Sage 300

Web Screens SDK 2017.1(PU1) to 2017.2 (PU2) Upgrade Guide

March 2017

The MIT License (MIT)

Copyright © 2017 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](#_Toc477353342)

[1.1 Required Version of Sage 300 4](#_Toc477353343)

[2. Update the Sage 300 SDK Wizard 5](#_Toc477353344)

[3. Install the Sage 300 Upgrade Wizard 6](#_Toc477353345)

[3.1 Solution and Projects Upgrade 6](#_Toc477353346)

[4. Accessing the Sage 300 Upgrade Wizard 7](#_Toc477353347)

[5. Using Sage 300 Upgrade Wizard 8](#_Toc477353348)

[5.1 Step 1 – Synchronize Web Files 9](#_Toc477353349)

[5.2 Step 2 – Refactor Razor Views to use the Global Layout 10](#_Toc477353350)

[5.3 Step 3 – Remove Deprecated Methods 11](#_Toc477353351)

[5.4 Step 4 – Address Resx Files for Blank Values 12](#_Toc477353352)

[5.5 Step 5 – Ensure Bundle Names are Correct 13](#_Toc477353353)

[5.6 Step 6 – Synchronize Accpac Libraries 14](#_Toc477353354)

[5.7 Step 7 – Convert Web Project to Contain Module ID (Optional) 15](#_Toc477353355)

[5.8 Step 8 – Confirmation 16](#_Toc477353356)

[5.9 Upgrade Completed 17](#_Toc477353357)

[6. Options Menu changes 18](#_Toc477353358)

[7. Recompiling the Solution 19](#_Toc477353359)

[8. Compilation Troubleshooting 20](#_Toc477353360)

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 2017.1 to SDK 2017.2.

Since Web Screens SDK 2017.1 was released, code changes were made to CSS, HTML, JavaScript, and other source files. To build with Sage 300 2017.2, some folders and files require updates.

**Note:** The steps in this document are required only for solutions and projects that are compatible with Web Screens SDK 2017.1. The Solution Wizard and Code Generation Wizard have the updated code files and libraries, so solutions and projects that were generated recently will have the latest files and libraries.

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

* Synchronize global web files
* Refactor razor views to use the new Global Layout
* Remove deprecated methods
* Address resx files for blank values
* Ensure bundle names are correct
* Synchronize Accpac libraries
* Convert web project to contain module id (optional)
  1. Required Version of Sage 300

To get started, install Sage 300 2017.2 with the Web Screens option selected.

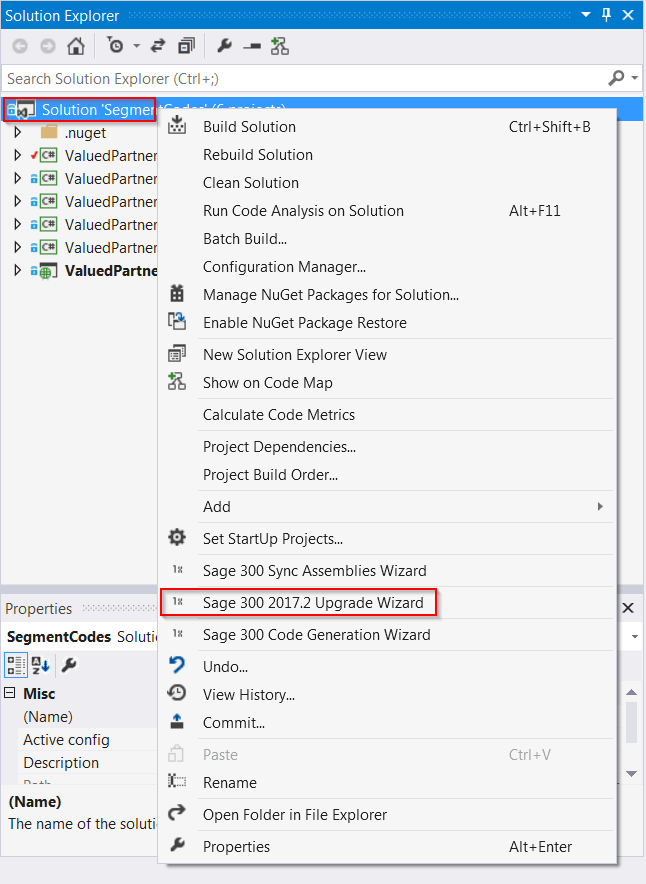
1. Install the Sage 300 Upgrade Wizard

The Upgrade Wizard is a Visual Studio Plugin that was developed in Visual Studio 2015 and is compatible with Visual Studio 2013.

To install the wizards:

1. Run Sage300UpgradeWizardPackage.vsix.
2. Select the Install the Wizard Package option.
3. Accessing the Sage 300 2017.2 Upgrade Wizard

This **Sage 300 2017.2 Upgrade Wizard** is a Visual Studio Plugin in Visual Studio 2015. To open it, right-click the solution, and on the context menu, click Sage 300 2017.2 Upgrade Wizard.



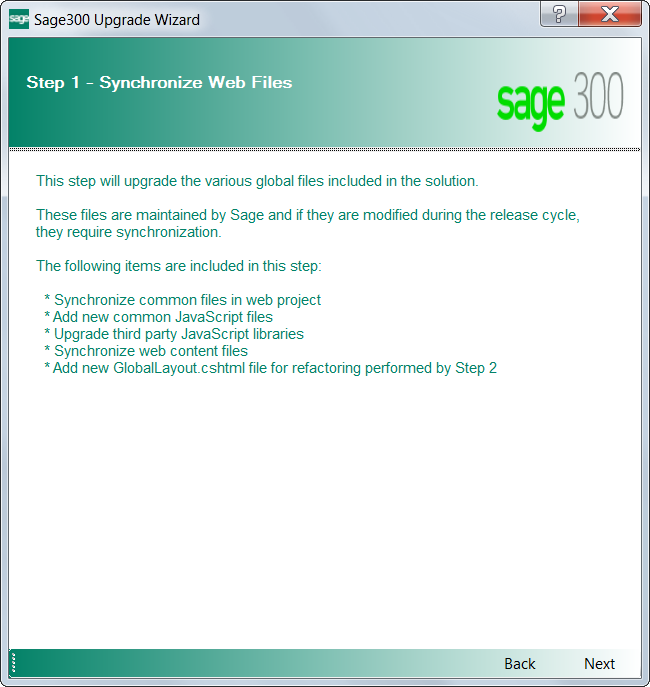
1. Using Sage 300 Upgrade Wizard



**Important:** Please backup the solution and projects before proceeding with the upgrade.

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

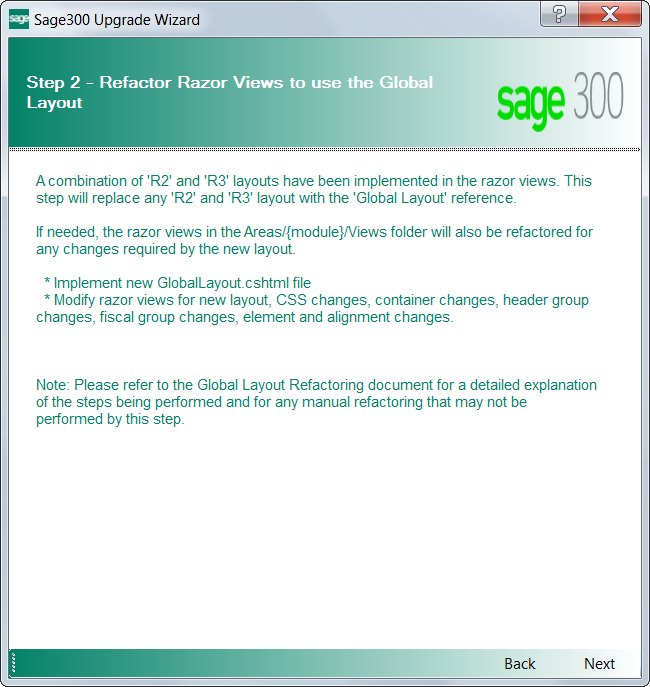
* 1. Step 1 – Synchronize Web Files



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.

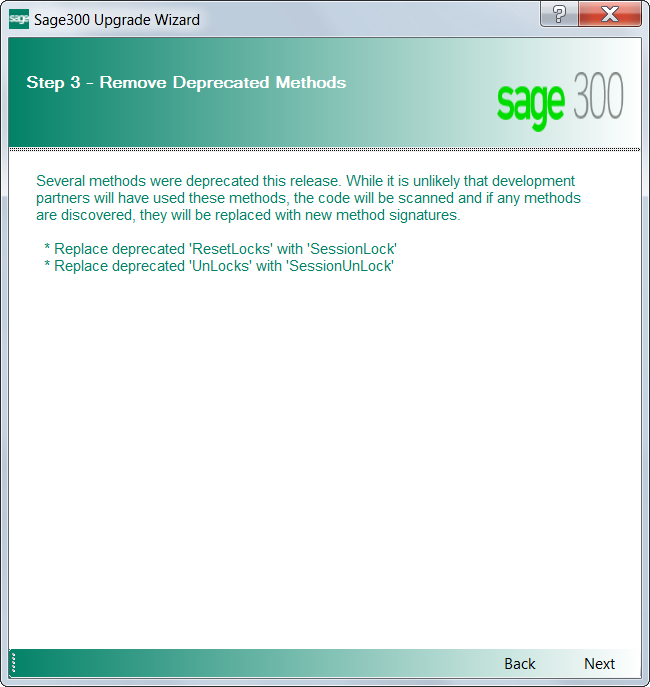
* 1. Step 2 – Refactor Razor Views to use the Global Layout



**Important:** The Sage 300 Upgrade Wizard will automatically perform most of the changes required for the refactoring of razor views, but some areas require manual work. It’s recommended to examine razor views following the steps detailed in **Sage300SDK\_GlobalLayoutRefactoringGuide.docx.**

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

* 1. Step 3 – Remove Deprecated Methods



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

* 1. Step 4 – Address Resx Files for Blank Values



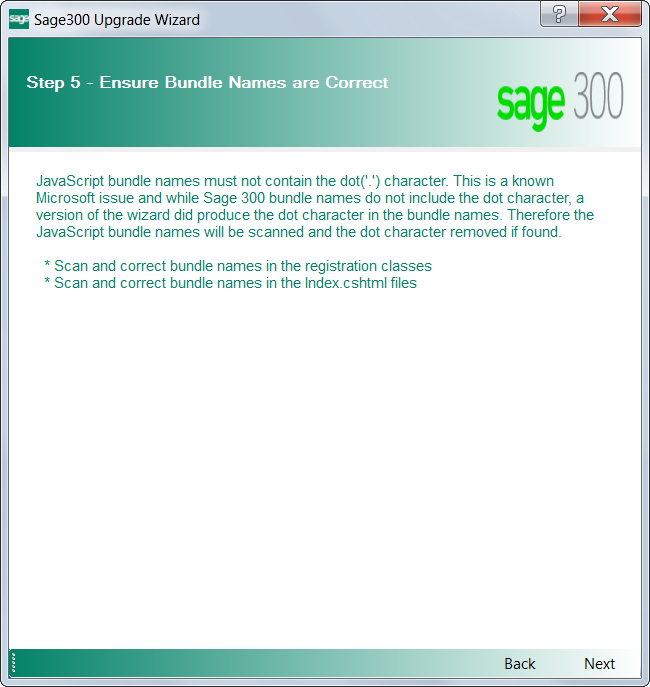
<https://jthomas903.wordpress.com/2017/01/24/sage-300-optional-resource-files/>

Blank values in a non-English Resx file can be resolved in one of two ways:

* If the partner is intending to support their screen in a Sage 300 supported locale, the locale’s Resx file must be translated with proper values
* If the partner is not intending to support their screen in a Sage 300 supported locale, the locale’s non-English Resx file can simply be deleted from the project while in the Visual Studio IDE.

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

* 1. Step 5 – Ensure Bundle Names are Correct



<https://jthomas903.wordpress.com/2017/01/23/sage-300-javascript-bundle-names/>

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

* 1. Step 6 – Synchronize Accpac Libraries



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

* 1. Step 7 – Convert Web Project to Contain Module ID (Optional)



**Important:** This step renames the web project folder and modifies some of the files in it, please ensure the web folder or the files in it are not open in another program.

During the upgrade process, this step fails if the web folder or any of the files in it are open in another program and an error message is recorded in the upgrade log file.

The upgrade process can be re-run if the error is observed in the upgrade log file.

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

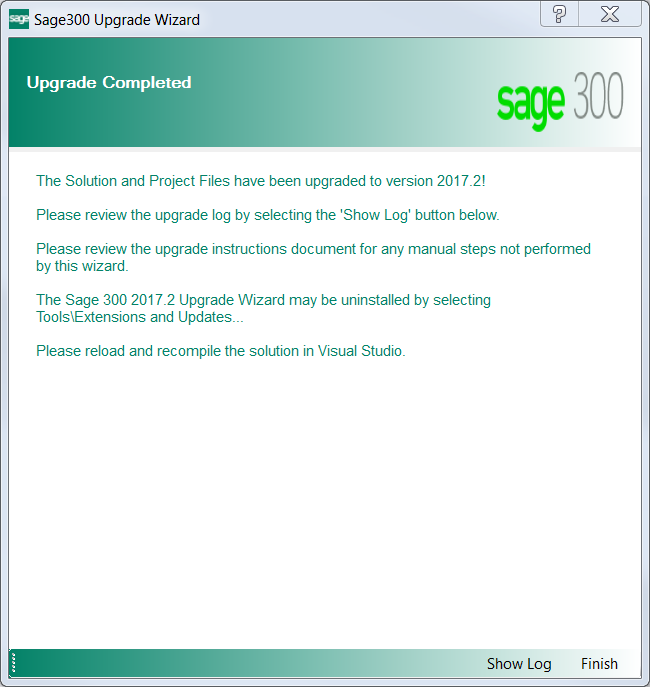
* 1. Step 8 – Confirmation



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. Upgrade Completed



The log file displays what files have been upgraded or modified. The upgrade is complete.

Click Show Log to display log file. Click Finish to exit the wizard.

1. Options Menu changes

The Web Projects Areas\Core\Views\Shared\\_OptionsMenu.cshtml has been updated to clean up redundant code. A brief description has been provided below, update the Options Menu parameters accordingly in your .cshtml files.

**Following options are not supported anymore**:

// OptionsMenu layout class use "options-menu" instead of "dropDown-Menu"

@Html.Partial(Core.OptionsMenu, Model.UserAccess, new ViewDataDictionary {{OptionsMenu.UseLessCss, true}})

// OptionsMenu Import/Export disabled based on Model

// Data.ImportOptionsDisable/Data.ExportOptionsDisable field binding

@Html.Partial(Core.OptionsMenu, Model.UserAccess, new ViewDataDictionary {{OptionsMenu.DisableBinding, true}})

// OptionsMenu Import/Export link enabled based on Model

// Data.ImportOptionsDisable/Data.ExportOptionsDisable css binding

@Html.Partial(Core.OptionsMenu, Model.UserAccess, new ViewDataDictionary {{OptionsMenu.CssBinding, true}})

Please view the file Areas\Core\Views\Shared\\_OptionsMenu.cshtml for the detailed implementation of these option values and how it can be used in your project.

1. Recompiling the Solution

After the solution that supported the 2017.1 (PU1) version of the SDK has been upgraded to support 2017.2 (PU2), please reload the solution and projects, and recompile the solution. For compilation troubleshooting, see the next section.

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

1. Compilation Troubleshooting

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

* Clean solution
* In Web project folder:
  + delete bin and obj folders
  + delete \*csproj.user file