



# ONC SITE DROPDOWN MENU COMPONENT

## Design Document

### [Abstract](#)

Design document for the page header component used in ONC SPO sites for all the modern style pages, to be consistent with the current site.

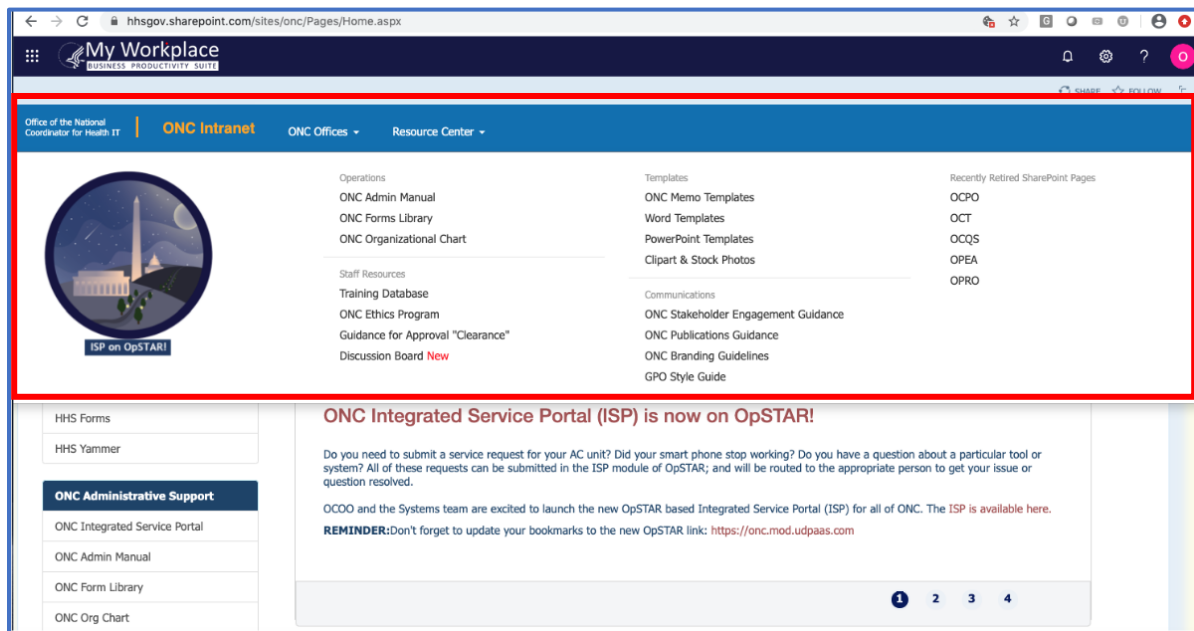
ONC Site Administrator  
farrah.darbouze@hhs.gov

## TABLE OF CONTENTS

.....	0
Goal.....	2
Design.....	3
Implementation Plan .....	4
Tools and Software Packages.....	5
Build, Test and Deployment.....	6
Key Code Logic and APIs .....	10
Maintenance and Change Control .....	12

## GOAL

In ONC SPO site, we have a dropdown menu on top of all the pages (as highlighted below):



This mega dropdown menu is implemented inside the site master page.

Recently when we are going through our site redesign, we want to use the more appealing modern style pages made available by Microsoft. However, master page or layout pages are currently no available in the modern style sites.

Our goal is to implement a new header dropdown menu component that utilizes the new SharePoint Framework to display the current ONC dropdown menu on all the modern style pages.

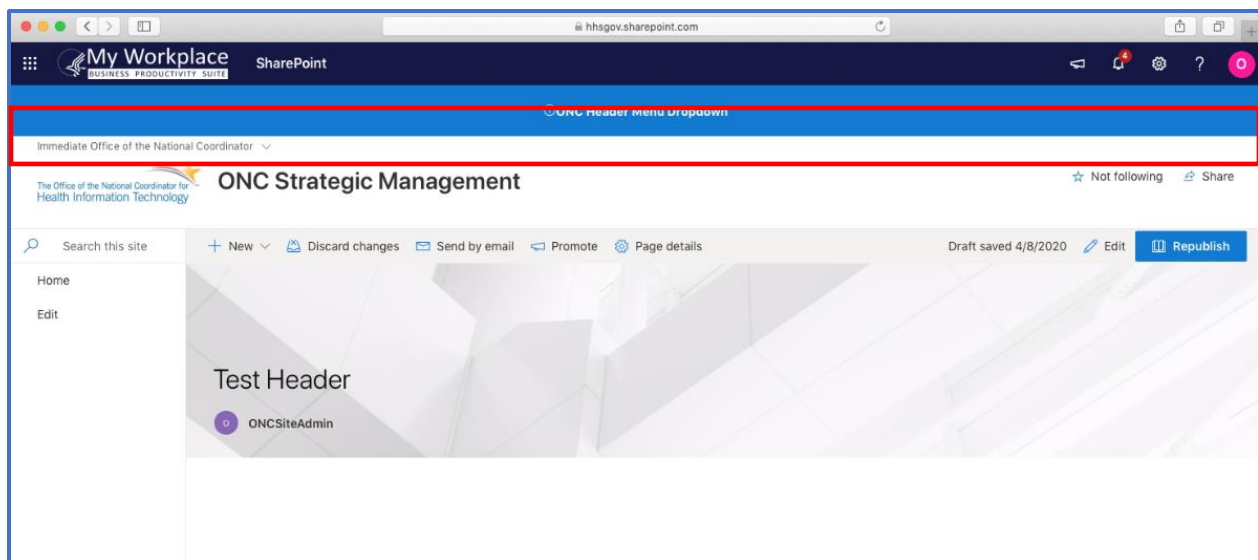
## DESIGN

SharePoint Framework Extensions allows user to use the familiar SharePoint Framework tools and libraries for client-side development to extend the SharePoint the modern pages and document libraries. Specifically, SharePoint Framework Extensions type of Application Customizers, allows user to customize the header (PlaceholderName .Top) and footer (PlaceholderName .Bottom) of all modern pages in the site with the extensions application applied to.

In this design, we choose to use Application Customizers load:

1. custom scripts: handle dropdown menu behaviors;
2. third-party JavaScript libraries: include jQuery and bootstrap;
3. styles: include bootstrap, fontawesome, and inline custom CSS;
4. and, custom HTML elements: display the dropdown menu contents.

And render the content in the placeholder “PlaceholderName .Top”:



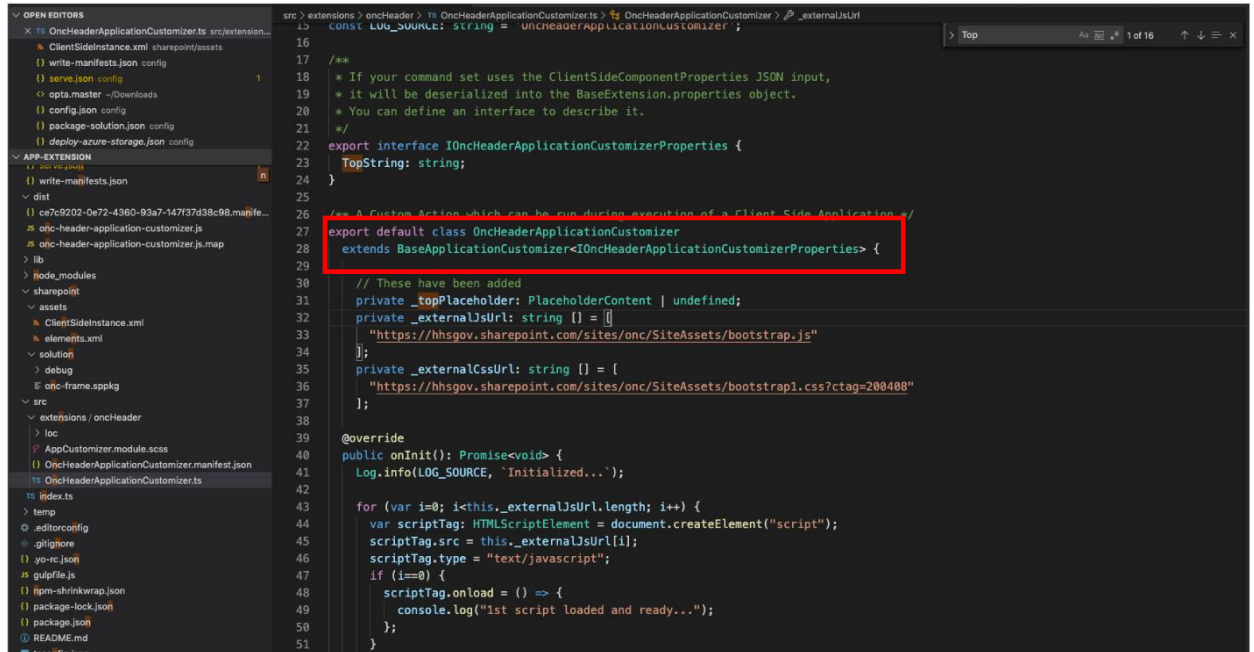
Lastly, Application Customizer extensions are supported with Site, Web, and List scopes. And we are planning to use it on the site level by associating the application customizer to the UserCustomAction collection in Site object level.

REFERENCE: <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/extensions/get-started/using-page-placeholder-with-extensions>

## IMPLEMENTATION PLAN

The implementation plan is:

1. Extract and assemble all necessary code, library, css and HTMLs from the current site for the dropdown menu so that we can have an independent component ready to be injected to the new site;
2. Create the SharePoint Framework Extension project for Application Customizer;
3. Extends the BaseApplicationCustomizer class to handle the Header placeholder rendering, and injects all the artifacts prepared in step #1;



```
16
17
18 /**
19  * If your command set uses the ClientSideComponentProperties JSON input,
20  * it will be deserialized into the BaseExtension.properties object.
21  * You can define an interface to describe it.
22  */
23 export interface IOncHeaderApplicationCustomizerProperties {
24   TopString: string;
25 }
26
27 /** A Custom Action which can be run during execution of a Client Side Application */
28 export default class OncHeaderApplicationCustomizer
29   extends BaseApplicationCustomizer<IOncHeaderApplicationCustomizerProperties> {
30
31   // These have been added
32   private _topPlaceholder: PlaceholderContent | undefined;
33   private _externalJsUrl: string[] = [
34     "https://hhsgov.sharepoint.com/sites/onc/SiteAssets/bootstrap.js"
35   ];
36   private _externalCssUrl: string[] = [
37     "https://hhsgov.sharepoint.com/sites/onc/SiteAssets/bootstrap1.css?ctag=200408"
38   ];
39
40   @override
41   public onInit(): Promise<void> {
42     Log.info(LOG_SOURCE, 'Initialized...');
43
44     for (var i=0; i<this._externalJsUrl.length; i++) {
45       var scriptTag: HTMLScriptElement = document.createElement("script");
46       scriptTag.src = this._externalJsUrl[i];
47       scriptTag.type = "text/javascript";
48       if (i==0) {
49         scriptTag.onload = () => {
50           console.log("1st script loaded and ready...");
51         };
52       }
53     }
54   }
55 }
```

4. Test the application;
5. Deploy the application.

NOTE: Per the Microsoft document online, to deploy the client-side application needs to be deployed in an application catalog site which we do not have privilege to operate on. That's where we need support from the tenant administrator.

## TOOLS AND SOFTWARE PACKAGES

As SharePoint recommended, for our development tools, we are using:

1. NodeJS for application coding;
2. Yeoman for project/framework generating
3. Visual Code for IDE; and,
4. Gulp for build

To reduce development efforts to the minimum, we want to reuse as much code from the current dropdown menu as possible. Some of the third-party libraries/styles used there include:

1. jQuery,
2. bootstrap, and
3. fontawesome.

## BUILD, TEST AND DEPLOYMENT

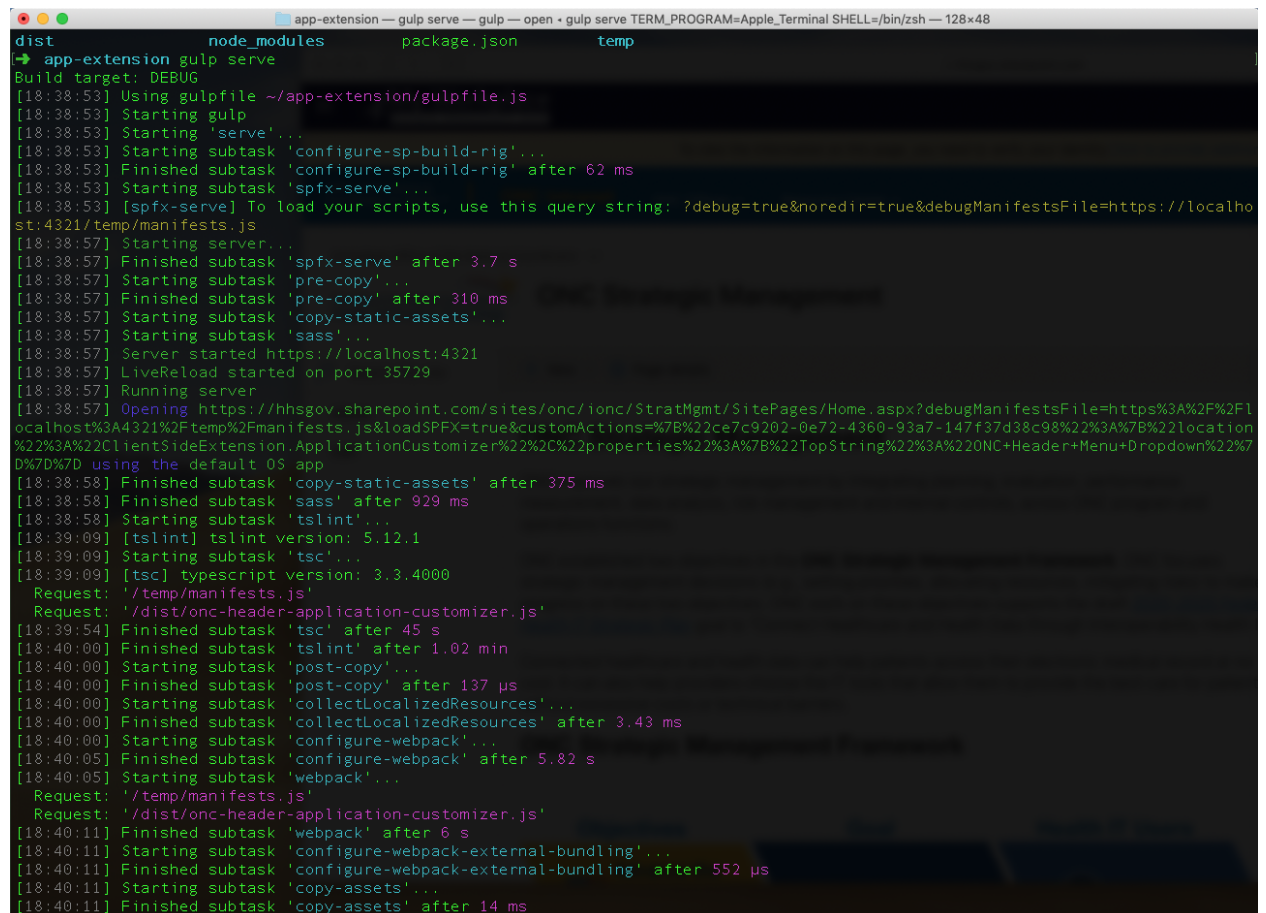
Once the SharePoint project is setup with Yeoman generator, we should be able to do:

1. prepare the source and development tools  
please refer to <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/set-up-your-development-environment> to get the required tools and software package set up.  
Then: try to clone the repo, install the NodeJS library, gulp and then test build:

```
git clone the repo
npm i
npm i -g gulp
gulp
```

2. build and test with command:

*gulp serve*



```
dist      node_modules      package.json      temp
→ app-extension gulp serve
Build target: DEBUG
[18:38:53] Using gulpfile ~/app-extension/gulpfile.js
[18:38:53] Starting gulp
[18:38:53] Starting 'serve'...
[18:38:53] Starting subtask 'configure-sp-build-rig'...
[18:38:53] Finished subtask 'configure-sp-build-rig' after 62 ms
[18:38:53] Starting subtask 'spfx-serve'...
[18:38:53] [spfx-serve] To load your scripts, use this query string: ?debug=true&noredir=true&debugManifestsFile=https://localhost:4321/temp/manifests.js
[18:38:57] Starting server...
[18:38:57] Finished subtask 'spfx-serve' after 3.7 s
[18:38:57] Starting subtask 'pre-copy'...
[18:38:57] Finished subtask 'pre-copy' after 310 ms
[18:38:57] Starting subtask 'copy-static-assets'...
[18:38:57] Starting subtask 'sass'...
[18:38:57] Server started https://localhost:4321
[18:38:57] LiveReload started on port 35729
[18:38:57] Running server
[18:38:57] Opening https://hhsgov.sharepoint.com/sites/onc/ionc/StratMgmt/SitePages/Home.aspx?debugManifestsFile=https%3A%2F%2Flocalhost%3A4321%2Ftemp%2Fmanifests.js&loadSPFX=true&customActions=%7B%22ce7c9202-0e72-4360-93a7-147f37d38c98%22%3A%7B%22location%22%3A%22ClientSideExtension.ApplicationCustomizer%22%2C%22properties%22%3A%7B%22TopString%22%3A%220NC+Header+Menu+Dropdown%22%7D%7D%7D using the default OS app
[18:38:58] Finished subtask 'copy-static-assets' after 375 ms
[18:38:58] Finished subtask 'sass' after 929 ms
[18:38:58] Starting subtask 'tslint'...
[18:39:09] [tslint] tslint version: 5.12.1
[18:39:09] Starting subtask 'tsc'...
[18:39:09] [tsc] typescript version: 3.3.4000
Request: '/temp/manifests.js'
Request: '/dist/onc-header-application-customizer.js'
[18:39:54] Finished subtask 'tsc' after 45 s
[18:40:00] Finished subtask 'tslint' after 1.02 min
[18:40:00] Starting subtask 'post-copy'...
[18:40:00] Finished subtask 'post-copy' after 137 µs
[18:40:00] Starting subtask 'collectLocalizedResources'...
[18:40:00] Finished subtask 'collectLocalizedResources' after 3.43 ms
[18:40:00] Starting subtask 'configure-webpack'...
[18:40:05] Finished subtask 'configure-webpack' after 5.82 s
[18:40:05] Starting subtask 'webpack'...
Request: '/temp/manifests.js'
Request: '/dist/onc-header-application-customizer.js'
[18:40:11] Finished subtask 'webpack' after 6 s
[18:40:11] Starting subtask 'configure-webpack-external-bundling'...
[18:40:11] Finished subtask 'configure-webpack-external-bundling' after 552 µs
[18:40:11] Starting subtask 'copy-assets'...
[18:40:11] Finished subtask 'copy-assets' after 14 ms
```

Once the browser is loaded, you can see the menu load with the page:

My Workplace  
BUSINESS PRODUCTIVITY SUITE

To view the information on this page, you need to verify your identity. [Click to provide additional credentials.](#)

Office of the National Coordinator for Health IT | **ONC Intranet** | ONC Offices | Resource Center

Immediate Office of the National Coordinator

The Office of the National Coordinator for Health Information Technology **ONC Strategic Management** ☆ Not following Share

Search this site + New Page details Published 4/14/2020 Edit

## ONC Strategic Management

ONC improves our strategic management by integrating planning, evaluation, performance measurement, data analysis, risk management and internal controls, across ONC program and operations functions.

ONC established two objectives in the **ONC Strategic Management Framework**. ONC focuses strategic management decisions (e.g., setting priorities, allocating resources, mitigating risks) to make progress on these two objectives. ONC work on these objectives supports the draft [2020-2025 Federal Health IT Strategic Plan](#) goal to "Connect Healthcare and Health Data through Interoperability Health IT."

Connected healthcare and health data can help patients access their electronic medical record at no cost. It can also help providers choose the IT tools that allow them to provide the best care for patients, without excessive costs or technical barriers.

### ONC Strategic Management Framework

Objectives Goal Health IT Users

Advance the

#### Links

Strategic Management Advisory Committee (Restricted Access)

#### Resources

See all

+ New ... All Documents

Name

NC Priorities 2020.pptx

ONC Objectives.pptx

ONC Strategic Mgmt Cheat She...

ONC Strategic Mgmt Framewor...

Click on the dropdown, you can check things working –

My Workplace  
BUSINESS PRODUCTIVITY SUITE

To view the information on this page, you need to verify your identity. [Click to provide additional credentials.](#)

Office of the National Coordinator for Health IT | **ONC Intranet** | **ONC Offices** | Resource Center

### IONC

Providing leadership to HHS, ONC, and health IT industry through strategic planning, management, communications, and stakeholder engagements. Divisions within the IO include:

- Immediate Office of the National Coordinator
- Public Affairs and Communications
- Chief Scientist
- Clinical Division

### OCOO

Providing the management infrastructure and administrative backbone for ONC to function at its highest levels. Divisions within OCOO include:

- Enterprise Risk Management
- Financial and Human Capital Management
- Operational Services
- Procurement and Grants
- Systems and Data

### OPOL

Accelerating progress to a nationwide interoperable health IT infrastructure through strategy, policy development, and coordination. Divisions within OPOL include:

- Immediate Office
- Strategic, Planning, and Coordination
- Regulatory Affairs
- Interoperability

### OTECH

Accelerating progress to a nationwide interoperable health IT infrastructure through standards coordination and development, sponsoring and encouraging science, research, and pilot projects, and supporting industry-wide health IT testing and transparency. Divisions within OTECH include:

- Immediate Office
- Technical Strategy and Analysis
- Standards
- Certification and Testing

ONC established two objectives in the **ONC Strategic Management Framework**. ONC focuses strategic management decisions (e.g., setting priorities, allocating resources, mitigating risks) to make progress on these two objectives. ONC work on these objectives supports the draft [2020-2025 Federal Health IT Strategic Plan](#) goal to "Connect Healthcare and Health Data through Interoperability Health IT."

Connected healthcare and health data can help patients access their electronic medical record at no cost. It can also help providers choose the IT tools that allow them to provide the best care for patients, without excessive costs or technical barriers.

### ONC Strategic Management Framework

Objectives Goal Health IT Users

Advance the

#### Resources

See all

+ New ... All Documents

Name

NC Priorities 2020.pptx

ONC Objectives.pptx

ONC Strategic Mgmt Cheat She...

ONC Strategic Mgmt Framewor...



3. to build a deployable package:

```
gulp bundle --ship  
gulp package-solution --ship
```

4. Generated directories after build:

*lib/\** - intermediate-stage common js build artifacts  
*dist/\** - the bundled script, along with other resources  
*deploy/\** - all resources which should be uploaded to a CDN.

For SharePoint deployment, you can use `SharePoint/solution/onc-frame.sppkg`.

5. to deploy the package:

Previously, all add-ins and SharePoint Framework solutions had to be managed centrally in the tenant app catalog. While tenant administrators could delegate the access to other people in the organization, a deployed package was visible on all site collections. SharePoint offered no supported way of deploying add-ins and SharePoint Framework solutions only to specific sites.

With the introduction of site collection app catalogs, tenant administrators can enable app catalog on the specific sites. Once enabled, site collection administrators can deploy SharePoint add-ins and SharePoint Framework solutions that will be available only in that particular site collection.

The following schema illustrates using site collection app catalogs:

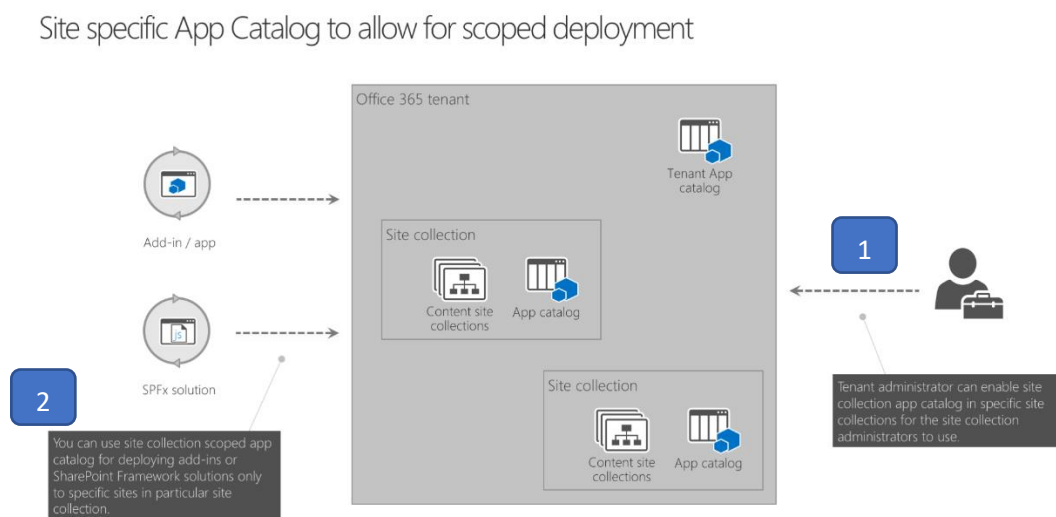


Diagram illustrating the concept of site collection app catalog

- a. Tenant administrator enables the site collection app catalogs for our site collection
- b. ONC developer upload deployable client-side solution package (SharePoint/solution/onc-frame.sppkg) to the site collection app catalog library.
- c. Add the customer solution to required ONC office sites and verify the solution.

## KEY CODE LOGIC AND APIS

This code does not use any APIs.

This code loads library and contents from the ONC site and renders it to the page to generate the dropdown menu. Contents and libraries loaded from the ONC site are:

The open source libraries such as jquery, bootstrap and fontawesome, they need to be downloaded and upload to the ONC site manually. The dropdown menu contents are used for the ONC staff to change the content of the menu without redeploying the component.

Dependencies	Purpose	File Name
1. jQuery	Open source JS library for HTML page, used in the current dropdown menu	jquery.min.js
2. bootstrap	Open source JS/CSS library for HTML page, used in the current dropdown menu	bootstrap.js, bootstrap1.css
3. fontawesome	Open source CSS library for HTML page fonts and icons, used in the current dropdown menu	all.css
4. dropdown menu contents.	dynamic Text/HTML contents, extracted from the current dropdown menu	header-dropdown-menu.txt

All these are located on the ONC site in the SharePoint library -

<https://hhs.gov.sharepoint.com/sites/onc/SiteAssets>

The screenshot shows the SharePoint 'Site Assets' library interface. The left sidebar contains navigation links: Home, Notebook, Documents, Recent, Site Pages, Site Contents, and Immediate Office of the... The main area displays a table of files with columns: Name, Modified, and Modified By. A blue box highlights the following files:

Name	Modified	Modified By
header-dropdown-menu.txt	A few seconds ago	ONCSiteAdmin
jquery.min.js	May 2	ONCSiteAdmin
bootstrap.css	May 1	Lazzaro, Victor (OS/ONC)
bootstrap1.css	May 10, 2019	ONCSiteAdmin
bootstrap.js	February 20, 2018	Lazzaro, Victor (OS/ONC)

All the above dependencies are dynamically loaded. The logic of loading these is in the code located at: `src/extensions/oncHeader/OncHeaderApplicationCustomizer.ts`.

Look into the code and check lines 125-142 shown as below. Those code perform dynamic loading of the library and contents:

```
125         function loadAllObjects() {
126             console.log('jquery loading completed....');
127             loadScript("https://hhsgov.sharepoint.com/sites/onc/SiteAssets/bootstrap.js", function(){console.log("bootstrap.js ready")});
128             locaCss("https://hhsgov.sharepoint.com/sites/onc/SiteAssets/bootstrap1.css", function(){console.log("bootstrap.css ready")});
129             locaCss("https://use.fontawesome.com/releases/v5.5.0/css/all.css", function(){console.log("fontawesome.css ready")});
130             jQuery(document).ready(function(){
131                 jQuery.get('https://hhsgov.sharepoint.com/sites/onc/SiteAssets/header-dropdown-menu.txt', function(tdata) {
132                     $('#oncmenuubar').html(tdata);
133                 });
134                 $(".dropdown").hover(
135                     function() { $('#.dropdown-menu', this).stop().fadeIn("fast");
136                     },
137                     function() { $('#.dropdown-menu', this).stop().fadeOut("fast");
138                     });
139             });
140         }
141         loadScript("https://hhsgov.sharepoint.com/sites/onc/SiteAssets/jquery.min.js", loadAllObjects);
142         console.log('jquery loading started....');
```

After libraries loaded. Line #132 render the dropdown menu dynamically into the top bar placeholder.

## MAINTENANCE AND CHANGE CONTROL

The contents header dropdown menu component is rather static. We'll have some of the content pulling from our current site. This allows us to reduce the needs of updating the component while maintain control of the displayed information in the meantime. Use other words, once test and deployed, the component should not require much maintenance.

Nevertheless, there can be requirements for major update of the component. In that case, we'll follow the current build, test and deployment procedure to redeploy the component.