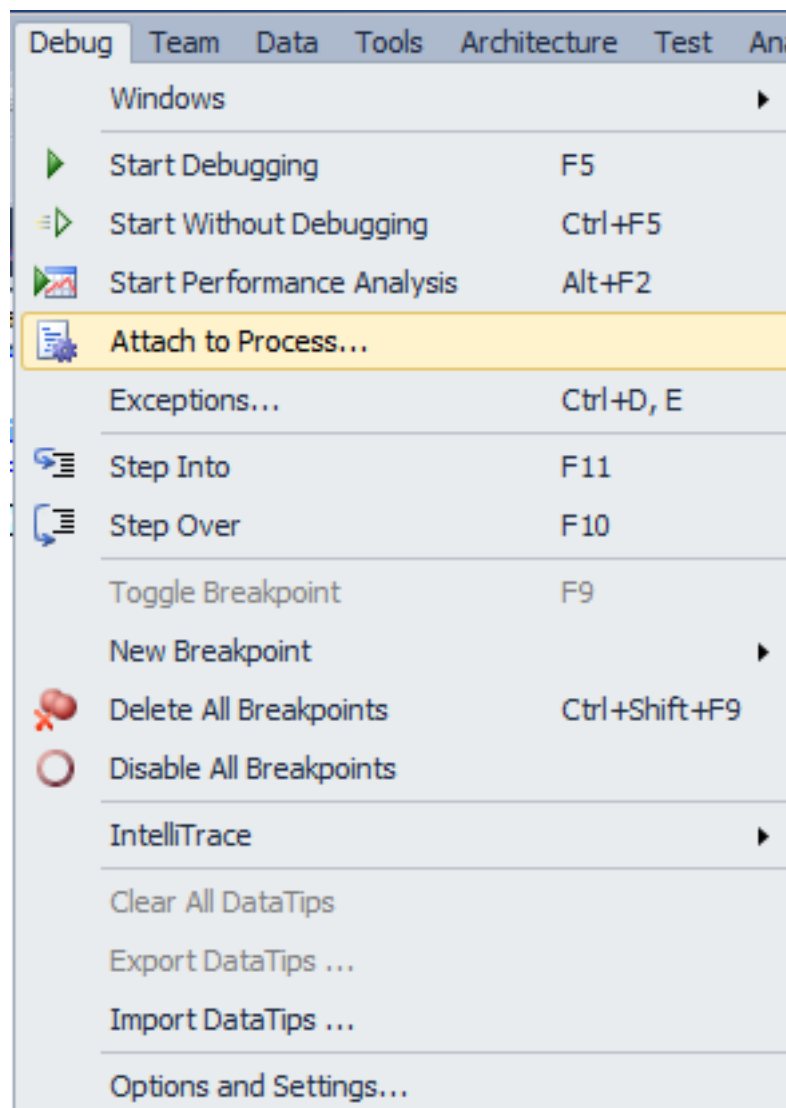


مواردی وجود دارد که نیاز به Attach کردن یک پروسس به Application خود دارید. برای این منظور باید از بین w3wp‌های موجود که IIS اجرا کرده پردازش مرتبط را یافته و آن را Attach نمایید در غیر این صورت امکان debug کردن Application مشکل خواهد بود. در این پست راه حلی برای این مورد بیان شده است .
 فرض کنید می‌خواهید از بین تعدادی پروسس یکی را برای debug کردن انتخاب نمایید .
 (برای انتخاب پروسس از منوی Debug روی Attach to Process کلیک کنید و تیک نمایش تمام پروسس‌ها را بزنید)



کدام w3wp.exe مرتبط با Application جاری من است ؟



Process	ID	Title	Type	User Name	Session
w3wp.exe	13568		T-SQL, Man...	SEIFOLLAHI\admi...	0
w3wp.exe	10568		T-SQL, Man...	SEIFOLLAHI\admi...	0
w3wp.exe	9308		T-SQL, Man...	SEIFOLLAHI\admi...	0
w3wp.exe	9280		T-SQL, Man...	SEIFOLLAHI\admi...	0
w3wp.exe	4748		T-SQL, Man...	SEIFOLLAHI\SPS_...	0
w3wp.exe	5556		T-SQL, Man...	SEIFOLLAHI\admi...	0
w3wp.exe	9204		T-SQL, Man...	IIS APPPOOL\Def...	0
w3wp.exe	7380		T-SQL, Man...	SEIFOLLAHI\admi...	0
w3wp.exe	4476		T-SQL, Man...	SEIFOLLAHI\admi...	0
WebAnalyticsServi...	4152		T-SQL, Man...	SEIFOLLAHI\admi...	0

☒ Show processes from all users ☒ Show processes in all sessions Refresh

برای پیدا کردن اینکه کدام پروسس متعلق به Application مورد نظر ماست باید از برنامه [Process Explorer](#) کمک بگیریم . پس از اجرای این برنامه روی ستون‌های آن کلیک سمت راست کنید و Select Column را انتخاب نمایید



گزینه Command Line را انتخاب نمایید و پس از OK کردن به دنبال پردازش‌های w3wp در حال اجرا بگردید .



حال می‌توانید پروسه مورد نظر خود را براحتی بیابید و شناسه پردازش را از آنجا بخوانید

Process	PID	Command Line
svchost.exe	3400	C:\Windows\system32\svchost.exe -k iissvcs
w3wp.exe	4476	c:\windows\system32\inetsrv\w3wp.exe -ap "SecurityTokenServiceApplicationPool" -v "v2.0" -f ...
w3wp.exe	7380	c:\windows\system32\inetsrv\w3wp.exe -ap "6f3e04326881487090916c07abd51b7b" -v "v2.0" -f ...
w3wp.exe	9204	c:\windows\system32\inetsrv\w3wp.exe -ap "DefaultAppPool" -v "v2.0" -f "webengine4.dll" -a ...
w3wp.exe	5556	c:\windows\system32\inetsrv\w3wp.exe -ap "TSF8080" -v "v2.0" -f "webengine4.dll" -a ...
w3wp.exe	4748	c:\windows\system32\inetsrv\w3wp.exe -ap "SPS2010 - 3000 - AP" -v "v2.0" -f "webengine4.dll" ...
w3wp.exe	9280	c:\windows\system32\inetsrv\w3wp.exe -ap "SharePoint - MySiteCollection2000" -v "v2.0" -f "we...
w3wp.exe	9308	c:\windows\system32\inetsrv\w3wp.exe -ap "SharePoint - 9090" -v "v2.0" -f "webengine4.dll" -a ...
w3wp.exe	10568	c:\windows\system32\inetsrv\w3wp.exe -ap "SharePoint Central Administration v4" -v "v2.0" -f "w...
w3wp.exe	13568	c:\windows\system32\inetsrv\w3wp.exe -ap "SharePoint - 4000" -v "v2.0" -f "webengine4.dll" -a ...

اکنون شناسه مشخص شده و می‌توانید به Debug کردن پردازش

vas.exe	5792
vssphost4.exe	12640
w3wp.exe	13568
w3wp.exe	10568
w3wp.exe	9308
w3wp.exe	9280
w3wp.exe	4748
w3wp.exe	5556
w3wp.exe	9204
w3wp.exe	7380

[موفق باشید](#)

نظرات خوانندگان

نویسنده: مهدی موسوی
تاریخ: ۱۶:۸ ۱۳۹۲/۰۱/۲۳

سلام.

روش‌های ساده‌تری هم برای اینکار وجود داره. کافیه تا اونجاییکه علاقمند هستید کدتون break بخوره، این کد رو بنویسید:

```
if (Debugger.IsAttached)
    Debugger.Break();
else
    Debugger.Launch();
```

بدین ترتیب هر وقت اجرا به این خط برسه، پنجره Visual Studio Just-In-Time Debugger باز میشه و Debugger بطور خودکار به App شما Attach میشه و ...

موفق باشید.

نویسنده: محمد باقر سیف‌اللهی
تاریخ: ۱۸:۴۷ ۱۳۹۲/۰۱/۲۳

به خاطر بعضی دست کاری هایی که روی IIS انجام داده بودم (+) VS در مود Debug قرار نمی‌گرفت و پس از فشردن F5 پیغام خطا نمایش می‌داد (با این مضمون که امکان Attach کردن به پروسس وجود ندارد).

نویسنده: مهدی
تاریخ: ۱۳:۵۱ ۱۳۹۲/۰۱/۲۴

راه حلی دیگر

```
cd c:\windows\system32\inetsrv
appcmd list wp
```

نویسنده: حمید
تاریخ: ۱۸:۳۱ ۱۳۹۲/۰۱/۲۹

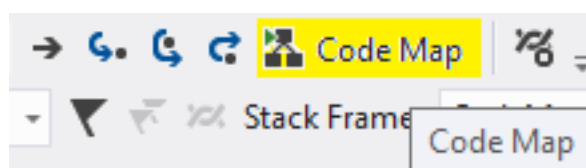
بجای Process Explorer از تسک منیجر هم میتونید استفاده کنید، ستون‌های مورد نظر رو فقط شو کنید.

Code Map چیست؟

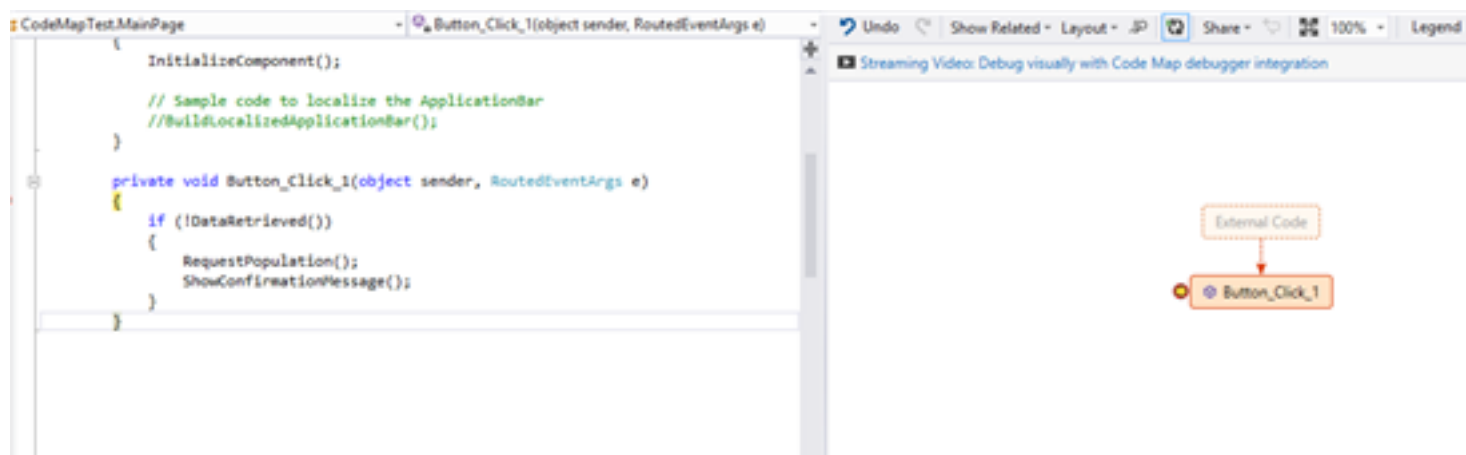
در نسخه 1 Visual Studio 2012 Ultimate Update Code Map اضافه گردید که امکان تصویر سازی، روابط کدها را فراهم می‌سازد.

در نسخه 2 Visual Studio 2012 Update Code Map را توسعه داد و با پشتیبانی از اشکال زدایی، با Code Map نیز می‌توانید قدم به قدم کد را اشکال زایی نمایید. به زبان ساده‌تر Code Map فلوچارت اجرای برنامه است که در آن ارتباط بین متودها نمایش داده شده است.

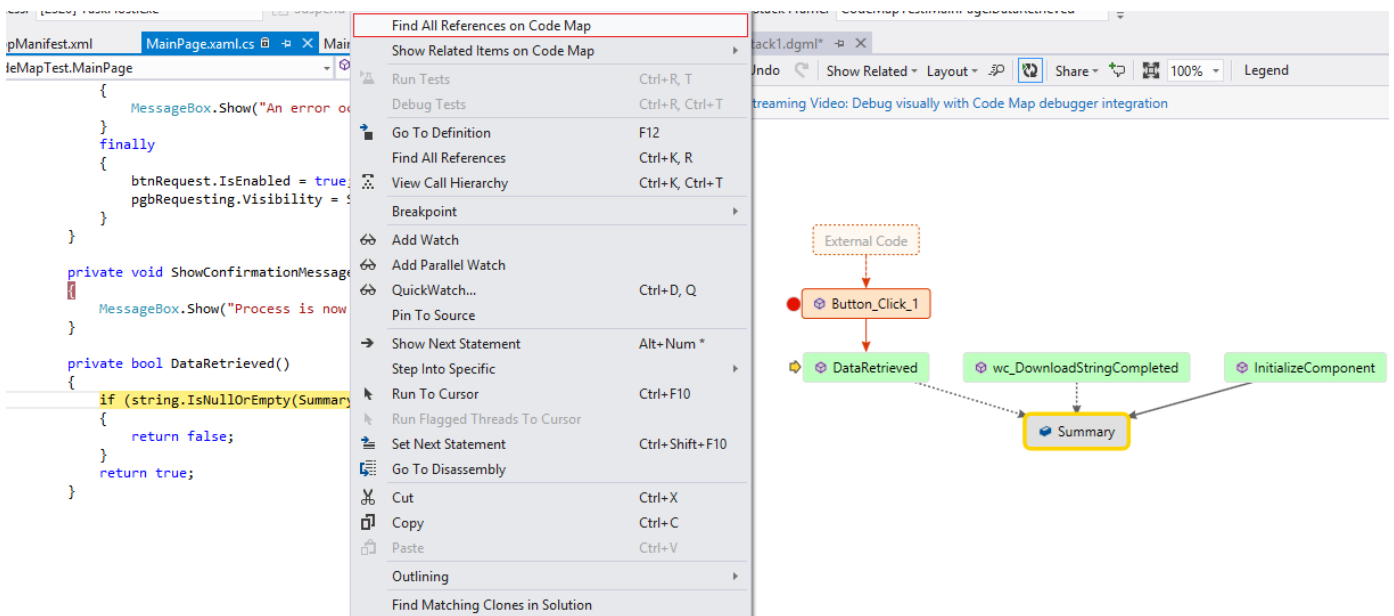
چگونه از Code Map Debugging استفاده نماییم؟



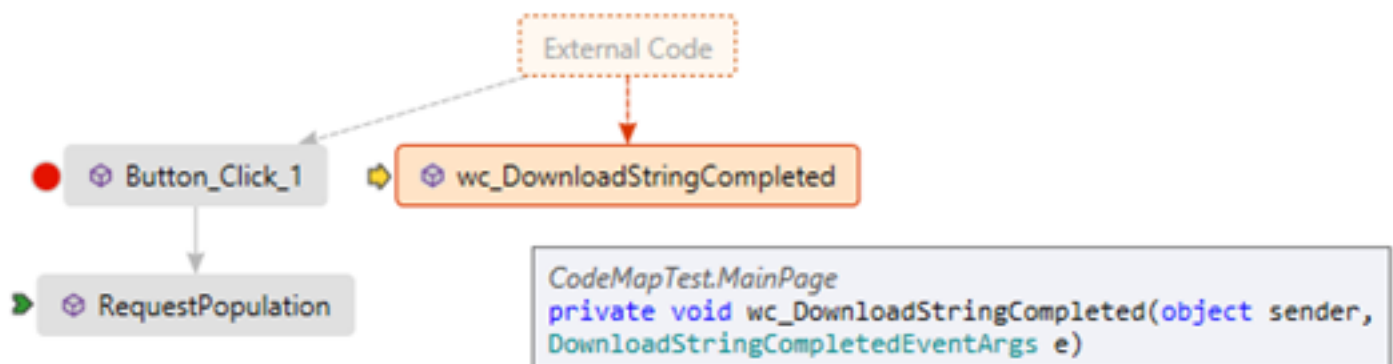
در خطی از برنامه Break Point گذاشته و برنامه را اجرا نمایید. وقتی برنامه به خط مورد نظر رسید، نوار ابزار Debug را فعال و گزینه Code Map را انتخاب کنید تا پنجره Code Map باز شده و فلوچارت کد مورد نظر را رسم نماید. نوار قرمز رنگ دور گره، قسمتی از کد که در حال حاضر در حال اشکال زدایی است را نشان می‌دهد. وقتی با کلید F11 به درون متودی می‌روید گره متناظر آن نیز رسم می‌شود.



اگر بر روی فیلد Summary راست کلیک و گزینه Find All References on Code Map را انتخاب کنید، همه ارجاع‌های که این فیلد در آنها صدا زده شده است را نمایش می‌دهد.

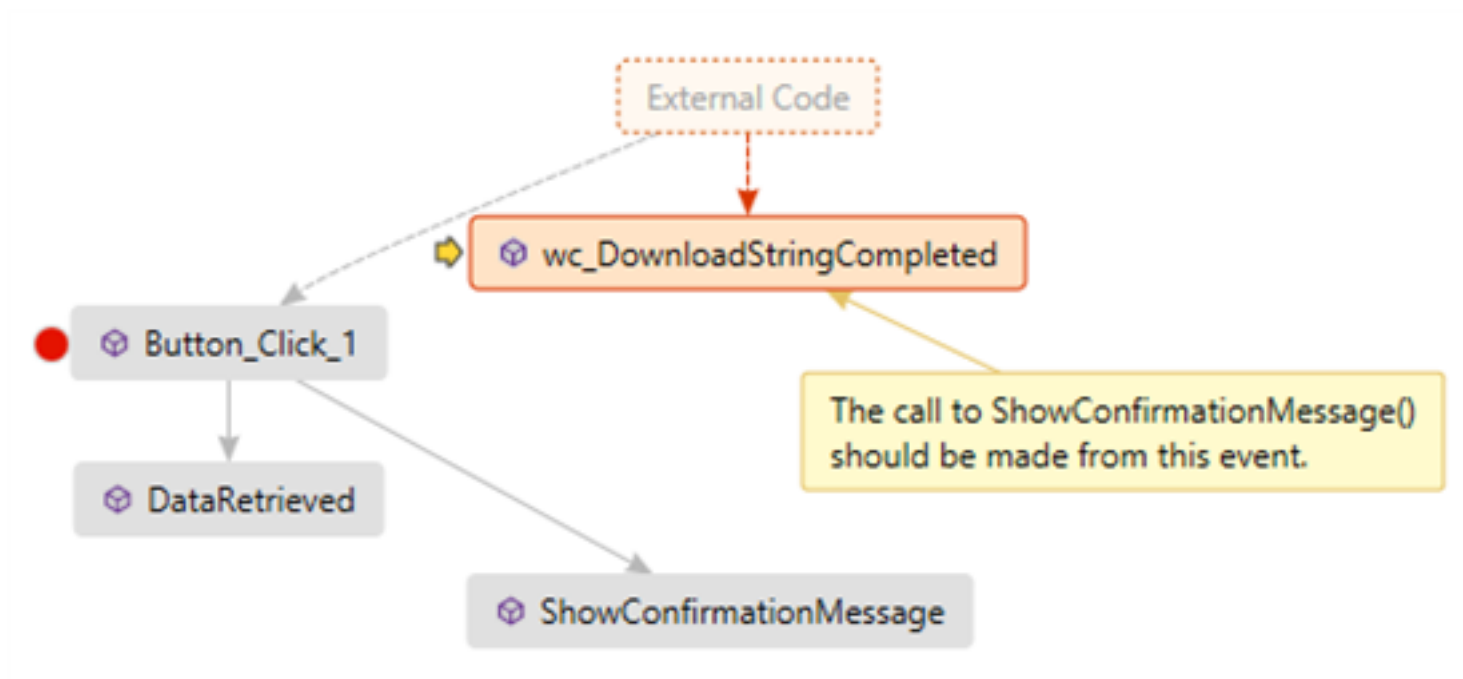


اگر بر روی هر کدام از گره‌ها موس خود را نگه دارید پنجره‌ای باز شده و اطلاعات درباره آن گره به شما نمایش می‌دهد. مثلاً برای متودها امضای متود را نمایش می‌دهد.

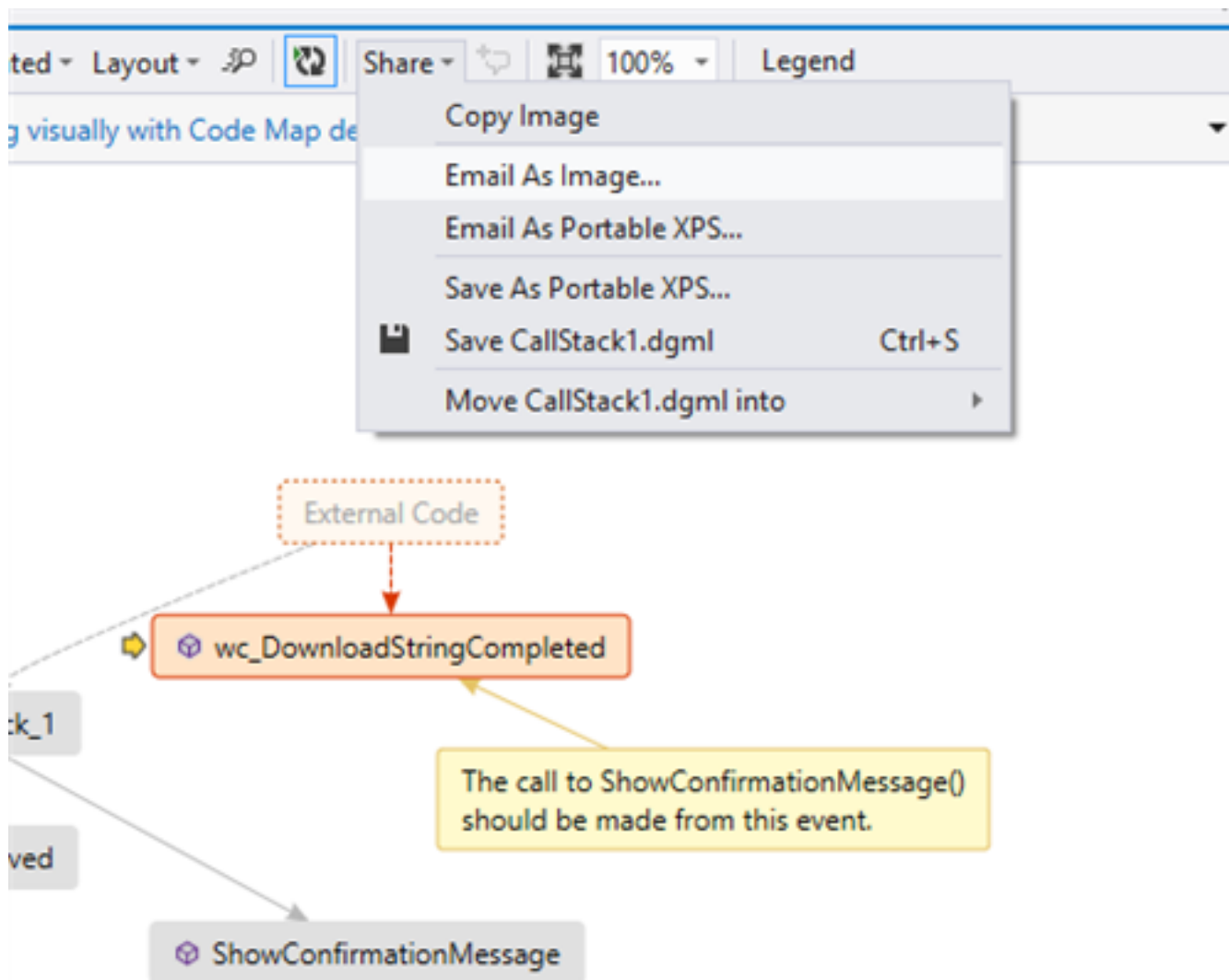


بررسی ویژگی‌های Code Map:

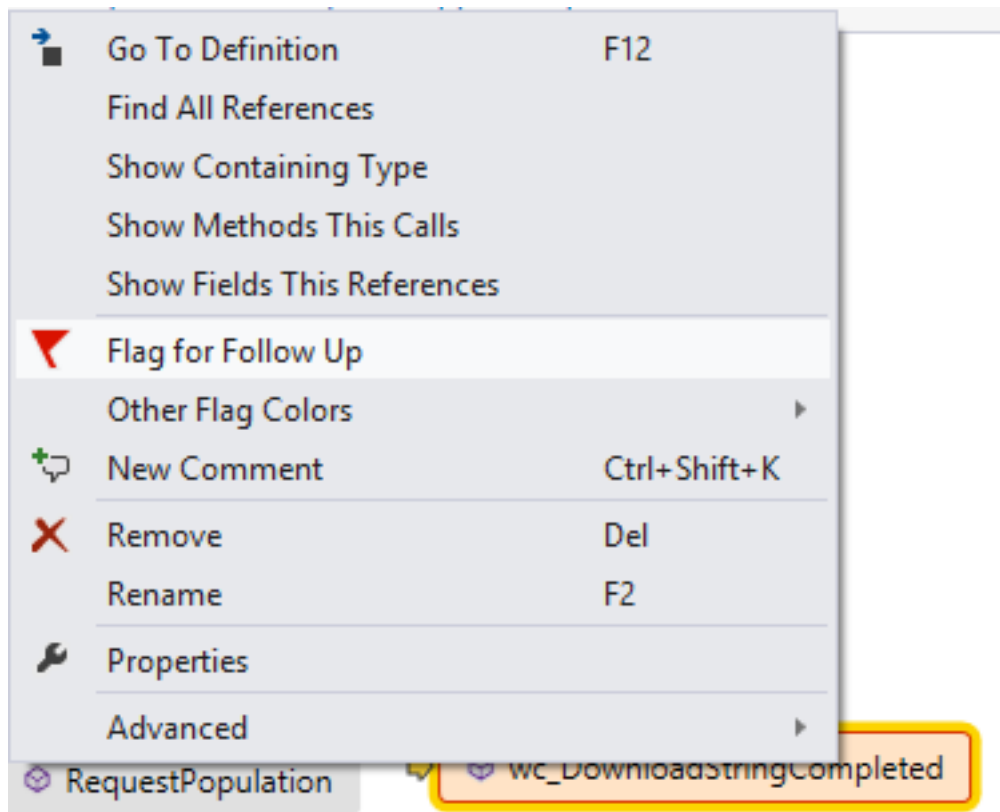
برای هر گره در صورت لزوم می‌توانید یادداشتی اضافه کنید برای این کار گره مورد نظر را انتخاب تا نوار ابزاری در کنار آن ظاهر شود و گزینه create new comment node را انتخاب و یادداشت خود را ثبت نمایید.



می توانید code map را با دیگران از طریق ایمیل به اشتراک گذارید یا آن را ذخیره نمایید. برای این کار در بالای پنجره code map گزینه share را انتخاب و یکی از آینه‌ها مورد نظر را کلیک نمایید.



علاوه بر این برای تعقیب کردن یک گره آن می‌توانید آن را با یک پرچم و تغییر رنگ آن در پنجره code map از دیگر گره‌ها مجزا و تعقیب نمایید. برای این کار بر روی گره مورد نظر راست کلیک و گزینه Flag for Follow Up را انتخاب نمایید. از گزینه Other Flag Colors رنگ پیش فرض آن را می‌توانید تغییر دهید.



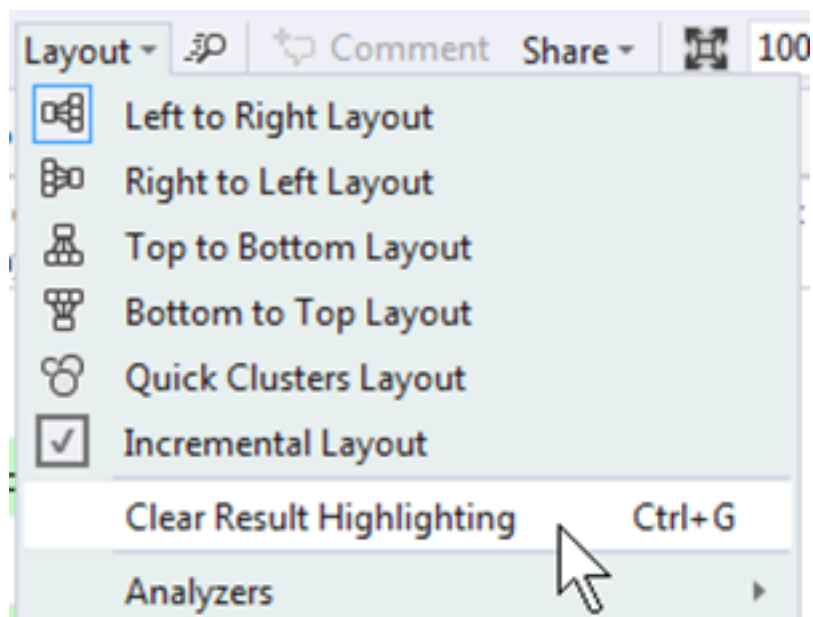
بازبینی اشاره گر ها:

در کنار هر گره علامتی نمایان می‌شود که وضعیت code amp در آن لحظه برای آن گره را نمایش می‌دهد.

- Current location of editor insertion point is in this item.
- A breakpoint is set in this item.
- ⚙ The next statement that will be executed is in this item (instruction pointer).
- ⚙ A breakpoint is set in this item. The next statement that will be executed is in this item (instruction pointer).
- A breakpoint is set in this item. Current location of editor insertion point is in this item.

تغییر نمای دیاگرام:

برای تغییر نمای گراف code map و نمایش بهتر آن از گزینه layout یکی از حالات موجود را می‌توانید انتخاب نمایید. هر وقت گره‌ای اضافه شود که قبلاً اضافه شده باشد رنگ آن سبز می‌شود برای پاک کردن این چنین گره‌های گزینه clear Result Highlighting را انتخاب نمایید.



نظرات خوانندگان

نویسنده: سی شارپ 2012
تاریخ: ۱۴:۱۴ ۱۳۹۲/۰۸/۲۲

آیا قابلیت code map در ورژن VS 2013 وجود دارد؟
لطفا راهنمایی کنید

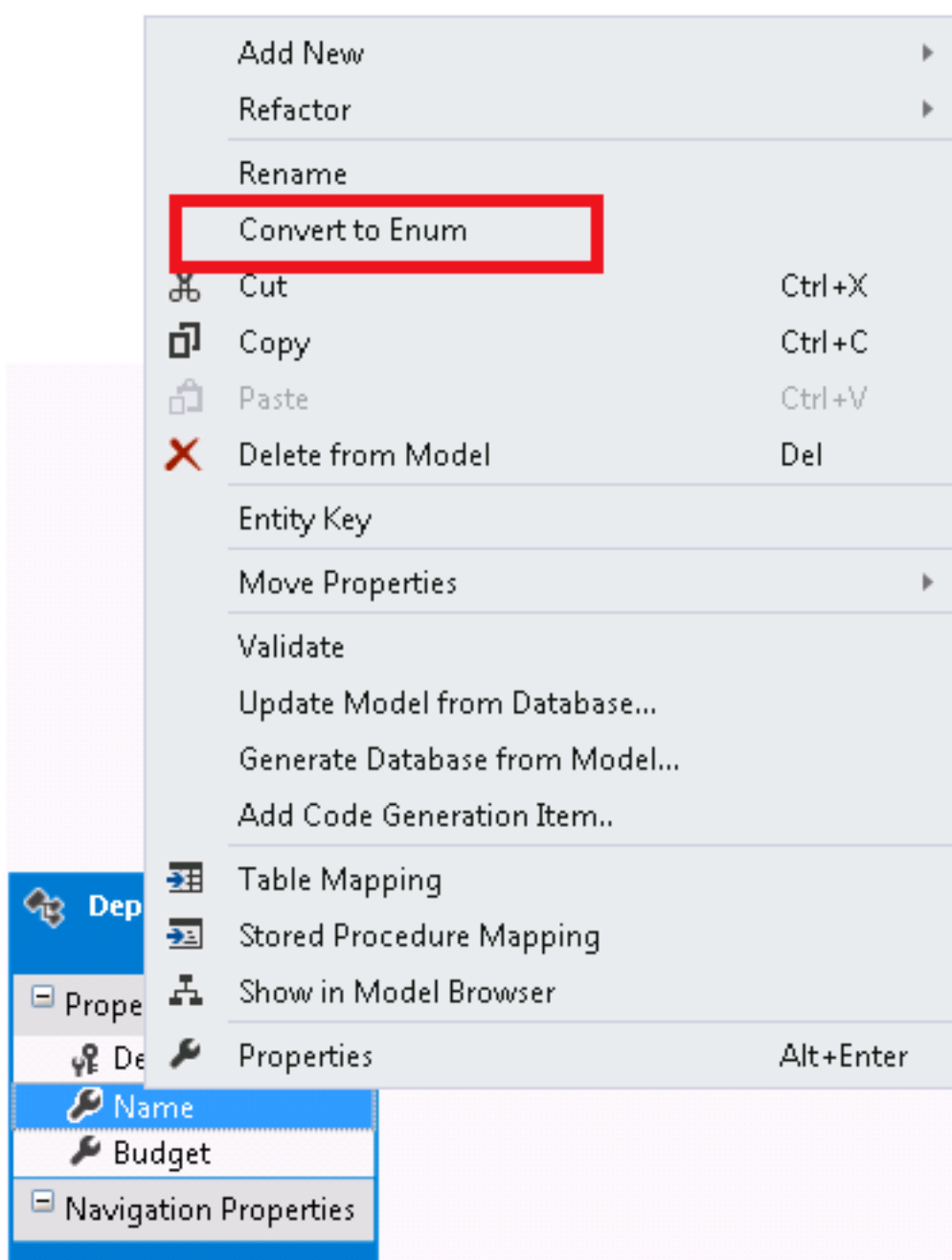
نویسنده: محسن خان
تاریخ: ۱۴:۵۱ ۱۳۹۲/۰۸/۲۲

بله. [نگارش ultimate آن این قابلیت را دارد](#) .

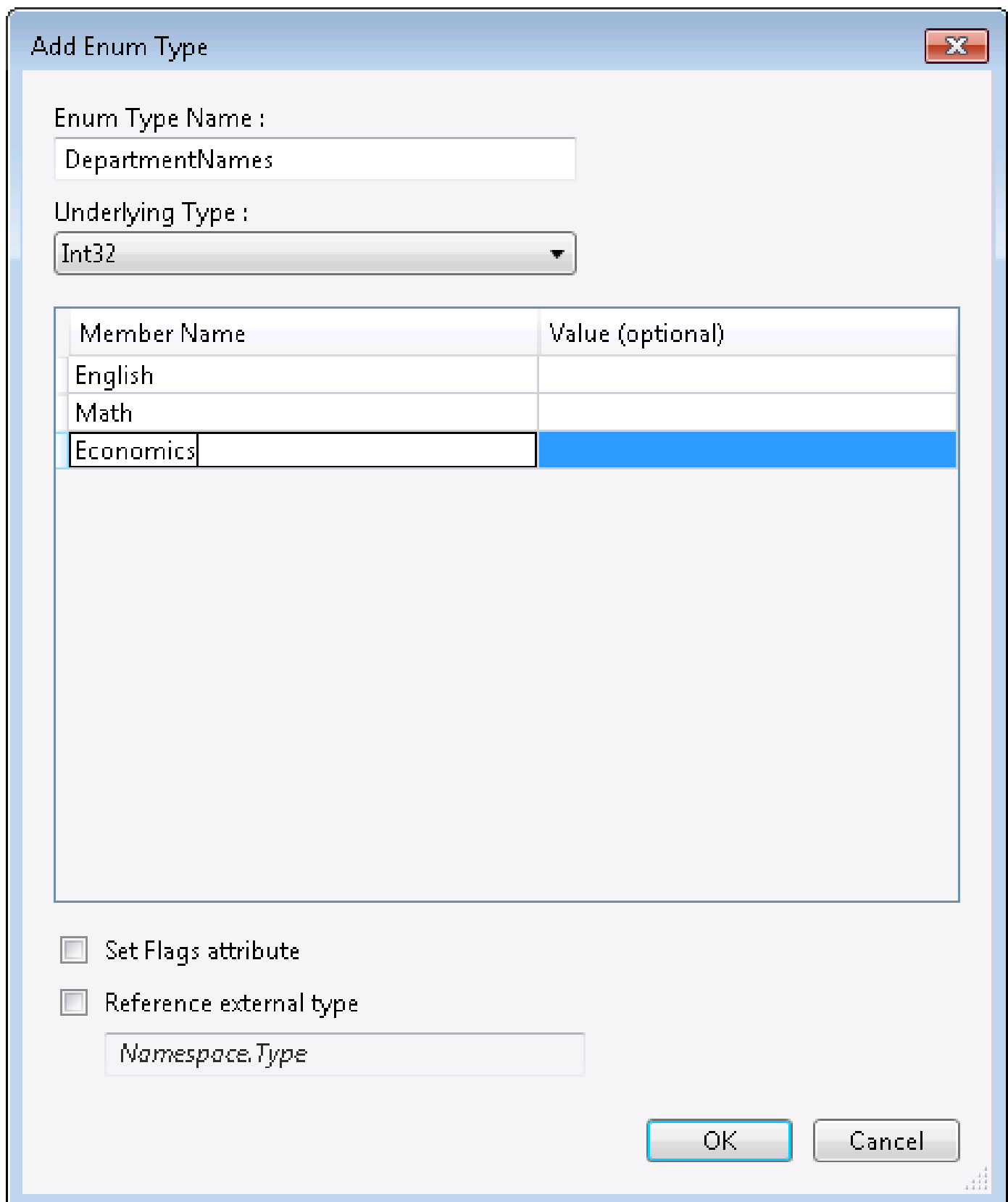
ویرایش 2012 ابزار Visual Studio جهت کار با EF امکانات جدیدی دارد که سعی دارم به طور خلاصه چند مورد آنرا توضیح دهم.

پشتیبانی از Enum

در نسخه‌های قبل از EF 5 پشتیبانی توکاری از Enumها وجود نداشت و برنامه نویس برای استفاده از آنها مجبور بود از روش‌های دیگری استفاده کند؛ مانند [^](#) استفاده کند. در نسخه 5 این امکان بر راحتی قابل اعمال است. بدین منظور کافی است: 1- در Designer بر روی خصوصییتی که قصد تبدیل آنرا به enum دارید راست کلیک کرده و گزینه Convert to enum را انتخاب کنید.



2- در پنجره Add enum ابتدا در قسمت Underlying type نوع Int32 را انتخاب کنید سپس می‌توان نام enum و اعضای آنرا و تعیین کرد.



Add Enum Type

Enum Type Name :
DepartmentNames

Underlying Type :
Int32

Member Name	Value (optional)
English	
Math	
Economics	

☐ Set Flags attribute

☐ Reference external type

Namespace.Type

OK Cancel

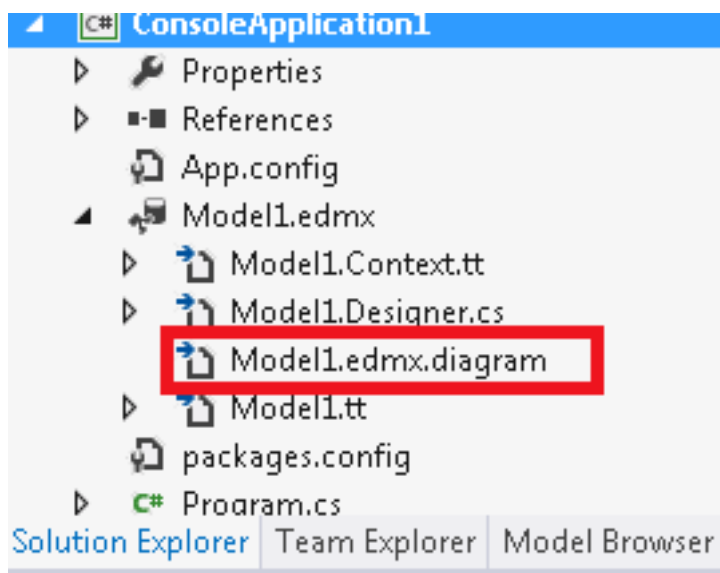
3- دکمه ok را کلیک کنید و سپس پروژه را Build کنید.

در ادامه به راحتی می‌توان از آن در برنامه به صورت زیر استفاده کرد:

```
var department = (from d in context.Departments
                  where d.Name == DepartmentNames.English
                  select d).FirstOrDefault();
```

تقسیم یک مدل در Entity Framework به چند دیاگرام

هنگامی که یک دیاگرام جدید ایجاد می‌کنید این دیاگرام به طور پیش فرض با نام Diagram1 به پوشه Diagrams اضافه می‌شود. اطلاعات مربوط به ظاهر موجودیت مانند رنگ و شکل و روابط آنها نیز در فایل با پسوند edmx.diagram قرار می‌گیرند. شما بصورت دستی نمی‌توانید این فایل را تغییر دهید چون اطلاعات آن دوباره توسط جنریتور رونویسی می‌شود. لذا تغییر در دیاگرام به روش دستی مورد اطمینان نیست!



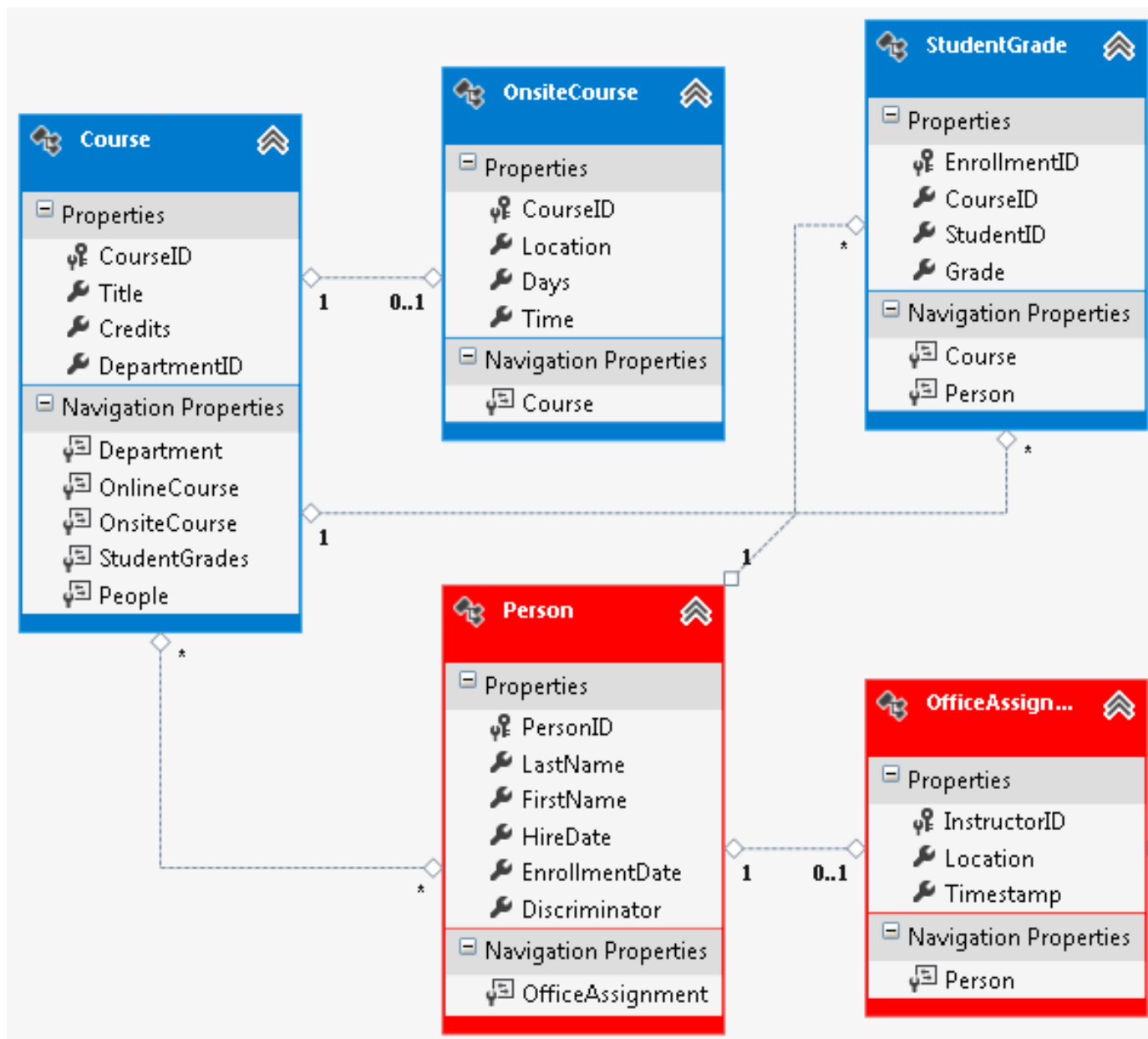
حتما برای کسانی که از EF Designer استفاده می‌کنند پیش آمده که بخواهند مدل موجودیت هایشان را بجای یک فایل در چند فایل قرار دهند. اینکار مخصوصا زمانی که تعداد موجودیت‌ها زیاد است لازم به نظر می‌رسد بعلاوه اینکه مدیریت و مرور موجودیت‌ها را در پروژه‌های بزرگ آسانتر می‌کند. خوشبختانه این امکان در Visual Studio 2012 ایجاد شده است.

بدین منظور در دیاگرام برنامه موجودیت هایی را که می‌خواهید به دیاگرام دیگری انتقال دهید با کلیک و شیفت انتخاب کنید. سپس راست کلیک کرده و گزینه Move to new Diagram را انتخاب کنید. دیاگرام جدیدی ایجاد شده و موجودیت انتخاب شده به آنجا انتقال داده می‌شود. بهتر است موجودیت هایی که برای انتقال انتخاب کرده اید به صورت یک گروه مستقل باشند یعنی با موجودیت‌های دیگر رابطه نداشته باشند.

کار انتقال به یک دیاگرام جدید را می‌توان به کمک کلیدهای Ctrl+C و Ctrl+X نیز انجام داد، باید توجه داشت در حالتی که موجودیت را کپی می‌کنید، نام موجودیت جدید با اضافه شدن یک عدد از موجودیت موجود جدا می‌شود.

تغییر رنگ موجودیت

روش دیگری که جهت متمایز و جدا کردن موجودیت‌ها می‌توان از آن استفاده کرد، تغییر رنگ آنهاست. بدین منظور پس از انتخاب موجودیت‌ها می‌توانید با تغییر مقدار Fill Color در پنجره Properties رنگ موجودیت‌های انتخاب شده را تغییر داد.



نظرات خوانندگان

نویسنده:

میثم

تاریخ:

۱۷:۵۵ ۱۳۹۲/۰۵/۱۴

پشتیبانی از Enum در linq2sql وجود ندارد؟

نویسنده:

سیروس

تاریخ:

۲۳:۱۸ ۱۳۹۲/۰۵/۱۴

به شکل مطرح شده خیر، اما بصورت دستی می‌توان اینکار را انجام داد. به [^](#) و [^](#) مراجعه کنید.

Zen Coding روشی سریع برای نوشتن کدهای HTML با استفاده از گرامر selectorهای CSS است. این روش برای اولین بار در سال 2009 توسط Sergey Chikuyonok معرفی شد و در طول این مدت، تبدیل به روشی عالی برای نوشتن کدهای HTML تکراری و یکنواخت شده است.

برای استفاده از این روش در ویژوال استادیو 2012، احتیاج به نصب افزونه‌ی محبوب و پر طرفدار Web Essentials است. این افزونه که توسط Mads Kristensen توسعه پیدا کرده است، علاوه بر Zen Coding، ویژگی‌های بسیار زیادی در زمینه‌ی توسعه وب به ویژوال استادیو اضافه می‌کند و توصیه‌ی اکید دارم که حتما این افزونه‌ی فوق العاده را نصب کنید و از امکانات دیگر آن بهره ببرید.

گرامر Zen Coding، یک سری عملگر دارد که مرجعی از همه‌ی آن‌ها را در زیر مشاهده می‌کنید.

ایجاد خاصیت id

. ایجاد خاصیت class

[] ایجاد خاصیت دلخواه

> ایجاد عنصر فرزند

+ ایجاد عنصر برادر یا خواهر (Sibling)

^ رجوع به والد

* تکثیر کننده‌ی عنصر است. هر عنصری را می‌تواند n بار تکرار کند

\$ با شمارنده جایگزین می‌شود

\$\$ با شمارنده دارای padding جایگزین می‌شود

{ } متن داخل آکولاد را در عنصر مورد نظر قرار می‌دهد

() امکان گروه بندی عبارات را می‌دهد.

lorem اطلاعات ساختگی برای آزمایش برنامه ایجاد می‌کند

مشخص است که بسیاری از عملگرها با selectorهای CSS یکسان اند. قبل از اینکه به بررسی دقیق‌تر هر کدام از این عملگرها

بپردازیم، دو مثال را با هم بررسی می‌کنیم تا به قدرت آن پی ببریم.

پس از نصب افزونه‌ی Web Essentials، یک صفحه‌ی html یا cshtml و یا هر قسمتی که بتوان در آن کد html نوشت را گشوده و

سپس کد زیر را در آن بنویسید و در نهایت کلید Tab را فشار دهید.

```
table#tblUsers.table.table-striped.table-bordered.table-hover>thead>tr>th{row}+th{Name}+th{Last
Name}+th{Operations}^tbody>tr*6>(td{row}+td{FirstName}+td{LastName}+td>button.btn.btn-
primary{Edit}+button.btn.btn-danger{Delete})
```

پس از فشردن کلید Tab، خروجی زیر را مشاهده خواهید کرد:

```
<table id="tblUsers" class="table table-striped table-bordered table-hover">
  <thead>
    <tr>
      <th>row</th>
      <th>Name</th>
      <th>Last Name</th>
      <th>Operations</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>row</td>
      <td>FirstName</td>
      <td>LastName</td>
      <td>
        <button>Edit</button>
        <button>Delete</button>
      </td>
    </tr>
  </tbody>
```

```

        <td>row</td>
        <td>FirstName</td>
        <td>LastName</td>
        <td>
            <button class="btn btn-primary">Edit</button>
            <button class="btn btn-danger">Delete</button></td>
    </tr>
    <tr>
        <td>row</td>
        <td>FirstName</td>
        <td>LastName</td>
        <td>
            <button class="btn btn-primary">Edit</button>
            <button class="btn btn-danger">Delete</button></td>
    </tr>
    <tr>
        <td>row</td>
        <td>FirstName</td>
        <td>LastName</td>
        <td>
            <button class="btn btn-primary">Edit</button>
            <button class="btn btn-danger">Delete</button></td>
    </tr>
    <tr>
        <td>row</td>
        <td>FirstName</td>
        <td>LastName</td>
        <td>
            <button class="btn btn-primary">Edit</button>
            <button class="btn btn-danger">Delete</button></td>
    </tr>
    <tr>
        <td>row</td>
        <td>FirstName</td>
        <td>LastName</td>
        <td>
            <button class="btn btn-primary">Edit</button>
            <button class="btn btn-danger">Delete</button></td>
    </tr>
</tbody>
</table>

```

به نظر پیچیده می‌آید و حتماً با خودتان می‌گویید که همان HTML خام را بنویسم، راحت‌تر هستم. شاید الان برای شما پیچیده به نظر آید ولی اگر تا انتهای این مقاله را مطالعه کنید، این قول را به شما می‌دهم که برای شما نوشتن این گونه کدها بسیار ساده خواهد بود و بعید به نظر می‌آید که شما دیگر HTML را به شیوه‌های قدیمی بنویسید، همان طور که بنده نیز با چند بار آزمون و خطا، توانستم کد بالا را بنویسم.

یک مثال دیگر می‌تواند به این شکل باشد:

```
ul>li[ng-repeat="user in Users"]>p[ng-bind="{{user.UserName}}"]+a{Details}
```

که خروجی زیر را در بر دارد:

```

<ul>
    <li ng-repeat="user in Users">
        <p ng-bind="{{user.UserName}}"></p>
        <a href="">Details</a>
    </li>
</ul>

```

از کدهای بالا این نتیجه را می‌توان گرفت که Zen Coding؛ بسیار انعطاف پذیر، ساده و قدرتمند است. در ادامه نگاهی دقیق‌تر به عملگرهای استفاده شده در کدهای بالا می‌اندازیم.

خواص id (#) و class (.)

به مانند گرامر CSS، به راحتی می‌توانید id یا class خاصی را به عنصر مورد نظر نسبت دهید:

```
<!-- Type this -->
```

```
div.container
```

```
<!-- Creates this -->
<div class="container"></div>
```

```
<!-- Type this -->
ul#userList
```

```
<!-- Creates this -->
<ul id="userList"></ul>
```

```
<!-- Type this -->
div.container#wrapper
```

```
<!-- Creates this -->
<div class="container" id="wrapper"></div>
```

```
<!-- Type this -->
button.btn.btn-primary
```

```
<!-- Creates this -->
<button class="btn btn-primary"></button>
```

نکته: اگر شما عنصر مورد نظری را که می‌خواهید بر روی آن کلاس و یا آیدی را اعمال کنید، مشخص نکنید به طور پیش فرض، Zen Coding آن عنصر را div در نظر می‌گیرد.
مثال:

```
<!-- Type this -->
.container
```

```
<!-- Creates this -->
<div class="container"></div>
```

> عناصر فرزند (Child Elements)

با استفاده از عملگر (>)، عنصر مورد نظر و عناصر فرزند درون آن را می‌توانید ایجاد کنید.

```
<!-- Type this -->
div.container>div#header
```

```
<!-- Creates this -->
<div class="container">
  <div id="header"></div>
</div>
```

```
<!-- Type this -->
div.navbar>div.navbar-inner>ul.navbar
```

```
<!-- Creates this -->
<div class="navbar">
  <div class="navbar-inner">
    <ul class="navbar"></ul>
  </div>
</div>
```

[] خواص سفارشی (Custom Attributes)

با نوشتن خاصیت سفارشی خود درون براکت [], به راحت می‌توانید آن عنصر و خاصیت سفارشی مورد نظر خود را ایجاد کنید.

```
<!-- Type this -->
div[title]
```

```
<!-- Creates this -->
<div title=""></div>
```

و یا اگر بخواهید چندین خاصیت را به صورت همزمان اعمال کنید:

```
<!-- Type this -->
input[type="text" placeholder="First Name"]

<!-- Creates this -->
<input type="text" value="" placeholder="First Name" />
```

```
<!-- Type this -->
ul[ng-repeat="user in Users"]>li[ng-model="user.FirstName"]

<!-- Creates this -->
<ul ng-repeat="user in Users">
  <li ng-model="user.FirstName"></li>
</ul>
```

{ } متن (Text)

این عملگر، متن مورد نظر شما را داخل عنصر قرار می‌دهد.

```
<!-- Type this -->
div>h1{My Title}

<!-- Creates this -->
<div>
  <h1>My Title</h1>
</div>
```

+ عناصر خواهر و برادر (Siblings Elements)

با قرار دادن عملگر + ، عناصر خواهر و برادر را ایجاد کنید.

```
<!-- Type this -->
form>input[type="text"]+input[type="checkbox"]

<!-- Creates this -->
<form>
  <input type="text" value="" />
  <input type="checkbox" value="" />
</form>
```

```
<!-- Type this -->
div#header>img+h1{Title}

<!-- Creates this -->
<div id="header">
  <img src="" alt="" />
  <h1>Title</h1>
</div>
```

^ بالا رفتن از عناصر (Climbing Elements)

این عملگر تقریباً برعکس عملگر > عمل می‌کند. با عملگر ^ شما می‌توانید به والد یک عنصر تا هر تعداد سطحی که بخواهید برسید. اگر شما به طور مثال بخواهید که دو سطح به والد بالاتر برگردید، باید از ^^ استفاده کنید.

```
<!-- Type this -->
table>thead>tr>th{row}^^tbody>tr>td{row1}

<!-- Creates this -->
<table>
  <thead>
```

```

        <tr>
            <th>row</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>row1</td>
        </tr>
    </tbody>
</table>

```

* تکثیر عناصر (Elements Multiplication)

با عملگر *, هر تعداد عنصر را که می‌خواهید تکثیر کنید.

```

<!-- Type this -->
ul>li*3>p{Hello}

<!-- Creates this -->
<ul>
    <li>
        <p>
            Hello
        </p>
    </li>
    <li>
        <p>
            Hello
        </p>
    </li>
    <li>
        <p>
            Hello
        </p>
    </li>
</ul>

```

```

<!-- Type this -->
(div.container>h1{title}+div.content{Some Text Here})*2

<!-- Creates this -->
<div class="container">
    <h1>title</h1>
    <div class="content">
        Some Text Here
    </div>
</div>
<div class="container">
    <h1>title</h1>
    <div class="content">
        Some Text Here
    </div>
</div>

```

\$ شماره گذاری آیتم‌ها (Item Numbering)

هنگامی که شما با کمک *, از روی یک عنصر n عنصر را تکثیر می‌کنید، می‌توانید با استفاده از عملگر \$, به عدد متناظر با آن حلقه‌ی تکرار دست پیدا کنید.

```

<!-- Type this -->
ul>li*2>p{item $}

<!-- Creates this -->
<ul>
    <li>
        <p>
            item 1
        </p>
    </li>
    <li>

```

```

        <p>
            item 2
        </p>
    </li>
</ul>

```

نکته: استفاده از چند عملگر \$، باعث قرار گرفتن همان تعداد صفر، پیش از عدد می‌شود.

```

<!-- Type this -->
ul>li*2>p{item $$$}

<!-- Creates this -->
<ul>
    <li>
        <p>
            item 001
        </p>
    </li>
    <li>
        <p>
            item 002
        </p>
    </li>
</ul>

```

() گروه بندی (Grouping)

گروه بندی یکی از امکانات فوق العاده‌ی Zen Coding است که به شما امکان نوشتن عبارات پیچیده را می‌دهد. کاربرد آن در مثال زیر کاملا واضح است، و به راحتی با یک عبارت، کل ساختار Dom را پیاده سازی شده است.

```

<!-- Type this -->
div>(header>div)+section>(ul>li*2>a)+footer>(div>span)

<!-- Creates this -->
<div>
    <header>
        <div></div>
    </header>
    <section>
        <ul>
            <li><a href=""></a></li>
            <li><a href=""></a></li>
        </ul>
        <footer>
            <div>
                <span></span>
            </div>
        </footer>
    </section>
</div>

```

Lorem داده‌های ساختگی (Dummy Data)

حتما بارها و بارها این اتفاق افتاده است که برای تست قالب برنامه‌ی خود، آن را با داده‌های ساختگی پر کرده باشید. معمولا شما این کار را به صورت دستی انجام می‌دادید. اما این بار عبارت lorem را نوشته و کلید Tab را فشار دهید. به طور پیش فرض برای شما 30 کلمه می‌نویسد. اگر تعداد مشخصی کلمه می‌خواهید کافی است تعداد آن را جلوی کلمه‌ی lorem بنویسید. برای مثال lorem4.

حتی از lorem می‌توانید در عبارات Zen Coding استفاده کنید:

```

<!-- Type this -->
div>(h1>lorem5)+(h3>lorem3)

<!-- Creates this -->
<div>
    <h1>Lorem ipsum dolor sit amet.</h1>
    <h3>Lorem ipsum dolor.</h3>
</div>

```

نظرات خوانندگان

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۵/۲۳ ۷:۵۸

با عرض پوزش؛ گویا در انتقال کدها به سایت، کدهای html دارای خاصیت class حذف شده بودند، که الان به صورتی دستی اصلاح کردم.

نویسنده: emad
تاریخ: ۱۳۹۲/۰۵/۲۳ ۱۳:۴۵

<http://visualstudiogallery.msdn.microsoft.com/6ed4c78f-a23e-49ad-b5fd-369af0c2107f>

اینم لینک برا vs2010

نویسنده: R.M
تاریخ: ۱۳۹۲/۰۵/۲۳ ۱۹:۵۷

ظاهرا با Resharper سازگار نیست. درسته؟

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۵/۲۳ ۲۳:۱۴

مشکلش دقیقا چیه؟

من که با visual studio2012 update3 و resharper 8 مشکلی ندارم.
احتمالا از نسخه‌ی reshaper هست. نسخه‌ی 8 نباید مشکلی داشته باشه.