Sage 300 Web Screens SDK

Upgrade Guide for 2051.1 to 2025.2

June 2025

The MIT License (MIT)

Copyright © 2025 The Sage Group plc or its licensors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

[1. Overview 4](#_Toc201592794)

[1.1 Required Version of Sage 300 5](#_Toc201592795)

[2. Installing the Wizard 6](#_Toc201592796)

[3. Accessing the Wizard 7](#_Toc201592797)

[4. Using the Wizard 8](#_Toc201592798)

[4.1 Step 1 – Synchronize Kendo Files 9](#_Toc201592799)

[4.2 Step 2 – Synchronize Web Files 10](#_Toc201592800)

[4.3 Step 3 – Synchronize Accpac Libraries 11](#_Toc201592801)

[4.4 Step 4 – Confirmation 12](#_Toc201592802)

[4.5 Step 5 – Recompile 13](#_Toc201592803)

[4.6 Step 6 – Manual Steps Required 14](#_Toc201592804)

[4.6.1 Business Repository – app.config File 14](#_Toc201592805)

[4.6.2 Business Repository – packages.config File 14](#_Toc201592806)

[4.6.3 Services – app.config File 15](#_Toc201592807)

[4.6.4 Services – packages.config File 15](#_Toc201592808)

[4.6.5 Web – packages.config File 15](#_Toc201592809)

[4.6.6 Web – web.config File 15](#_Toc201592810)

[4.6.7 Business Repository – \*.csproj File 16](#_Toc201592811)

[4.6.8 Web – \*.csproj File 16](#_Toc201592812)

[4.6.9 Web – Global.asax.cs File 20](#_Toc201592813)

[4.6.10 Complete 20](#_Toc201592814)

[5. Upgrade Log 21](#_Toc201592815)

[6. Compilation Troubleshooting 22](#_Toc201592816)

1. Overview

This document is intended to serve as a guide for illustrating how to use the Sage 300 Upgrade Wizard to upgrade Visual Studio solutions and projects that are compatible with the Web Screens SDK 2025.1 to Web Screens SDK 2025.2

The wizard will upgrade the solution and projects in following steps:

* Synchronize Kendo Files
* Synchronize Web Files
* Synchronize Accpac Libraries
  1. Required Version of Sage 300

To get started, install Sage 300 2025.2 with the Web Screens option selected.

1. Installing the Wizard

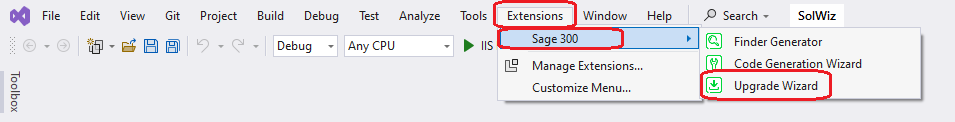
The Upgrade Wizard is a Visual Studio Plugin and is compatible with Visual Studio 2022. The Sage 300 Upgrade Wizard tool is bundled with the Sage 300 Wizard Package (Code Generation Wizard, Finder Generator, Upgrade Wizard, Language Wizard, and Sync Assemblies Wizard)

**Note:** If the previous Sage 300 Wizard Package is installed, it must be uninstalled first (Visual Studio - Tools, Extensions, Uninstall option, select the Sage 300 Wizard Package to uninstall).

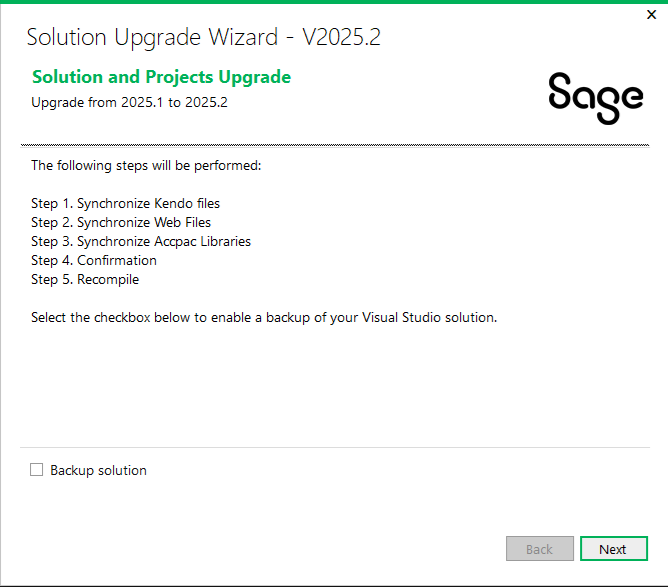
To install the utility:

1. Run Sage.CA.SBS.ERP.Sage300.WizardPackage.vsix.
2. Select the Install the Wizard Package option.
3. Note: Installing this package installs the above-mentioned wizards since they are combined in the same package.
4. Accessing the Wizard

The wizard is embedded in Visual Studio 2022 as a plugin. To open it, click the **Extensions** menu on the Visual Studio toolbar, select the **Sage 300** menu, and click **Upgrade** Wizard.



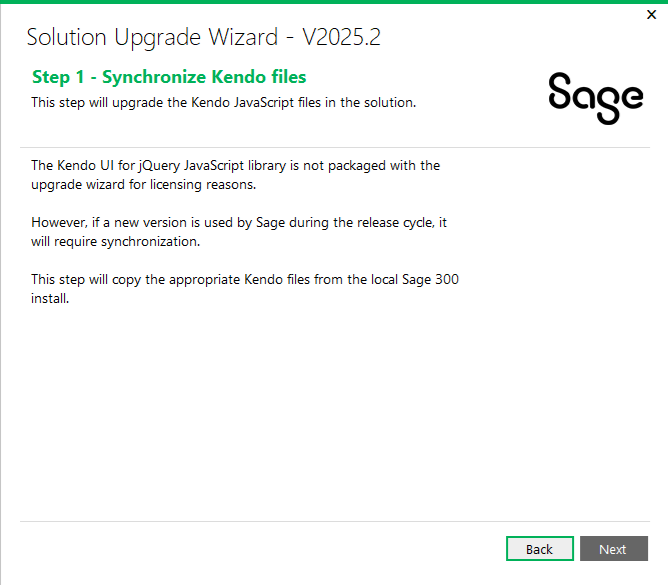
1. Using the Wizard

­

**Important:** Please backup the solution and projects before proceeding with the upgrade. If you would like the Upgrade Wizard to make a backup of your solution, click the ‘Backup solution’ checkbox.

Click **Next** to continue to the next step.

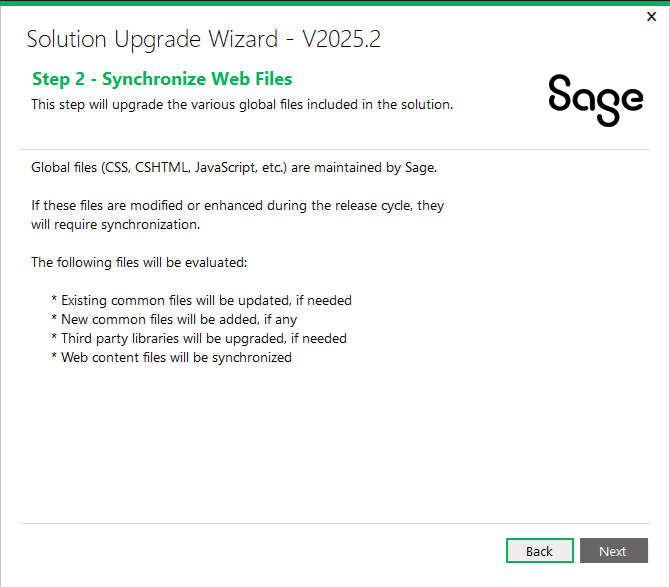
* 1. Step 1 – Synchronize Kendo Files



A copy of the Kendo UI for jQuery JavaScript library will be copied into the solution from the local Sage 300 installation.

Click Next to proceed or Back to go back to the previous step.

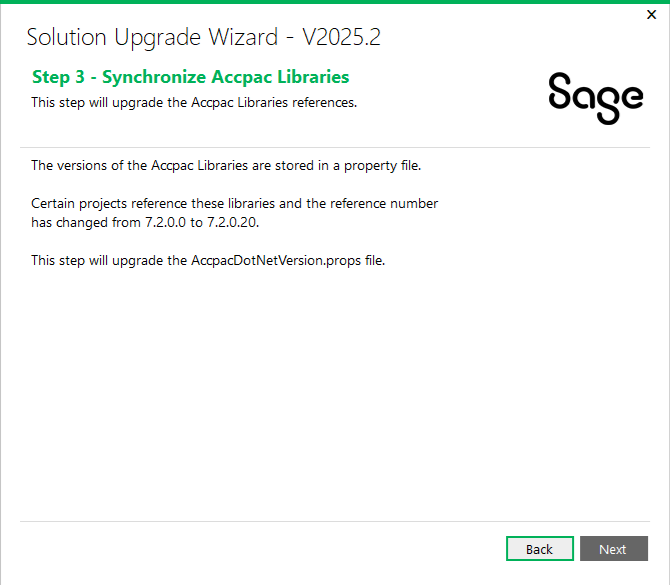
* 1. Step 2 – Synchronize Web Files



Contents in the Web Projects Areas\Shared, Areas\Core, Views, Scripts, Content, and Assets folders are upgraded in this step.

Click Next to proceed or Back to go back to the previous step.

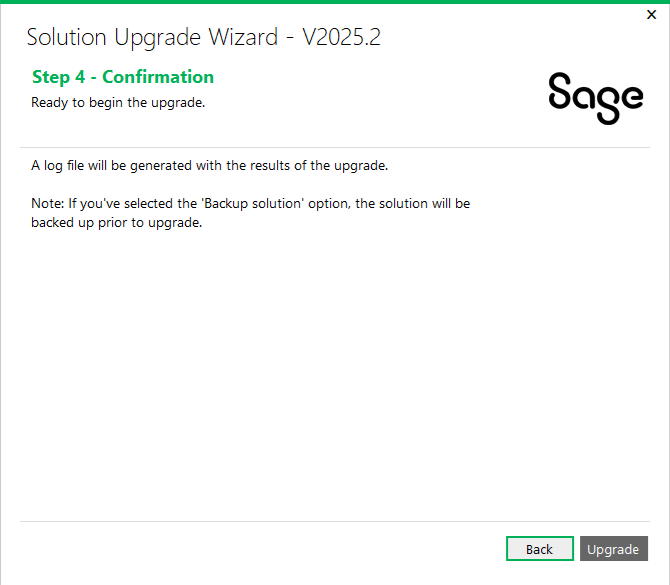
* 1. Step 3 – Synchronize Accpac Libraries



Accpac libraries are updated from 7.2.0.0 to 7.2.0.20.

Click Next to proceed or Back to go back to the previous step.

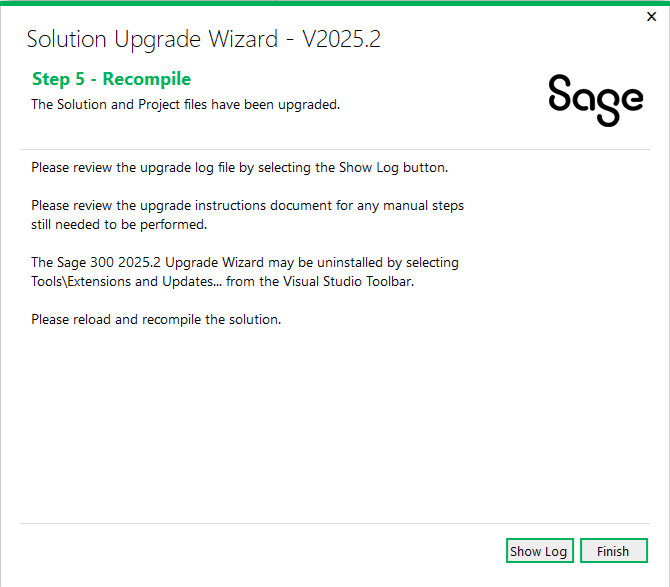
* 1. Step 4 – Confirmation



The solution and projects are ready to be upgraded.

Click Upgrade to upgrade the solution and projects or Back to go back to the previous step.

* 1. Step 5 – Recompile



Click **Show Log** to display log file. The log file displays what files have been upgraded or modified. At this point, the upgrade is now complete.

Click **Finish** to exit the wizard.

After the solution that supported Sage 300 2025.1 has been upgraded to support 2025.2, you need to reload and recompile the solution. For compilation troubleshooting, see the final section.

**Note:** This upgrade has manual steps that must be completed before compiling. Please see the next section to complete the manual steps.

**Note:** Before running the new solution, clear the browser cache to replace cached JavaScript and CSS files.

* 1. Step 6 – Manual Steps Required

The Upgrade Wizard for 2025.2 was not able to automatically make changes to all the files in the solution that require modification. Therefore, developer intervention is required for numerous files.

We apologize in advance for not incorporating these into the wizard for automatic upgrades. In the next release, there will be additional changes as well and it is our desire to have the wizard perform these modifications automatically and without developer intervention.

The web screen samples in the Web SDK’s ***samples*** folder have been updated and are a great source to see the resulting files after the changes have been made. Some of the modified files can be copied from the samples to override content in the solution files. Be sure the solution has been backed up in case any developer added entries need to be restored.

* + 1. Business Repository – app.config File

The app.config file contains binding redirects for the included packages in the assembly.

Copy the **app.config** file from any of the sample’s Business Repository folders (i.e., samples\ClearStatistics\ValuedPartner.TU.BusinessRepository\app.config) and overwrite the app.config file in the Business Repository project.

Note: Examine the app.config file prior to overwriting to ensure any partner added redirect statements are manually added once the copy is complete.

* + 1. Business Repository – packages.config File

The packages.config file contains the packages in the assembly.

Copy the **packages.config** file from any of the sample’s Business Repository folders (i.e., samples\ClearStatistics\ValuedPartner.TU.BusinessRepository\packages.config) and overwrite the packages.config file in the Business Repository project.

Note: Examine the packages.config file prior to overwriting to ensure any partner added packages are manually added once the copy is complete.

* + 1. Services – app.config File

The app.config file contains binding redirects for the included packages in the assembly.

Copy the **app.config** file from any of the sample’s Services folders (i.e., samples\ClearStatistics\ValuedPartner.TU.Services\app.config) and overwrite the app.config file in the Services project.

Note: Examine the app.config file prior to overwriting to ensure any partner added redirect statements are manually added once the copy is complete.

* + 1. Services – packages.config File

The packages.config file contains the packages in the assembly.

Copy the **packages.config** file from any of the sample’s Services folders (i.e., samples\ClearStatistics\ValuedPartner.TU.Services\packages.config) and overwrite the packages.config file in the Services project.

Note: Examine the packages.config file prior to overwriting to ensure any partner added packages are manually added once the copy is complete.

* + 1. Web – packages.config File

The packages.config file contains the packages in the assembly.

Copy the **packages.config** file from any of the sample’s Web folders (i.e., samples\ClearStatistics\ValuedPartner.TU.Web\packages.config) and overwrite the packages.config file in the Web project.

Note: Examine the packages.config file prior to overwriting to ensure any partner added packages are manually added once the copy is complete.

* + 1. Web – web.config File

The web.config file contains binding redirects for the included packages in the assembly.

Copy the <assemblyBinding> … </assemblyBinding> **entirely** file from any of the sample’s web.config files (i.e., samples\ClearStatistics\ValuedPartner.TU.Web\packages.config) and overwrite the existing <assemblyBinding> section in the web.config file of the Web project..

Note: Examine the <assemblyBinding> tag prior to overwriting to ensure any partner added packages are manually added once the copy is complete.

* + 1. Business Repository – \*.csproj File

The **Newtonsoft.Json** package is no longer required by the Business Repository

Load the Business Repository’s \*.csproj file and delete the reference tag for Newtonsoft.Json if it exists. For example:

<Reference Include="Newtonsoft.Json, Version=9.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed, processorArchitecture=MSIL">

<HintPath>..\packages\Newtonsoft.Json.9.0.1\lib\net45\Newtonsoft.Json.dll</HintPath>

</Reference>

* + 1. Web – \*.csproj File

The Web’s \*.csproj file will require some references to be added, some replaced, and some deleted.

Note: The following changes are to add a reference after this or that to match the order in the sample web.config files. But the order does not really matter.

Load the Web’s \*.csproj file in an editor and make the following changes:

* **Add** the following reference after the **Antlr3.runtime** reference:

<Reference Include="CEDataServiceContract">

<HintPath>$(Sage300WebDir)\bin\CEDataServiceContract.dll</HintPath>

<Private>True</Private>

</Reference>

* **Add** the following references after the **EntityFramework.SqlServer** reference:

<Reference Include="Microsoft.AspNet.SignalR.Core">

<HintPath>$(Sage300WebDir)\bin\Microsoft.AspNet.SignalR.Core.dll</HintPath>

</Reference>

<Reference Include="Microsoft.AspNet.SignalR.SystemWeb">

<HintPath>$(Sage300WebDir)\bin\Microsoft.AspNet.SignalR.SystemWeb.dll</HintPath>

</Reference>

* **Delete** the following references:

<Reference Include="Microsoft.Data.OData, Version=5.6.4.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">

<SpecificVersion>False</SpecificVersion>

<HintPath>$(SolutionDir)packages\Microsoft.Data.OData.5.6.4\lib\portable-net45+wp8+win8+wpa\Microsoft.Data.OData.dll</HintPath>

</Reference>

<Reference Include="Microsoft.Data.Services.Client, Version=5.6.4.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">

<SpecificVersion>False</SpecificVersion>

<HintPath>$(SolutionDir)packages\Microsoft.Data.Services.Client.5.6.4\lib\portable-net45+wp8+win8+wpa\Microsoft.Data.Services.Client.dll</HintPath>

</Reference>

* **Add** the following references after the **Microsoft.Data.Edm** reference:

<Reference Include="Microsoft.Owin">

<HintPath>$(SolutionDir)packages\Microsoft.Owin.4.0.1\lib\net45\Microsoft.Owin.dll</HintPath>

</Reference>

<Reference Include="Microsoft.Owin.Host.SystemWeb">

<HintPath>$(SolutionDir)packages\Microsoft.Owin.Host.SystemWeb.2.1.0\lib\net45\Microsoft.Owin.Host.SystemWeb.dll</HintPath>

</Reference>

<Reference Include="Microsoft.Owin.Security">

<HintPath>$(SolutionDir)packages\Microsoft.Owin.Security.3.0.0\lib\net45\Microsoft.Owin.Security.dll</HintPath>

</Reference>

* **Modify** the following reference:

**OLD**:

<Reference Include="Newtonsoft.Json, Version=9.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed, processorArchitecture=MSIL">

<HintPath>..\packages\Newtonsoft.Json.9.0.1\lib\net45\Newtonsoft.Json.dll</HintPath>

</Reference>

**NEW**:

<Reference Include="Newtonsoft.Json">

<HintPath>$(SolutionDir)packages\Newtonsoft.Json.13.0.3\lib\net45\Newtonsoft.Json.dll</HintPath>

</Reference>

* **Add** the following references after the **Newtonsoft.Json** reference:

<Reference Include="Owin">

<HintPath>$(Sage300WebDir)\bin\Owin.dll</HintPath>

</Reference>

<Reference Include="RouteDebugger">

<HintPath>$(Sage300WebDir)\bin\RouteDebugger.dll</HintPath>

</Reference>

* **Add** the following reference after the **System** reference:

<Reference Include="System.Buffers">

<HintPath>$(Sage300WebDir)\bin\System.Buffers.dll</HintPath>

</Reference>

* **Add** the following references after the **System.Data.DataSetExtensions** reference:

<Reference Include="System.Data.Services.Client" />

<Reference Include="System.DirectoryServices" />

<Reference Include="System.Drawing" />

<Reference Include="System.IO.Compression" />

<Reference Include="System.Memory">

<HintPath>$(Sage300WebDir)\bin\System.Memory.dll</HintPath>

</Reference>

* **Add** the following references after the **System.Net.Http.Primitives** reference:

<Reference Include="System.Numerics" />

<Reference Include="System.Numerics.Vectors">

<HintPath>$(Sage300WebDir)\bin\System.Numerics.Vectors.dll</HintPath>

</Reference>

<Reference Include="System.Runtime.Caching" />

<Reference Include="System.Runtime.CompilerServices.Unsafe">

<HintPath>$(Sage300WebDir)\bin\System.Runtime.CompilerServices.Unsafe.dll</HintPath>

</Reference>

<Reference Include="System.Runtime.Serialization" />

<Reference Include="System.Security" />

<Reference Include="System.ServiceModel.DomainServices.EntityFramework, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL" />

<Reference Include="System.ServiceModel.Web" />

* **Add** the following references after the **System.Spatial** reference:

<Reference Include="System.Text.Encodings.Web">

<HintPath>$(Sage300WebDir)\bin\System.Text.Encodings.Web.dll</HintPath>

</Reference>

<Reference Include="System.Threading.Tasks.Extensions">

<HintPath>$(Sage300WebDir)\bin\System.Threading.Tasks.Extensions.dll</HintPath>

</Reference>

<Reference Include="System.ValueTuple">

<HintPath>$(Sage300WebDir)\bin\System.ValueTuple.dll</HintPath>

</Reference>

* **Add** the following reference after the **System.Web.Abstractions** reference:

<Reference Include="System.Web.DataVisualization" />

* **Modify** the following reference:

**OLD**:

<Reference Include="System.Web.Optimization, Version=1.1.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">

<SpecificVersion>False</SpecificVersion>

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.Web.Optimization.1.1.2\lib\net40\System.Web.Optimization.dll</HintPath>

</Reference>

**NEW**:

<Reference Include="System.Web.Optimization">

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.Web.Optimization.1.1.3\lib\net40\System.Web.Optimization.dll</HintPath>

</Reference>

* **Modify** the following reference:

**OLD**:

<Reference Include="System.Web.Razor, Version=2.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">

<SpecificVersion>False</SpecificVersion>

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.Razor.3.2.3\lib\net45\System.Web.Razor.dll</HintPath>

</Reference>

**NEW**:

<Reference Include="System.Web.Razor">

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.Razor.3.2.7\lib\net45\System.Web.Razor.dll</HintPath>

</Reference>

* **Modify** the following references:

**OLD**:

<Reference Include="WebMatrix.Data, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">

<SpecificVersion>False</SpecificVersion>

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.WebPages.Data.3.1.2\lib\net45\WebMatrix.Data.dll</HintPath>

</Reference>

<Reference Include="WebMatrix.WebData, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">

<SpecificVersion>False</SpecificVersion>

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.WebPages.WebData.3.1.2\lib\net45\WebMatrix.WebData.dll</HintPath>

</Reference>

**NEW**:

<Reference Include="WebMatrix.Data">

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.WebPages.Data.3.2.3\lib\net45\WebMatrix.Data.dll</HintPath>

</Reference>

<Reference Include="WebMatrix.WebData">

<HintPath>$(SolutionDir)packages\Microsoft.AspNet.WebPages.WebData.3.2.3\lib\net45\WebMatrix.WebData.dll</HintPath>

</Reference>

* **Add** the following reference after the **WebMatrix.WebData** reference:

<Reference Include="WindowsBase" />

* + 1. Web – Global.asax.cs File

The UMClientConfig object has been deprecated.

Load the Global.asax.cs file in an editor and make the following change:

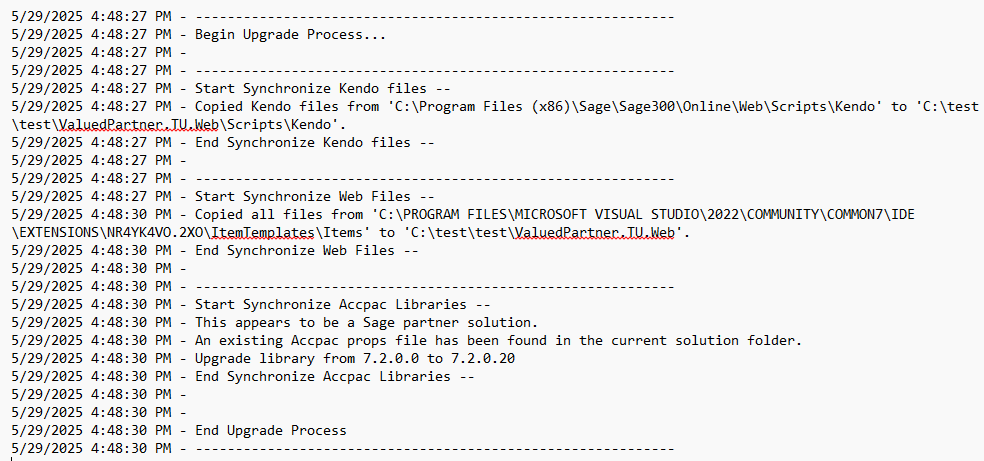
* **Delete** UMClientConfig.Register();
  + 1. Complete

The manual steps are complete. Load the project’s solution and re-compile to ensure all files have been modified successfully.

Note: Any partner Nuget packages that have been added to the project files since being generated by the Solution Wizard must be manually re-added.

1. Upgrade Log

Once the Upgrade Wizard has completed you have the option to ‘Show Log’ to view the upgrade log. The following is a screenshot of a typical upgrade log.



1. Compilation Troubleshooting

**Parser Error when loading Web project in browser**

* Clean solution
* In Web project folder:
  + delete bin and obj folders