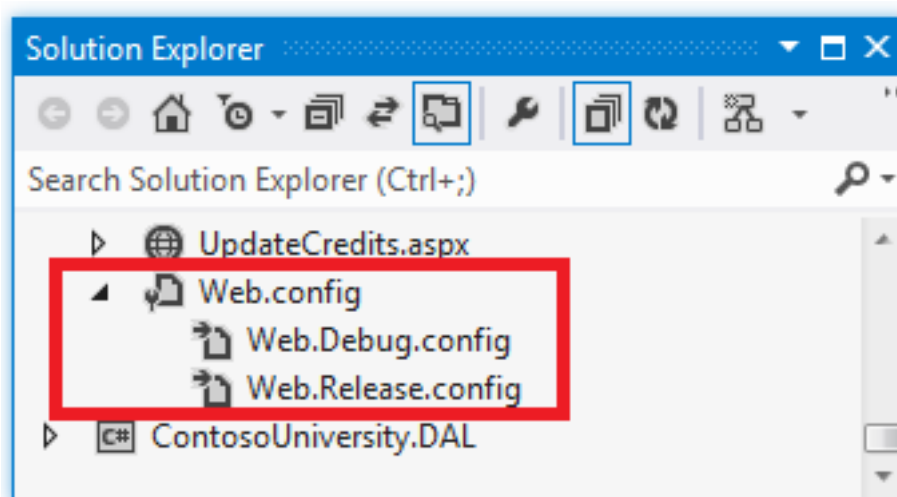


یکی از مشکلات برنامه نویسان اختلاف بین فایل web.config تولید شده در سیستم خودشان و مقصد نهایی برنامه می‌باشد. در این مطلب به نحوه خودکار سازی تغییرات، برای توسعه بر روی مقصد نهایی برنامه می‌پردازیم. اکثر برنامه‌ها تنظیماتی در فایل web.config خود دارند که زمان عرضه برای مقصد نهایی می‌بایست تغییر کنند. پردازش خودکار این تغییرات کمک می‌کند تا از خطاهای ناشی از تغییرات دستی در زمان عرضه نهایی جلوگیری شود.

### فایل‌های پیش فرض انتقالی

در پنجره Solution Explorer فایل web.config را بوسیله ایکن کنار آن باز کنید تا دو فایل پیش فرض web.Debug.config و web.Release.config که برای build configuration های برنامه ایجاد شده‌اند، مشاهده نمایید.



شما می‌توانید برای ایجاد build configuration دیگری، با راست کلیک کردن بر روی فایل web.config و انتخاب گزینه Add Config Transforms یک فایل دیگر را ایجاد نمایید. البته اگر این گزینه غیر فعال است، ابتدا می‌بایست یک build configuration جدید را ایجاد نمایید.

### غیرفعال کردن حالت اشکال زدایی (debug)

در زمان عرضه نهایی برنامه دیگر لازم نیست تا امکان اشکال زدایی فعال باشد؛ به همین خاطر در فایل web.Release.config تگ compilation مانند کد زیر از دستور RemoveAttributes استفاده می‌کنیم تا آن خاصیت را حذف نماید.

```
<compilation xdt:Transform="RemoveAttributes(debug)" />
```

### محدود کردن صفحات خطای برنامه فقط برای برنامه نویسی

یکی دیگر از تنظیمات، عدم نمایش خطاهای برنامه به کاربر نهایی و انتقال آن به یک صفحه رخداد خطا می‌باشد:

```
<customErrors mode="Off" defaultRedirect="~/GenericErrorPage.aspx">
```

```
<error statusCode="404" redirect="~/GenericErrorPage.aspx" />
</customErrors>
```

اما در زمان کد نویسی در محیط ویژوال استادیو می‌خواهیم mode را بر روی Off تنظیم نماییم تا خطا نمایش داده شود:

```
<customErrors mode="RemoteOnly" xdt:Transform="Replace" defaultRedirect="~/GenericErrorPage.aspx">
  <error statusCode="404" redirect="~/GenericErrorPage.aspx" />
</customErrors>
```

برای این کار از دستور Replace استفاده می‌کنیم تا تگ customErrors با مد RemoteOnly را جایگزین مد Off در زمان Release نماید.

### تنظیم رشته اتصال پایگاه داده اصلی

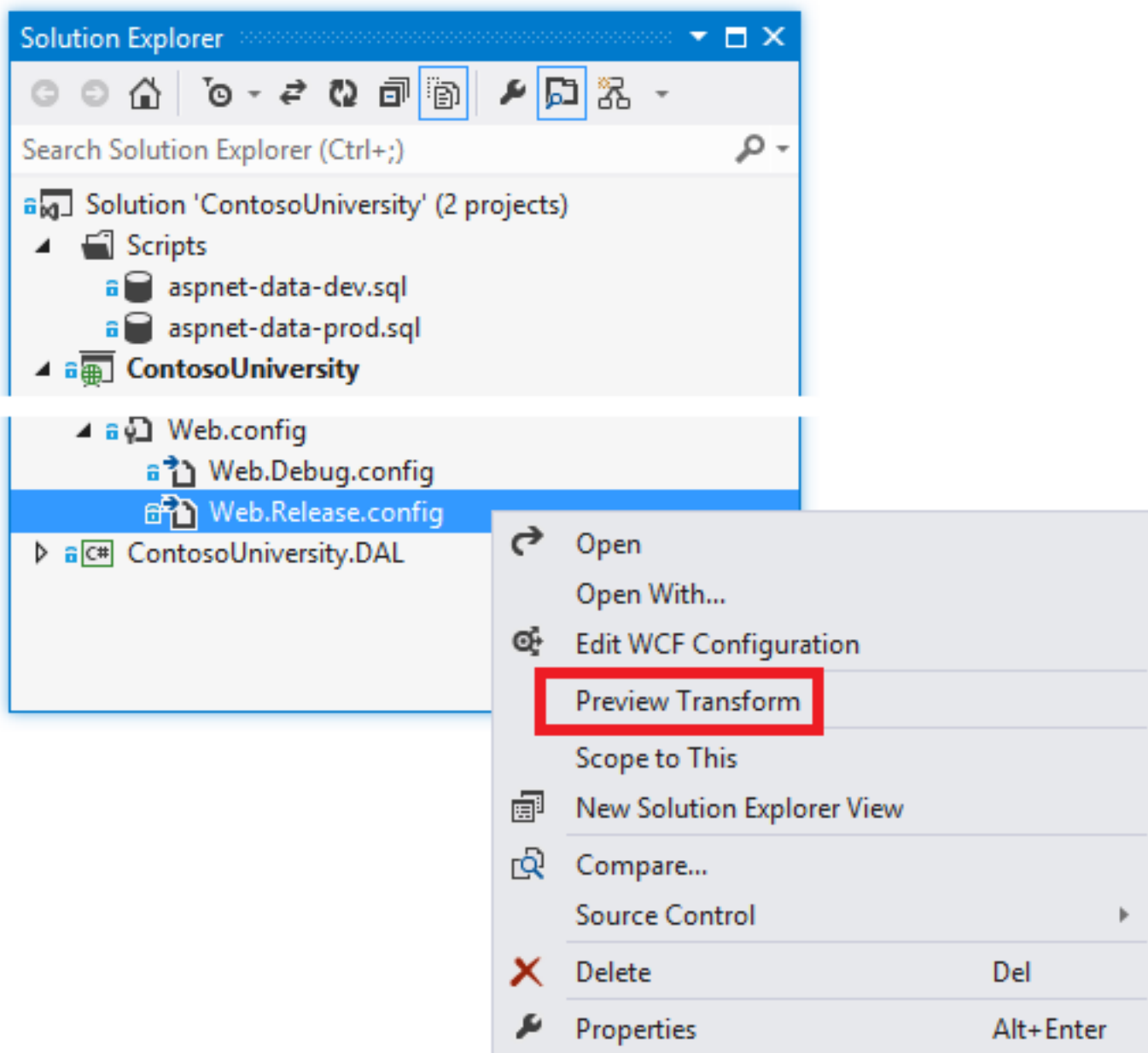
مهمترین قسمت فایل web.config تگ connection string می‌باشد که باید به رشته اتصال پایگاه داده نهایی برنامه، تغییر یابد.

```
<connectionStrings>
  <add name="TestContext"
    connectionString="Data Source=Server1;Password=****;User ID=sa; Initial Catalog=Test;Integrated
Security=True"
    xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>
</connectionStrings>
```

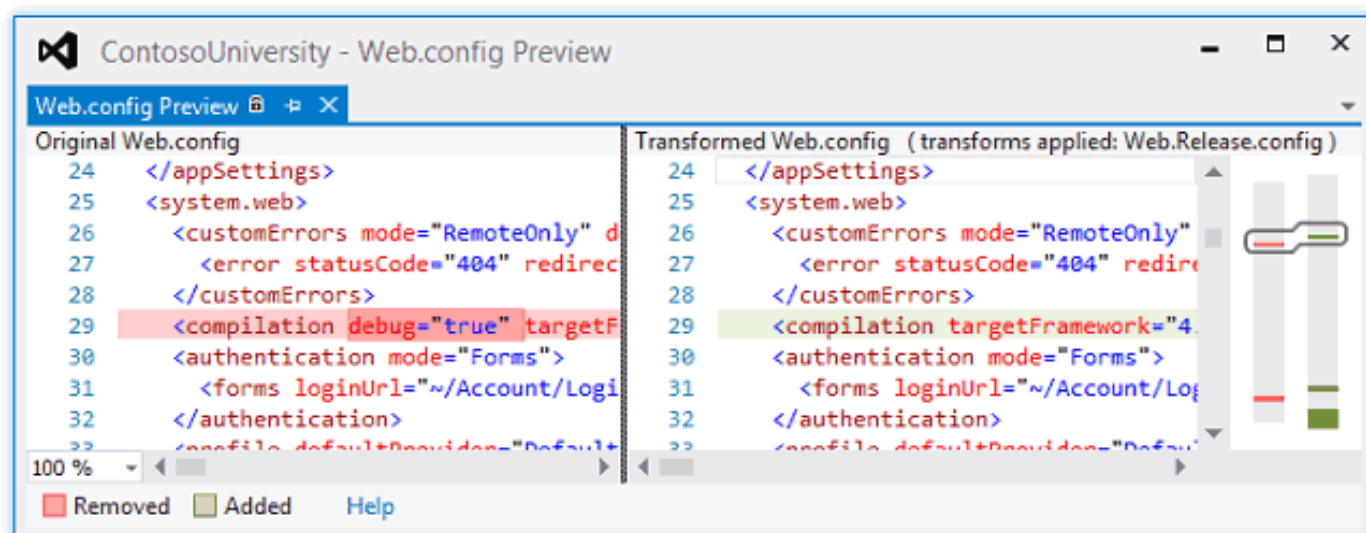
برای این کار از دستور SetAttributes مانند بالا استفاده می‌کنیم. اما چون ممکن است چندین connection string داشته باشیم، از دستور Match با مقدار name استفاده می‌کنیم، تا فقط رشته اتصالی که نام آن برابر رشته جاری می‌باشد را تغییر دهد که در اینجا TestContext را جستجو می‌نماید.

### پیش نمایش connection string نهایی

برای مشاهده پیش نمایش نهایی web.config، بر روی فایل web.Release.config در پنجره Solution Explorer راست کلیک کنید و گزینه Preview Transform را انتخاب کنید.



هر دو فایل web.config اصلی و نهایی در کنار هم آورده می‌شود و اختلاف بین این دو را که برجسته‌تر شده است، می‌توانید مشاهده نمایید.



ادامه دارد...

## نظرات خوانندگان

نویسنده: امیر

تاریخ: ۲۱:۲۴ ۱۳۹۱/۱۲/۲۲

یعنی ما تنظیمات رو روی سرور داریم بعد میایم خودمون تنظیم میکنیم، یا اینکه با این کار خودش خودکار روی سرور ست میشه. لطفا کمی توضیح بدین

نویسنده: سید امیر سجادی

تاریخ: ۹:۱۷ ۱۳۹۲/۰۲/۰۵

این کار زمان publish سایت انجام میشه و تنظیماتی که روی فایل web.config انجام شده رو اعمال میکنه.

نویسنده: آتوسا فتوحی

تاریخ: ۲۲:۲۹ ۱۳۹۲/۱۰/۲۲

با سلام.

این تنظیمات رو باید در Web.Config نوشت یا در Web.Release.Config

نویسنده: مجتبی کاویانی

تاریخ: ۱۶:۲۱ ۱۳۹۲/۱۰/۲۳

دستورات انتقال را می‌توانید در فایل Web.Debug.Config و یا Web.Release.Config بنویسید

نویسنده: آتوسا فتوحی

تاریخ: ۱۱:۳۴ ۱۳۹۲/۱۰/۲۴

سلام.

چجوری Connection String رو هنگام انتقال فایل Web.Config باید Encrypt کرد ؟

در مطلب قبلی [Web.config File Transformation #1](#) با مفهوم انتقال وب کانفیگ و برخی از روش‌های آن آشنا شدید در ادامه به موارد دیگری خواهیم پرداخت.

### قواعد انتقال وب کانفیگ

در کل دو ویژگی اصلی در انتقال وب کانفیگ وجود دارد که یک xdt:Transform و دیگری xdt:Locator می باشد. این دو در واقع چگونگی تغییر فایل انتقالی در زمان deploy آن را تعیین می‌کنند. این ویژگی‌ها از نوع xml می‌باشد که در فضای نام XML-Document- Transform تعریف شده و با پسوند xdt شروع می‌شود.

قاعده ویژگی Locator

این ویژگی برای جستجو در فایل وب کانفیگ استفاده می‌شود

:Condition

عبارت condition برای تعیین شرط قاعده Locator استفاده می‌شود

```
Locator="Condition(XPath expression)"
```

مثال:

```
<configuration xmlns:xdt="...">
  <connectionStrings>
    <add name="AWLT" connectionString="newstring"
      providerName="newprovider"
      xdt:Transform="Replace"
      xdt:Locator="Condition(@name='oldname'
        or @providerName='oldprovider')"/>
  </connectionStrings>
</configuration>
```

مثال بالا نشان می‌دهد که چگونه دنبال connectionstring ی بگردیم که نام آن oldname یا نام ارائه دهنده آن oldprovider باشد.

:Match

برای انتخاب عنصر یا عناصری که مقدار آن برابر ویژگی یا ویژگی‌های تعیین شده باشد. در صورتی که چندین ویژگی یافت شود، فقط عناصری که همه ویژگی‌های تعیین شده آن‌ها برابر باشد انتخاب می‌شود. نام ویژگی‌ها را می‌بایست با کما از هم جدا نمود.

```
Locator="Match(comma-delimited list of one or more attribute names)"
```

مثال:

```
<configuration xmlns:xdt="...">
  <connectionStrings>
    <add name="AWLT" connectionString="newstring"
      providerName="newprovider"
      xdt:Transform="Replace"
      xdt:Locator="Match(name)"/>
  </connectionStrings>
</configuration>
```

مثال فوق دنبال عناصری که ویژگی نام آنها برابر AWLT باشد می‌گردد.

:XPath

عبارتی از مسیرهای عناصر xml را جستجو می‌کند.

```
Locator="XPath(XPath expression)"
```

مثال:

```
<configuration xmlns:xdt="...">
  <connectionStrings>
    <add name="AWLT" connectionString="newstring"
      providerName="newprovider"
      xdt:Transform="Replace"
      xdt:Locator="XPath(configuration/connectionStrings[@name='AWLT'
        or @providerName='System.Data.SqlClient'])" />
  </connectionStrings>
</configuration>
```

در این مثال همانند مثال Condition همان عناصر را با استفاده از XPath جستجو می‌نماید.

قاعده ویژگی Transform

این ویژگی نحوه انتقال عنصر در فایل وب کانفیگ را مشخص می‌سازد.

:Replace

عناصر تعیین شده با عناصر انتقالی جایگزین می‌شود

```
Transform="Replace"
```

:Insert

عنصر انتقالی به عنوان فرزند عنصر تعیین شده اضافه می‌گردد.

```
Transform="Insert"
```

مثال:

```
<configuration xmlns:xdt="...">
  <connectionStrings>
    <add name="AWLT" connectionString="newstring"
      providerName="newprovider"
      xdt:Transform="Insert" />
  </connectionStrings>
</configuration>
```

connectionstring انتقالی را به لیست رشته‌های اتصال فایل انتقالی اضافه می‌نماید.

:InsertBefore

عنصر انتقالی را قبل از مسیر تعیین شده با XPath قرار می‌دهد.

```
Transform="InsertBefore(XPath expression)"
```

مثال:

```
<configuration xmlns:xdt="...">
  <authorization>
    <allow roles="Admins"
      xdt:Transform="InsertBefore(/configuration/system.web/authorization/deny[@users='*'])" />
  </authorization>
</configuration>
```

عنصر انتقالی که همان allow می‌باشد که قبل از عنصر مسیر configuration/system.web/authorization/deny برای همه کاربر

است قرار می‌دهد.

:InsertAfter

عنصر تعیین شده را بعد از مسیر تعیین شده با XPath قرار می‌دهد.

```
Transform="InsertAfter(XPath expression)"
```

مثال:

```
<configuration xmlns:xdt="...">
  <authorization>
    <deny users="UserName"
      xdt:Transform="InsertAfter
        (/configuration/system.web/authorization/allow[@roles='Admins'])" />
    </authorization>
  </configuration>
```

:Remove

عنصر تعیین شده را از فایل انتقالی حذف می‌نماید. اگر چندین عنصر یافت شود اولین عنصر حذف خواهد شد.

```
Transform="Remove"
```

مثال

```
<configuration xmlns:xdt="...">
  <connectionStrings>
    <add xdt:Transform="Remove" />
  </connectionStrings>
</configuration>
```

همه عناصر add عنصر connectionstring را انتخاب و اولین add را حذف می‌نماید.

:RemoveAll

عنصر یا عناصر تعیین شده را از فایل انتقالی حذف می‌نماید.

```
Transform="RemoveAll"
```

مثال:

```
<configuration xmlns:xdt="...">
  <connectionStrings>
    <add xdt:Transform="RemoveAll" />
  </connectionStrings>
</configuration>
```

همه عناصر add را از فایل انتقالی حذف می‌نماید.

:RemoveAttribute

ویژگی تعیین شده را از عناصر انتخاب شده حذف می‌نماید.

```
Transform="RemoveAttributes(comma-delimited list of one or more attribute names)"
```

مثال:

```
<configuration xmlns:xdt="...">
```



```
<compilation
  xdt:Transform="RemoveAttributes(debug,batch)">
</compilation>
</configuration>
```

ویژگی debug و batch را از عنصر compilation وب کانفیگ انتقالی حذف می‌نماید.

:SetAttribute

ویژگی تعیین شده را با مقادیر انتقالی مقدار دهی می‌کند.

```
Transform="SetAttributes(comma-delimited list of one or more attribute names)"
```

مثال:

```
<configuration xmlns:xdt="...">
  <compilation
    batch="false"
    xdt:Transform="SetAttributes(batch)">
  </compilation>
</configuration>
```

ویژگی batch وب کانفیگ انتقالی را با مقدار false مقدار دهی می‌کند.

ادامه دارد...