

Customizing Your OEM Build

- [Things to note](#)
- [How to Customize the Images](#)
- [How to Customize the labels](#)
- [How to Customize the Settings Purchase License link](#)
- [How to Customize the App Center Download from Marketplace link](#)
- [How to Customize the App Center](#)
- [How to Customize Default Apps:](#)
- [How to Blacklist Java classes](#)
- [Artifacts By Folder and Type located in the OEM Package](#)
 - [/src/main/webapp/](#)
 - [/src/main/resources](#)
 - [/src/main/java](#)
 - [/src/main/webapp/WEB-INF/lib](#)

Things to note

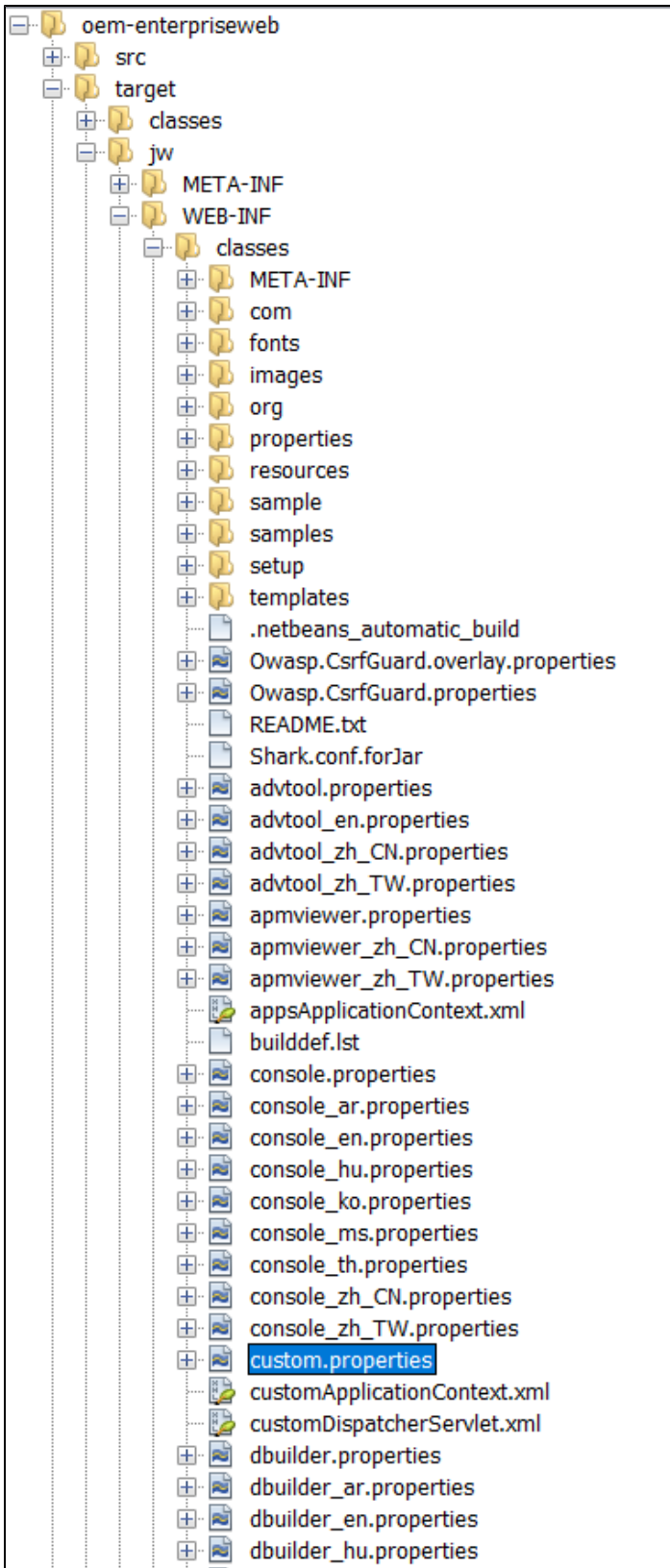
- By default, the OEM project replaces the standard artifacts with a default implementation (blue theme).
- Replace and customize the source artifacts within the project as required.
- The Java interface implementations provided in the OEM package are samples only. It is recommended to implement classes in a separate JAR project.

How to Customize the Images

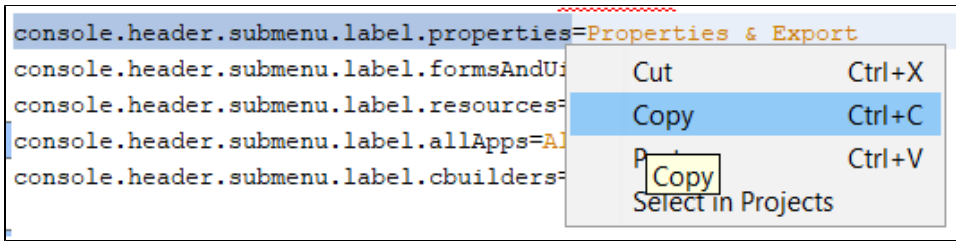
Custom images are already in the correct dimensions, edit the images directly in **OEM package folder\src\main\webapp\images**

How to Customize the labels

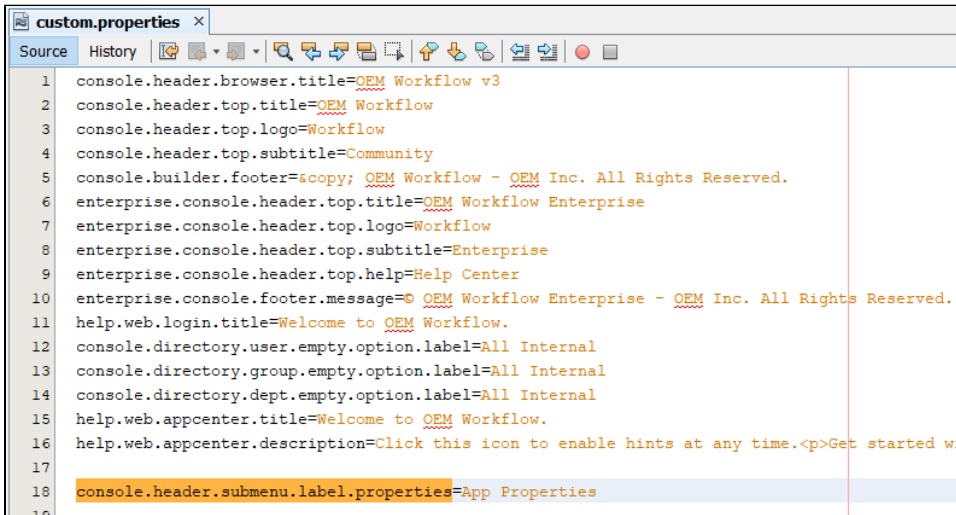
1. Once you have performed step 3 in [Overall Build Flow For OEM Build](#), you can locate the Property Resource Bundles in **OEM package folder\target\w\WEB-INF\classes**. Open the appropriate .properties file via a text editor or your preferred IDE.



2. Find the label you wish to customize, and copy the key.



3. Paste the key into OEM package folder\src\main\resources\console.properties and edit its value.



How to Customize the Settings Purchase License link

Customize the following keys in OEM package folder\src\main\resources\custom.properties:

license.purchase - The label for the Purchase License link

license.purchase.url - The URL to go to when the Purchase License link is clicked, empty hides the link

How to Customize the App Center Download from Marketplace link

Customize the following keys in OEM package folder\src\main\resources\custom.properties:

appCenter.link.marketplace.trusted - Comma-delimited list of trusted URLs for Marketplace installation

If you using the default App Center app, the link to the Marketplace is defined in the first **Custom HTML** element in the **Published Apps** form.

How to Customize the App Center

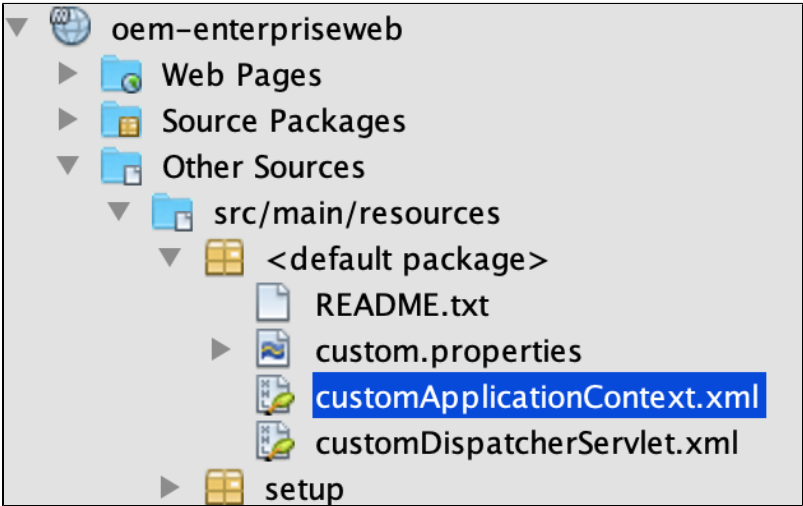
The App Center is installed from the OEM package folder\src\main\resources\setup\apps\APP_appcenter-1.zip app during the first initialization. Replace this file with your own app as required, but you must use the same filename.

How to Customize Default Apps:

1. Place your default apps in oem package folder /src/main/resources/setup/apps/
2. Edit /src/main/resources/setup/setup.properties and define the paths to your apps.
3. You should also customize the default database bundled in your custom OEM installer.

How to Blacklist Java classes

Should you wish to remove a certain form element or any java classes from the OEM, go to OEM package folder\src\main\resources\customApplicationContext.xml



uncomment the `<property name="blackList">` element and insert the class into its `<set>` child element

```
<!-- Customized blacklist and classpath scanning for the PluginManager -->
<bean id="pluginManager" class="org.joget.plugin.base.PluginManager" destroy-method="shutdown">
  <!--
  <property name="blackList">
    <set>
      <value>com.acme.plugin.SamplePlugin</value>
      <value>com.acme.plugin.TestPlugin</value>
    </set>
  </property>
  <property name="scanPackageList">
    <set>
      <value>com.acme</value>
      <value>org.acme</value>
    </set>
  </property>
  -->
</bean>
```

Artifacts By Folder and Type located in the OEM Package

pom.xml - Maven POM

/src/main/webapp/

| JSP Files | Description |
|----------------------------------|----------------------------------------------------|
| /index.jsp | index page |
| /home/* | home folder containing the index.html landing page |
| /WEB-INF/jsp/console/home.jsp | home screen after logging in the web console |
| /WEB-INF/jsp/console/welcome.jsp | welcome area on the right in the home screen |
| /WEB-INF/jsp/error404.jsp | 404 not found error page |
| /WEB-INF/jsp/error500.jsp | 500 internal server error page |

| | |
|---------------------|--------------------|
| /mobile/offline.jsp | offline error page |
|---------------------|--------------------|

You can also override any of the existing jsp pages by matching the original file location.

For example, you would like to customize how the admin bar is presented, create the file **"adminBar.jsp"** in **"/src/main/webapp/WEB-INF/jsp/console/apps/"** folder.

| CSS Files | Description |
|---------------------------|--------------------------------|
| /css/admin_bar_custom.css | custom admin bar CSS |
| /css/builder_custom.css | custom CSS for the builders |
| /css/console_custom.css | custom CSS for the web console |

| Images | Description |
|------------------------|---------------------------------|
| /images/custom/* | custom images used |
| /images/favicon.ico | browser favicon for web console |
| /images/favicon_uv.ico | browser favicon for usersviews |

/src/main/resources

| Files | Description |
|------------------------------|-------------------------------------------------------------------------------|
| customApplicationContext.xml | custom Spring application context for custom bean and property initialization |
| custom.properties | custom resource bundle for labels |

/src/main/java

| Files | Description |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sample/SampleDataEncryptionImpl.java | A sample implementation of the DataEncryption plugin, using SHA-256 with salts for hashing and encryption using the Jasypt library with PBKDF2WithMD5AndDES algorithm and 1000 key obtention iterations. |
| sample/SamplePropertiesImpl.java | Sample datasource Properties implementation that simply logs encrypted and decrypted values. |

/src/main/webapp/WEB-INF/lib

You may place additional plugins that you have developed in .jar file extensions. You can also place OSGi plugins into this folder too but they will be loaded using the standard class loader. You may read more at [Introduction to Plugin Architecture](#). By placing them here, they cannot be removed dynamically and treated as "core" through [Manage Plugins](#) as they are not dynamically loaded.

You may also want to consider adding the dynamic OSGi plugins into "wflow/app_plugins" as part of the installation bundle creation. Please check out the **"wflow-install"** project and [Creating a Windows Installer for Your OEM Build](#) for Windows installer.