Sage 300 Web Screens SDK

Web Screen Customization Wizards

July 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](#_Toc487546584)

[2. Customization Wizards 5](#_Toc487546585)

[2.1 Customization Wizard - Standalone 6](#_Toc487546586)

[2.2 Customization Wizard – Plug-in 6](#_Toc487546587)

[3. Development Environment Prerequisites 7](#_Toc487546588)

[3.1 Install Sage 300 7](#_Toc487546589)

[3.2 Install the Web Customization Package (plug-in) 7](#_Toc487546590)

[4. Creating the Customization 8](#_Toc487546591)

[4.1 Standalone Customization Wizard 8](#_Toc487546592)

[4.1.1 Step 1 – Add/Edit a Customization 8](#_Toc487546593)

[4.1.2 Step 2 – Add/Edit Screens 9](#_Toc487546594)

[4.1.3 Step 3 – Add/Edit Controls 10](#_Toc487546595)

[4.1.4 Step 4 – Generate Customization Files 12](#_Toc487546596)

[4.2 Plug-in Customization Wizard (optional) 13](#_Toc487546597)

[4.2.1 Step 1 – Create a New Project 13](#_Toc487546598)

[4.2.2 Step 2 – Generate a Customization Solution 13](#_Toc487546599)

[4.3 Debugging 15](#_Toc487546600)

1. Overview

This document is intended to serve as a guide for illustrating how to perform a Sage 300 web screen customization (customization).

The first, and most crucial step in the customization process, is to become familiar with the web screen to be customized.

Customization steps:

1. Familiarity with web screen being customized
2. Creation of JSON Manifest file (customization definition)
3. Optional creation of XML file(s) (control(s) for screen(s))
4. Optional creation of XSD file (based upon XML file creation)
5. Creation of JavaScript file(s) (business logic and/or control behavior)
6. Optional creation of Visual Studio Solution (MVC for extending functionality)

This guide performs a mock customization of the Order Entry screen:

* Adding controls
* Binding new controls to existing fields and newly added fields
* Intercepting events of existing controls and newly added controls

1. Customization Wizards

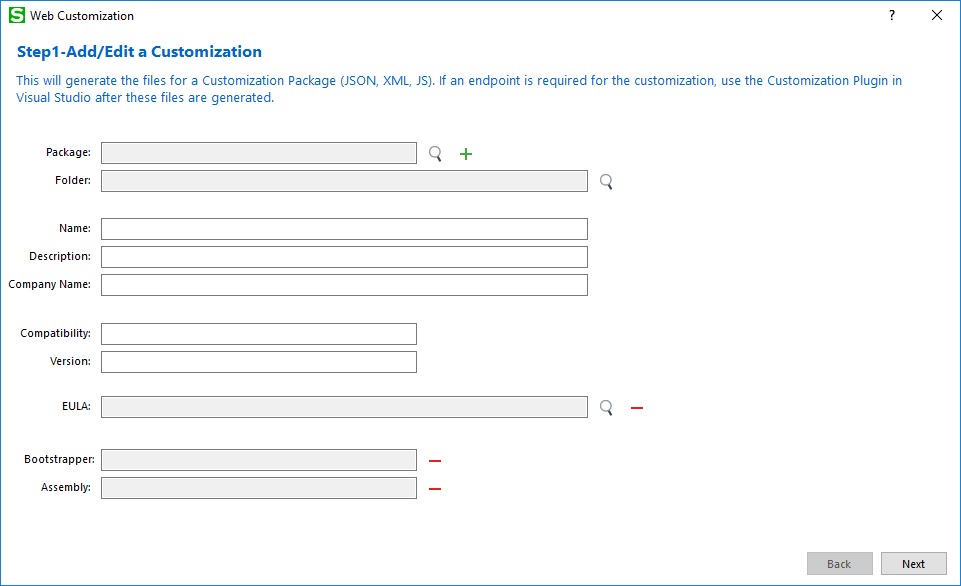
Prior to Sage 300 2018, the Customization Wizard was a Visual Studio Plug-in. The wizard created the JavaScript Template file, the XSD Schema file and the Visual Studio Solution for extending functionality via an MVC project.

However, this approach had numerous draw backs:

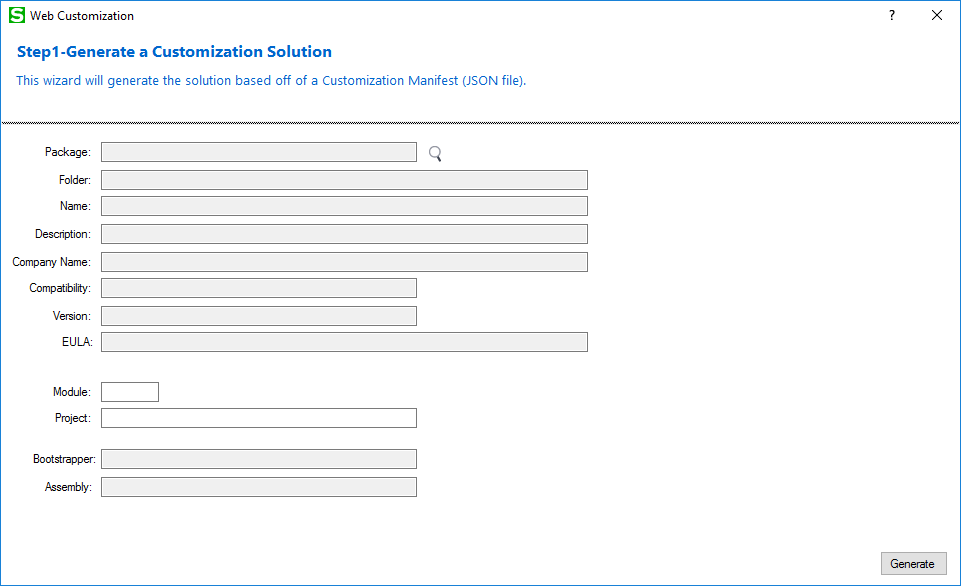
* The developer was forced to go into Visual Studio and created a solution even though an MVC project was not required if the customization was only going to be intercepting Sage control event and/or adding controls to a screen
* The developer was forced to manually create the JSON Manifest file
* The developer was forced to manually create the XML file(s) for controls that will be added to a screen

Sage 300 2018 introduced a meaningful change to the Customization process by splitting the existing Customization Wizard into two wizards:

* Customization Wizard – Standalone
  + This new standalone wizard simplifies and enhances the customization experience:
    - Developers no longer forced into Visual Studio to create and setup a customization that does not require a Visual Studio Solution
    - WYSWYG editor for JSON and XML files. No more creating these files by hand
    - Validation of JSON and XML files
* Customization Wizard – Plug-in
  + This revised plug-in wizard simplifies the solution creation by reading the JSON manifest created in the standalone wizard
  + Developers only access this plug-in if the customization requires an MVC project
  1. Customization Wizard - Standalone



* 1. Customization Wizard – Plug-in



1. Development Environment Prerequisites

Before starting the customization, Sage 300 must be installed on the developer machine along with the Plug-in Customization Wizard.

Familiarity with the web screen to be customized is a requirement. Additionally, knowledge and expertise in the following technologies are required:

* Microsoft ASP.NET MVC 5
* Microsoft C#
* JavaScript
* JQuery and JQuery Ajax (<https://jquery.com/>, <http://api.jquery.com/jquery.ajax/>)
* Kendo UI (<http://www.telerik.com/kendo-ui>)
* Knockout JavaScript Library (<http://knockoutjs.com/documentation/introduction.html>)
  1. Install Sage 300

Install Sage 300 on the developer machine. After installation, change the web application’s web.config file’s debug mode:

1. In {*Sage 300 installation folder*}\Online\Web, open the web.config file.
2. Locate the <System.web> entry and the <compilation debug="**false**" targetFramework="4.x.x" /> tag, where 4.x.x is version number, e.g. 4.5.1
3. Change the debug attribute to <compilation debug="**true**" targetFramework="4.x.x" />
   1. Install the Web Customization Package (plug-in)

The Sage 300 UI Customization Wizard (Sage300CustomProject.vsix) will install this plug-in to Visual Studio and is compatible with Visual Studio Versions 2013 and 2015.

Selecting this file will invoke a dialog, prompting to be installed. Select **OK** to install the plugin.

Note: If the plug-in is already installed, it must be uninstalled first and this is accomplished in Visual Studio by selecting Tools/Extensions and Updates, locating the plug-in and selecting the Uninstall option.

Note: Install this plug-in even though it may not be used if an MVC project is not required for the customization.

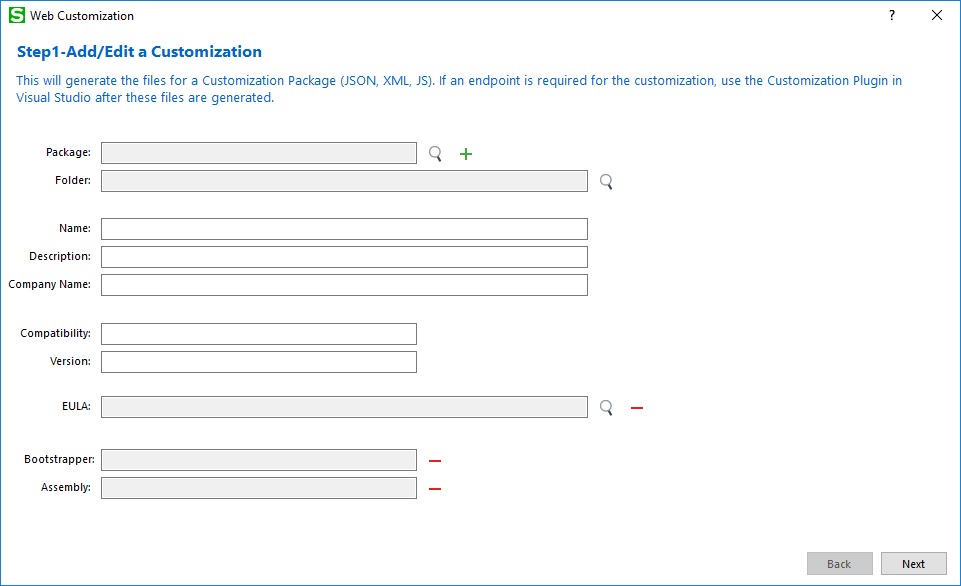
1. Creating the Customization
   1. Standalone Customization Wizard

Creates the JSON, XML and JavaScript files: (refer to *Sample1 - OE Order Entry Customization*).

Locate the standalone customization wizard (located in the SDK’s bin folder (bin\Sage.CA.SBS.ERP.Sage300.CustomizationWizard.exe).

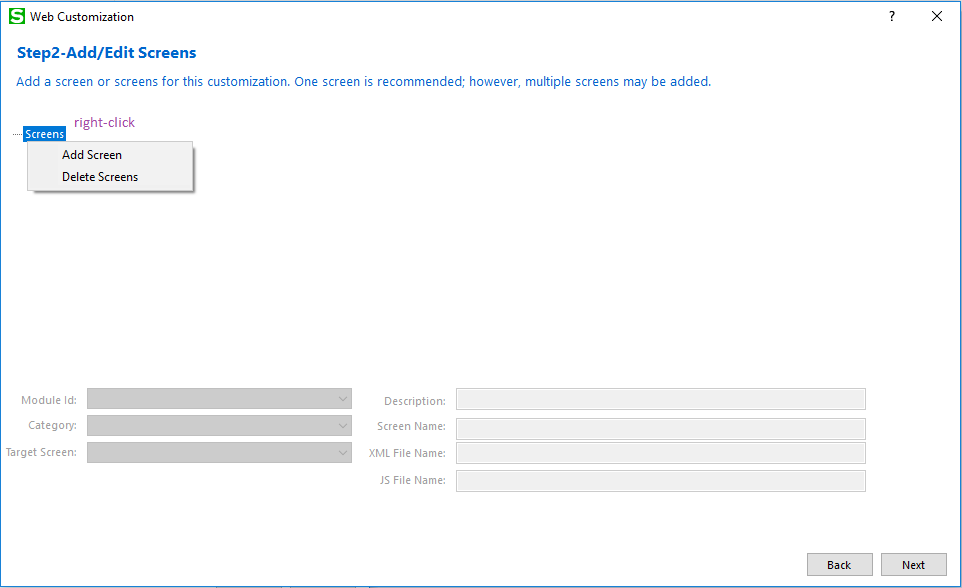
* + 1. Step 1 – Add/Edit a Customization

Allows a customization to be retrieved for modification or a new one created.

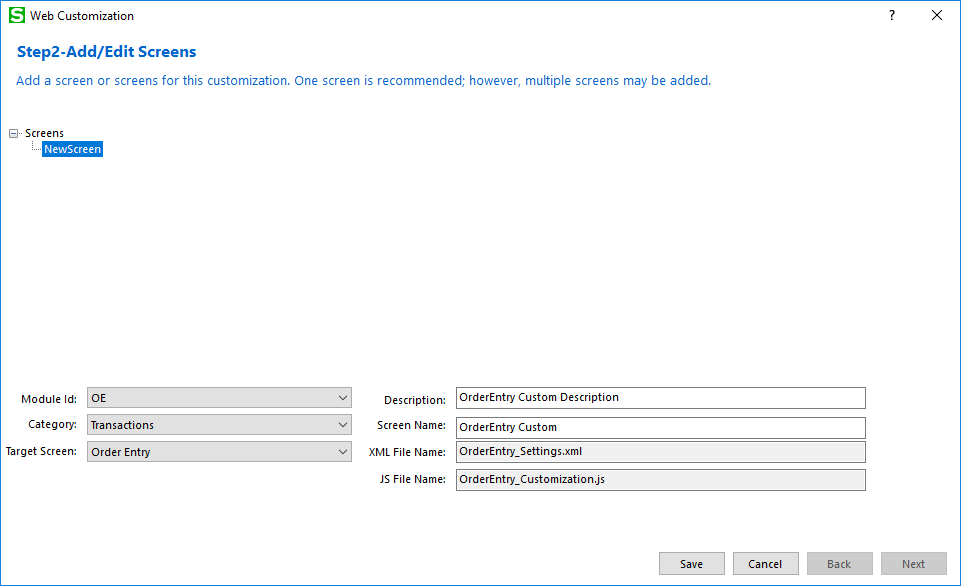


1. Populate the fields. Click on the magnifying glass next to Package to look up an existing customization or click the green plus button to generate a GUID for a new customization.
2. Enter the name, description, and company name for the customization. Ensure the compatibility and version are as desired. An EULA can also be specified at this stage.
3. If a bootstrapper and assembly are required, the developer should proceed to the Visual Studio plugin wizard upon completion of this wizard (for the creation of the MVC project where the plug-in wizard will update these fields). Modification of these two fields are disabled in this wizard.
4. Click **Next** to proceed.
   * 1. Step 2 – Add/Edit Screens

Allows a screen or screens to be added to or removed from the customization.

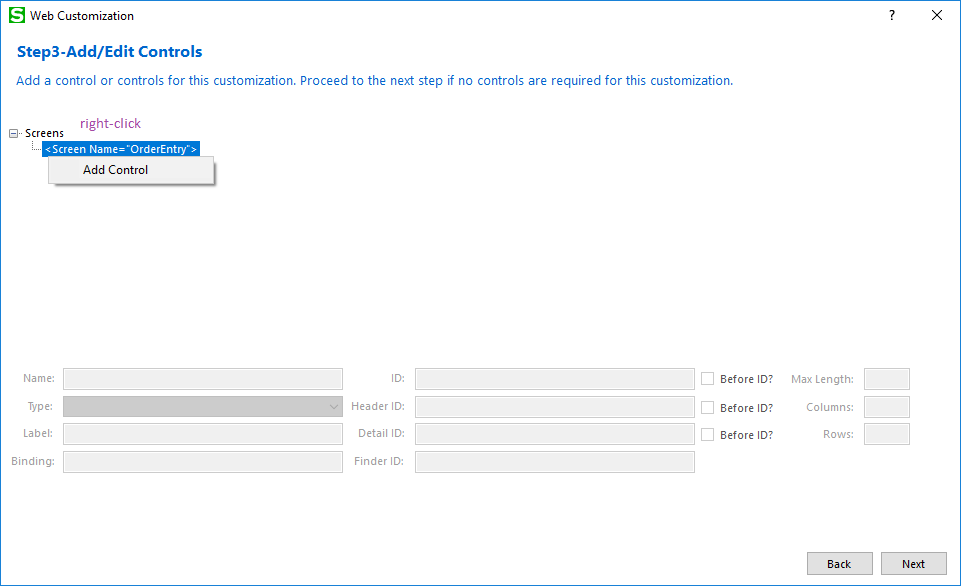


1. Right click the Screens node for options.

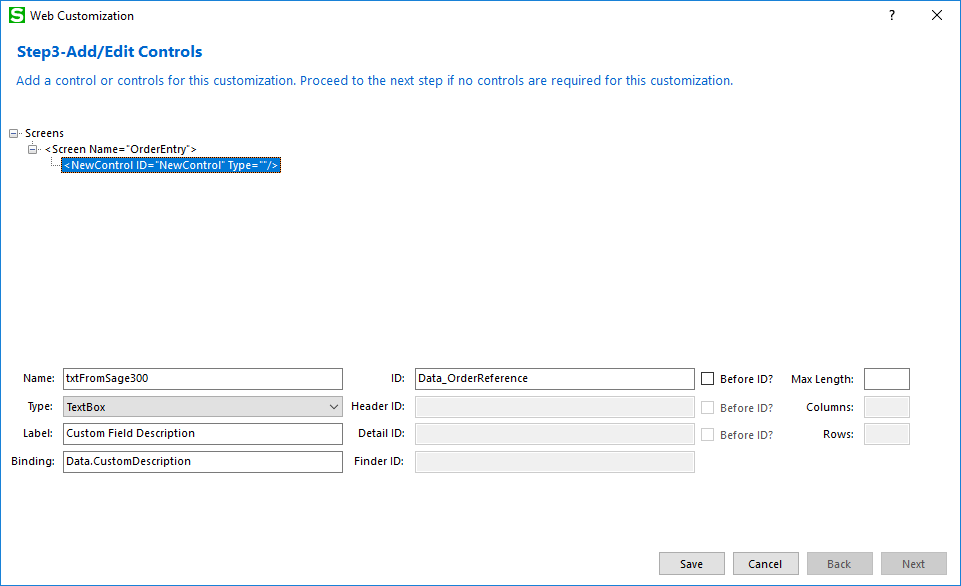


1. Select the module id, category and target screen as desired. The rest of the fields will autofill themselves. Only the Description and Screen Name fields are modifiable, if needed. The XML and JavaScript file names are not modifiable as the customization implementation has a naming convention that must be followed.
2. When finished, click **Save**. Repeat these steps to add another screen to the customization manifest if desired.
3. Press **Next** to proceed or **Back** to the previous step.
   * 1. Step 3 – Add/Edit Controls

Allows a control or controls to be added to or removed from the customization.



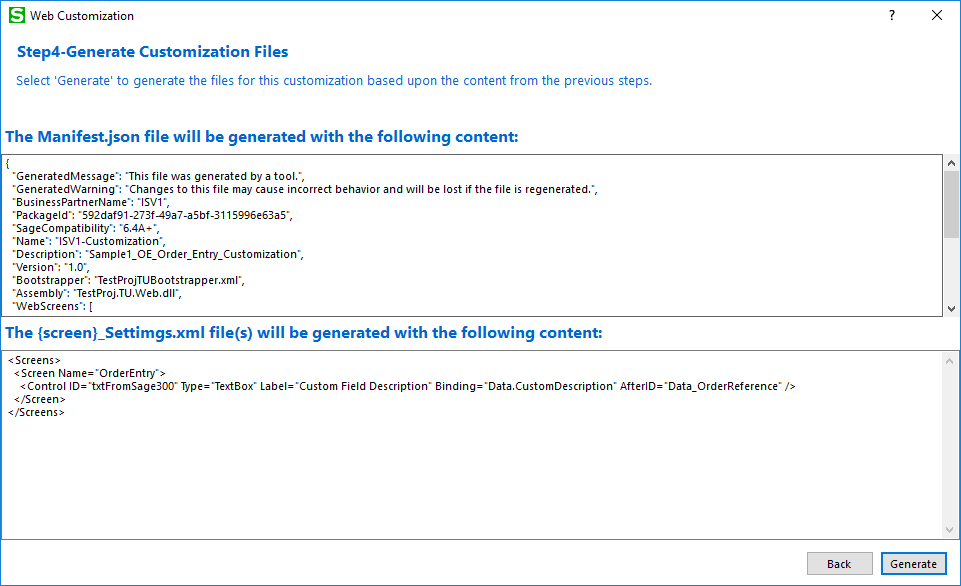
1. Right click the Screen or Control node for options.



1. Fill in the required fields. When finished, click **Save**. Repeat these steps to add another control if desired.
2. Press **Next** to proceed or **Back** to the previous Step.
   * 1. Step 4 – Generate Customization Files

Allows the manifest to be previewed before created or being modified.

1. If there are any corrections to be made, press **Back** to return to the previous step otherwise press **Generate** to generate the customization files.



Template files are generated in the Folder specified in Step 1:

* Control Settings file(s): *{screen name}\_*Settings.xml, such as OrderEntry\_Settings.xml
* JavaScript file(s): *{screen name}\_*Customization.js, such as OrderEntry\_Customization.js
* XML Schema file: ScreenConfig.xsd

**Note:** Do not change the names of the generated files. These determine which screen the customization will be applied to.

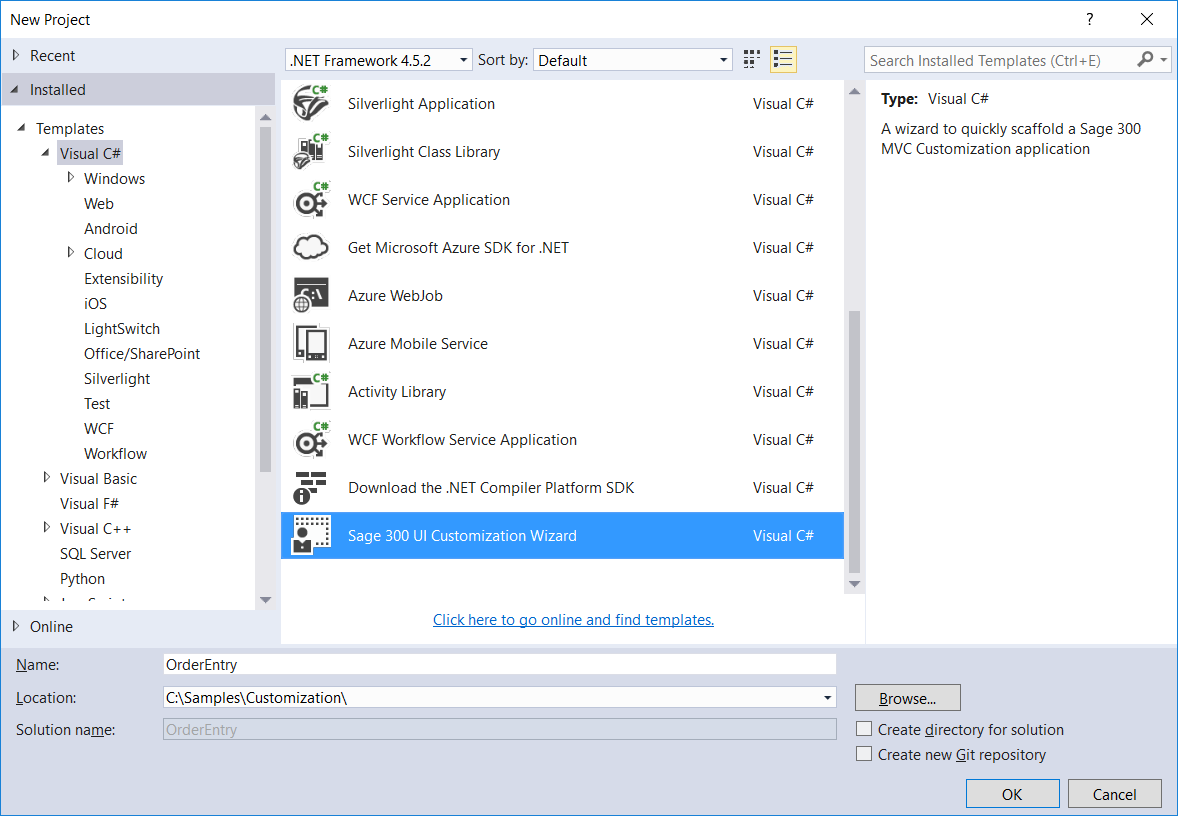
For detailed descriptions of all JSON and XML fields, refer to the XML File Specification document and the JSON File Specification document.

At this point, if a solution for endpoint customization is required, proceed to the Visual Studio plugin wizard.

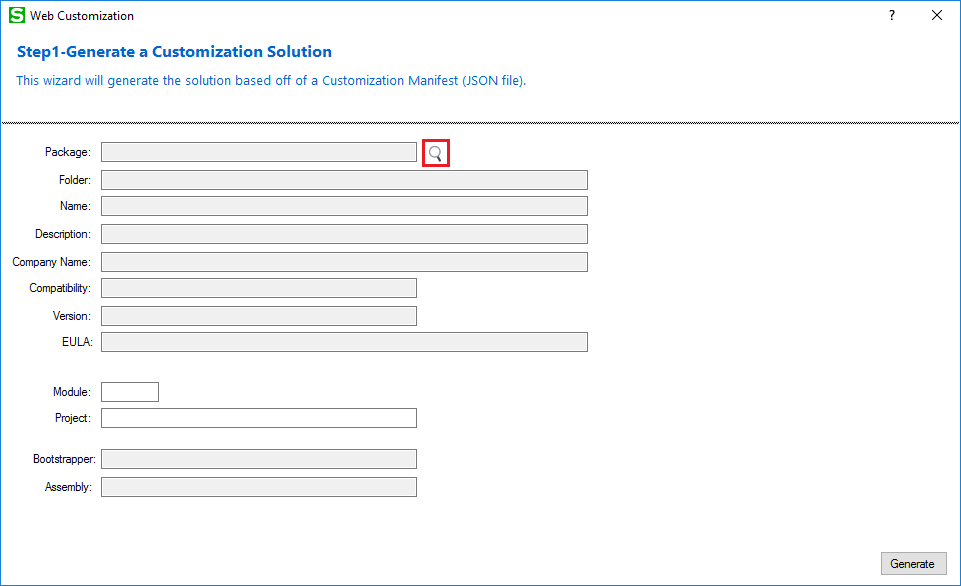
* 1. Plug-in Customization Wizard (optional)

Creates the customization project: (refer to *Sample1 - OE Order Entry Customization*).

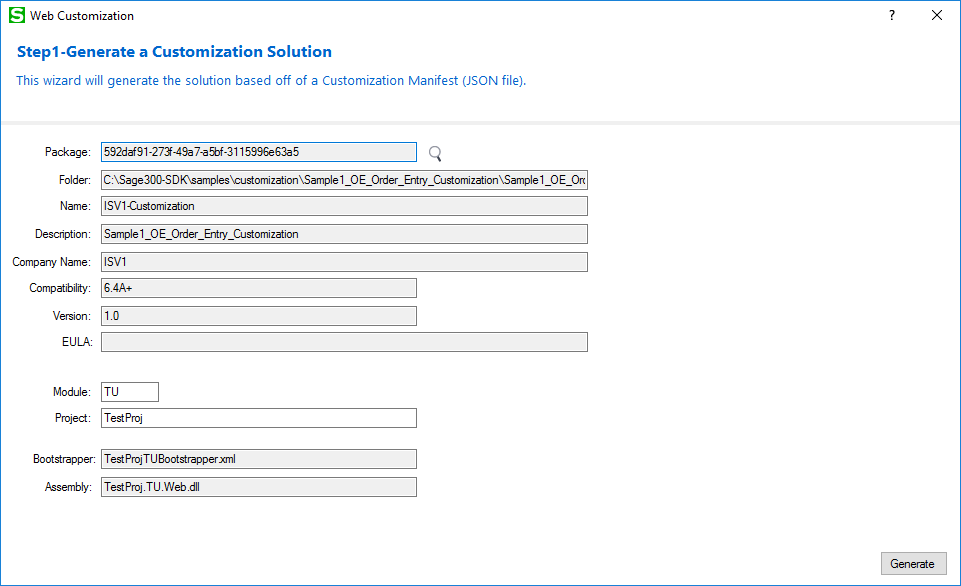
* + 1. Step 1 – Create a New Project



1. Open Visual Studio to create a new project,
2. In the new project dialog box, select Sage 300 UI Customization Wizard.
3. Enter the name and location, and then click OK.
   * 1. Step 2 – Generate a Customization Solution



1. Click on the magnifying glass next to Package to select the manifest JSON file (see the Manifest JSON File Specifications document for additional details).
2. The rest of the fields will populate according to the contents of the file:

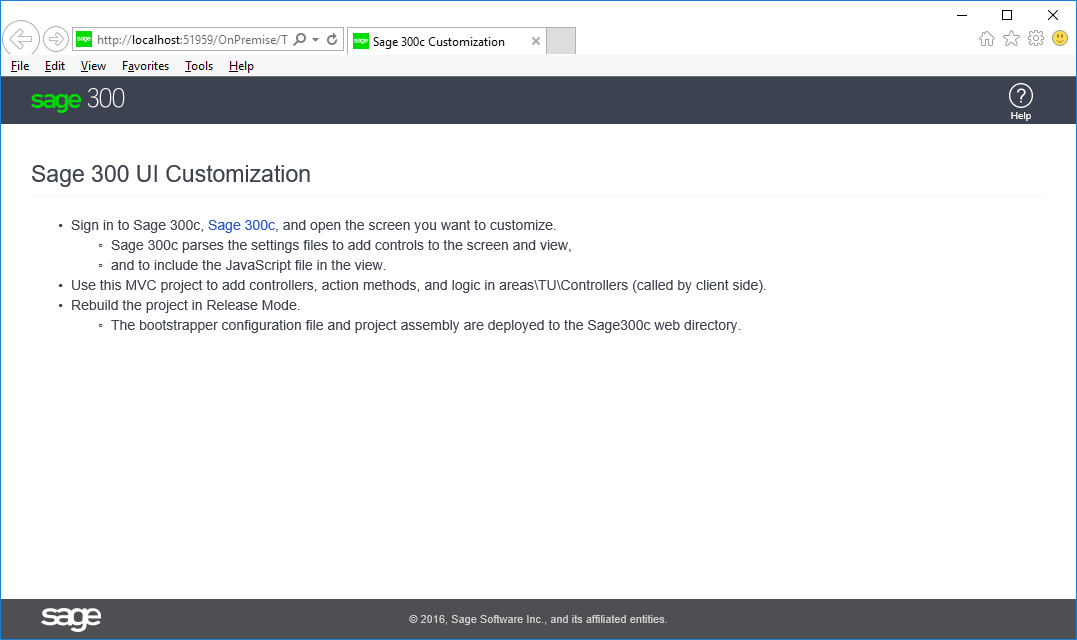


1. lick **Generate** to create the solution.
2. Select Nuget Package Manager and restore Nuget Packages
3. Compile the project and make sure it does not have any compilation errors.
   1. Debugging

Local debugging can be accomplished by copying the XML Settings file(s) and JavaScript file(s) to the shared data folder’s Customization folder of the local installation *{Sage300 installed shared data folder}\Customization folder*

To start debugging:

1. Open the above generated screen settings file OrderEntry\_Settings.xml and JavaScript file OrderEntry\_Customization.js in the code editor of VS 2013 / VS2015.
2. In the JS file, set a breakpoint at the initial entry point ISV1OrderEntryCustomizationUI.init()
3. Run the project with Internet Explorer.
4. When the default page appears, click the [Sage 300c application](http://localhost/Sage300) link to switch to the local Sage 300c web application.



1. Sign in to the Sage 300c web application and navigate to the O/E Order Entry screen. As this screen rendering html document includes the OrderEntry\_Customization.js, when the O/E Order Entry document screen is ready, it will hit the breakpoint.

For additional information, see the Customization Example Reference. The contents of that document mirror the contents of this one and may provide supplementary assistance.