

قبل از شروع بحث، سورس کامل پروژه را از [اینجا](#) دریافت کنید (یک پروژه VSTO از نوع outlook add-in در VS.Net 2008 SP1).

توضیحات مربوطه را به دو قسمت تقسیم کرده‌ام. قسمت اول یافتن تاریخ‌های sent و فارسی کردن آنها و قسمت بعدی نحوه اضافه کردن یک ستون و مقدار دهی آن (در روزی دیگر).

متن ایمیل‌های دریافتی در آوت‌لوک‌های جدید عموماً به دو فرمت HTML و یا RichText دریافت می‌شوند. حالت‌های دیگری هم مانند plain و unspecified هم موجود هستند که حتی اگر ایمیلی را به صورت plain ارسال نمائید، با فرمت RichText نمایش داده خواهد شد (بنابراین بر اساس آزمایشات من بررسی این دو فرمت کفایت می‌کند).

برای اینکه قسمت‌های sent را پیدا کنیم در ابتدا باید سورس صفحه را بررسی نمائیم (کلیک راست و view source). در حالت فرمت HTML داریم:

```
<p class=MsoNormal><b><span style='font-size:10.0pt;font-family:"Tahoma", "sans-serif"'>From:</span></b><span style='font-size:10.0pt;font-family:"Tahoma", "sans-serif"'> Nasiri, Vahid <br><b>Sent:</b> <span lang=AR-SA dir=RTL>our date goes here</span><br><b>To:</b> xyz<br><b>Subject:</b> our subject</p>
```

و در حالت ایمیل‌های RichText خواهیم داشت:

```
From: tst@tst.net<br>Sent:<span lang=AR-SA dir=RTL>our date goes here</span><br>To: Nasiri, Vahid<br>Subject: <span lang=AR-SA dir=RTL>xyz</span><br>
```

خوب، برای پیدا کردن عبارت تاریخ قسمت sent چه باید کرد؟ (our date goes here در اینجا) استفاده از روش‌های متداول کار با رشته‌ها در اینجا به علت انبوهی از تگ‌های HTML اصلاً مقرون به صرفه نیست و کند خواهد بود. خوشبختانه با وجود کتابخانه regular expressions در دات نت، پیدا کردن عباراتی که از یک الگوی خاص پیروی می‌کنند به سادگی و با سرعت بسیار بالایی قابل انجام است. پیشنهاد من برای دو فرمت بالا به صورت زیر بوده: (شاید شما الگوی دیگری را یافتید، زیبایی اوپن سورس :)

```
private const string REGEXHTMLPATTERN = @"(?s)>\s(.+?)<br>";
private const string REGEXPLAINTEXTPATTERN = "(?s)Sent:(.+?)<br>";
```

برای مثال در حالت دوم هر چیزی که بین sent و br قرار می‌گیرد در کل متن بررسی خواهد شد (با استفاده از MatchCollection فضای نام System.Text.RegularExpressions). در اینجا اگر Convert.ToDateTime موفق بود یعنی تاریخ قابل تبدیل است (البته قبل از تبدیل تمام تگ‌های HTML احتمالی هم تمیز خواهند شد) و ما آنرا با استفاده از تابع DateTimeToFarsiStr در کلاس cDate به نمونه شمسی تبدیل کرده (لطفاً به سورس برنامه مراجعه کنید) و نهایتاً آنرا در متن جایگزین می‌کنیم. سرعت استفاده از RegularExpressions فوق العاده بالا است و برای نمونه در ایمیلی با بیش از 20 رپیلای در کسری از ثانیه کل

این عملیات انجام خواهد شد.

تا اینجا بررسی کلی الگوریتم مورد استفاده قسمت اول به پایان می‌رسد.

بیشترین وقتی که در این پروژه صرف شد نحوه پیدا کردن شیء MailItem جاری باز شده با استفاده از رخدادهای آوت‌لوک بود (مدت مدیدی را برای این مورد وقت گذاشتم! چون عملاً در هیچ کتابی به این مباحث پرداخته نمی‌شود و باید کل نت را زیر و رو کرد). دو مورد را باید بررسی کرد. الف) inspector ها (صفحه‌ای که جهت ایجاد یک ایمیل جدید یا رپیلای به ایمیل جاری باز می‌شود، inspector نام دارد) ب) ActiveExplorer ها (صفحه‌ای که لیست ایمیل‌ها را نمایش می‌دهد و این صفحه می‌تواند در فولدرهای مختلفی که شما ایجاد کرده‌اید نیز نمایش داده شود بنابراین بررسی inbox به تنهایی کافی نیست) نحوه ایجاد اشیاء مربوطه و تحت نظر قرار دادن آنها را در روال ThisAddIn\_Startup فایل ThisAddIn.cs می‌توانید مشاهده نمائید. نکته مهمی که اینجا وجود دارد، تعریف این اشیاء در سطح کلاس است. در غیراینصورت با اولین خانه تکانی garbage collector، اشیاء شما (بدلیل نبود ارجاعی فعال به آنها) معدوم خواهند شد(!) و دیگر روال‌های رخداد گردان تعریف شده کار نخواهند.