Sage 300

Web Screens SDK 2017 to 2017.1 (PU1)

Upgrade Guide

December 2016

The MIT License (MIT)

Copyright © 2016 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](#_Toc467659999)

[1.1 Required Version of Sage 300 4](#_Toc467660000)

[2. Update the Sage 300 SDK Wizard 5](#_Toc467660001)

[3. Update the Projects and Reference Assemblies 6](#_Toc467660002)

[3.1 Load PowerShell scripts for the update 6](#_Toc467660003)

[3.2 Update the target .NET Framework 6](#_Toc467660004)

[3.3 Update the solution’s \*.csproj and \*.config files 6](#_Toc467660005)

[3.4 Update the Web project’s (\*.csproj) ItemGroup contents 7](#_Toc467660006)

[3.5 Update the Web Artifact files 7](#_Toc467660007)

[4. Options Menu changes 8](#_Toc467660008)

[5. Optional - Adding Report functionality to the Web Project 9](#_Toc467660009)

[5.1 Copy WebForms folder 9](#_Toc467660010)

[5.2 Add CrystalDecisions references to the Web project 9](#_Toc467660011)

[6. Recompiling the Solution 10](#_Toc467660012)

[7. Compilation Troubleshooting 11](#_Toc467660013)

1. Overview

This Upgrade Guide provides instructions for upgrading Visual Studio solutions and projects that are compatible with the Web Screens SDK 2017.

Since Web Screens SDK 2017 was released, code changes were made to CSS, HTML, JavaScript, and other source files. To build with Sage 300 2017.1, 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. 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.

* 1. Required Version of Sage 300

This guide is for updating to support Sage 300 2017.1.

To get started, install Sage 300 2017.1 with the Web Screens option selected.

1. Update the Sage 300 SDK Wizard

The Sage 300 SDK wizards have been updated. You must uninstall any existing wizards, and then install the new wizards.

To uninstall existing wizards:

1. In Visual Studio, select Tools\Extensions and Updates.
2. In the Installed section, search for Sage 300 UI Wizard Package.
3. Select the Uninstall option.
4. Select Visual Studio to Restart.
5. Exit Visual Studio.

To install the new wizards:

1. Run Sage300WizardPackage.vsix.
2. Select the Install the Wizard Package option.
3. Update the Projects and Reference Assemblies

Some project files, binary and text files must be updated. To start this process, generate a temporary Visual Studio solution from the Sage 300 SDK wizard with the same namespace and module as the target solution to upgrade. This temporary solution acts as a source for file comparison and for files to copy from. After confirming that your solution is updated correctly, you can delete the temporary solution.

* 1. Load PowerShell scripts for the update

Open a PowerShell command line window and navigate the current directory to the ***directory of the solution to upgrade***. Copy and paste the PowerShell script file *2017.1-load-update-scripts.ps1* into the same current directory as PowerShell command window. Leave this PowerShell command window open for the entire duration of this update process.

Run the PowerShell script file *2017.1-load-update-scripts.ps1* to load the required script blocks for this update. The script accepts two arguments.

For details on how to run the script, run this command in the command window.

|  |
| --- |
| get-help .\2017.1-load-update-scripts.ps1 -detailed |

* 1. Update the target .NET Framework

The target .NET Framework has been updated to v4.6.2.

Run this script block in the PowerShell command window to update the target framework in the \*.csproj files

|  |
| --- |
| & $UpdateDotNetFramework |

* 1. Update the solution’s \*.csproj and \*.config files

Use a diff program (e.g. Meld, WinMerge, KDiff) and at the solution directory level, compare the temporarily generated 2017.1 SDK solution against the target solution.

From the diff, at minimum, merge the contents from any of the following files as required to update the reference assemblies. For the \*.csproj files the main focus is the <Reference> tag.

* \*BusinessRepository.csproj
* \<any directory>\app.config
* \<any directory>\packages.config
* \*Services.csproj
* \<Web project directory>\Web.config
* \*Web.csproj

During the merge of the file contents, you can compare against the *2017.1-update-references.diff* file to ensure the minimum required assemblies are updated.

* 1. Update the Web project’s (\*.csproj) ItemGroup contents

Contents in the Web Projects Areas\Shared, Areas\Core, Views, Scripts, Content, and Assets

Folders have been updated which require updates to the Web project’s .csproj file.

Run this script block in the PowerShell command window

|  |
| --- |
| & $UpdateWebProject |

* 1. Update the Web Artifact files

Contents in the Web Projects Areas\Shared, Areas\Core, Views, Scripts, Content, and Assets folders have been updated which need to be copied to the Web project’s folder from the temporarily generated 2017.1 Sage 300 SDK solution.

Run this script block in the PowerShell command window

|  |
| --- |
| & $UpdateWebArtifactsFiles |

If an error is observed in the output, this command can be re-run. The script removes the existing folders and copies new content from the generated solution.

1. Options Menu changes

The Web Projects Areas\Core\Views\Shared\\_OptionsMenu.cshtml has been updated which could cause alignment issues with the 2017 implementation of the Options Menu in the .cshtml files. A brief description has been provided below, update the Options Menu parameters accordingly in your .cshtml files.

// OptionsMenu Import/Export link always enabled

@Html.Partial(Core.OptionsMenu, Model.UserAccess)

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

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

// OptionsMenu Import/Export link enabled based on Model

// Data.ImportOptionsEnabled/Data.ExportOptionsEnabled field binding

@Html.Partial(Core.OptionsMenu, Model.UserAccess, new ViewDataDictionary {{OptionsMenu.EnableBinding, 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. Optional - Adding Report functionality to the Web Project
   1. Copy WebForms folder

If you require Report and Report debugging functionality and only if you have not already added this to your solution, the contents of the \WebForms folder from the temporary generated Sage 300 SDK solution needs to be copied to the upgrade target solution. The Powershell commands that have been loaded from *2017.1-load-update-scripts.ps1* will:

* Copy the \WebForms folder from the temporary generated Sage 300 SDK solution
* Update the \*Web.csproj project file to add new ItemGroup items for compile/content includes.

NOTE: Only run this command if you don’t already have WebForms added to your Web project or there will be compilation errors caused by the duplication of files in the .cproj file.

Run this script block in the PowerShell command window

|  |
| --- |
| & $AddReportFunctionality |

* 1. Add CrystalDecisions references to the Web project

Using the Diff program again, compare the temporarily generated 2017.1 Sage 300 SDK solution’s Web Project’s .csproj file with the Web Project file to update.

Copy over all the CrystalDecisions.\* <Reference> tags from the temp generated 2017.1 SDK solution.

1. Recompiling the Solution

The solution that supported the 2017 version of the SDK has now been manually upgraded to support 2017.1 (PU1)

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

**Issues with new GridField init properties**

* headerAttributes no longer accepts just a string type and requires the use of the type FinderConstant in the namepace Sage.CA.SBS.ERP.Sage300.Common.Web.Utilities

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

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