Sage 300 Web Screens SDK

Upgrade Guide for 2019.2 to 2020.0

August 2019

The MIT License (MIT)

Copyright © 1994-2019 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](#_Toc14101888)

[1.1 Required Version of Sage 300 4](#_Toc14101889)

[2. Installing the Sage 300 Upgrade Wizard 5](#_Toc14101890)

[3. Accessing the Sage 300 2020.0 Upgrade Wizard 6](#_Toc14101891)

[4. Using the Sage 300 Upgrade Wizard 7](#_Toc14101892)

[4.1 Step 1 – Synchronize Kendo Files 8](#_Toc14101893)

[4.2 Step 2 – Synchronize Web Files 9](#_Toc14101894)

[4.3 Step 3 – Synchronize Accpac Libraries 10](#_Toc14101895)

[4.4 Step 4 – Multiple Sessions framework changes 11](#_Toc14101896)

[4.5 Step 5 – Confirmation 12](#_Toc14101897)

[4.6 Step 6 – Recompile 13](#_Toc14101898)

[5. Upgrade Log 14](#_Toc14101899)

[6. Multiple Sessions framework changes 15](#_Toc14101900)

[7. Compilation Troubleshooting 20](#_Toc14101901)

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 2019.2 to Web Screens SDK 2020.0.

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

* Sychronize Kendo Files
* Synchronize Web Files
* Synchronize Accpac Libraries
* Multiple Sessions framework changes

* 1. Required Version of Sage 300

To get started, install Sage 300 2020.0 with the Web Screens option selected.

1. Installing the Sage 300 Upgrade Wizard

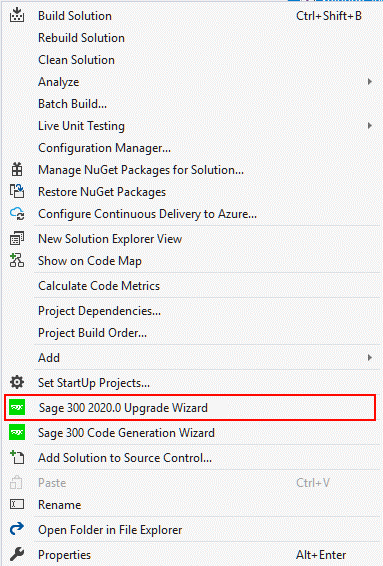
The Upgrade Wizard is a Visual Studio Plugin that was developed in Visual Studio 2017 and is compatible with Visual Studio 2017 only.

**Note:** If the previous Upgrade Wizard is installed, it must be uninstalled first (In Visual Studio - Tools, Components and Extensions, Uninstall option once the Upgrade Wizard is selected).

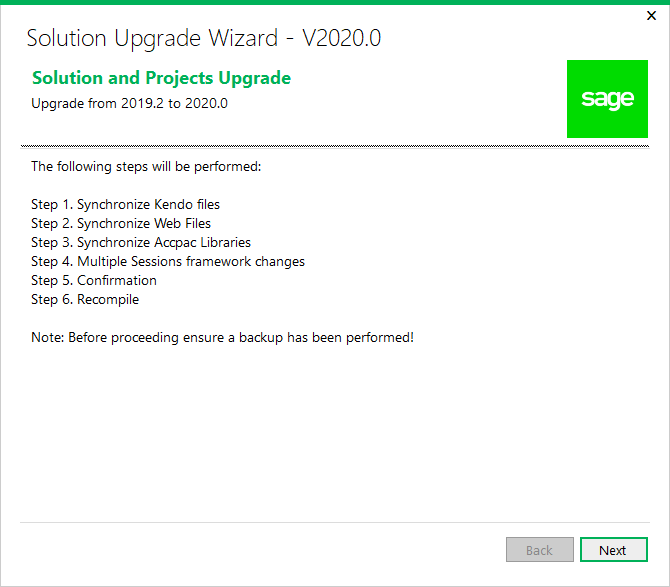
To install the wizard:

1. Run Sage300UpgradeWizardPackage.vsix.
2. Select the Install the Wizard Package option.
3. Accessing the Sage 300 2020.0 Upgrade Wizard

The **Sage 300 2020.0 Upgrade Wizard** is a Visual Studio Plugin. To open it, with the partner solution loaded in Visual Studio, right-click the solution, and on the context menu, click Sage 300 2020.0 Upgrade Wizard.



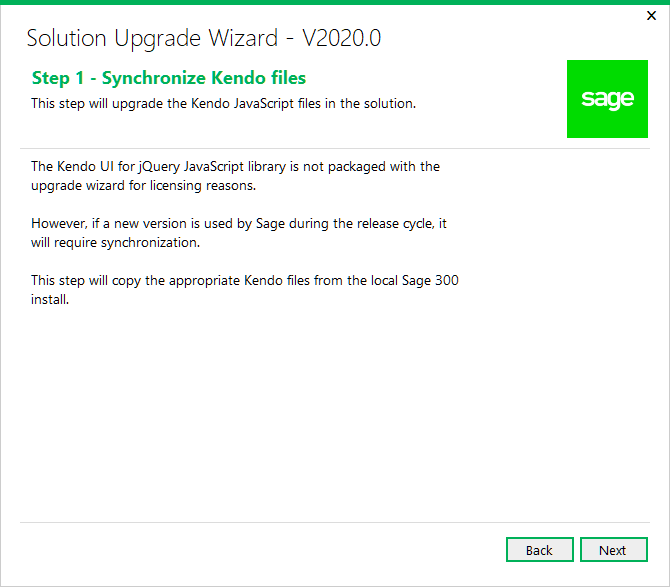
1. Using the Sage 300 Upgrade Wizard



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

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

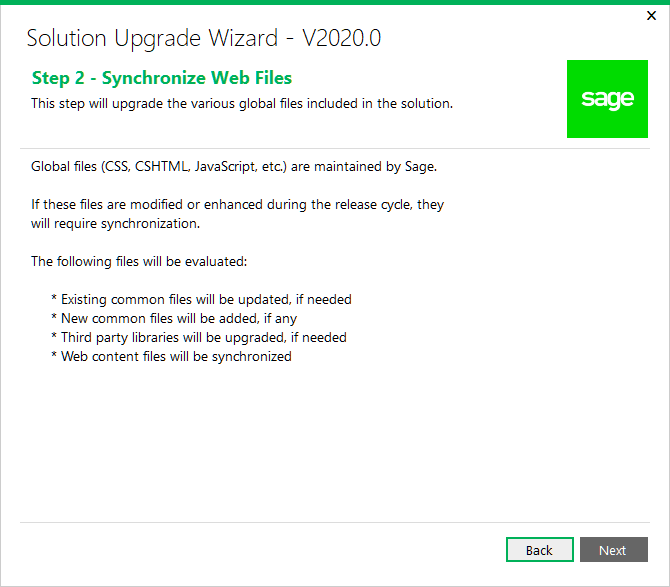
* 1. Step 1 – Synchronize Kendo Files



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.

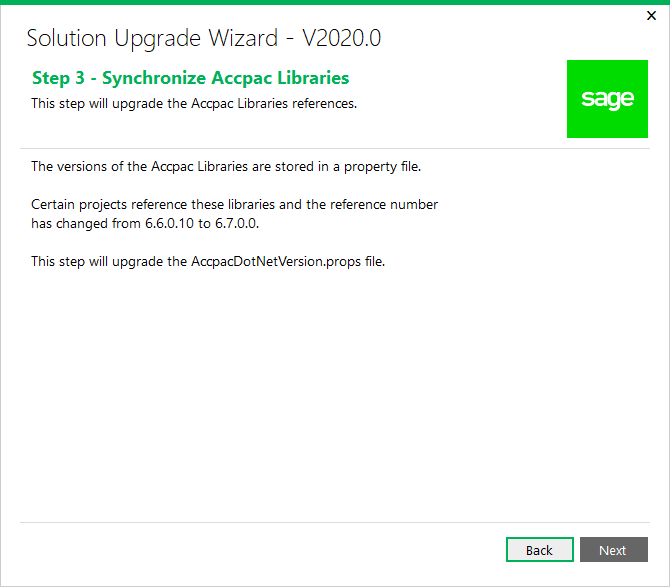
* 1. Step 2 – Synchronize Web Files



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



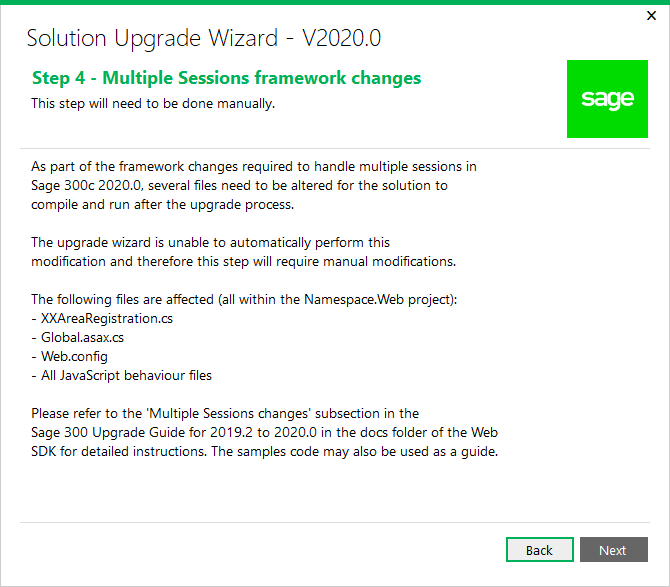
The upgrade wizard will search for copies of the **AccpacDotNetVersion.props** file in the following locations:

* + Root of the Solution folder
    - If found, it will stop looking. If not found, it will then search in the root of each project folder within the solution.

If the **AccpacDotNetVersion.props** file is found in the root of the solution folder, it will be upgraded as normal. If the **AccpacDotNetVersion.props** file is not found in the root of the solution folder but is found in one or more project folders, the wizard will update the project file (.csproj) in each of these folders with the correct path to the **AccpacDotNetVersion.props** file located in the solution folder and then remove the copy of the **AccpacDotNetVersion.props** file in the project folder. When this process has completed, the solution should contain only a single **AccpacDotNetVersion.props** file located in the root of the solution folder.

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

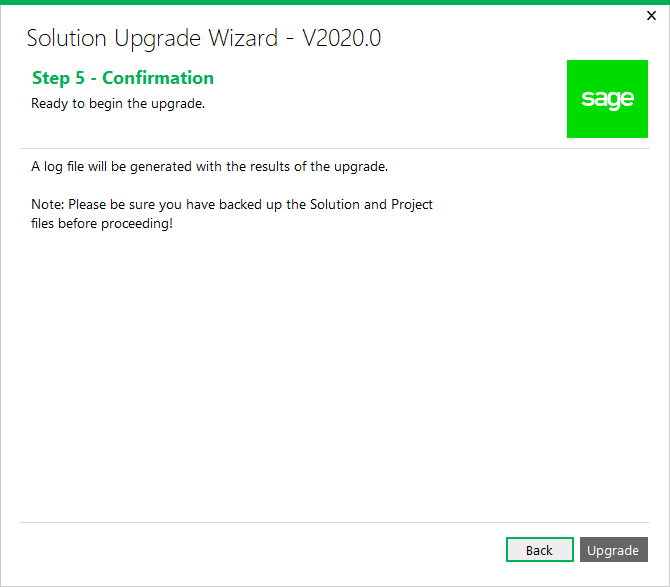
* 1. Step 4 – Multiple Sessions framework changes



This is simply an informational step. See Section 6 – Multiple Sessions framework changes for full details on the manual modifications required.

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

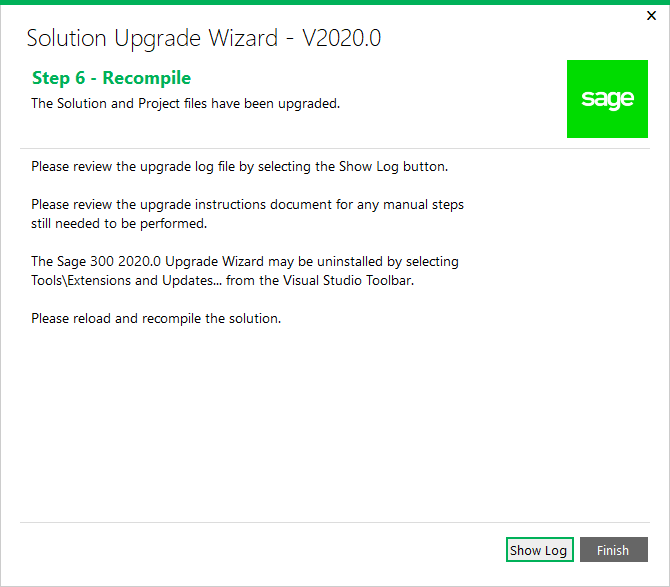
* 1. Step 5 – Confirmation



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 6 – 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.

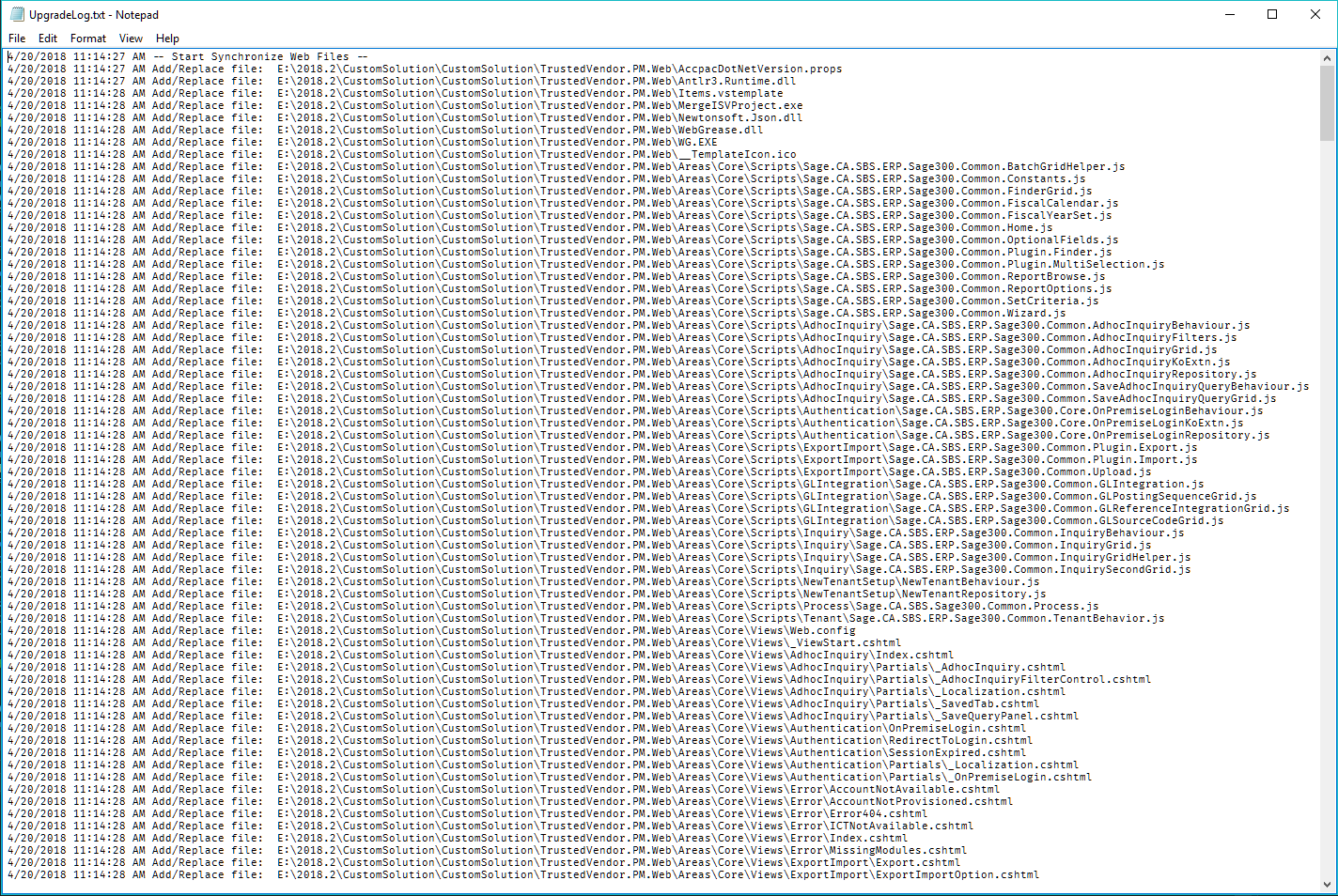
Click **Finish** to exit the wizard.

After the solution that supported Sage 300 2019.2 has been upgraded to support 2020.0, you need to reload 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. 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. Multiple Sessions framework changes

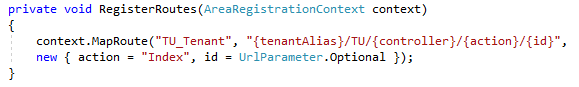
* **Files affected**

{Namespace.Web}\{Namespace}AreaRegistration.cs

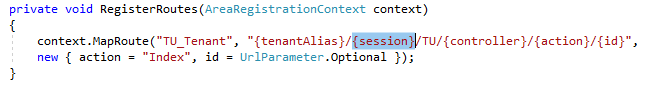
* **Description**

Change the routing configuration to add a session id to the URL pattern

* **Before Change**



* **After Change**



* **Files affected**

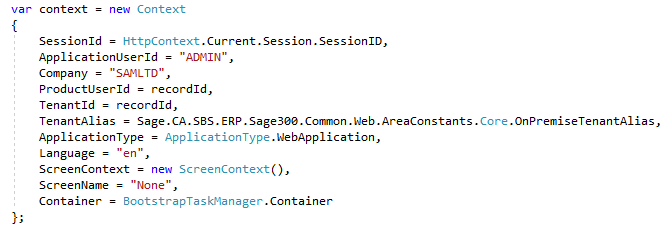
{Namespace.Web}\Global.asax.cs

Any other place that creates a new Context object

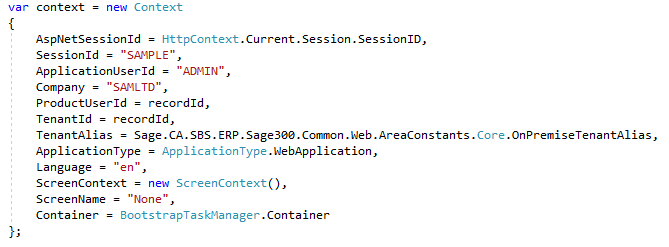
* **Description**

Repurpose the use of the old property SessionId and additionally pass in a new property called AspNetSessionId in the Context object.

* **Before Change** 
  + Value of Context.SessionId was the ASP.NET session id
  + Context.AspNetSessionId does not exist



* **After Change**
  + Value of Context.SessionId is the unique session id that is assigned during the login process to each unique user/company login.
  + Value of Context.AspNetSessionId is the ASP.NET session id



* **Files affected**

{Namespace.Web}\Global.asax.cs

* **Description**

Remove the sessionId parameter from the AuthenticationManager.LoginResult function

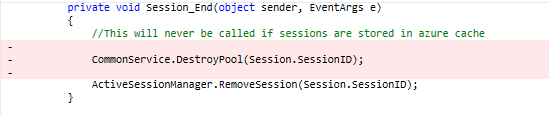
Remove the CommonService.DestroyPool function call in Session\_End

* **Before Change**



* **After Change**





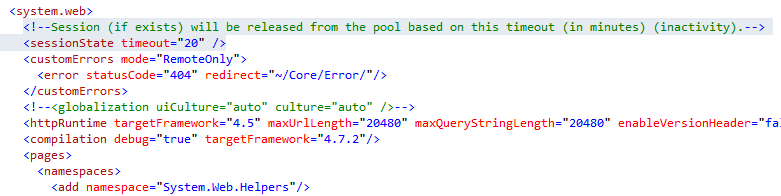
* **Files affected**

{Namespace.Web}\Web.config

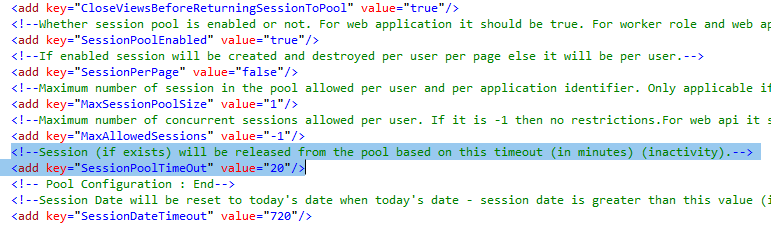
* **Description**

Add a new line to Web.config, this is the value that both the client-side and server-side timeout now respects. Remove the old timeout value (benign but serves no purpose anymore).

* **Add the highlighted line**



* **Remove the highlighted line**



* **Files affected**

All JavaScript behaviour files

{Namespace.Web}\Areas\{Namespace}\Scripts\{ScreenName}\Behaviour.js

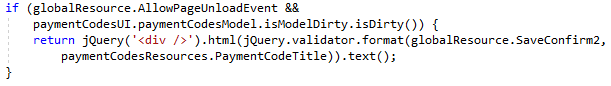
* **Description**

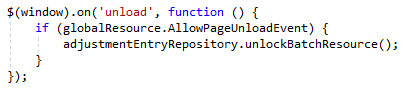
Change the dirty flag call in the ‘beforeunload’ event. This will usually be found at the end of the behaviour file. You need to extract out the isDirty check (if there is more than one, OR them together) and pass it into the new sg.utls.isPageUnloadEventEnabled function. Then, return a value through the new helper function sg.utls.getDirtyMessage.

Additionally, if there is cleanup done in the ‘unload’ event that is gated by the globalResource.AllowPageUnloadEvent, remove that check.

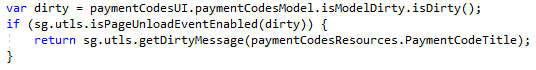
See below:

* **Before Change**





* **After Change**





* **Files affected**

{Namespace.Web}\WebForms\BaseWebPage.cs

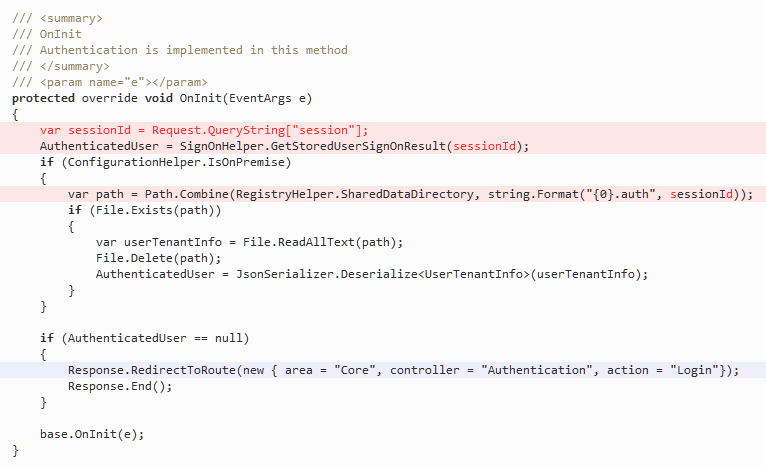
* **Description**

Add code to get and use the sessionId from the query string in the BaseWebPage.OnInit method.

* **Before Change**



* **After Change**



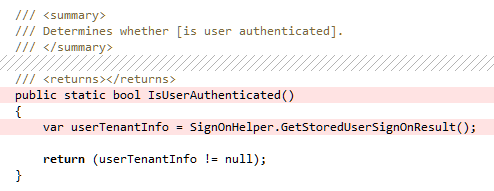
* **Files affected**

{Namespace.Web}\WebForms\BaseWebPage.cs

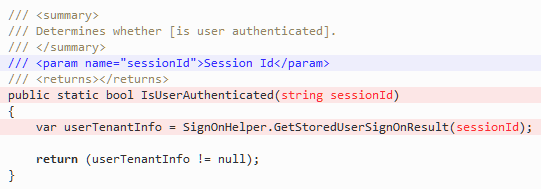
* **Description**

Add code to pass in the sessionId as a parameter to the BaseWebPage.IsUserAuthenticated method.

* **Before Change**



* **After Change**



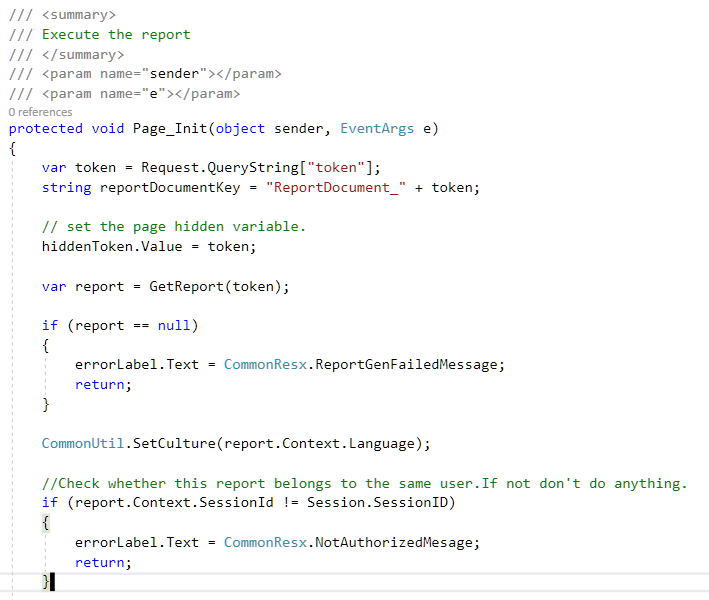
* **Files affected**

{Namespace.Web}\WebForms\ReportViewer.aspx.cs (multiple locations in file)

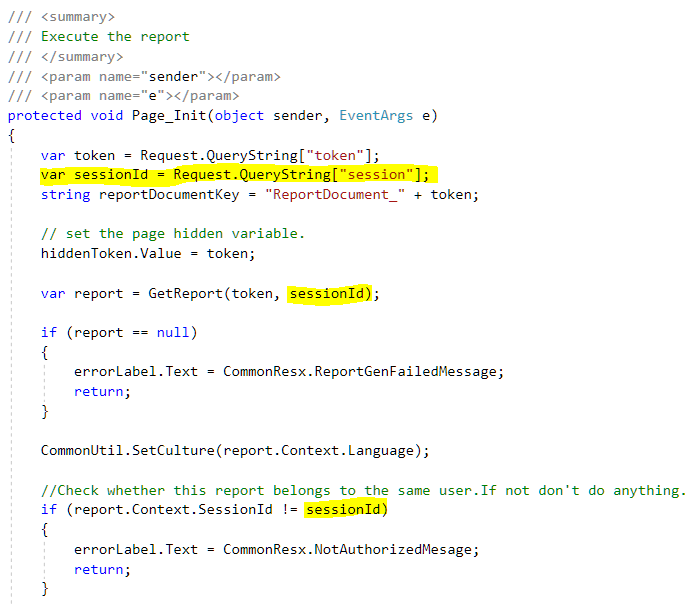
* **Description**

Add code to handle sessionId query string parameter in ReportViewer.Page\_Init method.

* **Before Change**



* **After Change**



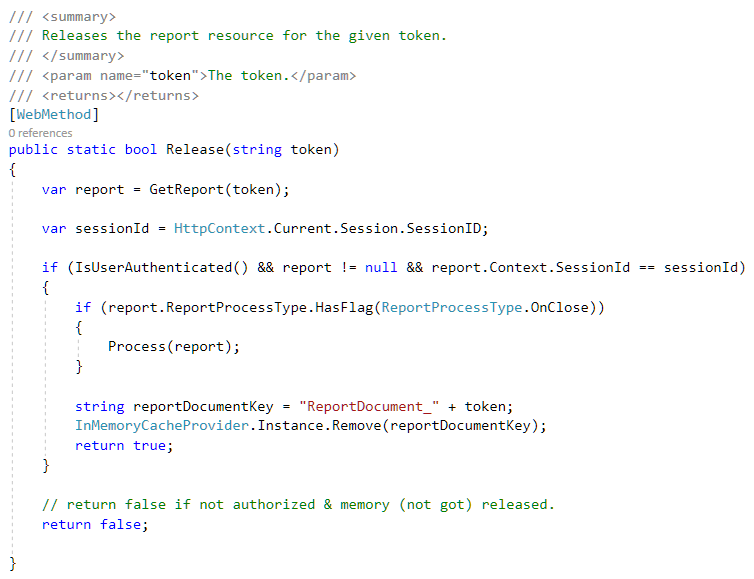
* **Files affected**

{Namespace.Web}\WebForms\ReportViewer.aspx.cs (multiple locations in file)

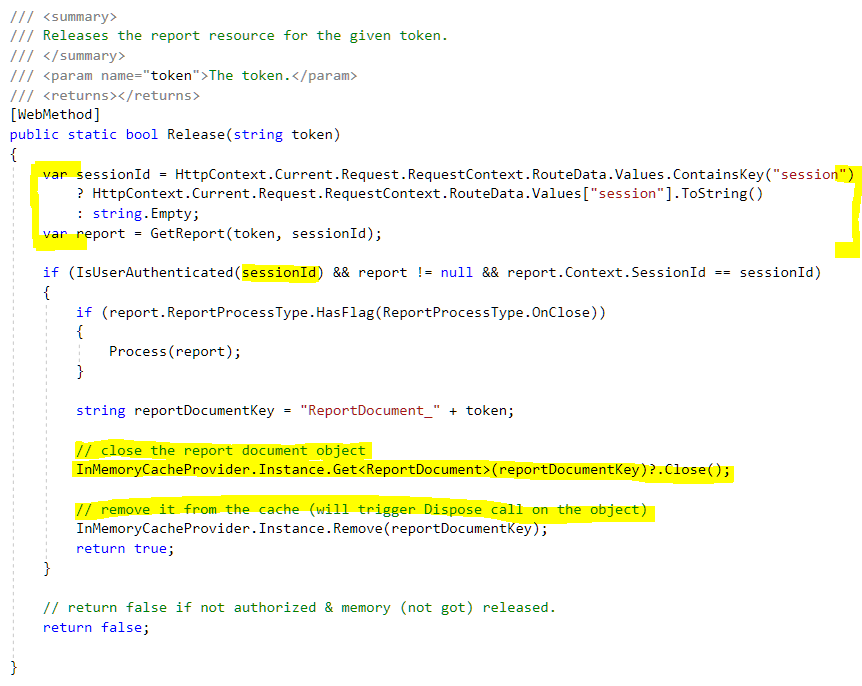
* **Description**

Add code to handle new sessionId variable in ReportViewer.Release method.

* **Before Change**



* **After Change**



* **Files affected**

{Namespace.Web}\WebForms\ReportViewer.aspx.cs (multiple locations in file)

* **Description**

Add code to handle new sessionId variable in ReportViewer.GetReport method.

* **Before Change**



* **After Change**



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