**How to Upgrade an ASP.NET MVC 4 and Web API Project to ASP.NET MVC 5 and Web API 2**

By [Rick Anderson](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#author-20106) Rick Anderson October 17, 2013

[Print](javascript:window.print();)

ASP.NET MVC 5 and Web API 2 bring a host of new features, including attribute routing, authentication filters, and much more. See <http://www.asp.net/vnext> for more details.

This walkthrough will guide you with the steps required to upgrade your application to the latest version.

Note: Please see [ASP.NET and Web Tools for Visual Studio 2013 Release Notes](http://www.asp.net/visual-studio/overview/2013/release-notes)   for information on breaking changes from MVC 4 and Web API to the next version.

This article was written by Youngjune Hong and  Rick Anderson ( [@RickAndMSFT](https://twitter.com/#!/RickAndMSFT) )

**Upgrade Steps**

1. Backup your project. This walkthrough will require you to make changes  to your project file, package configuration, and web.config files.
2. For upgrading from Web API to Web API 2, in global.asax, change:

WebApiConfigRegisterGlobalConfigurationConfiguration

to

GlobalConfigurationConfigureWebApiConfigRegister

1. Make sure all the packages that your projects use are compatible with  MVC 5 and Web API 2. The following table shows the MVC 4 and Web API related  packages than need to be changed. If you have a package that is dependent on  one of the packages listed below, please contact the publishers to get the  newer versions that are compatible with MVC 5 and Web API 2. If you have the  source code for those packages, you should recompile them with the new  assemblies of MVC 5 and Web API 2.

|  |  |  |
| --- | --- | --- |
| **Package Id** | **Old version** | **New version** |
| Microsoft.AspNet.Razor | 2.0.x.x | 3.0.0 |
| Microsoft.AspNet.WebPages | 2.0.x.x | 3.0.0 |
| Microsoft.AspNet.WebPages.WebData | 2.0.x.x | 3.0.0 |
| Microsoft.AspNet.WebPages.OAuth | 2.0.x.x | 3.0.0 |
| Microsoft.AspNet.Mvc | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.Mvc.Facebook | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.Core | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.SelfHost | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.Client | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.OData | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.WebHost | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.Tracing | 4.0.x.x | 5.0.0 |
| Microsoft.AspNet.WebApi.HelpPage | 4.0.x.x | 5.0.0 |
| Microsoft.Net.Http | 2.0.x. | 2.2.x. |
| Microsoft.Data.OData | 5.2.x | 5.6.x |
| System.Spatial | 5.2.x | 5.6.x |
| Microsoft.Data.Edm | 5.2.x | 5.6.x |
| Microsoft.AspNet.Mvc.FixedDisplayModes |  | Removed |
| Microsoft.AspNet.WebPages.Administration |  | Removed |
| Microsoft-Web-Helpers |  | Microsoft.AspNet.WebHelpers |

1. **Note:**Microsoft-Web-Helpers has been replaced  with Microsoft.AspNet.WebHelpers. You should remove the old package first,  and then install the newer package.   
     
   There is no cross version  compatibility among major ASP.NET packages. For example, MVC 5 is compatible  with only Razor 3, and not Razor 2.
2. Open your project in Visual Studio 2013.
3. Remove any of the following ASP.NET NuGet packages that are installed.  You will remove these using the Package Manager Console (PMC). To open the  PMC, select the**Tools** menu and then select **Library  Package Manager,** then select **Package Manager Console**.  Your project might not include all of these.
   1. Microsoft.AspNet.WebPages.Administration   
      This package is typically added when upgrading from MVC 3 to MVC 4.  To remove it, run the following command in the PMC:  
      Uninstall-Package -Id Microsoft.AspNet.WebPages.Administration
   2. Microsoft-Web-Helpers   
      This  package has been rebranded as Microsoft.AspNet.WebHelpers.  To remove it, run the following command in the PMC:  
      Uninstall-Package -Id Microsoft-Web-Helpers
   3. Microsoft.AspNet.Mvc.FixedDisplayMode    
      This package contains a work around for a bug in MVC 4 that has been  fixed in MVC 5. To remove it, run the following command in the PMC:  
      Uninstall-Package -Id Microsoft.AspNet.Mvc.FixedDisplayModes
4. Upgrade all the ASP.NET NuGet packages using the PMC. In the PMC, run  the following command:  
   Update-Package  
   The Update-Package command without any parameters will update every  package. You can update packages individually by using the ID argument. For  more information about the update command, run get-help update-package.

**Update the Application *web.config* File**

Be sure to make these changes in the app *web.config* file, not the *web.config* file in the *Views*folder.

Locate the <runtime>/<assemblyBinding> section, and  make the following changes:

1. In the elements with the name attribute “System.Web.Mvc”, change the  version number from “4.0.0.0” to “5.0.0.0”. (Two changes in that element.)
2. In elements with the name attribute "System.Web.Helpers” and  "System.Web.WebPages" change the version number from “2.0.0.0” to “3.0.0.0”.  Four changes will occur, two in each of the elements.
3. <runtime><assemblyBindingxmlns"urn:schemas-microsoft-com:asm.v1"<!--Two elements removed for Clarity --><dependentAssembly><assemblyIdentity"System.Web.Helpers"publicKeyToken"31bf3856ad364e35"/><bindingRedirectoldVersion"1.0.0.0-.0.0.0newVersion.0.0.0/></dependentAssembly><dependentAssembly><assemblyIdentity"System.Web.Mvc"publicKeyToken"31bf3856ad364e35"/><bindingRedirectoldVersion"1.0.0.0-.0.0.0newVersion.0.0.0/></dependentAssembly><dependentAssembly><assemblyIdentity"System.Web.WebPages"publicKeyToken"31bf3856ad364e35"/><bindingRedirectoldVersion"1.0.0.0-3.0.0.0newVersion3.0.0.0/></dependentAssembly><!--WebGrease element removed for Clarity -->

  </assemblyBinding

1. Locate the <appSettings> section and update the  webpages:version from 2.0.0.0.0 to 3.0.0.0 as shown below:

<appSettings><add"webpages:Version"value3.0.0.0/><add"webpages:Enabled"value"false"/><add"PreserveLoginUrl"value"true"/><add"ClientValidationEnabled"value"true"/><add"UnobtrusiveJavaScriptEnabled"value"true"/></appSettings>

1. Remove any trust levels other than Full. For example:

<securityPolicy><!--<trustLevel name="Medium"  policyFile="web\_mediumtrust.config"/>--></securityPolicy>

**Update the *web.config*files under the Views folder**

If your application is using areas, you will also need to update each *web.config* file in the *Views*sub-folder of each Area folder.

1. Update all elements that contain “System.Web.Mvc” from version “4.0.0.0”  to  version“5.0.0.0”.

<system.web.webPages.razor><hostfactoryType"System.Web.Mvc.MvcWebRazorHostFactory, System.Web.Mvc, Version=5.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/><pagespageBaseType"System.Web.Mvc.WebViewPage"<namespaces><addnamespace"System.Web.Mvc"/><!--Elements removed for Clarity.--></namespaces></pages></system.web.webPages.razor>

-->pages

      validateRequest"false"

      pageParserFilterType"System.Web.Mvc.ViewTypeParserFilter, System.Web.Mvc, Version=5.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"

      pageBaseType"System.Web.Mvc.ViewPage, System.Web.Mvc, Version=5.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"

      userControlBaseType"System.Web.Mvc.ViewUserControl, System.Web.Mvc, Version=5.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"<controls>add assembly"System.Web.Mvc, Version=5.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"namespace"System.Web.Mvc" tagPrefix"mvc"/></controls</pages</system

1. Update all elements that contain “System.Web.WebPages.Razor”  from  version “2.0.0.0”  to  version“3.0.0.0”. If this section contains “System.Web.WebPages”,  update those elements from  version “2.0.0.0” to  version“3.0.0.0”

<configuration><configSections><sectionGroup"system.web.webPages.razor""System.Web.WebPages.Razor.Configuration.RazorWebSectionGroup, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"<section"host""System.Web.WebPages.Razor.Configuration.HostSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"requirePermission"false"/><section"pages""System.Web.WebPages.Razor.Configuration.RazorPagesSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"requirePermission"false"/></sectionGroup></configSections>

1. If you removed the Microsoft-Web-Helpers NuGet package in a  previous step, installMicrosoft.AspNet.WebHelpers  with  the following command in the PMC:  
   Install-Package -Id  Microsoft.AspNet.WebHelpers
2. If your app uses the  [User.IsInRole()](http://msdn.microsoft.com/en-us/library/system.web.security.roleprincipal.isinrole(v=vs.110).aspx) method, add the following to the *Web.config* file.

<system.webServer><modules><remove"RoleManager"/></modules></system.webServer>

**Final Steps**

Build and test the application.

 Remove the MVC 4 project type GUID from the project files.

1. In Solution Explorer, right-click the project name and then select **Unload Project**.
2. Right-click the project  and select Edit ProjectName.csproj.
3. Locate the ProjectTypeGuids element and then remove  the MVC 4 project GUID, {E3E379DF-F4C6-4180-9B81-6769533ABE47}.
4. Save and close the open project file.
5. Right-click the project and select **Reload Project**.

This article was originally created on October 17, 2013

**Author Information**

**Rick Anderson** – Rick Anderson works as a programmer writer for Microsoft, focusing on ASP.NET MVC, Windows Azure and Entity Framework. You can follow him on twitter via @RickAndMSFT.