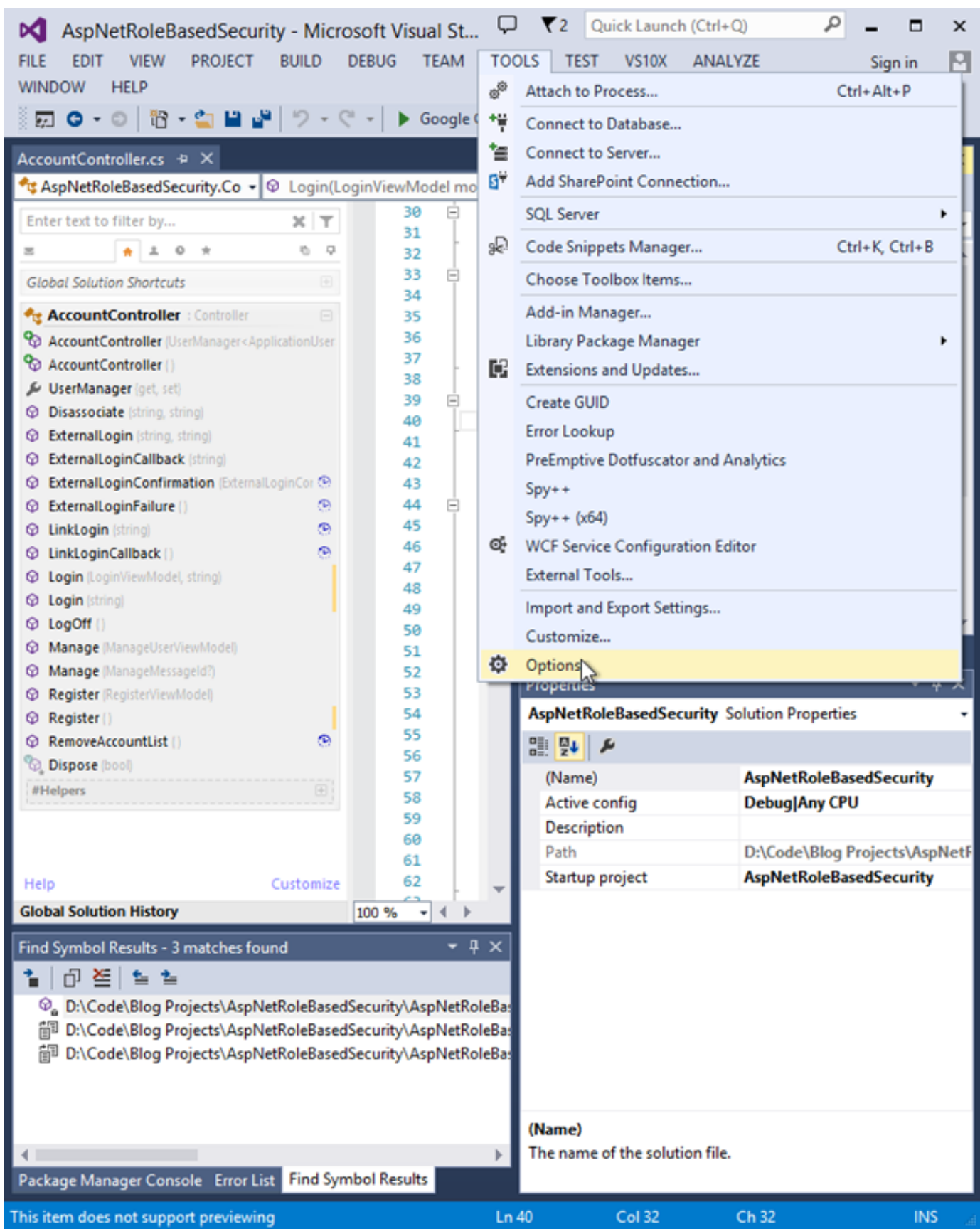


ابزار NuGet بسیار کار آمد و مفید است. یکی از مشکلات رایج هنگامی پیش می آید که پروژه را به همراه بسته های نصب شده به سورس کنترل push می کنید. با این کار حجم زیادی از فایل ها را به مخزن سورس کنترل آپلود می کنید و هنگام clone کردن پروژه توسط هر شخصی، این اطلاعات باید دریافت شوند. بدتر از این هنگامی است که برخی از بسته ها از سورس حذف می شوند و باید به اعضای تیم پروژه اطلاع دهید که چه بسته هایی باید دریافت و نصب شوند.

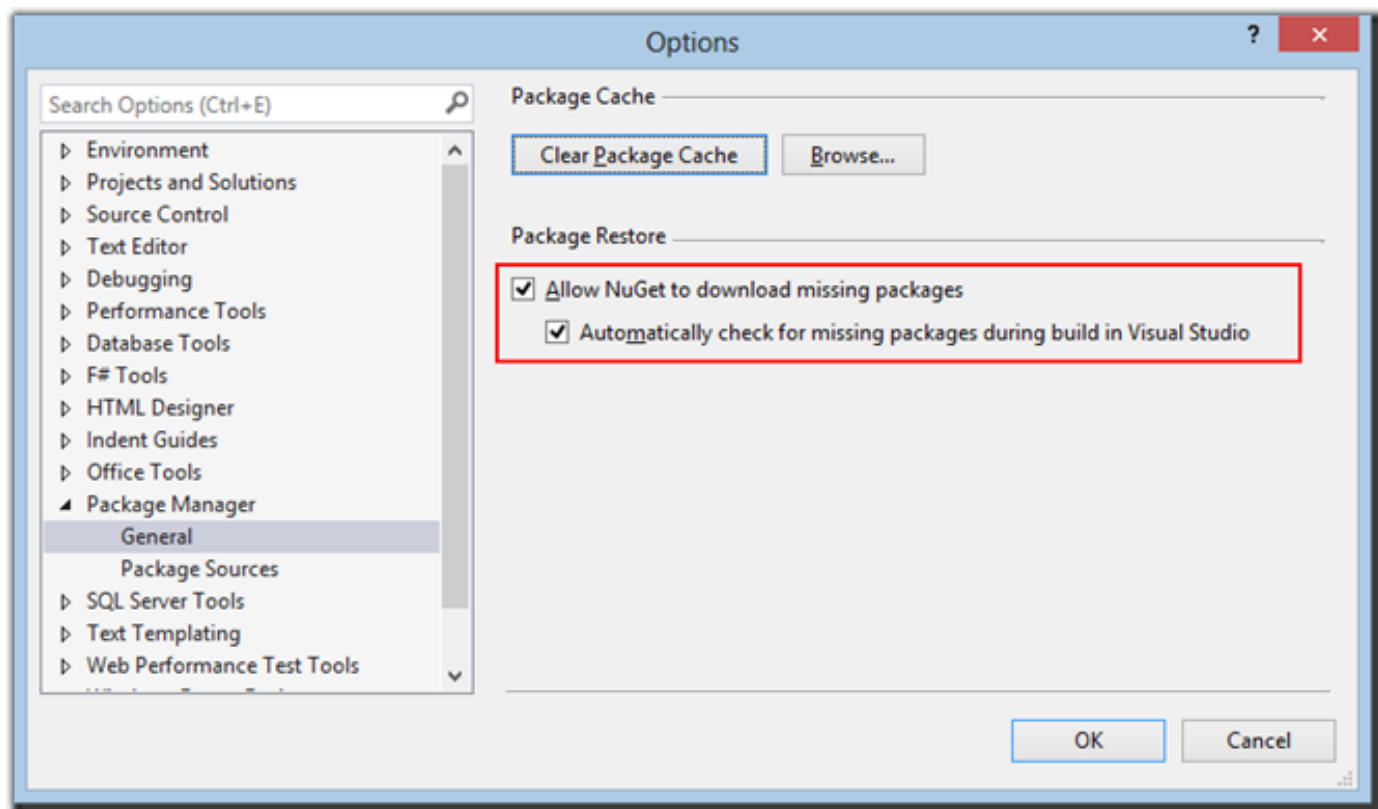
برای رفع این موارد به [NuGet Package Restore](#) وارد شوید.

به ویژوال استودیو اجازه دهید بسته های NuGet را در صورت لزوم احیا کند

پیش از آنکه بتوانیم از قابلیت [Package Restore](#) استفاده کنیم باید آن را روی ماشین خود فعال کنیم. این کار روی هر ماشین باید انجام شود (per-machine requirement). بدین منظور به منوی Package Manager -> Options -> Tools بروید.



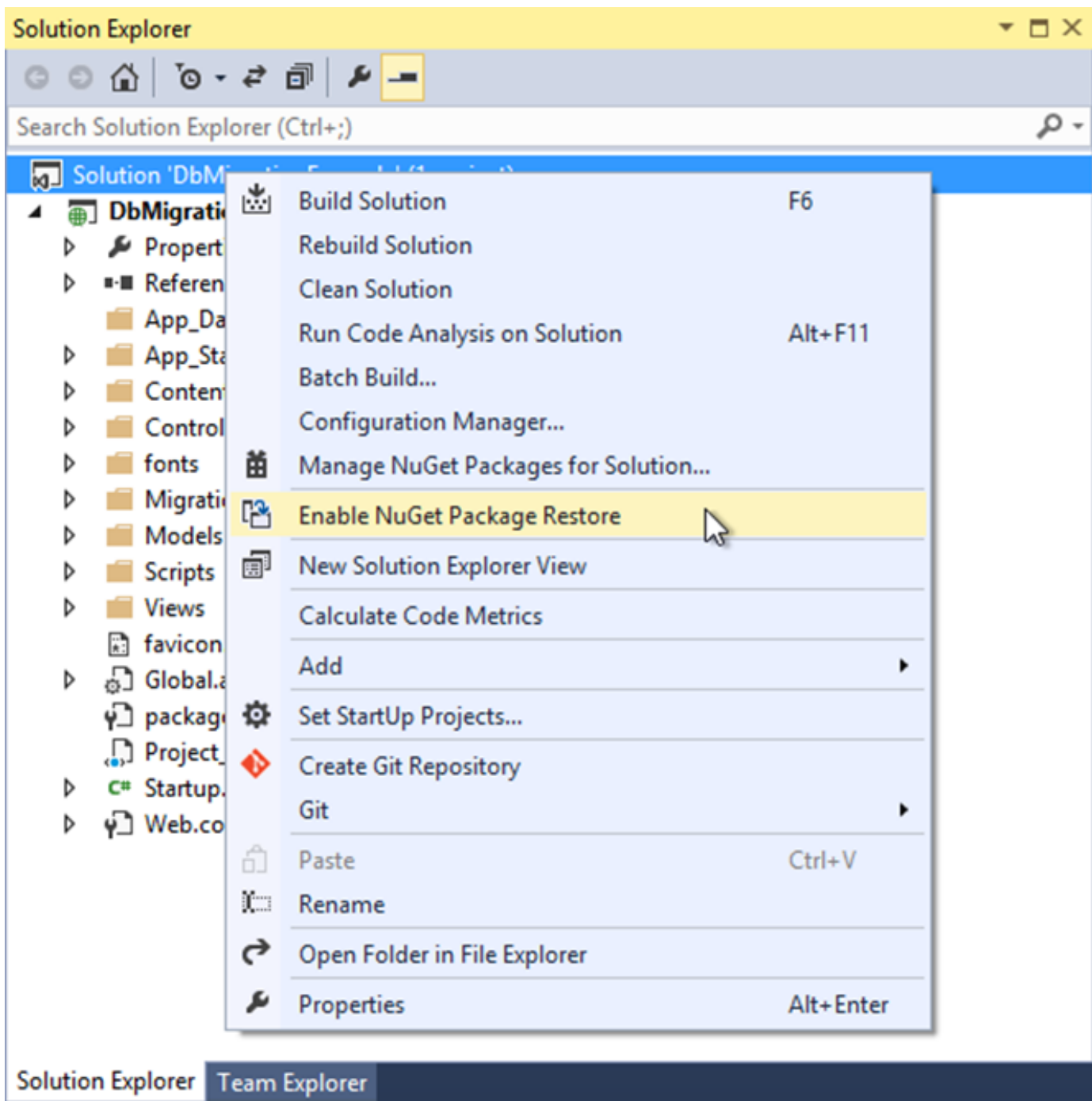
در دیالوگ باز شده تنظیمات مربوطه را مانند تصویر زیر بروز رسانی کنید.



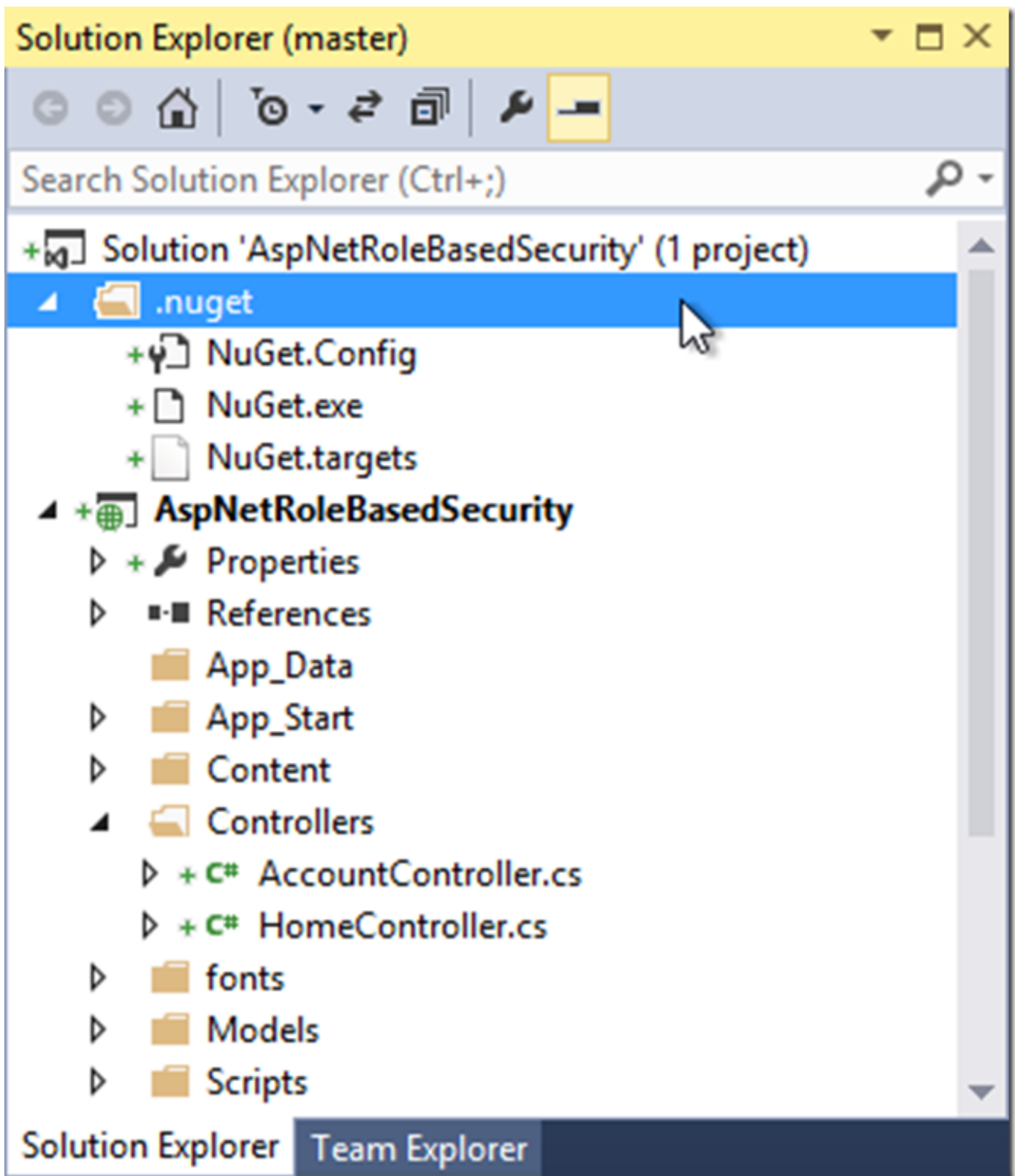
حال که ماشین ما برای بازیابی خودکار بسته‌های NuGet پیکربندی شده است، باید این قابلیت را برای Solution مورد نظر هم فعال کنیم.

فعال سازی NuGet Package Restore برای پروژه‌ها

بدین منظور روی Solution کلیک راست کنید و گزینه Enable Package Restore را انتخاب نمایید.



این کار ممکن است چند ثانیه زمان ببرد. پس از آنکه ویژوال استودیو پردازش های لازم را انجام داد، می توانید ببینید که پوشه جدیدی در مسیر ریشه پروژه ایجاد شده است.



همانطور که می بینید فایلی با نام NuGet.exe در این پوشه قرار دارد که باید به سورس کنترل آپلود شود. هنگامیکه شخصی پروژه شما را از سورس کنترل دریافت کند و بخواهد پروژه را Build کند، بسته های مورد نیاز توسط این ابزار بصورت خودکار دریافت و نصب خواهند شد.

مرحله بعد حذف کردن تمام بسته های NuGet از سورس کنترل است. برای اینکار باید فایل *gitignore* را ویرایش کنید. فرض بر

این است که سورس کنترل شما Git است، اما قواعد ذکر شده برای دیگر فریم ورک ها نیز صادق است. تنها کاری که باید انجام دهید این است که به سورس کنترل خود بگویید چه چیزهایی را در بر گیرد و از چه چیزهایی صرفنظر کند.

ویرایش فایل `gitignore` برای حذف بسته ها و شامل کردن `NuGet.exe`

یک پروژه معمولی ASP.NET MVC 5 که توسط قالب استاندارد VS 2013 ایجاد می شود شامل 161 فایل از بسته های مختلف می شود (در زمان تالیف این پست). این مقدار قابل توجهی است که حجم زیادی از اطلاعات غیر ضروری را به مخزن سورس کنترل اضافه می کند. با استفاده از نسخه پیش فرض فایل `gitignore` (یا فایل های مشابه دیگر برای سورس کنترل های مختلف مثل TFS) تعداد فایل هایی که در کل به مخزن سورس کنترل ارسال می شوند بیش از 200 آیتم خواهد بود. قابل ذکر است که این تعداد فایل شامل فایل های اجرایی (binary) و متعلق به ویژوال استودیو نیست. به بیان دیگر نزدیک به 75% از فایل های یک پروژه معمولی ASP.NET MVC 5 که توسط VS 2013 ساخته می شود را بسته های NuGet تشکیل می دهد، که حالا می توانند بجای ارسال شدن به مخزن سورس کنترل، بصورت خودکار بازبازی و نصب شوند.

برای حذف این فایل ها از سورس کنترل، فایل `gitignore` را ویرایش می کنیم. اگر از سورس کنترل های دیگری استفاده میکنید نام این فایل `hgignore` یا `tfsignore` یا غیره خواهد بود. محتوای فایل شما ممکن است با لیست زیر متفاوت باشد اما جای نگرانی نیست. تنها تغییرات اندکی بوجود خواهیم آورد و مابقی محتویات فایل مهم نیستند.

چشم پوشی از پوشه Packages

فایل `gitignore` را باز کنید و برای نادیده گرفتن پوشه بسته های NuGet در سورس، خط زیر را به آن اضافه کنید.

```
packages*/
```

استثنای برای در نظر گرفتن `NuGet.exe` ایجاد کنید

به احتمال زیاد فایل `gitignore` شما از فایل هایی با فرمت `exe` چشم پوشی می کند. برای اینکه بسته های NuGet بتوانند بصورت خودکار دریافت شوند باید استثنای تعریف کنیم. فایل `gitignore` خود را باز کنید و به دنبال خط زیر بگردید.

```
*.exe
```

سپس خط زیر را بعد از آن اضافه کنید. دقت داشته باشید که ترتیب قرارگیری این دستورات مهم است.

```
*.exe  
!NuGet.exe
```

دستورات بالا به Git می گوید که فایل های `exe` را نادیده بگیرد؛ اما برای فایل `NuGet.exe` استثناء قائل شود. انجام مرحله بالا انتخابی (optional) است. اگر کسی که پروژه را از مخزن سورس کنترل دریافت می کند قابلیت Package Restore را روی Solution فعال کند ابزار `NuGet.exe` دریافت می شود. اما با انجام این مراحل دیگر نیازی به این فعالسازی نخواهد بود، پس در کار اعضای تیم هم صرفه جویی کرده اید.

اطلاع رسانی به اعضای تیم و مشتریان بالقوه

دیگر نیاز نیست بسته های NuGet را به مخزن سورس کنترل ارسال کنیم. اما باید به مخاطبین خود اطلاع دهید تا پیکربندی های لازم برای استفاده از قابلیت Package Restore را انجام دهند (مثلا در فایل `README.txt` پروژه).

نظرات خوانندگان

نویسنده: Ara

تاریخ: ۲۲:۲۷ ۱۳۹۳/۰۲/۰۷

یک کار خوب داخل دیگه اینه که یک Local Package Source در شرکت داشته باشیم که دچار گیر کردن گاه به گاه ،مشکلات nuget تو ایران که بعضی وقتها گیر می‌کنه نیافتیم و package با سرعت بالا نصب بشوند

نویسنده: آرمین ضیاء

تاریخ: ۲۳:۲ ۱۳۹۳/۰۲/۰۷

میتونه رویکرد مناسبی باشه اما بهتر است که بسته‌های مورد نیاز از سرویس‌های معتبر مثل خود NuGet.org دریافت بشن تا انتشارات جدید در دسترس باشند. اگر منظورتون رو درست فهمیده باشم با این رویکرد یک کپی محلی از بسته‌ها خواهیم داشت. در صورتی که بسته‌ها نیاز به بروز رسانی داشته باشند نهایتا باز نیاز به دریافت پکیج‌ها از اینترنت است.

نویسنده: مسعود دانش پور

تاریخ: ۱۰:۱۱ ۱۳۹۳/۰۲/۰۸

به نظر بنده اگر به تایتل این نوشته مفید به "بیرون نگاه داشتن پکیج‌های NuGet از سورس کنترل Git" تغییر کنه بسیار عالی‌تر خواهد شد.

نویسنده: آرمین ضیاء

تاریخ: ۱۷:۴۹ ۱۳۹۳/۰۲/۰۸

با تشکر، عنوان بروز رسانی شد.

نویسنده: صابر فتح الهی

تاریخ: ۲۳:۵۸ ۱۳۹۴/۰۶/۲۱

سلام

اما من توی tfه هرکاری میکنم امکانش نیست فولدر packages چک این نشه. یه فایل ساختم توی ریشه پروژه به نام tfignore. و تنظیمات بهش دادم اما اعمال نشد.

نویسنده: عزیزخانی

تاریخ: ۱۳:۳۱ ۱۳۹۴/۰۶/۲۲

در ابتدا بهتر است با فایل‌های packages.config و repositories.config آشنا شویم. فایل packages.config در ازای هر پروژه ایجاد می‌شود و در این فایل اطلاعات package هایی که به پروژه اضافه شده اند نگهداری می‌شوند.

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Microsoft.Owin" version="3.0.0" targetFramework="net45" />
  <package id="RavenDB.Client" version="2.0.2375" targetFramework="net45" />
</packages>
```

فایل repositories.config در فولدر packages در مسیر اصلی solution وجود دارد، در این فایل مسیر فایل‌های packages.config هر پروژه نگهداری می‌شود.

```
<?xml version="1.0" encoding="utf-8"?>
<repositories>
  <repository path="..\Application\Test\packages.config" />
</repositories>
```

```
<repository path="..\ViewModel\Test\packages.config" />
</repositories>
```

- 1- ابتدا فولدر packages را از سورس کنترل حذف کنید، checkin کنید.
حال هر شخصی که پروژه را get میکند نیاز است با توجه به مطالب و روش های گفته شده در بالا -به ویژوال استودیو اجازه دهید بسته های NuGet را در صورت لزوم احیا کند- یا -فعال سازی NuGet Package Restore برای پروژه ها- package ها را دریافت کند.
بعد از انجام این کار فولدر packages و فایل repositories.config در local هر کاربر ایجاد می شوند.
توجه شود این فولدر و محتویات آن از طریق Add Item to folder دوباره به سورس کنترل اضافه نشود.
- 2- یا می توانید ابتدا محتویات فولدر packages به غیر از فایل repositories.config را از سورس کنترل پاک کنید (فولدرها و فایل های package ها). به فایل repositories.config برای مسیر فایل های packages.config هر پروژه نیاز داریم .
شما فولدر packages و فایل repositories.config را checkin کنید (در فولدر packages جز فایل repositories.config فایل یا فولدر دیگری وجود نداشته باشد).
و در هر بار Build کردن موجود بودن فایل های package دوباره چک می شوند و اگر موجود نباشند، دریافت می شوند.
نکته مهم این است که اگر Build definition تعریف کرده باشید نیاز به تنظیمات در سرور build برای دریافت package دارید.