**Is ASP.NET MVC 5 incompatible with the WebMatrix SimpleMembershipProvider?**

Posted by: [admin](https://exceptionshub.com/author/admin) November 25, 2017 [Leave a comment](https://exceptionshub.com/is-asp-net-mvc-5-incompatible-with-the-webmatrix-simplemembershipprovider.html#respond)

Questions:

We have an existing application that was build on ASP.NET MVC 4 & Web API. The admin parts of the site use Simple Membership. I’m interested in upgrading the application to MVC 5 / Web API 2, to take advantage of some of the new features that have been added. But it looks like they might be incompatible.

Specifically, after installing the RC packages from NuGet into one of the projects in my solution, and [updating the web.config information](http://www.asp.net/mvc/tutorials/mvc-5/how-to-upgrade-an-aspnet-mvc-4-and-web-api-project-to-aspnet-mvc-5-and-web-api-2), the application starts dying during startup on the line that calls WebSecurity.InitializeDatabaseConnection(), with this exception:

[MethodAccessException: Attempt by security transparent method 'WebMatrix.WebData.PreApplicationStartCode.OnConnectionOpened(System.Object, WebMatrix.Data.ConnectionEventArgs)' to access security critical method 'System.Web.WebPages.HttpContextExtensions.RegisterForDispose(System.Web.HttpContextBase, System.IDisposable)' failed.]

WebMatrix.WebData.PreApplicationStartCode.OnConnectionOpened(Object sender, ConnectionEventArgs e) +70

WebMatrix.Data.Database.OnConnectionOpened() +70

WebMatrix.Data.Database.EnsureConnectionOpen() +51

WebMatrix.Data.Database.QueryValue(String commandText, Object[] args) +63

WebMatrix.WebData.DatabaseWrapper.QueryValue(String commandText, Object[] parameters) +13

WebMatrix.WebData.SimpleMembershipProvider.GetUserId(IDatabase db, String userTableName, String userNameColumn, String userIdColumn, String userName) +206

WebMatrix.WebData.SimpleMembershipProvider.ValidateUserTable() +87

Other projects in the same solution using Simple Membership that I have *not* upgraded continue to work just fine.

Googling around for more information turns up lots of hits for that exception, of course, but nothing particular to WebMatrix.

FWIW: I know that Microsoft has introduced [(yet another) membership and identity solution](http://blogs.msdn.com/b/webdev/archive/2013/06/27/introducing-asp-net-identity-membership-system-for-asp-net-applications.aspx), but unless there’s a way to use that with the existing Simple Membership tables, or a seamless migration path for all of our existing user data, that’s not really an option for us.

**UPDATE (11 Oct)**

I just tried this again with a fresh checkout of the current trunk of our app. I’m using Visual Studio 2012, but otherwise followed the instructions from MS for upgrading an existing project. After updating to MVC 5 / Web API 2 / EF 6, the app started up an ran just fine.

There were no explicit trust requirements in the web.config to remove. I added the code from [this question](https://stackoverflow.com/questions/1064274/get-current-asp-net-trust-level-programmatically) to Global.asax.cs, and it reports that the app is running with full trust (in IIS Express, just F5-ed from VS).

Re-adding the same call to InitializeDatabaseConnection(), it starts dying with the exact same exception.

**SOLUTION (28 Oct)**

Trying the solution in @Kevin’s update from Friday, I found that it works. It was really strange to me that adding this apparently unrelated package would solve these security issues, and even *more* strange after I removed the package from my solution, and it *kept working*.

Taking a closer look at what was happening, I realized that the reason why this fixes the behavior is quite simple: the Microsoft.AspNet.WebHelpers package has two dependencies that were being added to my solution: Microsoft.AspNet.WebPages.Data and Microsoft.AspNet.WebPages.WebData. **Microsoft has moved the WebMatrix classes into new packages.**

So added the helpers package fixed the problem, not because of anything it was doing, but because **it was causing updated versions of the broken assemblies to be added** to my solution. The solution to the initial incompatibility, then, is to install these new packages when updating everything else from NuGet:

Install-Package Microsoft.AspNet.WebPages.WebData

**UPDATE (13 May 2015)**

It has been [suggested to me](https://stackoverflow.com/review/suggested-edits/8047475) that you may also need to manually install the second new package:

Install-Package Microsoft.AspNet.WebPages.Data

This *should not* be necessary, because this package is [an explicit dependency](http://www.nuget.org/packages/Microsoft.AspNet.WebPages.WebData/) of the first, and NuGet should be smart enough to install both. But if you get [an error when building](https://stackoverflow.com/questions/23181104/could-not-load-file-or-assembly-webmatrix-data-version-3-0-0-0), or don’t see NuGet add the dependency, it might help you.

Answers:

WebMatrix is compatible with MVC 5.

What I did was to take an empty MVC 5 project and incorporate WebMatrix SimpleMembershipProvider into it using [SimpleSecurity](https://simplesecurity.codeplex.com/), an open source project that [decouples SimpleMembership from your MVC application](http://kevin-junghans.blogspot.com/2013/03/decoupling-simplemembership-from-your.html). So far I am able to create the database, seed it, and log in and out. I plan on adding other features to this reference application, such as email confirmation and various tests. When I am done I will post the source code in the [SimpleSecurity Project](https://simplesecurity.codeplex.com/SourceControl/latest)

If I had to guess, your problem may be with the upgrade process. What process did you take to upgrade your MVC 4 project to MVC 5? [Did you follow this process](http://www.asp.net/mvc/tutorials/mvc-5/how-to-upgrade-an-aspnet-mvc-4-and-web-api-project-to-aspnet-mvc-5-and-web-api-2)? What version of the WebMatrix assemblies are you using? What version of Visual Studio are you using? I am using version 2.0.0.0 of WebMatrix and Visual Studio 2013 RC.

**Update (10/25/2013)**

I continued my experiment with adding SimpleMembership to an MVC 5 project and somewhere along the line it broke and I got the same results as @Sixten Otto. I did not test incrementally as I added things but I am suspicious it may have happened when I installed the Web API assemblies. They are not installed by default when creating a new MVC 5 project.

I did some more research on the error and came across this QA titled “[Attempt by security transparent method ‘WebMatrix.WebData.PreApplicationStartCode.Start()’](https://stackoverflow.com/questions/17926552/attempt-by-security-transparent-method-webmatrix-webdata-preapplicationstartcod)“. This is an old QA and originally someone was getting this same error when upgrading an MVC 3 app to MVC 4. But recently people have been adding answers in regards to upgrading to MVC 5 and one of the answers worked for me. **The solution for me was install the NuGet package *Microsoft.AspNet.WebHelpers*.** After installing this package everything worked fine.

A note about my research into migrating to the new ASP.NET Identity is that they do not use the same password hash, which precludes moving old members into a database used by ASP.NET Identity. ASP.NET Identity seems to be in real flux right now so maybe they will come up with a solution for this.

**Update (2/16/14)**

I erroneously reported that the hash algorithm for passwords was different in SimpleMembership and ASP.NET Identity. I assumed this based on a visual inspection of the hashed passwords, assuming that it was just the hashed password that was in the fields. After further research I found that SimpleMembership uses the System.Web.Helpers.Crypto class for hashing the password and what is stored in the password field is actually a 256 bit subkey and the salt. With that information I ran some tests to validate that ASP.NET Identity can verify passwords that are generated by SimpleMembership, and it passed. I was trying to find out what hash algorithm SimpleMembership used so I could plug in a password hasher in ASP.NET Identity that would allow me to migrate data from a SimpleMembership webiste to one that used ASP.NET Identity. Turns out it is not necessary. [I talk about the password hash and how to migrate the data from SimpleMembership to ASP.NET Identity in more detail in this article](http://kevin-junghans.blogspot.com/2014/02/migrating-existing-website-from.html).

Answers:

If you are getting the error

Attempt by security transparent method  
‘WebMatrix.WebData.PreApplicationStartCode.Start()’ to access security  
critical method  
‘System.Web.WebPages.Razor.WebPageRazorHost.AddGlobalImport(System.String)’  
failed.

In order to fix this install this package using NuGet package manager.

Install-Package Microsoft.AspNet.WebHelpers

After that , probably you will get another error

Cannot load WebMatrix.Data version 3.0.0.0 assembly

to fix this install this package using NuGet package manager.

Install-Package Microsoft.AspNet.WebPages.Data

Answers:

We are currently working on a migration doc for migrating from Simple Membership to ASP.NET Identity. Please stay tuned for a couple of weeks until we push this migration doc.  
For now you have to map your Simple Membership schema to Identity and change your application code to use OWIN for SignIN/ SIgnOut

Answers:

The above answers worked not until recent webpages 3.2.3. A new issue showed up for me. The current fix for me was by upgrading to .Net 4.5.3. I figured this out of frustration. This issue doesnt just affect MVC 5 but core Webmatrix projects as well after upgrading to webpages 3.2.3. I think it is a framework issue that will be fixed with the new Microsoft Identity. The current fix for me is below:  
*Note: Please use the property pages wizard in visual studio to change your target framework to .Net Framework 4.5.3. It will update your web.config*

<compilation debug="true" targetFramework="4.5.3">

<assemblies>

<add assembly="System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"/>

</assemblies>

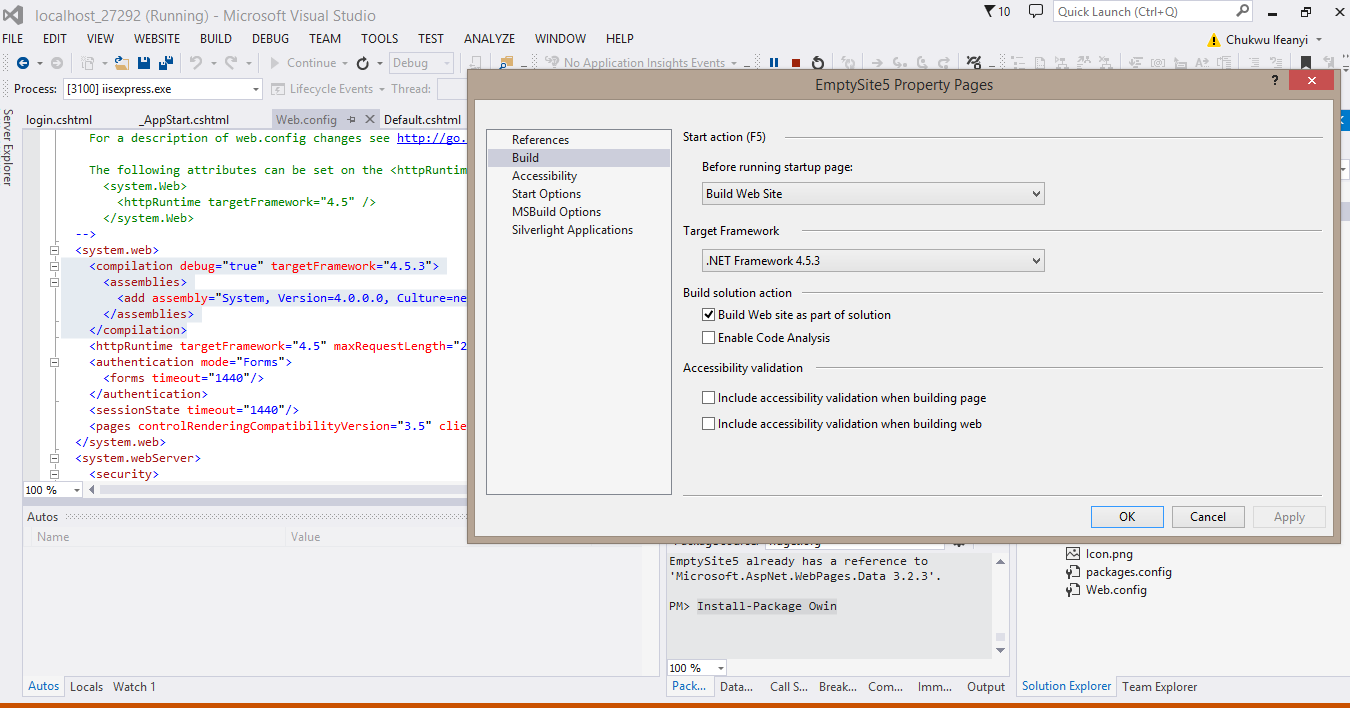
</compilation>

Step 1: Install-Package Microsoft.AspNet.WebHelpers

Step 2: Install-Package Microsoft.AspNet.WebPages.Data

Step 3: [Optional] Install-Package Owin

Step 4: Change targetFramework to .Net 4.5.3 via Property pages dialog box



[Optionally] Your Web.Config should look like the below

<?xml version="1.0"?>

<configuration>

<appSettings/>

<connectionStrings>

<add connectionString="Server=XTREMEGOSPEL;Database=portfolioDB;Trusted\_Connection=True" name="portDB" providerName="System.Data.SqlClient"/>

</connectionStrings>

<!--

For a description of web.config changes see http://go.microsoft.com/fwlink/?LinkId=235367.

The following attributes can be set on the <httpRuntime> tag.

<system.Web>

<httpRuntime targetFramework="4.5" />

</system.Web>

-->

<system.web>

<compilation debug="true" targetFramework="4.5.3">

<assemblies>

<add assembly="System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"/>

</assemblies>

</compilation>

<httpRuntime targetFramework="4.5" maxRequestLength="2097151"/>

<authentication mode="Forms">

<forms timeout="1440"/>

</authentication>

<sessionState timeout="1440"/>

<pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID"/>

</system.web>

<system.webServer>

<security>

<requestFiltering>

<requestLimits maxAllowedContentLength="4294967295"/>

</requestFiltering>

</security>

</system.webServer>

<runtime>

<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">

<dependentAssembly>

<assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35"/>

<bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0"/>

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35"/>

<bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0"/>

</dependentAssembly>

</assemblyBinding>

</runtime>

</configuration>

Answers:

I got same problem, not at my local computer but live site was having that.

I removed below lines from web config and it is working now.

<dependentAssembly>

<assemblyIdentity name="System.Web.WebPages.Razor" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" />

</dependentAssembly>