Portal Customization End-to-End Scenario

Last Modified on 11/10/2022 7:38 am EST

V10.1

Overview

You can customize the Cora Orchestration portal based on customer needs. You use a Visual Studio template project that simplifies the customization process and helps you apply the required custom configuration.

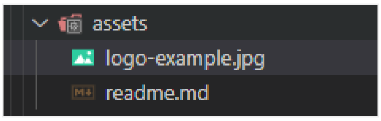
You can apply common custom code such as change logo and portal colors, add new pages and menu items. You can also apply more complex customization using React.

Common customization configurations

- Replace the logo
- Edit CSS styles
- Customize menu items
- Customize the grid and board display
- Customize the grid menu
- Change the behavior of the Close tab button
- Create a page using the FormViewer control
- Localize portal text
- Create a new page using React

After you have validated that development environment is up and running, you can start setting up the required customization configurations.

See below an example of how to implement three customization requests.

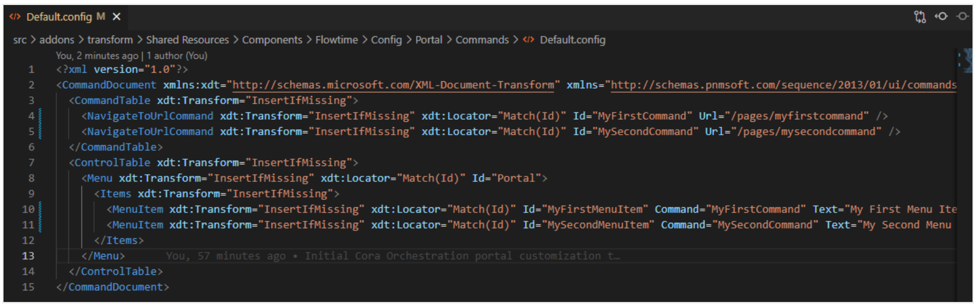
- Add custom menu item
- Change logo
  New logo location (example):
  

- Add a new page using the FormViewer control

Step 1: Apply the customization configuration in the template project

The first step is to open the template project with Visual Studio Code from the Template folder, and then:

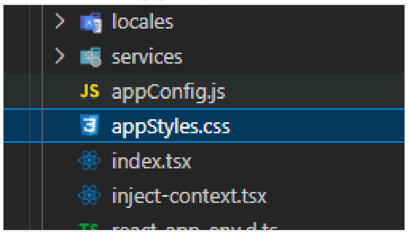
1. To add a custom menu item:
   a. Add an XML file with the new menu item in the **src\addons\transform** folder.
      Make sure that you keep the Shared Resources folder's hierarchy.


Default.config M
src > addons > transform > Shared Resources > Components > Flowtime > Config > Portal > Commands > Default.config
You, 2 minutes ago | 1 author (You)

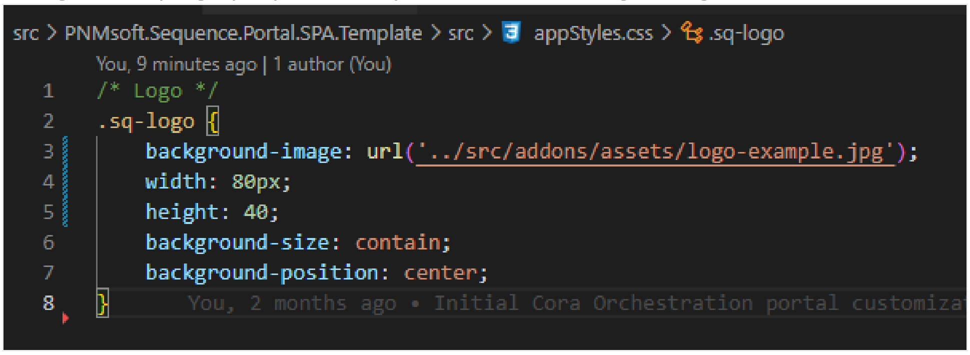
```
<?xml version="1.0"?>
<CommandDocument xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform" xmlns="http://schemas.pnmsoft.com/sequence/2013/01/ui/commands
  <CommandTable xdt:Transform="InsertIfMissing">
    <NavigateToUrlCommand xdt:Transform="InsertIfMissing" xdt:Locator="Match(Id)" Id="MyFirstCommand" Url="/pages/myfirstcommand" />
    <NavigateToUrlCommand xdt:Transform="InsertIfMissing" xdt:Locator="Match(Id)" Id="MySecondCommand" Url="/pages/mysecondcommand" />
  </CommandTable>
  <ControlTable xdt:Transform="InsertIfMissing">
    <Menu xdt:Transform="InsertIfMissing" xdt:Locator="Match(Id)" Id="Portal">
      <Items xdt:Transform="InsertIfMissing">
        <MenuItem xdt:Transform="InsertIfMissing" xdt:Locator="Match(Id)" Id="MyFirstMenuItem" Command="MyFirstCommand" Text="My First Menu Ite
        <MenuItem xdt:Transform="InsertIfMissing" xdt:Locator="Match(Id)" Id="MySecondMenuItem" Command="MySecondCommand" Text="My Second Menu
      </Items>
    </Menu>       You, 57 minutes ago • Initial Cora Orchestration portal customization t…
  </ControlTable>
</CommandDocument>
```

Note that this is a transformation file. Its content will be added to the original file without replacing it.

2. To change the logo:
   a. Locate the **appStyles.css** file in the **src** folder.

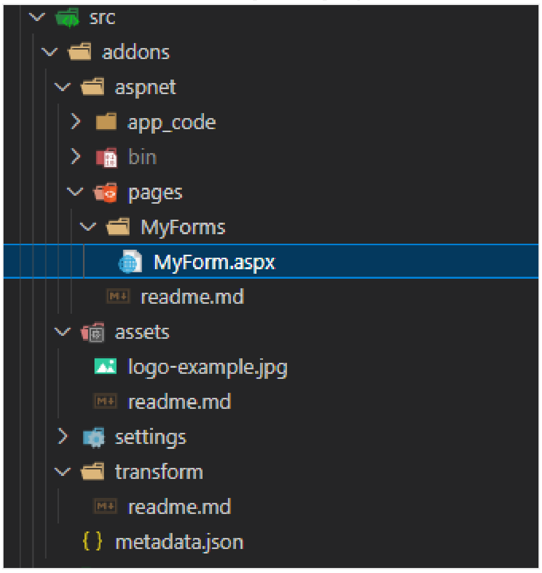


   b. Change the sq-logo properties to point to the new logo image location.


src > PNMsoft.Sequence.Portal.SPA.Template > src > appStyles.css > .sq-logo
You, 9 minutes ago | 1 author (You)

```
/* Logo */
.sq-logo {
    background-image: url('../src/addons/assets/logo-example.jpg');
    width: 80px;
    height: 40;
    background-size: contain;
    background-position: center;
}     You, 2 months ago • Initial Cora Orchestration portal customiza
```

3. To add a new page using the FormViewer control, add the ASPX page with the FormViewer control to the **src\addons\aspnet\pages** folder.



TIP

Learn more about transformation APIs:

- Web.config Transformation Syntax for Web Project Deployment Using Visual Studio | Microsoft Learn
- Web.config Transformation Syntax for Web Application Project Deployment | Microsoft Learn
- Xdt transform samples · projectkudu/kudu Wiki (github.com)

This step concludes the customization configuration, now you need to deploy the changes to IIS site.

Step 2: Deploy the customized site application on IIS

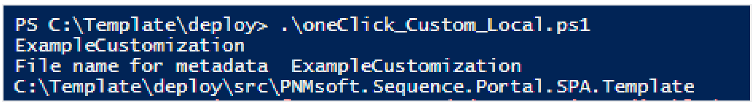
Run the **oneClick_Custom_Local.ps1** PowerShell function to:

- Apply your customization to a saved snapshot of the original portal version downloaded from the repository.
- Replace the **CoraSeQuenceFlowtime 1** site in IIS with the customized one.
- Prepare the custom snapshot.

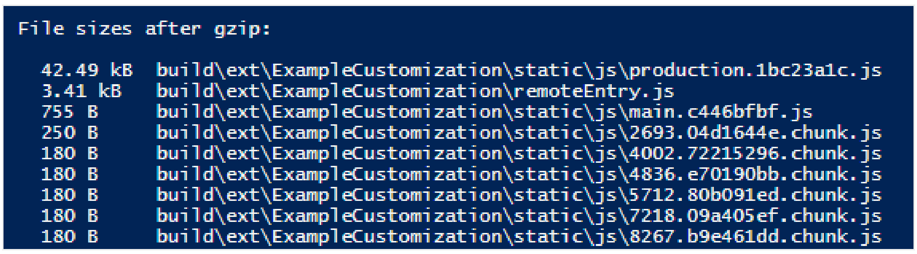
NOTE
The **oneClick_Custom_Local.ps1** function takes time to run as it works with snapshots.

TIP
You can maintain your template project in a source control tool. Your pull request then can trigger a pipeline that applies the customization and prepares a production-ready artifact for you.
Download a YAML of the pipeline.

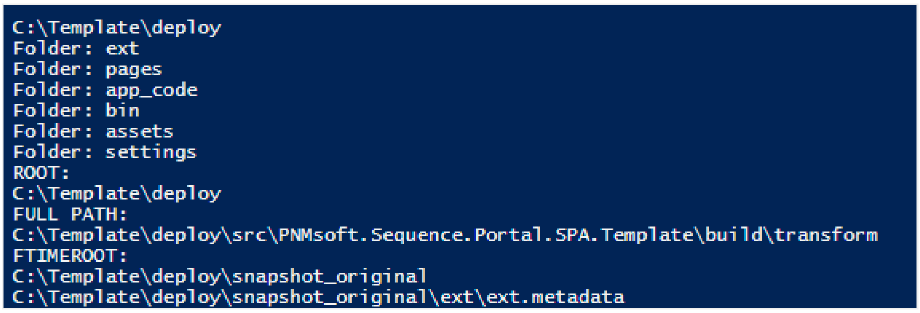
PowerShell function screens



```
PS C:\Template\deploy> .\oneClick_Custom_Local.ps1
ExampleCustomization
File name for metadata  ExampleCustomization
C:\Template\deploy\src\PNMsoft.Sequence.Portal.SPA.Template
```



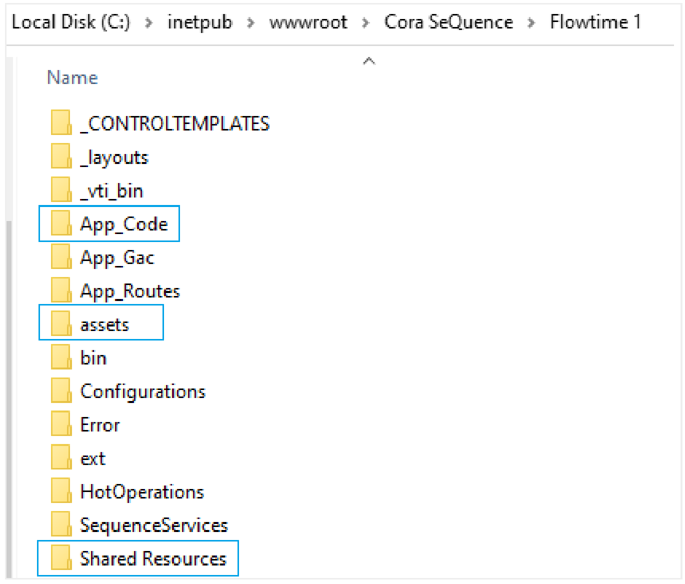
```
File sizes after gzip:

  42.49 kB  build\ext\ExampleCustomization\static\js\production.1bc23a1c.js
  3.41 kB   build\ext\ExampleCustomization\remoteEntry.js
  755 B     build\ext\ExampleCustomization\static\js\main.c446bfbf.js
  250 B     build\ext\ExampleCustomization\static\js\2693.04d1644e.chunk.js
  180 B     build\ext\ExampleCustomization\static\js\4002.72215296.chunk.js
  180 B     build\ext\ExampleCustomization\static\js\4836.e70190bb.chunk.js
  180 B     build\ext\ExampleCustomization\static\js\5712.80b091ed.chunk.js
  180 B     build\ext\ExampleCustomization\static\js\7218.09a405ef.chunk.js
  180 B     build\ext\ExampleCustomization\static\js\8267.b9e461dd.chunk.js
```



```
C:\Template\deploy
Folder: ext
Folder: pages
Folder: app_code
Folder: bin
Folder: assets
Folder: settings
ROOT:
C:\Template\deploy
FULL PATH:
C:\Template\deploy\src\PNMsoft.Sequence.Portal.SPA.Template\build\transform
FTIMEROOT:
C:\Template\deploy\snapshot_original
C:\Template\deploy\snapshot_original\ext\ext.metadata
```

Step 3: Validate the deployed changes

After the PowerShell function resumes running, check the IIS site structure.


Local Disk (C:) › inetpub › wwwroot › Cora SeQuence › Flowtime 1

Name

- _CONTROLTEMPLATES
- _layouts
- _vti_bin
- App_Code
- App_Gac
- App_Routes
- assets
- bin
- Configurations
- Error
- ext
- HotOperations
- SequenceServices
- Shared Resources

As expected for the scenario described here, the App_Code, assets, and the updated menu items XML file are now in the IIS site.

Refresh the browser page, and view your changes.

IMPORTANT
Make sure that you apply your custom code with the project template **only**. Manual changes made on the IIS site files are not saved and will be overwritten by the PowerShell functions.

TIP
Development sometimes is a repetitive *Do and test* process. To save time, and expedite the development process, you may choose to update the locally-running IIS with the changes you made only. In this case, run the **fast_Deploy_Local.ps1** function and refresh the page in the browser. The **fast_Deploy_Local.ps1** function does not create snapshots of your customization and can be used when you only want to quickly check the results of your customization during development.

You must run the **oneClick_Custom_Local.ps1** function once, as it builds the structure for the customization process. Afterwards, you can run the **fast_Deploy_Local.ps1** to save time when validating your changes.

Customize another site

The custom snapshot can be used as the base for another customization. In this case, edit the **2_download_Original_File_Local.ps1** function and specify the URL of the already customized portal.