

Oracle CPQ JET CSS: Branding and Styling

REVISION HISTORY:

Initial Document published	10/15/2018
Updated for 19B Library upgrade	8/15/2019

APPLIES TO:

Oracle CPQ Cloud Update 19B and later

PURPOSE:

The following information will aid existing CPQ Cloud customers in customizing the styling of the new JET UI's. This article covers Best Practices and basic tips for working with CSS and JET. JET library update in 19B changed some classes from the prior releases.

DETAILS:

What is the JET UI?

Oracle JavaScript Extension Toolkit (JET) is a modular open source toolkit based on modern JavaScript, CSS3 and HTML5 design and development principles. It's a collection of open source JavaScript libraries along with a set of Oracle contributed JavaScript libraries that make it as simple and efficient as possible to build and extend cloud-based, client-side applications. Oracle CPQ Cloud Release 18C introduces the JET Transaction UI - a modernized and fully responsive user experience for viewing and editing quotes and sales orders on any device and in any language. CPQ's JET Transaction UI supports virtually every feature available in the Legacy Commerce UI combined with full extensibility and improved usability. Upcoming releases will support additional CPQ modules such as Configuration and Homepage.

How do I customize the way JET UI's look?

If you are a new customer or have not yet implemented any customizations to styling your CPQ site, this document will help you understand how to add customizations that will apply to your JET Transaction UI for CPQ Cloud Commerce.

If you are an existing customer with styling customizations implemented on your legacy Commerce UIs, your custom UI styling will need to be recreated and applied to the JET Transaction UI using the new JET Stylesheet.

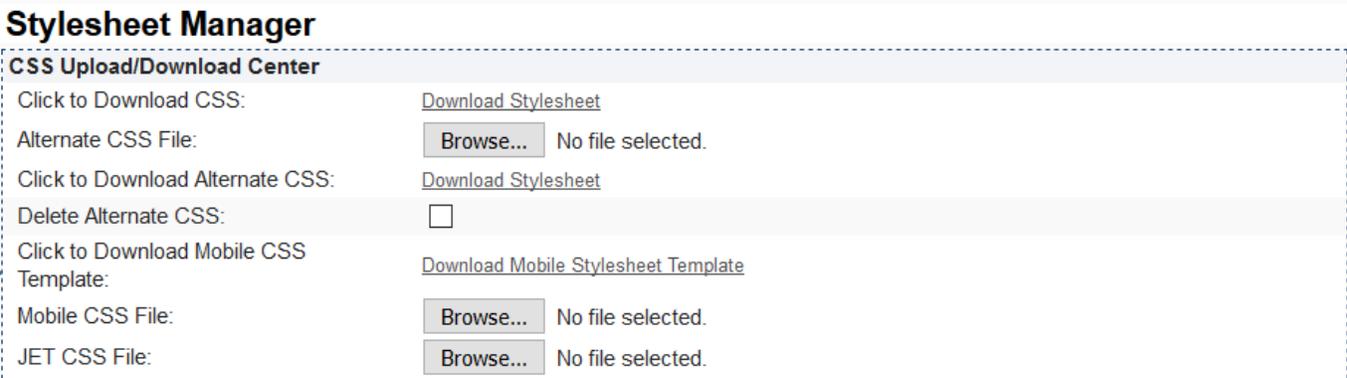
Unlike Legacy UI's, the new JET UI applies only a single custom Stylesheet in addition to the base framework-generated styling. Any existing customizations applied through the other legacy CSS files will not affect the new JET UIs. The new JET Stylesheet applies to all JET UIs in CPQ Cloud site-wide, rather than to a single UI or set of UIs. As a result, site-wide standards such as corporate styling can be implemented more easily, in one place. When additional JET UIs such as Configuration and Homepage are released, these pages will also apply the single JET Stylesheet.

Rather than downloading an initial CSS stylesheet, modifying it and uploading the modified file, customers will now reference the JET UI's system-generated CSS using a web development tool, such as Chrome Development Tools, create any desired modifications to the styling, and then upload a custom CSS file to revise the JET styling. For optimal performance, it is recommended that only modifications to the styling be included in this CSS file.

Note that styling modifications take effect immediately on the JET UIs and do not require either deployment or a refresh of browser cache.

The custom CSS file will be uploaded on the Stylesheet administration page. Navigate to the Stylesheet.

1. Click **Admin** to go to the Admin Home Page
2. Click **Stylesheet** in the Style and Templates section.
The **Stylesheet Manager** page displays as shown below.



Stylesheet Manager

CSS Upload/Download Center

Click to Download CSS:	Download Stylesheet
Alternate CSS File:	<input type="button" value="Browse..."/> No file selected.
Click to Download Alternate CSS:	Download Stylesheet
Delete Alternate CSS:	<input type="checkbox"/>
Click to Download Mobile CSS Template:	Download Mobile Stylesheet Template
Mobile CSS File:	<input type="button" value="Browse..."/> No file selected.
JET CSS File:	<input type="button" value="Browse..."/> No file selected.

3. Click Browse to upload and apply styling customizations to JET UIs

How do I create CSS for JET UI's?

As with Legacy UI, you can apply styling to Classes, ID's, and HTML elements. The ID tags created by CPQ for each attribute, action, product families, product lines, models, and custom classes all remain the same in the JET HTML structure. However, the HTML structure itself is very different from the Legacy UI because of the new framework used to create the HTML. Additionally, classes in JET are distinctly different from Legacy classes. Classes in JET for the different UI components and content are prefixed with the 'oj' for Oracle JET.

If you are new to creating CSS, here are some general guidelines and best practices:

- Classes can be reused as needed throughout your site.
 - In CSS a class name always begins with a "."
 - **Example:** '.group-list'
- ID tags are used to designate a unique element such as the individual attributes and actions that are only used once on a page.
 - In CSS an ID name always begins with a "#"
 - **Example:** '#attrib837'
- Elements are the standard HTML components
 - **Examples:**
 - h1 = Heading 1
 - p = paragraph
 - td = table cell
- Observe best practices regarding Class and ID naming conventions
 - Always use lower case
 - Always begin with a letter (CSS rules allow you to begin with an underscore_ but this is not a best practice and should be avoided)
 - **Do not use spaces:** Separate words 'with-a-dash'
 - No special characters
 - Short descriptive name
 - Try to create descriptive names that describe the function, not the appearance of the section.
 - **Example:** For an element that will be on the right hand side of the page but will hold a list of bundle names, consider naming that element 'bundle-list' instead of 'right-hand-column.'

Formatting best practices:

- For custom CSS styling, use Notepad++ or a CSS editor to write your CSS files. It will help you avoid bad syntax and improve your formatting.
- Use the tab key for indentation
- Curly brackets should open at the end of a line with the selector, and close at the beginning of the line following the last property and value
- Before changing or adding a stylesheet, it's always a good idea to download the existing or default stylesheet to your local computer, in case you need to revert or refer to the standard options.
- Be sure to properly Comment and Organize your CSS when you make additions or changes
- Be sure to document your changes, where you made them and why in your project's build tracker.
- Avoid using the `!important` tag. It's best to find the most specific styling possible, otherwise you may affect more elements than you intend. If possible, find a unique id or class for the page, grid or group that your element is within, and use that in your definition.
- Be aware that your user might be using an older browser and not all browsers follow the newest web standards. IE can be particularly troublesome. For the latest information on browser specific issues, visit: <http://caniuse.com/>
- For specifying colors, use the hex value instead of typing a color name. Most companies have a branding Style Guide with specific colors for their branding, this would contain exact hex codes to use.

- If you wish to remove the color for a selector, either comment out the property and value, or delete the whole line

- Do NOT leave a blank value after the #

- **GOOD:**

```
.bundle-list{  
    border-color: #000000;  
    border-style: solid;  
}
```

- **BAD:**

```
.bundle-list  
{  
border-color: black;  
border-style: solid;  
}
```

- **WORSE:**

```
.bundle-list {border-color: #; border-style: solid; }
```

How can I force content to stay in a fixed position? (Non-responsive behavior)

One of the features of the JET UI is the Responsive behavior. This means that content 'responds' to the viewer's browser size and adapts the content to fit the space available. In this way a viewer using a small screen, such as a mobile device, would likely see the content in a single column of attributes while a viewer using a wide screen sees the same content spread out across many columns of attributes. However, there may be content that you wish to remain fixed, regardless of the viewer's screen space. For example, there may be a series of rows for Group Discounts, where each row must remain fixed so that the row's content does not become disjointed.

- To set the content to remain fixed, it's recommended to create a custom CSS class in the Layout Editor as the selector to designate all columns that should be fixed. In the CSS example below, a custom class '.fixedcolumn' was assigned to all columns that should be fixed.
 - Use the min-width property, this disables responsive behavior, content will scale and stretch in the fixed position.

```
.fixedcolumn{
    min-width:100px;
}
```

How can I make Subtitles/Headers in my content?

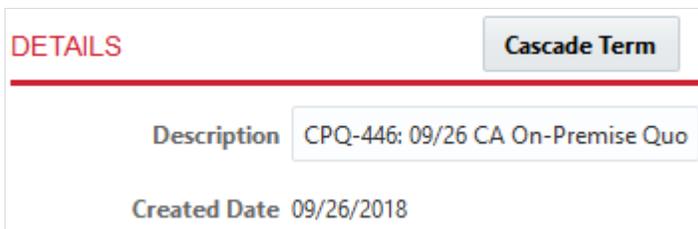
On Legacy screens, administrators typically used HTML attributes to create sub-headers over a column of attributes. Typically the HTML attribute was placed in a separate panel-row above the panel-row containing the attributes themselves. A custom class would then be assigned to the columns that would contain the sub-headers and then CSS was used to style attributes residing in a column with that class.

With the JET UI it is much easier to create sub-headers. Simply place a Toolbar where you wish a sub-header to display. In the Layout Editor settings for the Toolbar, you can assign a label. This label will serve as your sub-header text. It is also recommended that you set the toolbar alignment to Right aligned, and the Toolbar width to 100%.

The screenshot shows the JET UI Layout Editor interface. On the left, a 'Layout Elements' menu is open, listing options such as 'Add a new Panel', 'Add a new Tab', 'Add a new Panel Columns', 'Add a new Spacer', 'Add a new Toolbar', 'Add/Move a Line Item Table', and 'Add/Move the eSignature Grid'. A green arrow points from 'Add a new Toolbar' to a toolbar in the main editor. The toolbar is titled 'Toolbar' and contains a table with two columns: 'Label' and 'Action Mapping'. Below the table, there are configuration settings for the toolbar: 'Label' (set to 'Customer Details'), 'Hide Label' (unchecked), 'Toolbar Width (%)' (set to 100), and 'Toolbar Alignment' (set to Right). At the bottom right, there are buttons for 'New button', 'Ok', and 'Cancel'. The background shows a grid of panels with various attributes like 'Owner', 'Cancel Reason', and 'Created'.

In the JET CSS file you can easily style all or some of these Toolbars as desired to serve as sub-headers. Below is an example of the CSS for this and the rendered result.

```
.toolbar-with-label{
  border-bottom: 3px solid #000000;
  padding: 0px 10px 0 0;
  margin: 0 0 10px 5px;
  height: 35px;
}
.toolbar-with-label .oj-label-group{
  text-align: left;
  color: #CC2030!important;
  font-family: Helvetica, Arial, sans-serif;
  font-weight: 400;
  font-size: 14px;
  text-transform: uppercase;
}
```



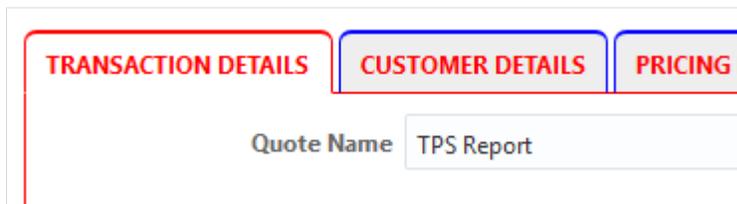
An additional benefit of using the Toolbar this way is that it allows the admin to place relevant actions into the Sub-header. This is the recommended method for placing actions in the layout as it gives context to the actions and makes it easier for the end user to find the buttons they need while viewing the related content. All buttons and attributes in the JET UI's can also be labeled independently of their definition's "Display Name", it is recommended to use this functionality to ensure that attribute and action labels are brief, clear and relevant.

How do I style tabs?

The HTML structure of tabs in the JET UI's is quite different from the HTML structure of tabs in the Legacy UI. In JET the HTML is simpler and easier to work with, additionally the Classes to be used as Selectors in JET are distinctly different. Because of this, CSS syntax used in Legacy is unlikely to work as expected if it's simply copied from the Legacy CSS file to the JET CSS file. Another change with JET is that the Layout Editor provides the option to display a set of tabs Vertically stacked, instead of laid out horizontally. When the Administrator wishes to change how tabs are styled, they can choose to change either or both of these tab sets.

Below is an example showing tab sets styled so that all tabs will use red, uppercase text for the labels, but the outline of horizontal tabs will look different from the outlines of vertical tabs.

```
.oj-tabbar-item-label {
  font-weight: 700;
  color: #FF0000;/*branding color*/
  text-transform: uppercase;
}
.oj-tabbar-horizontal .oj-tabbar-item {
  border-width: 2px 1px 1px 1px!important;
  border-radius: 6px 6px 0 0!important;
  border-color: #0000FF #0000FF #FF0000 #0000FF!important;
  background-color: #EEEEEE;
  height: 33px;
}
.oj-tabbar-horizontal .oj-tabbar-item.oj-selected {
  background-color: #FFFFFF;
  border-color: #FF0000 #FF0000 #FFFFFF #FF0000!important;
}
.oj-tabbar-vertical .oj-tabbar-item {
  border-width: 1px 1px 1px 2px!important;
}
.oj-tabbar-vertical .oj-tabbar-item.oj-selected {
  border-color: #FF0000 #FFFFFF #FF0000 #FF0000!important;
  border-width: 1px 1px 1px 3px!important;
}
.oj-tabbar-vertical > .oj-tabs-nav-root {
  padding: 0px 0 0;
}
.tab-content.oj-panel {
  border: 1px solid #FF0000!important;
  margin-top: -1px;
}
.oj-tabbar-horizontal{
  height: 33px;
  z-index: 999;
}
.oj-tabbar-divider{
  Display: none!important;
}
```



Each set of tabs is assigned by the system a unique ID, this can be used to style a particular set of tabs differently from the other tab sets.

How do I style buttons?

As with Tabs, JET UI Button HTML structure is quite different from the HTML structure of buttons in the Legacy UI. In JET the HTML is simpler and easier to work with, additionally the Classes to be used as Selectors in JET are distinctly different. Because of this, CSS syntax used in Legacy is unlikely to work as expected if it's simply copied from the Legacy CSS file to the JET CSS file. CPQ assigns a unique class to each action placed in the layout, based on the action's variable name. This can be used to style specific buttons differently from others. The Administrator can also choose to style only buttons in the page header toolbar, or buttons not in the page header toolbar, or simply style all buttons in the layout to be the same. Below is an example showing all buttons styled with uppercase, white text for the labels, and blue background, with a specific action set to show as a green button. The content ID was used to prefix the selectors so that only the buttons in the content area would be styled, and not the buttons in the Alta Navigation.

```
#content .oj-button {
  background-color: #0000FF!important;
  background-image: none!important;
  border-color: #0000FF;
  color: #FFFFFF;
}
#content .oj-button.action-submit_t {
  background-color: #00FF00!important;
  background-image: none!important;
  border-color: #00FF00;
}

#content .oj-button-label{
  color: #FFFFFF;
  font-weight: 700;
}
```



What else can I style?

Content text

To change the way text is styled in the UI, the best practice is to use the html elements as selectors. Below is an example of how the text could be styled for Headings and labels.

```
h1, h2, h3, h4, h5 {
  color: #00ff00;
  font-family: Courier;
  font-style: italic;
  font-weight: 700!important;
}
```



Panels and Page background

All of the non-tabbed panels in JET UI are generated with the same class 'oj-panel'. This makes it easy to set the same border and background settings for all panels, everywhere. Tabbed panels are all generated with the class 'oj-tabs-panel'. The main panel that contains all of the page content also has a unique ID 'main-panel' which allows you to style this one separately from the others. Finally, the title bar at the top of the page also has a unique ID 'top-toolbar'.

Example:

```
#content,  
#top-toolbar #top#toolbar-spacer,  
.oj-panel{  
    background:#FFFFFF;  
}  
#top-toolbar,  
#main-panel{  
    border: 0px solid transparent;  
    background: transparent;  
}
```

Attributes in the Line Item table

Individual columns in the Line Item table can be color-coded, or otherwise custom styled, simply by using the attribute variable name as the selector. By adding the prefix "col-" before the attribute name, this tells the JET UI that you wish to style table columns that contain that attribute.

Example:

```
.col-pricePeriod_1{  
    background: #ff0000!important;  
}
```

Specific tabs

Individual tabs can be styled differently from the others if desired. The most reliable way to do this is by using the Nth-child selector. Below is an example of styling the 3rd tab differently from the others.

Example:

```
li.oj-tabbar-item:nth-child(3) {  
    border-color: 2 #FF0000 solid;  
    background: #00FF00;  
}
```

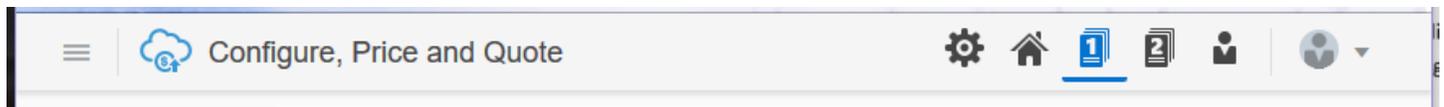
How do I change the icons in the Alta Navigation?

The Alta Navigation became available with Oracle CPQ release 2017 R2. This Navigation UI uses the JET framework and so shares many similarities to the JET Transaction UI. At this time, it is not affected by the JET CSS file that applies to the Transaction UI (and other JET UI's in the future). While this is likely to change in the future, for now site Administrators can use the previously available `_alt_nav.css` file to customize Alta navigation in the same way that Legacy navigation is styled.

Below is an example of how CSS can be applied to the Alta Navigation so that the commerce process icons render alternative images for each state, these images are uploaded using the File Manager. Additional styling centers the title content vertically.

```
#cpq-alta-header #cpq-header-tb .cpq-nav-threepages[aria-labelledby="ui-id-4"]
.cpq-nav-icon::before {
    content: url("../..//image/Icons/navi_process1_24_ena.png")!important;
}
#cpq-alta-header #cpq-header-tb .cpq-nav-threepages[aria-labelledby="ui-id-5"]
.cpq-nav-icon::before {
    content: url("../..//image/Icons/navi_process2_24_ena.png")!important;
}
#cpq-alta-header #cpq-header-tb a.oj-hover[aria-labelledby="ui-id-4"] .cpq-nav-
icon::before {
    content: url("../..//image/Icons/navi_process1_24_hov.png")!important;
}
#cpq-alta-header #cpq-header-tb a.oj-hover[aria-labelledby="ui-id-5"] .cpq-nav-
icon::before {
    content: url("../..//image/Icons/navi_process2_24_hov.png")!important;
}
#cpq-alta-header #cpq-header-tb a.cpq-selected[aria-labelledby="ui-id-4"] .cpq-
nav-icon::before {
    content: url("../..//image/Icons/navi_process1_24_act.png")!important;
}
#cpq-alta-header #cpq-header-tb a.cpq-selected[aria-labelledby="ui-id-5"] .cpq-
nav-icon::before {
    content: url("../..//image/Icons/navi_process2_24_act.png")!important;
}
#cpq-alta-header > span{
    align-items: center!important;
}

#cpq-alta-header > span > img,
#cpq-alta-header > span > p {
    margin: 0 5px!important;
}
```



How can I provide input to the CPQ UX direction?

CPQ Cloud's User Experience team conducts periodic, ongoing usability testing to gather feedback on our UX enhancements and to validate that our new UI design patterns are intuitive and easy to learn by your users. By participating in this testing your company has an ability to preview and provide input on CPQ's UX direction.

Please contact Shirin Lange (shirin.lange@oracle.com) to participate.

Where can I learn more?

- [Oracle CPQ Cloud Release 18C What's New](#)
- CPQ Cloud Administration Online Help
- [CPQ Cloud JET Transaction UI: Planning for Adoption \(Doc ID 2455080.1\)](#)
- [CPQ Cloud JET Configuration and Transaction UI: Refactoring Existing JavaScript Customizations \(Doc ID 2490016.1\)](#)