**COVE Build & Deployment Guide**

Version: 1.2

Date: 3rd April 2017

Author: *Ashfaque Hussain*

Table of Contents

[1 Goal 4](#_Toc478984019)

[2 Scope 4](#_Toc478984020)

[3 Build Tools 4](#_Toc478984021)

[3.1 List of Tools 4](#_Toc478984022)

[3.2 Installation Steps 4](#_Toc478984023)

[4 Application 5](#_Toc478984024)

[4.1 Components to be Build 5](#_Toc478984025)

[4.2 Code Structure 6](#_Toc478984026)

[4.3 Build Process 6](#_Toc478984027)

[4.4 Code Coverage 9](#_Toc478984028)

[4.5 Application components Deployment Process 9](#_Toc478984029)

[5 Configuration Files 14](#_Toc478984030)

[5.1 Webserver Configuration Files 14](#_Toc478984031)

[5.2 Application Server Configuration Files 15](#_Toc478984032)

[6 Backup Procedure for Property Files 16](#_Toc478984033)

[6.1 Web Server 16](#_Toc478984034)

[6.1.1 httpd.conf 16](#_Toc478984035)

[6.1.2 ssl.conf 16](#_Toc478984036)

[6.1.3 webgate.conf 16](#_Toc478984037)

[6.1.4 tccluster.conf 16](#_Toc478984038)

[6.1.5 htdocs folder 16](#_Toc478984039)

[6.1.6 Artifacts 16](#_Toc478984040)

[6.2 Application Server 17](#_Toc478984041)

[6.2.1 portal-setup-wizard.properties 17](#_Toc478984042)

[6.2.2 camel\_props.properties 17](#_Toc478984043)

[7. Cache Clearing 17](#_Toc478984044)

[8. Procedure to Update/Modify Properties 17](#_Toc478984045)

[a. Web Server 17](#_Toc478984046)

[b. Application Server 18](#_Toc478984047)

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 28-09-2016 | 0.1 | Initial Draft Version | Ashfaque Hussain Mohammed |
|  | 0.2 | 1) Updated portlets artifacts list  2) Node modules to be installed  3) Code Coverage for Liferay Portlets | Ashfaque Hussain Mohammed |
| 14-11-2016 | 0.3 | 1. Updated the latest jar/war information  2. Removed the unnecessary plugins/tools  3. Updated the build steps for both jar/war | Aneef Kandukuri |
| 23-12-2016 | 0.4 | 1. Added the Webserver Property Files Detail  2. Backup procedure added for the application and webserver files | Vranda Vyas |
| 03-01-2017 | 0.5 | 1. Updated the file locations in section **“b)Liferay Portlets and JAR Build”** page 6 | Vranda Vyas |
| 04-01-2017 | 0.6 | 1. Added the build procedure for blank-theme in section **“a) Liferay Themes Build”** page 6  Also updated a new war file related to Epic-2 named as **1-3-3-column-layout**  2. Added build procedure for 1-2-2-column-layout and 1-3-3-column-layout under the section **b) Liferay layout Build** page 7 | Vranda Vyas |
| 05-01-2017 | 0.7 | Added a section 8 for cache clearing locations | Vranda Vyas |
| 10-01-2017 | 0.8 | Updated the section of Jar deployment under section 4.5 a and b | Vranda Vyas |
| 11-01-2017 | 0.9 | Added the Change request war files detail | Vranda Vyas |
| 18-01-2017 | 1.0 | Updated as per the following points:  2) Liferay jar file Deployments  6.1) Webserver Configuration Files  6.2) Application Server Configuration Files  8) Cache Clearing  9.) Procedure to Update/Modify Properties | Vranda Vyas |
| 14-03-2017 | 1.1 | Updated section 4.3 c) section. Updated the list of war files to be generated and deployed | Vranda Vyas |
| 03-04-2017 | 1.2 | Additional portlet location updated in 4.3 c) and 4.5 1) | Vranda Vyas |
| 07-04-2017 | 1.3 | Additional portlet location updated in 4.3 c) and 4.5 1) | Sudhama Krishna Cheruvu |

# Goal

This document has the goal to aggregate all relevant information regarding the build and deployment process related to GOOP application. Also it includes the webserver related properties and files that need to be added/modified as part of deployment with their respective locations within the server.

# Scope

The scope of this document is the updating/modifying of the property files when required.

# Build Tools

Below are the list of tools that need to be installed and the installation process.

## List of Tools

1. NodeJS & NPM
2. YO & GULP
3. Apache Maven

## Installation Steps

Below are installation steps for the builds tools that are required as part of the build setup.

1. Installing NodeJS and NPM

$ tar -xvf node-v4.5.0-linux-x64.tar.xz

$ ln -s /opt/oracle/backup-soft/node-v4.5.0-linux-x64/bin/node /usr/bin/node

$ ln -s /opt/oracle/backup-soft/node-v4.5.0-linux-x64/bin/npm /usr/bin/npm

Checking Version Info

$ node -v

$ npm version

Set below Environment Variables. Include them in user’s bash profile

$ export NPM\_PACKAGES=/home/vodadmin/.npm-packages

$ export NODE\_PATH=NPM\_PACKAGES/lib/node\_modules

$ export PATH=${PATH}:${NPM\_PACKAGES}/bin

1. Installing Yeoman and gulp

$ npm install -g yo gulp

1. Installing Maven

$ sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

$ sudo yum install apache-maven

1. Installing Karma & Jasmine

$ npm install -g karma karma-cli karma-coverage –save-dev

$ npm install -g jasmine-core karma-jasmine karma-chrome-launcher –save-dev

# Application

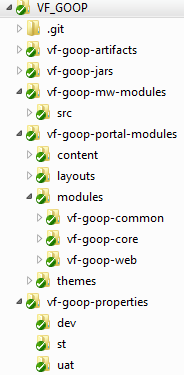
## Components to be Build

The components to be build is listed as below.

1. Liferay
   1. Portlets
   2. Themes
   3. Layouts
   4. OSGI Bundles
2. Camel
3. Liferay Content
4. Static Code/Content
5. Liferay Portal-setup-wizard.properties
6. Camel camel\_props.properties

## Code Structure

Below is the code structure in the repo.



## Build Process

1. **Liferay Themes Build**

**Code Base** – [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/goop-theme

**Artifacts** – [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/goop-theme/dist/**goop-theme.war**

For doing this build, we will need gulp tool. Use the below commands to do the build

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/goop-theme

$ gulp build

**Code Base** – [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/blank-theme

**Artifacts** –[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/blank-theme/dist/**blank-theme.war**

For doing this build, we will need gulp tool. Use the below commands to do the build

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/blank-theme

$ gulp build

1. **Liferay layout Build**

**Code Base** – [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/layouts/1-2-2-columns-layouttpl/

**Artifacts** –[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/layouts/1-2-2-columns-layouttpl/dist/**1-2-2-columns-layouttpl.war**

For doing this build, we will need gulp tool. Use the below commands to do the build

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/layouts/1-2-2-columns-layouttpl/

$ gulp build

**Code Base** – [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/layouts/1-3-3-columns-layouttpl/

**Artifacts** –[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/layouts/1-3-3-columns-layouttpl/dist/**1-3-3-columns-layouttpl.war**

For doing this build, we will need gulp tool. Use the below commands to do the build

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/layouts/1-3-3-columns-layouttpl/

$ gulp build

1. **Liferay Portlets and JAR Build**

**Code Base** – [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules

**Artifacts** –

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-common/target/**vf-goop-common-1.0.0.jar**

[REPO\_ROOT\_FOLDER]/ vf-goop-portal-modules/modules/vf-goop-core/vf-goop-components/vf-goop-autologin/target/**vf-goop-autologin-1.0.0.jar**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-core/vf-goop-webservices/vf-goop-webservices-service/target/**vf-goop-webservices-services-1.0.0.jar**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-core/vf-goop-webservices/vf-goop-webservices-api/target/**vf-goop-webservices-api-1.0.0.jar**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/goop-theme/dist/**goop-theme.war**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-web/vf-goop-welcome-portlet/target/**wp.war**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-web/vf-goop-csv-user-import-portlet/target/**csvip.war**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/themes/blank-theme/dist/**blank-theme.war**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-web/vf-goop-raise-incident-request-portlet/target/**rrp.war**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules/vf-goop-web/vf-cove-table-view-portlet/target/**ctvp.war**

[REPO\_ROOT\_FOLDER]/ vf-goop-portal-modules/modules/vf-goop-web/vf-cove-application/ target/ **vf-cove-application.war**

For doing Portlets build, we will need maven to be installed.

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules

$ mvn clean install

1. **Camel code Build**

**Code Base** – [REPO\_ROOT\_FOLDER]/vf-goop-mw-modules

**Artifacts** – [REPO\_ROOT\_FOLDER]/vf-goop-mw-modules/target/**vf-goop-mw-modules.war**

For doing Camel code build, we will need maven to be installed.

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-mw-modules

$ mvn clean install

1. **Liferay Content**

For Liferay content there is no build required, the artifacts will be available inside code base.

**Artifacts** –

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/content/document-library/**VF\_GOOP\_Documents\_-201609191540.portlet.lar**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/content/web/**VF\_GOOP\_Web\_Content-201609191542.portlet.lar**

[REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/content/web/**Web\_Content-201609191130.portlet.lar**

## Code Coverage

1. **Liferay Portlets**

Coverage Tool: **Cobertura**

For getting Cobertura report, we will need to run the below maven command.

$ cd [REPO\_ROOT\_FOLDER]/vf-goop-portal-modules/modules

$ mvn cobertura:cobertura cobertura.aggregate=true cobertura.report.format=xml

1. **AngularJS**

Coverage Tool: Karma/Jasmine

Code is still being checked in by DEV

## Application components Deployment Process

As of now, we are following only manual deployment procedure.

Note: We need to deploy the jar files first, followed by the Liferay war files and further camel war file.

1. Liferay war & jar files Deployments

To do the deployments in Liferay, you need to copy the below artifacts to Deploy folder of Liferay. As soon as you copy the files to deploy folder, they will be picked up by Liferay. No need to stop the server before the copy. Also no restarts needed.

|  |
| --- |
| **vf-goop-webservices-api-1.0.0.jar** |
| **vf-goop-webservices-services-1.0.0.jar** |
| **vf-goop-autologin-1.0.0.jar** |
| **blank-theme.war** |
| **goop-theme.war** |
| **csvip.war** |
| **wp.war** |
| **rrp.war** |
| **ctvp.war** |
| **vf-cove-application.war** |

Eg: /opt/SP/softwares/liferay-dxp-digital-enterprise-7.0-ga1/deploy/

The below jar needs to be copied to /osgi/static/ folder. Placing of this jar is a onetime activity and to be done on request.

|  |
| --- |
| **vf-goop-common-1.0.0.jar** |

Eg: /opt/SP/softwares/liferay-dxp-digital-enterprise-7.0-ga1/osgi/static/

1. Liferay jar file Deployments

Follow the below process in case of Liferay jar files deployment.

1. **First time deployment of all jar files**

Follow the below sequence while we are deploying jar files as below:

**vf-goop-webservices-api-1.0.0.jar**

**vf-goop-webservices-services-1.0.0.jar**

**vf-goop-autologin-1.0.0.jar**

Note: Even In case of deployment of **vf-goop-webservices-services-1.0.0.jar** individually, **we need to remove the “vf-goop-webservices-api-1.0.0.jar”** as well from the below location:

**$cd /opt/SP/weloadm/software/liferay-dxp-digital-enterprise-7.0-ga1/osgi/modules**

Than further first deploy “**vf-goop-webservices-api-1.0.0.jar**” in the deploy folder and following it deploy “**vf-goop-webservices-services-1.0.0.jar”**

**IMPORTANT: ALWAYS DEPLOY** “**vf-goop-webservices-api-1.0.0.jar**” FIRST AND THAN “**vf-goop-webservices-services-1.0.0.jar**”

1. **One Time Deployment of the below jar file**

This ja

1. Camel Deployments

To deploy the camel code, we need to copy the below artifact to Webapps folder of Tomcat. No need to stop the server before the copy. Also no restarts needed. Please make sure to remove the existing vf-goop-mw-modules.war file and vf-goop-mw-modules from the webapps folder of tomcat before placing the new file**.**

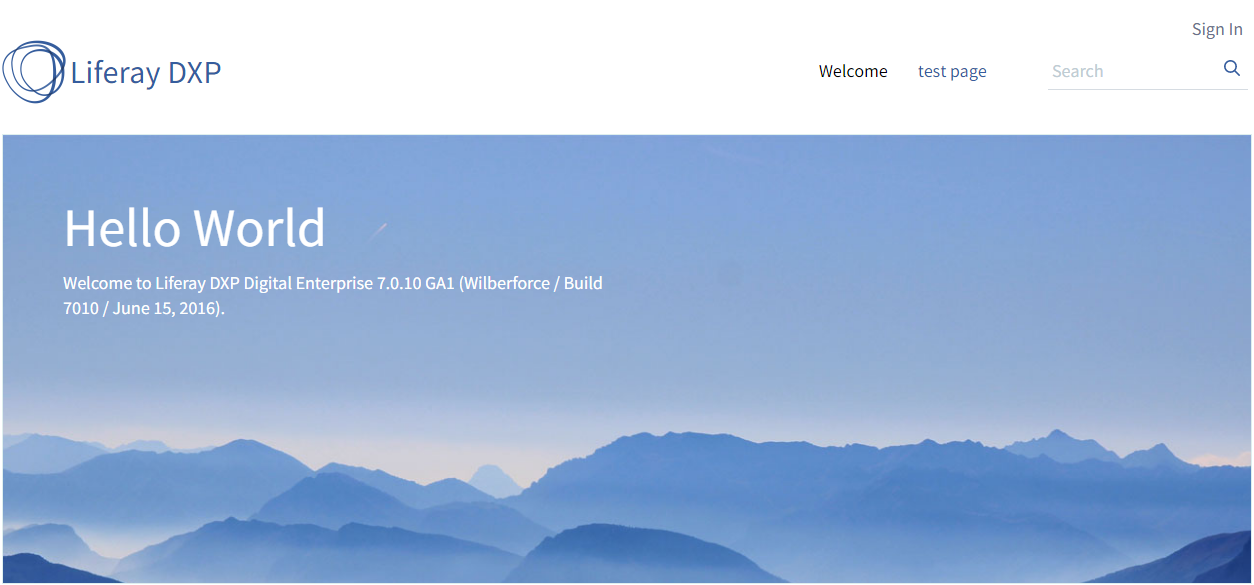
**vf-goop-mw-modules.war**

Eg: /opt/SP/softwares/liferay-dxp-digital-enterprise-7.0-ga1/tomcat-8.0.32/webapps/

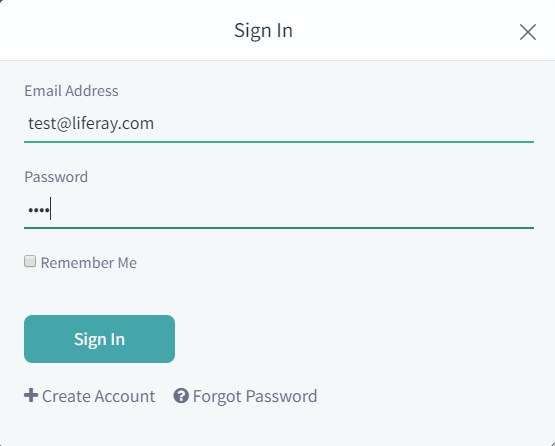
1. Liferay Content Import/Export

The Liferay Content deploy can be done by importing the LAR files in to the Liferay Portal. Please follow the below steps to perform this activity.

1. Once the application is up and running, open the Liferay portal by hitting the URL.



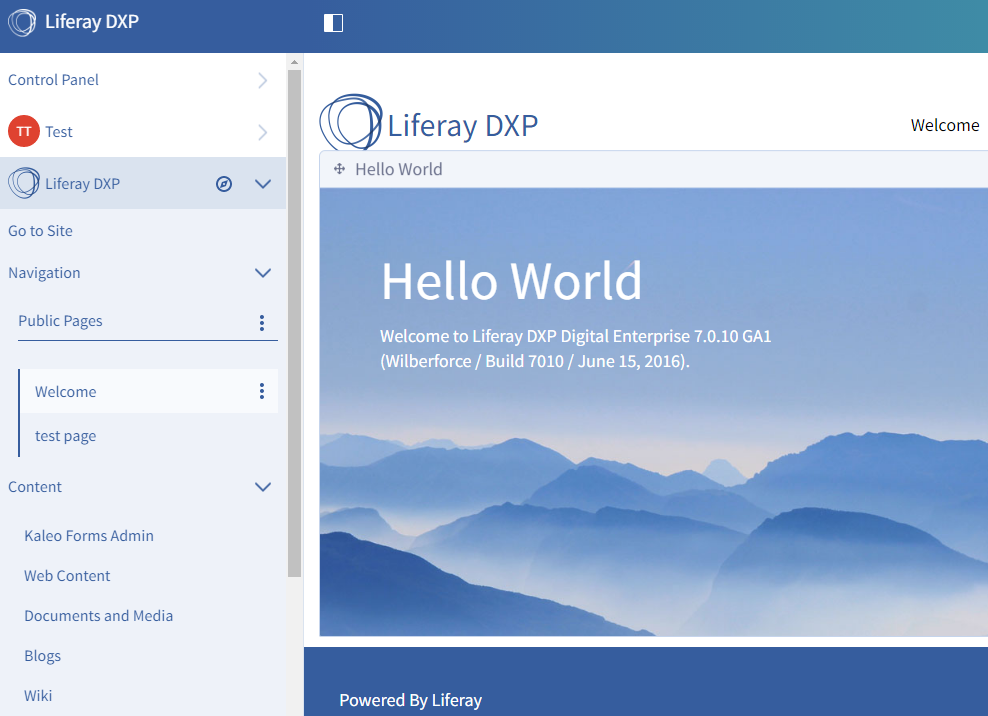
1. Click on Sign In and login to the portal with appropriate credentials.

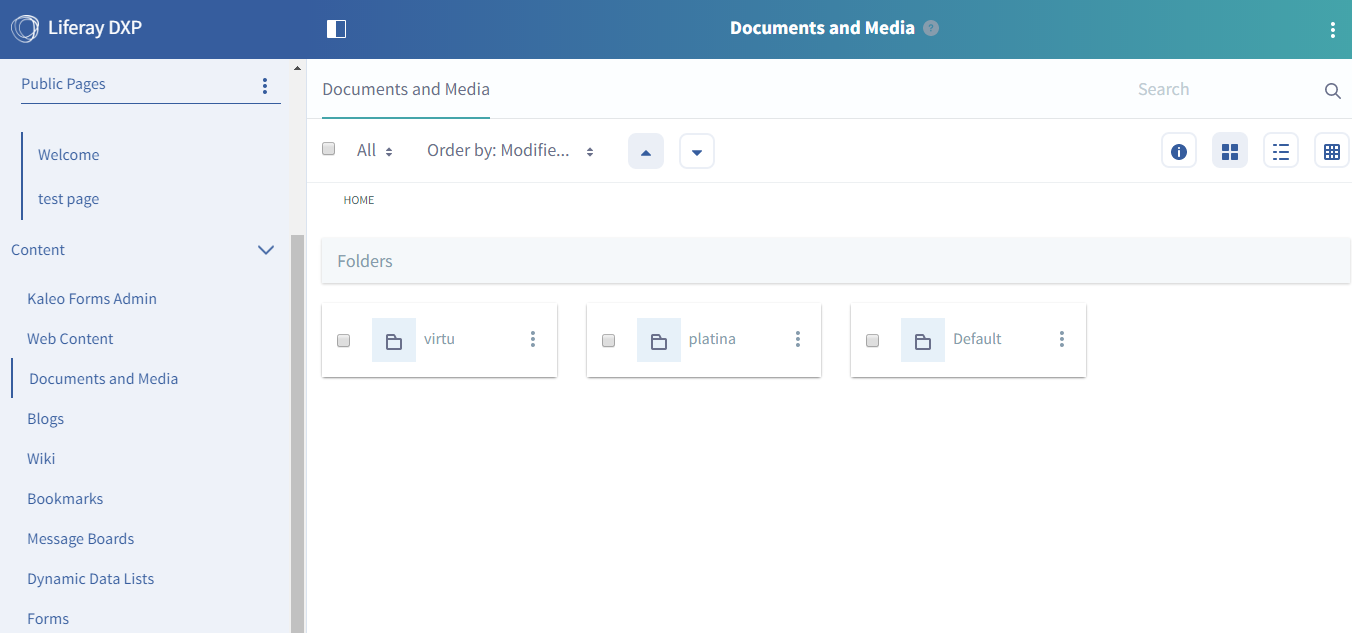


1. Once you sign in you will see the menu bar on left panel

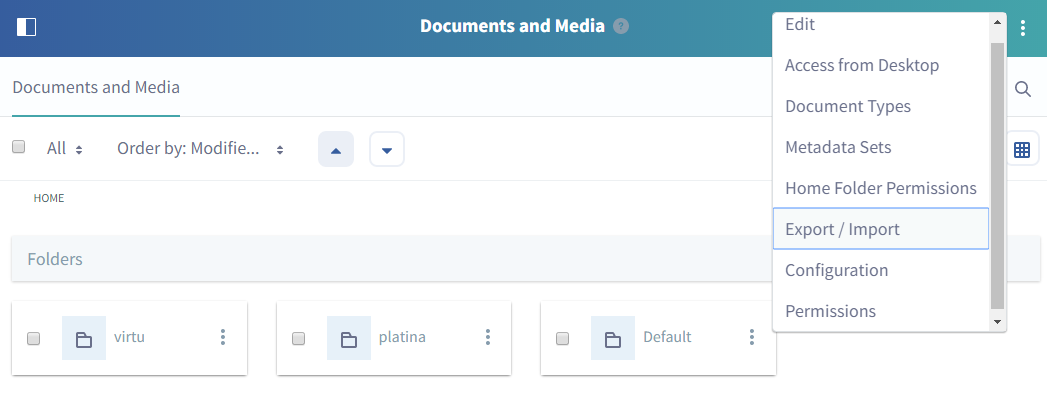


1. Click on Content -> Document and Media

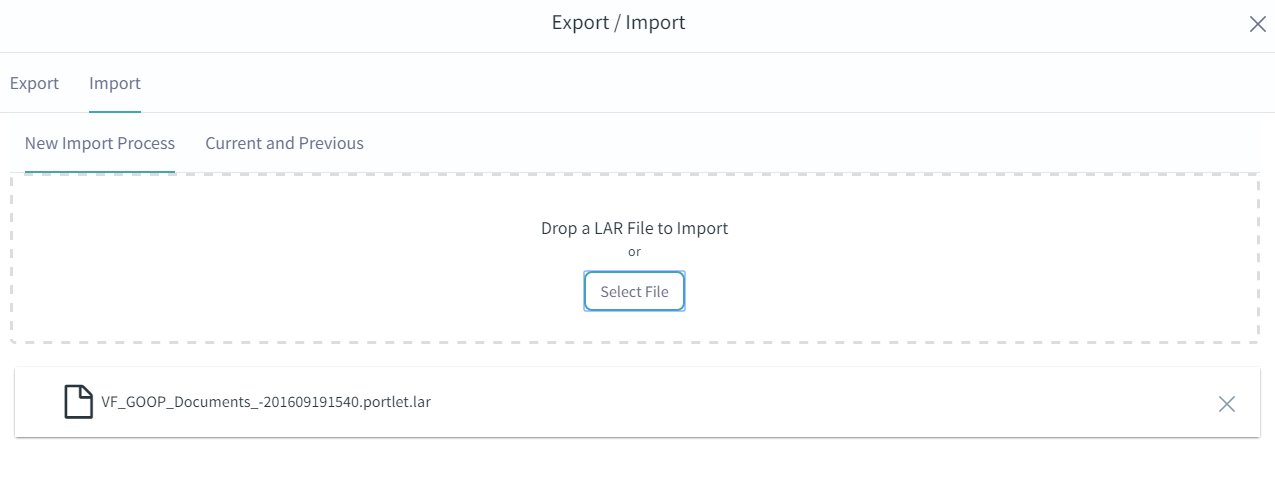




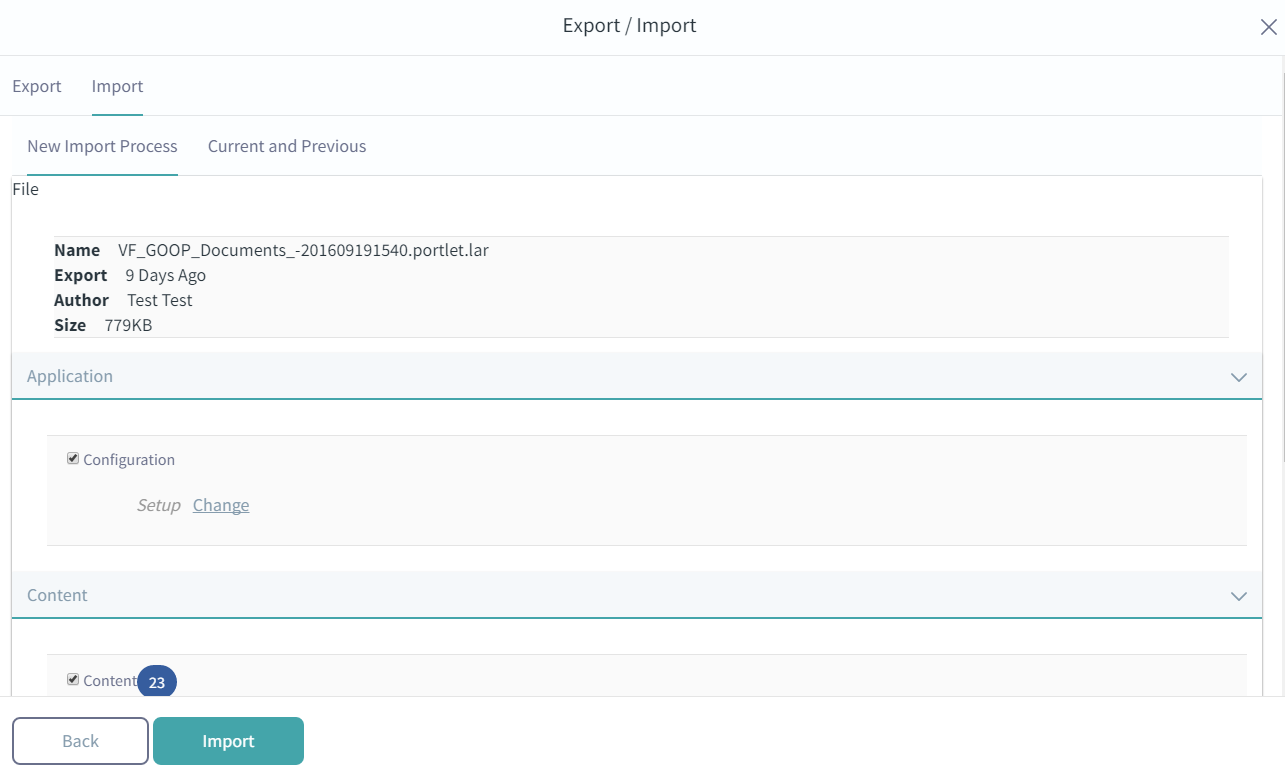
1. Once it is loaded, you can find the option from the top right corner. Select Export / Import from that list.



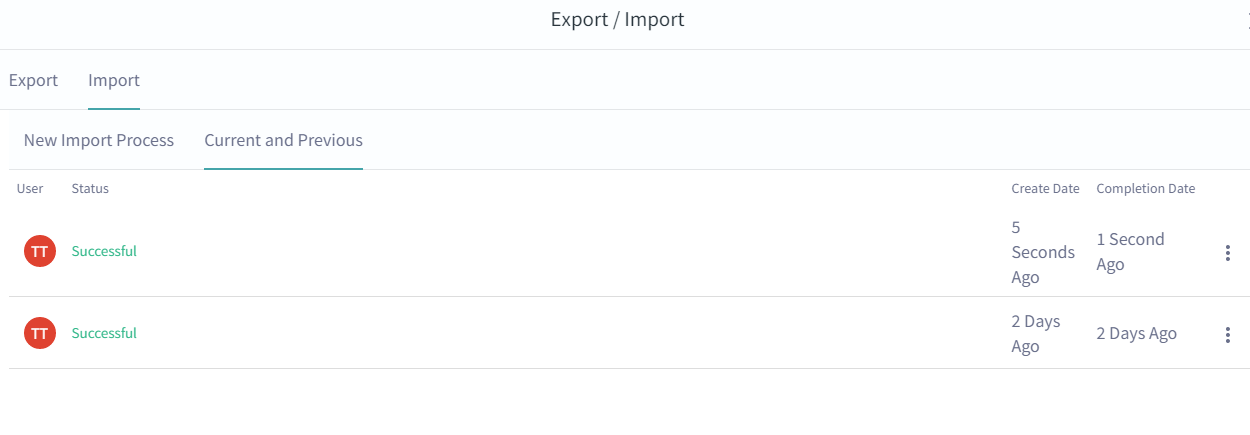
1. This will open Export / Import popup and from there you can select Import tab. Now select the file which you want to upload and this will add the file as shown below.



1. Now click on Continue button at the bottom and you will get the summary page for Import. You can verify the changes here and confirm by clicking on Import button.



1. Once the import is done, you will see the below screen that shows the Import status.



# Configuration Files

There are a list of configuration files that also need to be updated during the builds and deployment as per the requirement. Below given are the same list of files with their specific locations on the server so as to update them accordingly.

## Webserver Configuration Files

|  |  |  |
| --- | --- | --- |
| **S.No.** | **File Name** | **File Location** |
|  | httpd.conf | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
|  | ssl.conf | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
|  | tccluster.conf | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1/moduleconf |
|  | webgate.conf | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
|  | htdocs folder | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
|  | Web Server Artifacts | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1/webgate/config |
|  | Web Server Artifacts all three environments(**Gitlab Location)** | vf-goop-properties\dev\Webserver\GOOP\_APP\_ACTL  vf-goop-properties\st\Webserver\GOOP\_APP\_ACTL  vf-goop-properties\uat\Webserver\GOOP\_APP\_UAT\_ACTL |
|  |  |  |

Note: Webserver restart is required in case of update in the property files

## Application Server Configuration Files

|  |  |  |
| --- | --- | --- |
| **S.No.** | **File Name** | **File Location** |
|  | portal-setup-wizard.properties | **$LIFERAY\_HOME** |
|  | camel\_props.properties | /opt/SP/weloadm/software/camel\_prop |

Note: Application Server restart is required in case of update in the property files

# Backup Procedure for Property Files

For taking the backup of the property files follow the given steps:

1. Navigate to the following location.
2. Copy the file to the backup folder located at location: **/opt/SP/weloadm/software/backup**

## Web Server

### httpd.conf

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
| # | cp httpd.conf /opt/SP/weloadm/software/backup |

### ssl.conf

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
| # | cp ssl.conf/opt/SP/weloadm/software/backup |

### webgate.conf

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
| # | cp webgate.conf /opt/SP/weloadm/software/backup |

### tccluster.conf

|  |  |
| --- | --- |
| # | /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1/moduleconf |
| # | cp tccluster.conf /opt/SP/weloadm/software/backup |

### htdocs folder

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1 |
| # | cp –r htdocs /opt/SP/weloadm/software/backup |

### Artifacts

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/config/OHS/ohs1/webgate/config |
| # | cp \* /opt/SP/weloadm/software/backup |

## Application Server

### portal-setup-wizard.properties

|  |  |
| --- | --- |
| # | cd $LIFERAY\_HOME |
| # | cp portal-setup-wizard.properties /opt/SP/weloadm/software/backup |

### camel\_props.properties

|  |  |
| --- | --- |
| # | cd /opt/SP/weloadm/software/camel\_prop |
| # | cp camel\_props.properties /opt/SP/weloadm/software/backup |

# Cache Clearing

To avoid any issue (related to new changes being not picked up) it is a better practice to clear the cache before code deployment each time. Clear the cache from the below locations:

|  |  |
| --- | --- |
| # | <LIFERAY\_HOME>/work |
|  | <LIFERAY\_HOME>/osgi/state |
|  | <LIFERAY\_HOME><Tomcat\_HOME>/temp |
|  | <LIFERAY\_HOME><Tomcat\_HOME>/work |

# Procedure to Update/Modify Properties

To update or add new properties we need to follow a set of steps so as the updated or newly added properties to reflect. Follow the below steps to achieve the same.

Note: Restart is required in application server if the property files related to application server are updated and same in case of webserver. (Do not need to restart both each time even if we are not updating any property files related to them)

## Web Server

1. Stop the server by using the below command:

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/bin |
| # | ./opmnctl stopall |

1. Navigate to the file location as mentioned in **Step 3.**
2. Take the backup of the required property file as mentioned in **Step 5**
3. Update or modify the file as required and save it.
4. Start the server using the following command:

|  |  |
| --- | --- |
| # | cd /opt/SP/wwwadm/software/ohs/Oracle\_WT1/instances/goop-dev/bin |
| # | ./opmnctl startall |

## Application Server

1. Stop the server by using the below command:

|  |  |
| --- | --- |
| # | cd **$LIFERAY\_HOME/**tomcat-8.0.32/bin |
| # | ./shutdown.sh |

1. Navigate to the file location as mentioned in **Step 4.**
2. Take the backup of the required property file as mentioned in **Step 5.**
3. Update or modify the file as required and save it.
4. Start the server using the following command:

|  |  |
| --- | --- |
| # | cd **$LIFERAY\_HOME/**tomcat-8.0.32/bin |
| # | ./startup.sh |