کنید. هیچ تفاوتی ندارد. callback‌های معرفی شده در آن، دقیقاً محلی هستند که می‌توانید پارامترها را لاگ کنید.

نویسنده: دریا 69
تاریخ: ۱۷:۳۵ ۱۳۹۳/۰۸/۱۶

الان توی صفحه زیر

<http://www.codeproject.com/Articles/27637/EasyHook-The-reinvention-of-Windows-API-hooking>

بالای خط

a deep look under a hook

نوشته برای رهگیری اکسپلورر اونو در کامنت صدا بزنید. گفته میاد با صدازدن تابع "createfilew" توابع و پارامترای فایل اجرایی مورد نظر رو رهگیری میکنه

نویسنده: محسن خان

تابع `createfilew` جایی صدا زده نشده. مثل همین مقاله هوک شده. چرا هوک شده؟ چون می‌خواسته یک فایل مونیتور درست کنه. چرا این تابع رو هوک کرده؟ چون زمانیکه یک برنامه می‌خواد به فایلی در ویندوز دسترسی پیدا کنه یا درونش بنویسه، در اصل از `createfilew` استفاده می‌کنه. هوک کردن چیه؟ یعنی اینترسپت کردن. اینترسپت کردن چیه؟ یعنی وایسادن اون وسط گوش دادن. اول ویندوز میاد این تابع اینترسپت یا هوک شما رو صدا می‌زنه. بعد این وسط می‌تونی لاگش کنی. بعد که لاگش کردی باید تابع API اصلی رو صدا بزنی. چرا باید تابع API اصلی رو صدا بزنی؟ برای اینکه ویندوز نپوکه این وسط. برنامه بتونه به کار خودش ادامه بده.

اين مطلب دنباله‌ي « [تغيير عملکرد و يا رد يابي توابع ويندوز با استفاده از Hook هاى دات نتى](#) » است.

روش ارائه شده در آن با ويندوزهاى XP تا 7 نگارش‌هاى 32 بیتی و 64 بیتی، بدون مشکل کار می‌کند. اما تاثيری بر روی ويندوز 8 و نگارش‌هاى پس از آن نداشت.

تغييرات توابع GetTimeFormatW و GetDateFormatW در ويندوز اكسپلورر ويندوز 8

چه برنامه‌ی ExplorerPCa1 و چه API Monitor را اگر با فعال سازی توابع GetTimeFormatW و GetDateFormatW اجرا کنید، هیچ خروجی خاصی را مشاهده نخواهید کرد. در ابتدا به نظر می‌رسد که ساختار ويندوز شاید تغيير کرده‌است ... ولی اینطور نیست. فقط اينبار بجای فراخوانی اين توابع از kernel32.dll، از یک d11 مخفی در پوشه‌ی System32 استفاده می‌شود. روش پیدا کردن آن نیز به صورت زیر است:

```
"C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\bin\dumpbin.exe" /imports
c:\windows\explorer.exe > explorer.imports.txt
```

کار dumpbin.exe موجود در پوشه‌ی VC\bin و ویژوال استودیو، استخراج import table و export table یک فایل اجرایی و یا یک d11 بومی ويندوز است. به اين ترتيب می‌توان دریافت یک فایل exe، از چه d11 هاى استفاده می‌کند و همچنین از اين d11 ها، کدامیک از توابع آن‌ها را مورد استفاده قرار داده است.

اگر خروجی اين برنامه را که اکنون در فایل explorer.imports.txt ذخيره شده‌است، بررسی کنیم، به نتیجه‌ی زیر خواهیم رسید:

```
api-ms-win-core-datetime-l1-1-1.dll
14015E848 Import Address Table
1401613E0 Import Name Table
0 time date stamp
0 Index of first forwarder reference

2 GetDateFormatW
1 GetDateFormatEx
4 GetTimeFormatEx
```

بله. در ويندوزهاى سرى 8، ديگر از کرنل 32 برای دریافت GetDateFormatW استفاده نمی‌شود. اينبار از d11 اى به نام api-ms-win-core-datetime-l1-1-1.dll کمک گرفته شده‌است. اين d11 در پوشه‌ی System32 با خاصیت مخفی قرار دارد. بنابراین تنها تغييری که باید در برنامه‌ی ExplorerPCa1 داده شود، اضافه کردن مداخل جديد فوق است. در ويندوزهاى قبل از 8، از نگارش‌هاى Ex استفاده نمی‌شد. در اینجا هم از نگارش‌هاى W و هم Ex دار استفاده شده‌است.

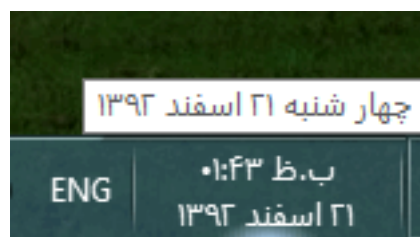
اگر خواستيد اين تغييرات را با برنامه‌ی API Monitor بررسی کنید، فایل جديد api-ms-win-core-datetime-l1-1-1.xml ذیل را در پوشه‌ی APIWindows آن کپی نمائيد تا مداخل api-ms-win-core-datetime-l1-1-1.dll نیز به مجموعه‌ی تعاریف آن اضافه شوند.

api-ms-win-core-datetime-l1-1-1.xml

حاصل نهایی، فایل‌هاى اجرایی و سورس بهبود یافته‌ی برنامه را از اینجا می‌توانید دریافت کنید:

[شمسى ساز تاريخ اكسپلورر ويندوز](#)

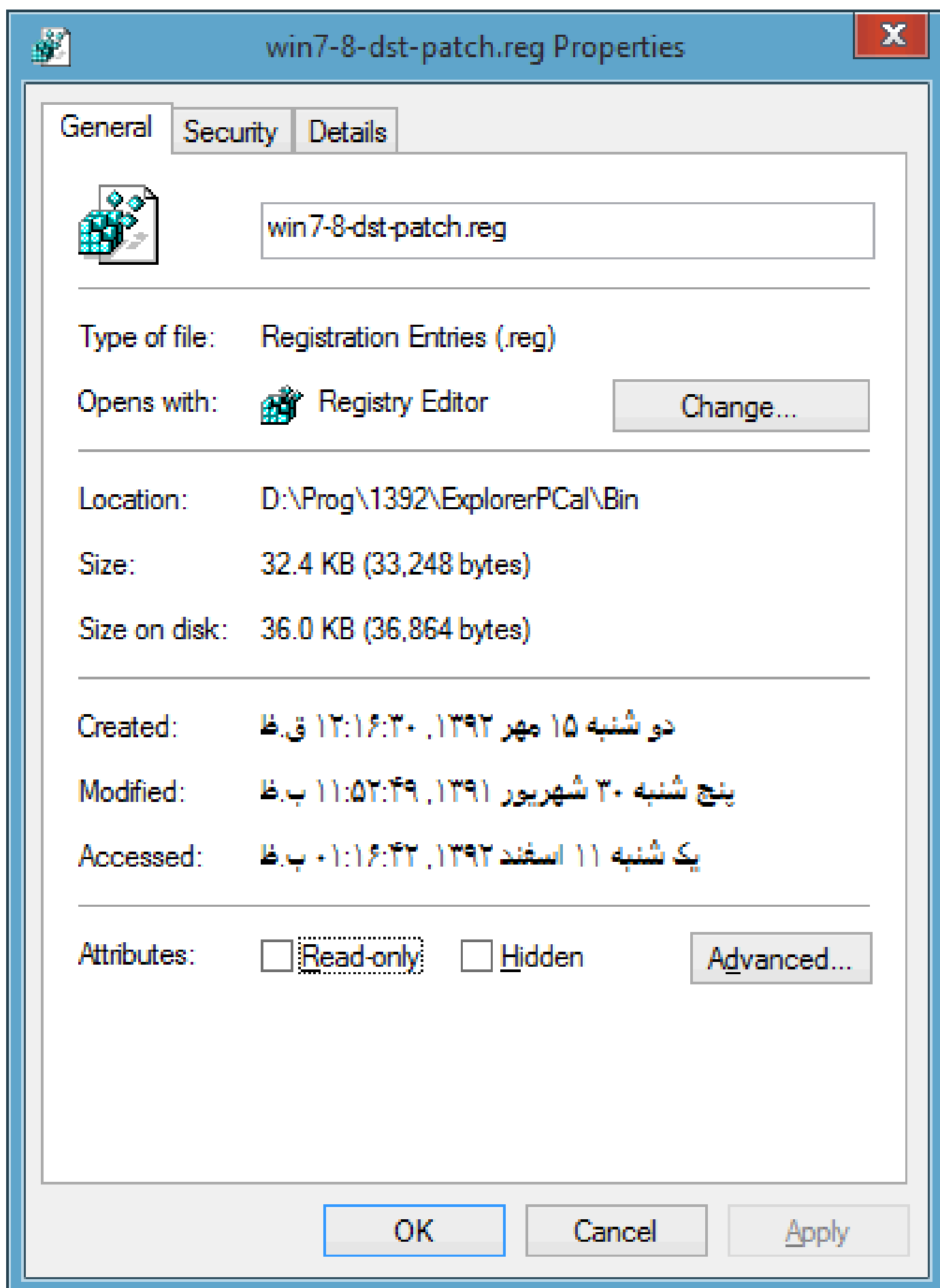
تاثير آن‌را نیز بر روی Explorer ويندوز 8، در تصاویر ذیل می‌توانید ملاحظه نمائيد:



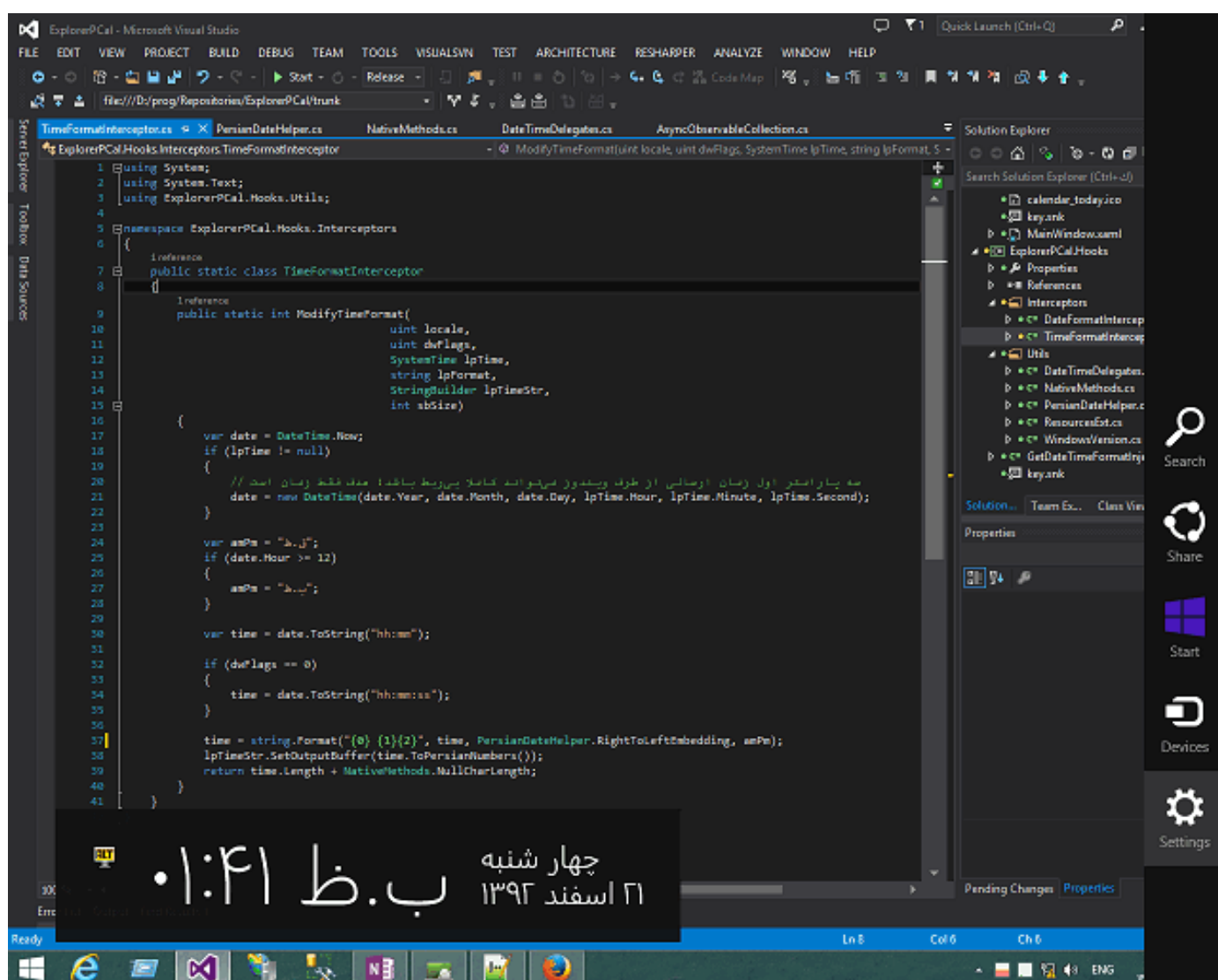
ساعت و تقویم نوار وظیفه‌ی ویندوز

Name	Date modified	Type	Size
_readme.txt	۱۰ مهر ۱۳۹۲ ۰۶:۱۸ ب.ظ	Text Document	1 KB
EasyHook.dll	۲۶ بهمن ۱۳۹۲ ۱۱:۵۶ ب.ظ	Application extens...	46 KB
EasyHook32.dll	۲۶ بهمن ۱۳۹۲ ۱۱:۵۶ ب.ظ	Application extens...	225 KB
EasyHook32Svc.exe	۲۶ بهمن ۱۳۹۲ ۱۱:۵۶ ب.ظ	Application	8 KB
EasyHook64.dll	۲۶ بهمن ۱۳۹۲ ۱۱:۵۶ ب.ظ	Application extens...	258 KB
EasyHook64Svc.exe	۲۶ بهمن ۱۳۹۲ ۱۱:۵۶ ب.ظ	Application	8 KB
ErrosLog.Log	۲۱ اسفند ۱۳۹۲ ۰۱:۳۴ ق.ظ	Text Document	79 KB
ExplorerPCal.exe	۱۸ اسفند ۱۳۹۲ ۰۶:۳۲ ب.ظ	Application	130 KB
ExplorerPCal.exe.Config	۲۱ اسفند ۱۳۹۲ ۰۱:۴۰ ب.ظ	CONFIG File	1 KB
ExplorerPCal.Hooks.dll	۲۱ اسفند ۱۳۹۲ ۰۱:۴۰ ب.ظ	Application extens...	15 KB
ExplorerPCal.Hooks.pdb	۲۱ اسفند ۱۳۹۲ ۰۱:۴۰ ب.ظ	Program Debug D...	32 KB
ExplorerPCal.pdb	۱۸ اسفند ۱۳۹۲ ۰۶:۳۲ ب.ظ	Program Debug D...	118 KB
ExplorerPCal.vshost.exe	۲۱ اسفند ۱۳۹۲ ۰۹:۲۷ ق.ظ	Application	24 KB
ExplorerPCal.vshost.exe.Config	۲۱ اسفند ۱۳۹۲ ۰۸:۴۲ ق.ظ	CONFIG File	1 KB
ExplorerPCal.vshost.exe.manifest	۲۸ خرداد ۱۳۹۲ ۰۴:۵۸ ب.ظ	MANIFEST File	1 KB
win7-8-dst-patch.reg	۳۰ شهریور ۱۳۹۱ ۱۱:۵۲ ب.ظ	Registration Entries	33 KB

تاریخ تغییرات فایل‌ها، در نمایش لیستی ویندوز اکسپلورر



تاریخ ایجاد و تغییرات یک فایل در خواص آن



تاریخ نمایش داده شده به همراه charm bar ویندوز 8

نظرات خوانندگان

نویسنده: میثم جوادی
تاریخ: ۱۴:۱۹ ۱۳۹۲/۱۲/۲۲

جدیدا مایکروسافت تغییراتی رو انجام میده که اطلاع رسانی خوبی درباره اش انجام نشده. علاوه بر موردی شما که اشاره کردید، یکپارچه سازی های Localizationی که انجام شده گاهی منتج به Break شدن نرم افزار میشه در حالیکه تغییر بسیار کوچک بوده مثلا در 8.1 به جای . از / برای Floating point فارسی پیش فرض استفاده شده که باعث میشه مقادیر ارسالی به دیتابیس دچار مشکل بشوند.

نویسنده: عباس
تاریخ: ۱۱:۲۹ ۱۳۹۳/۰۱/۱۴

با تشکر فراوان از شما خدا خیرتان دهد خیلی وقت بود که دنبال چنین برنامه ای می گشتم
(1) اگر تقویم taskbar ویندوز به صورت رقمی بود خیلی بهتر می شد. (مثلا 13/01/1393)
(2) اگر امکان داره برای تغییر تقویم صفحه lock ویندوز 8 هم این برنامه را بهبود بخشید.

نویسنده: علی مدرسی
تاریخ: ۱:۷ ۱۳۹۳/۰۸/۱۲

من کلی تلاش کردم که اون قسمت انتخاب ماه رو هم تو ویندوز فارسی کنم
اما نتونستم هوک مرتبط را پیدا کنم
ولی یه راه دیگه ای هست که هنگامی که برنامه تقویم ویندوز باز میشه (زمانی که رو ساعت کلیک می کنی) بهت خبر میده و میشه به جای صفحه تقویم برنامه خودمون را آورد بالا

می خواستم اگر کسی چیزی میدونه راهنمایی کنه که یه تقویم اپن سورس درست حسابی و کامل درست کنیم (:

در بیشتر وب سایت‌های شاهد نمایش تاریخ بر حسب تعداد روز/ ماه و یا سال گذشته شده از آن تاریخ هستیم. برای نمونه در سایت جاری تاریخ را بر همین اساس نمایش می‌دهند. نمونه‌ای از آن مانند «در ۲ سال قبل، چهارشنبه ۲۷ دی ۱۳۹۱، ساعت ۰۳:۳۵» می‌باشد. در این مقاله قصد دارم کدهایی را جهت انجام این کار ارائه کنم. در این مثال که در ادامه شاهد آن خواهیم بود، از یک پروژه‌ی Win form ساده، جهت نمایش بهتر استفاده کرده‌ام.

جهت اینکه درک کد و یا توضیح آن نیز ساده‌تر صورت بگیرد، به نظرم ابتدا متدهای مورد استفاده و کلاس‌هایی را که از آنها استفاده کرده‌ام، معرفی کنم بهتر باشد:

از کلاس PersianCalendar جهت گرفتن روز/ماه و سال استفاده شده و سه متد به شرح زیر دارد:

GetHour : از این متد برای گرفتن ساعت تاریخ مورد نظر استفاده می‌شود. کد این متد به صورت زیر است:

```
public string GetHour(DateTime lastdate)
{
    PersianCalendar pc = new PersianCalendar();
    string result = " ساعت " + (((pc.GetHour(lastdate)) < 10) ? ("0" +
pc.GetHour(lastdate).ToString()) : (pc.GetHour(lastdate).ToString()) + ":" +
(((pc.GetMinute(lastdate)) < 10) ? ("0" + pc.GetMinute(lastdate).ToString()) :
(pc.GetMinute(lastdate).ToString())));
    return result;
}
```

توضیح: اگر ساعت یا دقیقه تک رقمی باشد، یعنی کمتر از 10، برای نمایش بهتر آن یک صفر را به ابتدای آن اضافه می‌کنیم. یعنی ساعت 1:5 تبدیل می‌شود به 01:05

متد getDay : از این متد برای گرفتن نام روز مورد نظر استفاده می‌شود. ورودی این متد یک enum از نوع DayOfWeek است:

```
public string getDay(DayOfWeek day)
{
    string Result = "";
    switch (day)
    {
        case DayOfWeek.Friday:
            Result = "جمعه";
            break;
        case DayOfWeek.Monday:
            Result = "دوشنبه";
            break;
        case DayOfWeek.Saturday:
            Result = "شنبه";
            break;
        case DayOfWeek.Sunday:
            Result = "یکشنبه";
            break;
        case DayOfWeek.Thursday:
            Result = "پنج شنبه";
            break;
        case DayOfWeek.Tuesday:
            Result = "سه شنبه";
            break;
        case DayOfWeek.Wednesday:
            Result = "چهارشنبه";
            break;
        default:
            break;
    }
    return Result;
}
```

و در هر جایی نیاز به گرفتن تاریخ باشد، به صورت زیر عمل خواهیم کرد:

```
getDay(pc.GetDayOfWeek(LastDate))
```


pc یک متغیر از نوع persianclander می‌باشد.

متد GetMounth: همانطور که از نام این متد معلوم است، کار آن بازگشت نام ماه مورد استفاده است. کد آن نیز به صورت زیر می‌باشد:

```
public string GetMounth(int month)
{
    string[] monthInYear =
    { "فروردین", "اردیبهشت", "خرداد", "تیر", "مرداد", "شهریور", "مهر", "آبان", "آذر", "دی", "بهمن", "اسفند" };
    return monthInYear[month-1];
}
```

وجود 1- در هنگام return به این دلیل است که زمانیکه قصد دریافت شماره ماه را از شیء PersianClander داشته باشیم، از یک شروع می‌شود. یعنی برای ماه اسفند مقدار 12 و برای ماه فروردین مقدار 1 و در یک آرایه، ایندکس‌ها از صفر شروع می‌شوند. و اما کد کامل آن برای تبدیل تاریخ، به صورت رشته مورد نظر، به صورت زیر است:

```
private void btnGetDate_Click(object sender, EventArgs e)
{
    DateTime LastDate = DateTime.Parse(txtLastDate.Text);
    TimeSpan ts = DateTime.Now - LastDate;
    PersianCalendar pc = new PersianCalendar();
    int DifferenceYear = DateTime.Now.Year - LastDate.Year;
    int DifferenceMonth = DateTime.Now.Month - LastDate.Month;
    if(DateTime.Now.Month > LastDate.Month)
        DiffernceMonth = DateTime.Now.Month - LastDate.Month;
    else
        DiffernceMonth = LastDate.Month - DateTime.Now.Month;
    int DifferenceDays = ts.Days;

    StringBuilder Result = new System.Text.StringBuilder("");

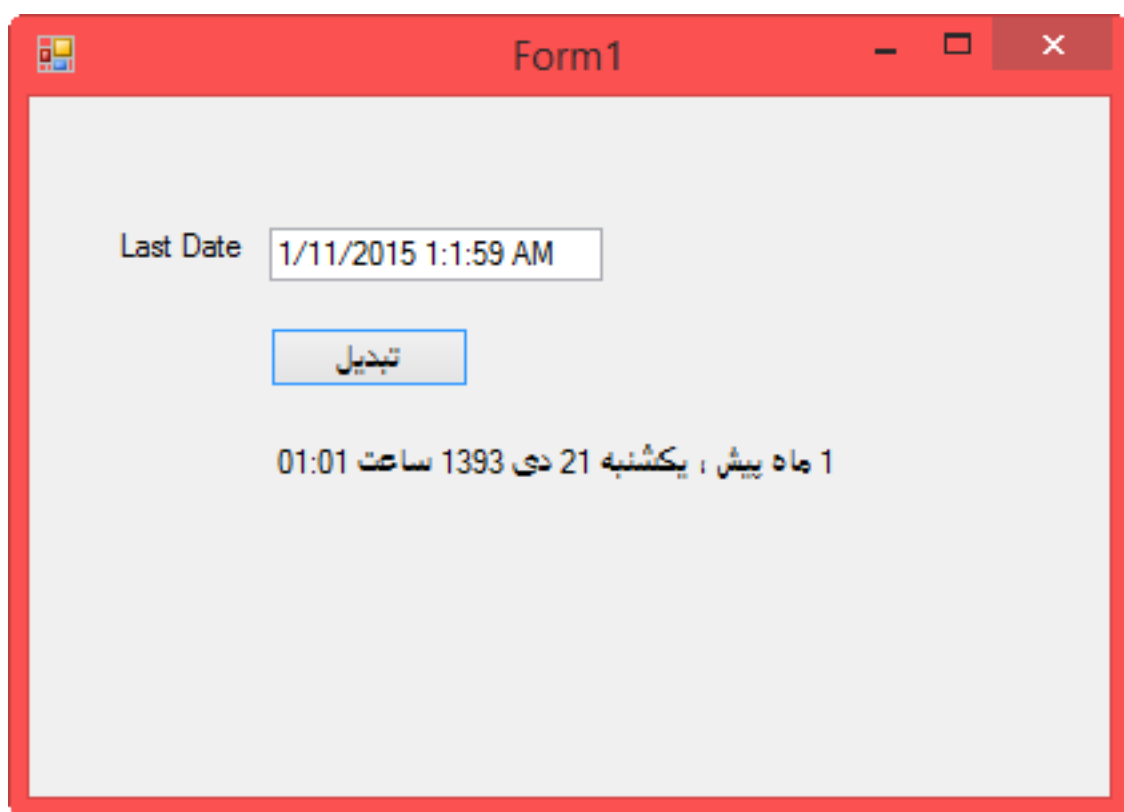
    if(DifferenceYear > 0)
    {
        Result.Append(DifferenceYear.ToString() + " سال پیش " ,
        "+getDay(pc.GetDayOfWeek(LastDate))+" " + pc.GetDayOfMonth(LastDate).ToString() + " " +
        GetMounth(pc.GetMonth(LastDate)) + " " + pc.GetYear(LastDate) + GetHour(LastDate));
    }
    else if(DiffernceMonth > 0)
    {
        Result.Append(DiffernceMonth.ToString() + " ماه پیش " , " " +
        getDay(pc.GetDayOfWeek(LastDate)) + " " + pc.GetDayOfMonth(LastDate).ToString() + " " +
        GetMounth(pc.GetMonth(LastDate)) + " " + pc.GetYear(LastDate) + GetHour(LastDate));
    }
    else if(DifferenceDays > 0)
    {
        Result.Append(DifferenceDays.ToString() + " روز پیش " , " " +
        getDay(pc.GetDayOfWeek(LastDate)) + " " + pc.GetDayOfMonth(LastDate).ToString() + " " +
        GetMounth(pc.GetMonth(LastDate)) + " " + pc.GetYear(LastDate) + GetHour(LastDate));
    }
    else if(DifferenceDays == 0)
    {
        Result.Append(" امروز " , " " + getDay(pc.GetDayOfWeek(LastDate)) + " " +
        pc.GetDayOfMonth(LastDate).ToString() + " " + GetMounth(pc.GetMonth(LastDate)) + " " +
        pc.GetYear(LastDate) + GetHour(LastDate));
    }

    lblResult.Text = Result.ToString();
}
```

کد زیر برای دریافت تعداد اختلاف بین ماه‌ها، از تاریخی گذشته تا تاریخ جاری است:

```
if(DateTime.Now.Month > LastDate.Month)
    DiffernceMonth = DateTime.Now.Month - LastDate.Month;
else
    DiffernceMonth = LastDate.Month - DateTime.Now.Month;
```

چرا از if استفاده شده است؟ فرض کنید تاریخ امروز 2015/12/2 باشد و تاریخی که قصد تبدیل آن را داریم 2014/12/10 است. تعداد اختلافی که بین تعداد ماه‌ها است 8 ماه است و اگر این بررسی چک کردن بزرگ بودن آن دو انجام نشود، مقدار 8- را بر میگرداند که برای کار ما نادرست است. نمونه‌ای از این تبدیل :



Form1

Last Date 1/11/2015 1:1:59 AM

تبدیل

1 ماه پیش ، یکشنبه 21 دی 1393 ساعت 01:01

[دانلود کدهای این مقاله](#)

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۵:۲۰ ۱۳۹۳/۱۱/۲۳

یک نکته: ("00") number.ToString به صورت خودکار 0 ماقبل اعداد دو رقمی رو اضافه می‌کنه.

نویسنده: عثمان رحیمی
تاریخ: ۱۶:۵۰ ۱۳۹۳/۱۱/۲۳

جهت تکمیل <<

در صورت نیاز به نمایش دقیقه :

```
double DifferenceMinute = ts.TotalMinutes;
```

و قبل از آخرین else if :

```
else if (DateTime.Now.Date == LastDate.Date && (DifferenceMinute<60 && (DateTime.Now.Hour-  
LastDate.Hour<=1)) )  
    Result.Append("در " + ConvertMinuteToString((int)DifferenceMinute) + " دقیقه قبل " + "  
" + GetHour(LastDate));
```

و تبدیل دقیقه به رشته :

```
public string ConvertMinuteToString(int minute)  
{  
    string Result = "";  
    string[] minLessTen = { "یک", "دو", "سه", "چهار", "پنج", "شش", "هفت", "هشت", "نه", "ده" };  
    string[] minLesstweny = { "یازده", "دوازده", "سیزده", "چهارده", "پانزده", "شانزده", "هفده", "هجده", "نوزده" };  
    Dictionary<int,string> minmore=new Dictionary<int,string>  
    {  
        {2,"بیست"},  
        {3,"سی"},  
        {4,"چهل"},  
        {5,"پنجاه"}  
    };  
    if (minute <= 10)  
        Result = minLessTen[minute - 1];  
    else if (minute < 20)  
        Result = minLesstweny[(minute%10)-1];  
    else if ((minute / 10) >= 2)  
    {  
        Result = minmore[minute / 10];  
        if (minute % 10 > 0)  
            Result += " و " + minLessTen[(minute % 10) - 1];  
    }  
    return Result;  
}
```

* بهتر است بجای String از StringBuilder استفاده شود .


```

        text = text.cleanTags().Trim();
        text = text.Replace("\t", " ");
        text = text.Replace("\n", " ");
        text = text.Replace("\r", " ");

        var words = text.Split(
            new[] { ' ', ',', ';', ':', '.', '!', '"', '(', ')', '?', ':', '\'', '«', '»', '+', '-' },
            StringSplitOptions.RemoveEmptyEntries);
        return words.Length;
    }

    private static string cleanTags(this string data)
    {
        return data.Replace("\n", "\n ").removeHtmlTags();
    }

    private static string removeHtmlTags(this string text)
    {
        return string.IsNullOrEmpty(text) ?
            string.Empty :
            _matchAllTags.Replace(text, " ").Replace("&nbsp;", " ");
    }
}

```

در اینجا حذف تگ‌های HTML و همچنین پردازش خطوط جدید و حروف غیرمجاز در نظر گرفته شده‌اند.

پس از اینکه موفق به شمارش تعداد کلمات یک متن HTML ایی شدیم، اکنون می‌توان این تعداد را تقسیم بر 180 (یک عدد معمول و متداول) کرد تا زمان خواندن کل متن بدست آید. سپس با استفاده از متد `toReadableString` می‌توان آن‌را به شکل قابل خواندن‌تری نمایش داد.

```

using System;

namespace ReadingTime
{
    public static class CalculateReadingTime
    {
        public static string MinReadTime(this string text, int wordsPerMinute = 180)
        {
            var wordsCount = text.WordsCount();
            var minutes = wordsCount / wordsPerMinute;
            return minutes == 0 ? "کمتر از یک دقیقه" : TimeSpan.FromMinutes(minutes).toReadableString();
        }

        private static string toReadableString(this TimeSpan span)
        {
            var formatted = string.Format("{0}{1}{2}{3}",
                span.Duration().Days > 0 ? string.Format("{0:0} روز و ", span.Days) : string.Empty,
                span.Duration().Hours > 0 ? string.Format("{0:0} ساعت و ", span.Hours) : string.Empty,
                span.Duration().Minutes > 0 ? string.Format("{0:0} دقیقه و ", span.Minutes) :
                span.Duration().Seconds > 0 ? string.Format("{0:0} ثانیه ", span.Seconds) :
                string.Empty,
                string.Empty);

            if (formatted.EndsWith(" و "))
            {
                formatted = formatted.Substring(0, formatted.Length - 2);
            }

            if (string.IsNullOrEmpty(formatted))
            {
                formatted = "ثانیه 0";
            }
            return formatted.Trim();
        }
    }
}

```

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

[ReadingTime.zip](#)

نظرات خوانندگان

نویسنده: صالح باقری

تاریخ: ۲۱:۴۶ ۱۳۹۴/۰۵/۳۱

الان بالای هر مطلب به تخمین خواندن میاد از همین کد استفاده شده؟

به نظرم باید یخورده اصلاحات صورت بگیره چون «کدخواندن» با «زمان خواندن» کمی تفاوت داره و برای خواندن کد عملیات دیباگ و کمپایل هم در مغز انجام میشه و زمان بیشتری صرف میکنه ... من خودم این مطلب رو 5 دقیقه ای خوندم اونم البته سرسری رد شدم ...

نویسنده: محسن خان

تاریخ: ۲۲:۲۳ ۱۳۹۴/۰۵/۳۱

خیلی از اوقات جنبه‌ی روانی این اعداد مهم‌تر هست تا دقت اون‌ها. همینقدر که می‌دونم یک مطلب رو می‌تونم در سه دقیقه تموم کنی، بهتر رغبت می‌کنی شروعش کنی.