Sage 300 Web Screens SDK

MergeISVProject Utility

July 2020

The MIT License (MIT)

Copyright © 1994-2020 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](#_Toc45550518)

[2. When to use this application 7](#_Toc45550519)

[3. How to use this application 8](#_Toc45550520)

[4. What does this program do? 10](#_Toc45550521)

[5. Example Deployment Folder Structure 11](#_Toc45550522)

[6. Deployment Flow 15](#_Toc45550523)

[6.1 Deployment Summary 19](#_Toc45550524)

[7. Post-Build Binary Locations 21](#_Toc45550525)

1. Overview

The **MergeISVProject** utility application is a C# project developed in Visual Studio 2017 and is compatible with Visual Studio 2017 and 2019. It is used bundle up all necessary files from a partner solution, minify JavaScript files and optionally deploy to a local Sage 300 installation. The following sections will outline what the program does and how/when to call it.

|  |  |  |
| --- | --- | --- |
| **Required Parameters (Default prefix for parameters is --)** | | |
| **Parameter Name** | **Parameter Type and Meaning** | **Example Usage** |
| mode | Integer:  0 – Target a full visual studio solution.  1 – Target a single project | --mode=0 |
| solutionpath | String:  This is the fully-qualified path to the Microsoft Visual Studio Solution  Note: Because the macro $(SolutionDir) expands out to include a trailing backslash, an extra backslash is required. See the example to the right. | --solutionpath=”$(SolutionDir)\” |
| webprojectpath | String:  This is the fully-qualified path to the Microsoft Visual Studio Solution Web project  Note: Because the macro $(ProjectDir) expands out to include a trailing backslash, an extra backslash is required. See the example to the right. | --webprojectpath=”$(ProjectDir)\” |
| menufilename | String:  This is the name of the Sage 300 menu definition file name. | --menufilename=”TUMenuDetail.xml” |
| Buildprofile | String:  This is the Microsoft Visual Studio project build configuration. (only Release is currently supported) | --buildprofile=”$(ConfigurationName)” |
| dotnetframeworkpath | String:  This is the fully-qualified path to the Microsoft .NET framework containing the aspnet\_compile.exe program. | --dotnetframeworkpath=”$(FrameworkDir)$(FrameworkVersion)” |
|  | | |
| **Optional Parameters (Default prefix for parameters is --)** | | |
| **Parameter Name** | **Parameter Type and Meaning** | **Example Usage** |
| minify | Boolean Flag:  If specified, instructs program to minify JavaScript files. | --minify |
| nodeploy | Boolean Flag:  If specified, instructs program to NOT deploy files to the local Sage 300 installation. | --nodeploy |
| testdeploy | Boolean Flag:  If specified, instructs program to SIMULATE deployment to the local Sage 300 installation. | --testdeploy |
| log | Boolean Flag:  If specified, instructs the program to log operations to a log file for diagnostic purposes. | --log |
| extraresourcelanguages | String:  If specified, instructs program to use the specified language resources, in addition to the ones Sage 300 handles by default (English, French, Spanish, Chinese -Simplified and Chinese – Traditional) | --extraresourcelanguages=”fr-ca,jp” |

1. When to use this application

This application is automatically called during the post-build step of the Web project when building a Sage 300 solution. There is no need to call this application manually.

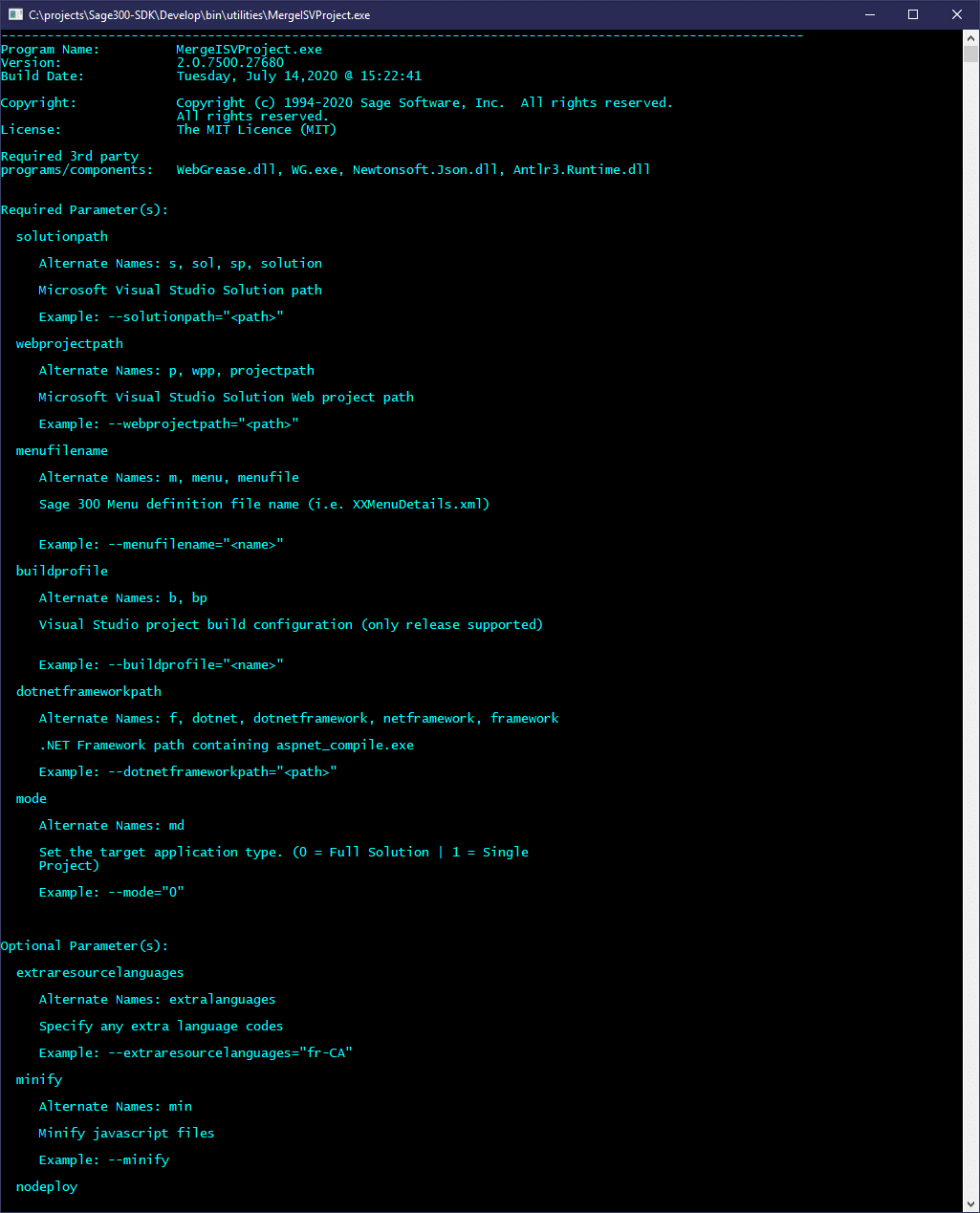
1. How to use this application

Upon generation of a new Sage 300 solution via the Sage 300 Solution Wizard, the Post-Build event of the <Namespace>.Web.csproj will contain the following:

|  |
| --- |
| <PropertyGroup>  <PostBuildEvent>  Call "$(ProjectDir)MergeISVProject.exe" --mode=0 --solutionpath="$(SolutionDir)\"  --webprojectpath="$(ProjectDir)\" --menufilename="$applicationid$MenuDetails.xml"  --buildprofile="$(ConfigurationName)" --dotnetframeworkpath="$(FrameworkDir)$(FrameworkVersion)"  --minify –log --extraresourcelanguages=”fr-ca,jp”  </PostBuildEvent>  </PropertyGroup> |

Most of the parameters are automatically picked up from the Microsoft Visual Studio build environment and there is no need to change any of them.

Running the MergeISVProject.exe program without any command-line parameters will display help for the program. This is illustrated in the following screen-shot:



1. What does this program do?

This program performs the following steps:

|  |  |
| --- | --- |
| **Full Solution (mode=0)** | **Single Project (mode=1)** |
| Staging Phase Steps (Copy files to staging folder(s)) | |
| * Web Config * Areas * Bin * Compile Views * Bootstrapper * Menus * Minify Javascript Files (Optional Step) * Images * Resource Satellite Files | * Bin * Bootstrapper |
| Deploy Phase Steps (Optional) (Copy files to local Sage 300 installation) | |
| * Bootstrapper * Menu Details * Images * Area Scripts * Compiled Views * Bin Folders * Resource Satellite Files | * Bootstrapper * Bin Folders |

1. Example Deployment Folder Structure

The following is an example of the folder/file structure that this application creates. This list contains only a subset of files that are created and some of the filenames are vendor-specific and will have a different name for other vendors. Files and Folder names that contain bolded text between square brackets [] are variable and will depend on the module id and/or Vendor name. For this example, the Vendor name is ‘Vendor’ and the Module ID is ‘XX’

\_\_\_DEPLOY\_\_\_

│

├───Compiled

│ │ PrecompiledApp.config

│ │ Web.config

│ │

│ ├───Areas

│ │ └───**[XX]**

│ │ ├───Scripts

│ │ │ ├───Common

│ │ │ │ **[Vendor]**CommonControlHelpers.js

│ │ │ │ **[Vendor]**CommonFinderHelper.js

│ │ │ │ **[Vendor]**CommonGridHelpers.js

│ │ │ │ **[Vendor]**CommonIntellisense.js

│ │ │ │ **[Vendor]**CommonJavaConstants.js

│ │ │ │ **[Vendor]**CommonKoExtn.js

│ │ │ │ **[Vendor]**CommonRespository.js

│ │ │ │ **[Vendor]**CommonSegmentsGrid.js

│ │ │ │

│ │ │ └───**[CompanyOptions]**

│ │ │ **[VendorCompanyOptions]**Behaviour.js

│ │ │ **[VendorCompanyOptions]**KoExtn.js

│ │ │ **[VendorCompanyOptions]**Repository.js

│ │ │

│ │ │

│ │ |───Views

│ │ | │ Web.config

│ │ | │ \_ViewStart.cshtml

│ │ | │

│ │ | └───**[CompanyOptions]**

│ │ | │ Index.cshtml

│ │ | │

│ │ | └───Partials

│ │ | \_**[CompanyOptions]**.cshtml

│ │ | \_Localization.cshtml

| | |

│ │ │

│ │ └───ExternalContent

│ │ │ bg\_menu\_xx.jpg

│ │ │ icon\_xx.png

│ │

│ └───bin

│ index.cshtml.fc773fc8.compiled

│ Microsoft.Web.Infrastructure.dll

│ Sage.CA.SBS.ERP.Sage300.AP.BusinessRepository.dll

│ Sage.CA.SBS.ERP.Sage300.AP.Interfaces.dll

│ Sage.CA.SBS.ERP.Sage300.Common.Web.dll

│ Sage.CA.SBS.ERP.Sage300.Web.dll

│ Sage.CA.SBS.ERP.Sage300.Workflow.DataAccess.dll

│ System.Spatial.dll

│ System.Web.WebPages.Deployment.dll

│ **[Vendor].[XX]**.BusinessRepository.dll

│ **[Vendor].[XX]**.Interfaces.dll

│ **[Vendor].[XX]**.Models.dll

│ **[Vendor].[XX]**.Resources.dll

│ **[Vendor].[XX]**.Services.dll

│ **[Vendor].[XX]**.Web.dll

│ \_**[companyoptions]**.cshtml.b0f8ce23.compiled

│ \_localization.cshtml.f0fa7b9.compiled

│ \_viewstart.cshtml.c780fcea.compiled

│

│

│

├───Staging

│ │ **[XX]**bootstrapper.xml

│ │ Web.config

│ │

│ ├───App\_Data

│ │ └───MenuDetail

│ │ **[XX]**MenuDetails.xml

│ │

│ ├───Areas

│ │ └───**[XX]**

│ │ ├───Scripts

│ │ │ ├───Common

│ │ │ │ **[Vendor]**CommonControlHelpers.js

│ │ │ │ **[Vendor]**CommonFinderHelper.js

│ │ │ │ **[Vendor]**CommonGridHelpers.js

│ │ │ │ **[Vendor]**CommonIntellisense.js

│ │ │ │ **[Vendor]**CommonJavaConstants.js

│ │ │ │ **[Vendor]**CommonKoExtn.js

│ │ │ │ **[Vendor]**CommonRespository.js

│ │ │ │ **[Vendor]**CommonSegmentsGrid.js

│ │ │ │

│ │ │ └───**[CompanyOptions]**

│ │ │ **[VendorCompanyOptions]**Behaviour.js

│ │ │ **[VendorCompanyOptions]**KoExtn.js

│ │ │ **[VendorCompanyOptions]**Repository.js

│ │ │

│ │ │

│ │ |───Views

│ │ | │ Web.config

│ │ | │ \_ViewStart.cshtml

│ │ | │

│ │ | └───**[CompanyOptions]**

│ │ | │ Index.cshtml

│ │ | │

│ │ | └───Partials

│ │ | \_**[CompanyOptions]**.cshtml

│ │ | \_Localization.cshtml

│ │ |

| | |

│ │ │

│ │ └───ExternalContent

│ │ │ bg\_menu\_xx.jpg

│ │ │ icon\_xx.png

│ │

│ ├───bin

│ │ │ App\_Web\_index.cshtml.fc773fc8.dll

│ │ │ App\_Web\_\_**[companyoptions]**.cshtml.b0f8ce23.dll

│ │ │ index.cshtml.146321df.compiled

│ │ │ **[Vendor]**.**[XX]**.BusinessRepository.dll

│ │ │ **[Vendor]**.**[XX]**.Interfaces.dll

│ │ │ **[Vendor]**.**[XX]**.Resources.dll

│ │ │ **[Vendor]**.**[XX]**.Services.dll

│ │ │ **[Vendor]**.**[XX]**.Web.dll

│ │ │ \_**[companyoptions]**.cshtml.b0f8ce23.compiled

│ │ │ \_localization.cshtml.d21b552a.compiled

│ │ │

│ │ ├───es

│ │ │ **[Vendor]**.**[XX]**.Resources.resources.dll

│ │ │

│ │ ├───zh-Hans

│ │ │ **[Vendor]**.**[XX]**.Resources.resources.dll

│ │ │

│ │ └───zh-Hant

│ │ **[Vendor]**.**[XX]**.Resources.resources.dll

│

│

└───\_\_READYTODEPLOY\_\_

│

├───Web

│ │ **[XX]**bootstrapper.xml

│ ├─App\_Data

│ │ └───MenuDetail

│ │ **[XX]**MenuDetails.xml

│ │

│ ├─Areas

│ │ └─**[XX]**

│ │ ├───Scripts

│ │ │ ├───Common

│ │ │ │ **[Vendor]**CommonControlHelpers.js

│ │ │ │ **[Vendor]**CommonFinderHelper.js

│ │ │ │ **[Vendor]**CommonGridHelpers.js

│ │ │ │ **[Vendor]**CommonIntellisense.js

│ │ │ │ **[Vendor]**CommonJavaConstants.js

│ │ │ │ **[Vendor]**CommonKoExtn.js

│ │ │ │ **[Vendor]**CommonRespository.js

│ │ │ │ **[Vendor]**CommonSegmentsGrid.js

│ │ │ │

│ │ │ └───**[CompanyOptions]**

│ │ │ **[VendorCompanyOptions]**Behaviour.js

│ │ │ **[VendorCompanyOptions]**KoExtn.js

│ │ │ **[VendorCompanyOptions]**Repository.js

│ │ │

│ │ ├───Views

│ │ │ │ Web.config

│ │ │ │ \_ViewStart.cshtml

│ │ │ │

│ │ │ └───**[CompanyOptions]**

│ │ │ │ Index.cshtml

│ │ │ │

│ │ │ └───Partials

│ │ │ \_**[CompanyOptions]**.cshtml

│ │ │ \_Localization.cshtml

│ │ │

│ │ │

│ │ │

│ │ └───ExternalContent

│ │ bg\_menu\_xx.jpg

│ │ icon\_xx.png

│ │

│ └─bin

│ │ App\_Web\_index.cshtml.e08d07b5.dll

│ │ App\_Web\_\_**[companyoptions]**.cshtml.b0f8ce23.dll

│ │ App\_Web\_\_localization.cshtml.ec026c03.dll

│ │ index.cshtml.8cd49998.compiled

│ │ \_localization.cshtml.2a78d606.compiled

│ │ **[Vendor].[XX]**.BusinessRepository.dll

│ │ **[Vendor].[XX]**.Interfaces.dll

│ │ **[Vendor].[XX]**.Models.dll

│ │ **[Vendor].[XX]**.Resources.dll

│ │ **[Vendor].[XX]**.Services.dll

│ │ **[Vendor].[XX]**.Web.dll

│ │

│ │

│ ├──es

│ │ **[Vendor].[XX]**.Resources.resources.dll

│ │

│ ├──zh-Hans

│ │ **[Vendor].[XX]**.Resources.resources.dll

│ │

│ └──zh-Hant

│ **[Vendor].[XX]**.Resources.resources.dll

└───Worker

**[XX]**bootstrapper.xml

**[Vendor].[XX]**.BusinessRepository.dll

**[Vendor].[XX]**.Interfaces.dll

**[Vendor].[XX]**.Models.dll

**[Vendor].[XX]**.Resources.dll

**[Vendor].[XX]**.Services.dll

1. Deployment Flow

The following table is a step-by-step description of the internal function calls that this application executes dependent on the mode setting specified on the command-line.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Notes:**  **Modes**  0 = Full Solution  1 = Single project only   |  |  | | --- | --- | | **Placeholder** | **Description** | | **[XX]** | Two letter module id | | **[SOURCE]** | …**[XX]**.Web folder (Web project folder) | | **[DEPLOY]** | [SOURCE]\\_\_\_DEPLOY\_\_\_\ | | **[FINAL]** | [DEPLOY]\\_\_READTODEPLOY\_\_\ | | **[SAGE300WEB]** | The location of the live Sage 300 installation Web folder on the local computer. The actual location will be dependent on how it was installed. For this example we’ll assume it is as follows: **C:\Sage\Sage300\Online\Web\** | | **[SAGE300WORKER]** | The location of the live Sage 300 installation Worker folder on the local computer. The actual location will be dependent on how it was installed. For this example we’ll assume it is as follows: **C:\Sage\Sage300\Online\Worker\** | | | |
| **‘Staging’ phase** | | |
| **The ‘Staging’ phase of the application is responsible for creating the deployment folder structure within the \_\_\_DEPLOY\_\_\_ folder that is located in the [SOURCE] folder.**  **Once the staging phase has completed, the final artifacts will be located in the following folder (and sub-folders):**  **[FINAL]\Web and [FINAL]\Worker**  **If the end user is planning to create an installer via some mechanism, the folder noted above (in blue) is where one will find the necessary components.** | | |
| **Mode** | **Method** | **Description** |
| 0 | \_Stage\_WebConfig() | Copy the Web.config file from the **[SOURCE]** folder to **[DEPLOY]**\Staging\ folder |
| 0 | \_Stage\_Areas() | Copy all folders and files from **[SOURCE]**\Areas\\* to **[DEPLOY]**\Staging\Areas\\*  Copies the ‘Views’, ‘Scripts’ and ‘ExternalContent’ folders (and any subfolders) |
| 0 & 1 | \_Stage\_Bin() | Copy filtered file list from **[SOURCE]**\bin\ to **[DEPLOY]**\Staging\bin\  "System.\*.dll",  "Sage.CA.SBS.ERP.\*.dll",  "\*.Web.Infrastructure.dll",  "\*.Web.dll",  $"\*.{XX}.\*.dll"  **Note**: If there exists a file named ‘**BinInclude.txt**’ in the root of the Web project AND this file contains a list of valid file names, then these files will also be copied to the **[DEPLOY]**\Staging\Bin\ folder. |
| 0 | \_Stage\_CompileViewsAndMoveResultsToStagingFolder() | Compile the Razor views into the ‘Compiled’ folder and compile them.  Clear out all files from the **[DEPLOY]**\Staging\bin\ folder  Copy all files from **[DEPLOY]**\Compiled\bin\ to **[DEPLOY]**\Staging\bin\  Clear out all files from the **[DEPLOY]**\Staging\Areas\ folder  Copy all files from **[DEPLOY]**\Compiled\Areas\ to **[DEPLOY]**\Staging\Areas\ |
| 0 & 1 | \_Stage\_Bootstrapper() | Copy one (or more) bootstrapper files from **[SOURCE]** folder to **[DEPLOY]**\Staging\ folder. The files are of the format **[XX]**Bootstrapper.xml where [**XX]** is a two letter module Id. |
| 0 | \_Stage\_Menus() | Copy menu definition file from **[SOURCE]** to **[DEPLOY]**\Staging\App\_Data\MenuDetail\ folder. The filename is of the format **[XX]**MenuDetails.xml |
| 0 | [Optional Step] \_Stage\_MinifyJavascripts() | Minify JavaScript files in the **[DEPLOY]**\Staging\Areas\**[XX]**\Scripts\[SomeModuleName] folder. This minification process is done in the same folder as the source JavaScript files. Once the minification process has completed, the unminified files are deleted from the folder. The minified file names are of the format:  **[VendorName].[XX].[SomeFunctionality].min.js**  If the solution contains multiple Module Id folders, they are all minified. |
| 0 | \_Stage\_ResourceSatelliteFiles() | Copy all resource satellite DLL’s from **[SOURCE]**\Bin\**[languge]**\ to **[DEPLOY]**\Staging\Bin\**[language]**\ folder. This is done for the following languages ‘es’, ‘fr’, ‘zh-hans’ and ‘zh-hant’.  The file pattern used is as follows: “\*.[XX].\*.dll”  If the command-line parameter –extraresourcelanguages has been defined, extra languages will be staged as well. |
| 0 & 1 | \_Stage\_CopyAllToFinal() | **This is the final stage in the STAGING process.**  Copy all staging files and folders to the final staging location.  This is done in two steps. The first step will copy all the necessary files for the Web deployment into the Web folder.  **[DEPLOY]**\Staging\ 🡪 **[FINAL]**\Web\  The second step will only the files necessary for the Worker deployment  **[DEPLOY]**\Staging\ 🡪 **[FINAL]**\Worker\ |
| **‘Deploy’ Phase (Optional)** | | |
| **The optional ‘Deploy’ phase of the application is responsible for copying the artifacts from the [FINAL]\**Web **folder to a locally hosted Sage 300 installation [SAGE300WEB] for testing purposes. Depending on the file and mode specified, some files are also copied to the [SAGE300WORKER] folder as well.** | | |
| **Mode** | **Method** | **Description** |
| 0 & 1 | \_Deploy\_Bootstrapper() | Copy one (or more) bootstrapper files from **[FINAL]**\Web folder to **[SAGE300WEB]**\ folder. The files are of the format **[XX]**Bootstrapper.xml where [**XX]** is a two letter module Id. There may be multiple bootstrapper files.  If mode = 0 (Full Solution), the same bootstrapper defined above is also copied to the **[SAGE300WORKER]**\ folder. |
| 0 | \_Deploy\_MenuDetails() | Copy menu definition file from **[FINAL]**\Web\App\_Data\ to **[SAGE300WEB]**\App\_Data\MenuDetail\ folder. The filename is of the format **[XX]**MenuDetails.xml.  In addition to copying it to the above folder, the program will also copy the menu details file into all sub-folders based on existing database (company) names. There can be one or more of these sub-folders. |
| 0 | \_Deploy\_AreaScripts() | Recursively copy all JavaScript folders and files from **[FINAL]**\Web\Areas\**[XX]**\Scripts\ to **[SAGE300WEB]**\Areas\**[XX]**\Scripts\ |
| 0 | \_Deploy\_CompiledViews() | Recursively copy compiled Razor views from **[FINAL]**\Web\Areas\**[XX]**\Views\ to **[SAGE300WEB]**\Areas\**[XX]**\Views\  Nothing is copied if the source folder does not exist. This usually means there was a problem during the compilation of the Razor views. |
| 0 | \_Deploy\_ExternalContent() | Recursively copy all ExternalContent folders and files from **[FINAL]**\Web\Areas\**[XX]**\ExternalContent\ to **[SAGE300WEB]**\Areas\**[XX]**\ExternalContent\ |
| 0 & 1 | \_Deploy\_BinFolders() | Copy the binary artifacts from the **[FINAL]**\Web\Bin\ folder to the **[SAGE300WEB]**\Bin.  **If Mode = 0 Then**  File types to be copied to Web folder  \*.compiled  App\_Web\_\*.dll  \*.Web.dll  \*.[XX].\*.dll  These files are copied to the **[SAGE300WEB]**\bin\ folder.  File types to be copied to Worker folder  \*.[XX].\*.dll  These files are copied to the **[SAGE300WORKER]**\ folder.  **Note**: If there exists a file named ‘**BinInclude.txt**’ in the root of the Web project AND this file contains a list of valid file names, then these files will also be copied to the **[SAGE300WEB]**\Bin\ and **[SAGE300WORKER]**\ folders.  **If Mode = 1 Then**  File types to be copied to Web folder  \*.\*.Web.dll  These files are copied to the **[SAGE300WEB]**\bin\ folder.  **Note**: If there exists a file named ‘**BinInclude.txt**’ in the root of the Web project AND this file contains a list of valid file names, then these files will also be copied to the **[SAGE300WEB]**\Bin\ folder. |
| 0 | \_Deploy\_ResourceSatelliteFiles() | Copy all resource satellite DLL’s from **[FINAL]**\Web\Bin\**[language]**\ to the following folders:  **[SAGE300WEB]**\Bin\**[language]**\  **[SAGE300WORKER]**\**[language]**\  This is done for the following languages ‘es’, ‘fr’, ‘zh-hans’ and ‘zh-hant’.  If the command-line parameter –extraresourcelanguages has been defined, extra languages will be deployed as well.  The file pattern used is as follows: “\*.[XX].\*.dll” |

* 1. Deployment Summary

The purpose of this section is to summarize what files need to be copied where when deploying them into a live Sage 300 installation. More information is available in the table defined in the previous Section 6.

1. **Deploy the bootstrappers:**
   1. [Mode == 0 and 1]

Copy the [XX]Bootstrapper.xml file from **[FINAL]**\Webto **[SAGE300WEB].**

* 1. [Mode == 0]

Copy the Copy the [XX]Bootstrapper.xml file from **[FINAL]**\Webto **[SAGE300WORKER].**

1. **Deploy the menu definition file:**
   1. [Mode == 0 only]

Copy the menu definition file [XX]MenuDetails.xml from **[FINAL]**\Webto **[SAGE300WEB]**\App\_Data\MenuDetail\

*In addition to copying it to the above folder, the program will also copy the menu details file into all sub-folders based on existing database (company) names. There can be one or more of these sub-folders*.

1. **Deploy Areas/Scripts:**
   1. [Mode == 0 only]

Recursively copy all JavaScript folders and files from **[FINAL]**\Web\Areas\**[XX]**\Scripts\ to **[SAGE300WEB]**\Areas\**[XX]**\Scripts\

1. **Deploy Areas/ExternalContent:**
   1. [Mode == 0 only]

Recursively copy all ExternalContent folders and files from **[FINAL]**\Web\Areas\**[XX]**\ExternalContent\ to **[SAGE300WEB]**\Areas\**[XX]**\ExternalContent\

1. **Deploy Compiled Views:**
   1. [Mode == 0 only]

Recursively copy compiled Razor views from **[FINAL]**\Web\Areas\**[XX]**\Views\ to **[SAGE300WEB]**\Areas\**[XX]**\Views\

*Nothing is copied if the source folder does not exist. This usually means there was a problem during the compilation of the Razor views.*

1. **Deploy Bin Folders:**
   1. [Mode == 0 only]

Copy the following file types from **[FINAL]**\Web\Bin\ to **[SAGE300WEB]**\Bin\

* + - \*.compiled
    - App\_Web\_\*.dll
    - \*.Web.dll
    - \*.[XX].\*.dll

Copy the following file types from **[FINAL]**\Web\Bin\ to **[SAGE300WORKER]**\

* + - \*.[XX].\*.dll

**Note**: If there exists a file named ‘**BinInclude.txt**’ in the root of the Web project AND this file contains a list of valid file names, then these files will also be copied to the **[SAGE300WEB]**\Bin\ and **[SAGE300WORKER]**\ folders.

* 1. [Mode == 1 only]

Copy the following file types from **[FINAL]**\Web\Bin\ to **[SAGE300WEB]**\Bin\

* + - \*.\*.Web.dll

**Note**: If there exists a file named ‘**BinInclude.txt**’ in the root of the Web project AND this file contains a list of valid file names, then these files will also be copied to the **[SAGE300WEB]**\Bin\ folder.

1. **Deploy Resource Satellite Files:**
   1. [Mode == 0 only]

Copy all resource satellite DLL’s (\*.[XX].\*.dll) from **[FINAL]**\Web\Bin\**[language]**\ to the following folders:

**[SAGE300WEB]**\Bin\**[language]**\

**[SAGE300WORKER]**\**[language]**\

This is done for the following languages ‘es’, ‘fr’, ‘zh-hans’ and ‘zh\_hant’, if the resources have been defined.

If the command-line parameter –extraresourcelanguages has been defined, extra languages will be deployed as well.

1. Post-Build Binary Locations

There is a post-build step that will copy the **Release** version of MergeISVProject.exe into the following folders::

* \bin\utilities\
* \src\wizards\UpgradeWizardTemplates\Items\
* \src\wizards\Templates\Web\

Once the MergeISVProject.exe file has been copied into the above folders, the following items will need to be rebuilt:

* Sage300UIWizardPackage.vsix
  + Contains both the Sage 300 Solution Wizard and Sage 300 Code Generation Wizard
* Sage300UpgradeWizardPackage.vsix
  + Contains the Sage 300 Upgrade Wizard