

**Liberty Health**

**Release Management Process**

**Project: Commissions Self-Build**

Prepared by: Jaco Bosman

LIFE INVESTMENTS HEALTH CORPORATE PROPERTIES ADVICE

DOCUMENT HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Amended by** | **Changes** |
| 1.0 | 01-04-2018 | Draft | Jaco Bosman |  |
| 1.0.1 | 10-04-2018 | Correction | Sarel Eybers | First Check and Update |

TABLE OF CONTENTS

Contents

[Purpose 2](#_Toc511048378)

[1.1. Background 2](#_Toc511048379)

[1.2. Compiling and generate .Jar files 3](#_Toc511048380)

[1.3. Compile/Deploy Home application 4](#_Toc511048381)

[1.4. Deploying to Weblogic 5](#_Toc511048382)

[1.4.1 Deploying to production 5](#_Toc511048383)

[1.4.2 Deploying to non-production env 5](#_Toc511048384)

[1.5. BitBucket - Promote code 9](#_Toc511048385)

# Purpose

The purpose of this document is to illustrate the release management process for the ADF Commissions system and will also serve as part of the handover process.

## Background

The Commission ADF application is currently split into 3 different applications, with various different projects within. These 3 are CommissionHome, CommissionViewController and CommissionBtf. Each serves a different purpose as outlined below:

**CommissionHome:**

This application contains the shell/templates of the application. This is where the look & feel and the roles and security of the application are set. The home project needs to include all the .jar files that are used in any of the 3 projects and the Home application is then deployed as an .ear file, which can be deployed onto the WebLogic server.

**CommissionViewController:**

This application contains all landing pages (or better known as search screens) in the View Controller project, as well as the watch list functionality. This project should never have a Model/database project. The project gets deployed as a .jar file.

**CommissionBtf:**

This application holds the Model project and all BTF projects. BTF projects are setup per business function and this is where the CRUD operations occur. These projects are deployed as .jar files and then incorporated into the CommissionViewController task flows.

## Compiling and generating .jar files

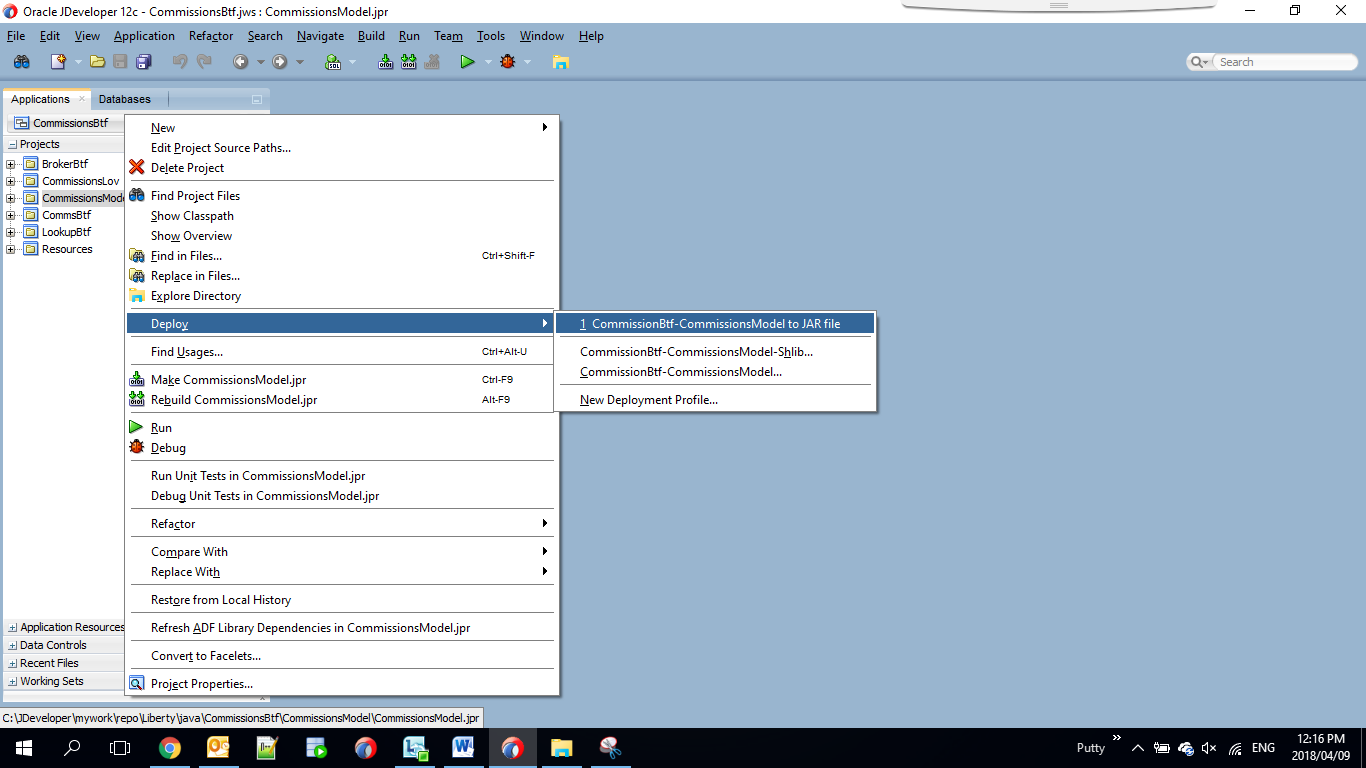
In order to get the application ready for deployment, ensure that all code is up to date from BitBucket. Open JDeveloper and compile each of the 3 applications mentioned above.

All of the .jar files required for the application are stored in the lib-repo folder of the Commissions project.

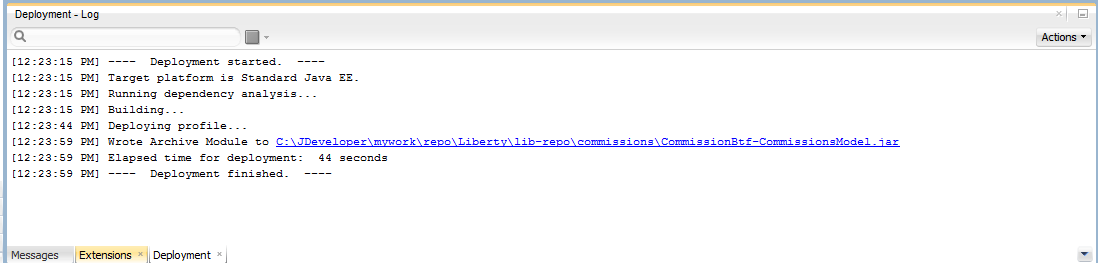
In order to generate the .jar files for the CommissionsBtf project as well as the CommissionViewController project, repeat the below steps for each project in the application:

1. Right click the project and select the deploy menu item, followed by the CommissionBtf-[ProjectName] option.

NB! The deployment profile ending in Shlib should not be used as the current architecture of the deployments are not setup for this.



Once the jar file has been generated successfully, you'll see the following in the log window.



<snip> All JAR files deployed and EAR file also deployed

[03:07:44 PM] ---- Deployment started. ----

[03:07:44 PM] Target platform is Standard Java EE.

[03:07:44 PM] Running dependency analysis...

[03:07:44 PM] Building...

[03:08:40 PM] Deploying profile...

[03:08:42 PM] Moving WEB-INF/adfc-config.xml to META-INF/adfc-config.xml

[03:08:49 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Liberty\lib-repo\commissions\CommissionsViewController-CommissionsViewController.jar

[03:08:49 PM] Elapsed time for deployment: 1 minute, 5 seconds

[03:08:49 PM] ---- Deployment finished. ----

[03:09:51 PM] ---- Deployment started. ----

[03:09:51 PM] Target platform is Standard Java EE.

[03:09:51 PM] Running dependency analysis...

[03:09:51 PM] Building...

[03:09:52 PM] Deploying profile...

[03:09:52 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Common\lib-repo\common\core-Resources-Resources.jar

[03:09:52 PM] Elapsed time for deployment: 1 second

[03:09:52 PM] ---- Deployment finished. ----

[03:10:08 PM] ---- Deployment started. ----

[03:10:08 PM] Target platform is Standard Java EE.

[03:10:08 PM] Running dependency analysis...

[03:10:08 PM] Building...

[03:11:14 PM] Deploying profile...

[03:11:14 PM] Moving WEB-INF/adfc-config.xml to META-INF/adfc-config.xml

[03:11:15 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Liberty\lib-repo\commissions\CommissionsBtf-LookupBtf.jar

[03:11:15 PM] Elapsed time for deployment: 1 minute, 6 seconds

[03:11:15 PM] ---- Deployment finished. ----

[03:15:23 PM] ---- Deployment started. ----

[03:15:23 PM] Target platform is Standard Java EE.

[03:15:23 PM] Running dependency analysis...

[03:15:23 PM] Building...

[03:16:11 PM] Deploying profile...

[03:16:11 PM] Moving WEB-INF/adfc-config.xml to META-INF/adfc-config.xml

[03:16:11 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Liberty\lib-repo\commissions\CommissionBtf-CommsBtf.jar

[03:16:11 PM] Elapsed time for deployment: 48 seconds

[03:16:11 PM] ---- Deployment finished. ----

[03:18:14 PM] ---- Deployment started. ----

[03:18:14 PM] Target platform is Standard Java EE.

[03:18:14 PM] Running dependency analysis...

[03:18:25 PM] Deploying profile...

[03:18:29 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Liberty\lib-repo\commissions\CommissionBtf-CommissionsModel.jar

[03:18:29 PM] Elapsed time for deployment: 15 seconds

[03:18:29 PM] ---- Deployment finished. ----

[03:18:53 PM] ---- Deployment started. ----

[03:18:53 PM] Target platform is Standard Java EE.

[03:18:53 PM] Running dependency analysis...

[03:19:01 PM] Deploying profile...

[03:19:02 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Liberty\lib-repo\commissions\CommissionsBtf-CommissionsLov.jar

[03:19:02 PM] Elapsed time for deployment: 9 seconds

[03:19:02 PM] ---- Deployment finished. ----

[03:19:57 PM] ---- Deployment started. ----

[03:19:57 PM] Target platform is Standard Java EE.

[03:19:57 PM] Running dependency analysis...

[03:20:47 PM] Deploying profile...

[03:20:47 PM] Moving WEB-INF/adfc-config.xml to META-INF/adfc-config.xml

[03:20:48 PM] Wrote Archive Module to C:\JDeveloper\mywork\repo\Liberty\lib-repo\commissions\CommissionBtf-BrokersBtf.jar

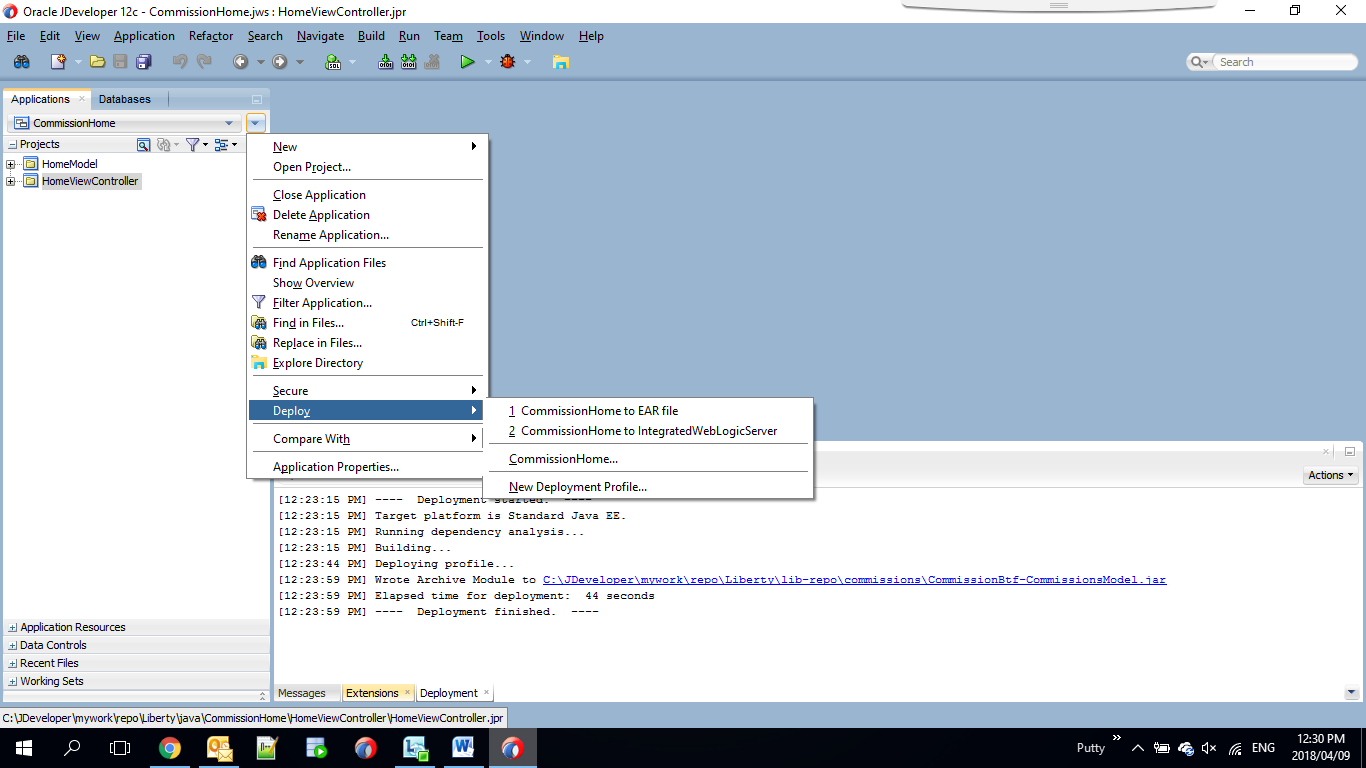
[03:20:48 PM] Elapsed time for deployment: 51 seconds

[03:20:48 PM] ---- Deployment finished. ----

</snip>

## Compile/Deploy Home application

In order to generate the .ear file open the CommissionHome application, select the deploy menu item and select the CommissionHome to EAR file.



Once the .ear file has been generated, you'll be notified of the location of the .ear file in the log window, as depicted below:

<snip> All JAR files deployed and EAR file also deployed

[03:21:23 PM] ---- Deployment started. ----

[03:21:23 PM] Target platform is (Weblogic 12.x).

[03:21:23 PM] Running dependency analysis...

[03:21:45 PM] Deploying 2 profiles...

[03:21:46 PM] Wrote Web Application Module to C:\JDeveloper\mywork\repo\Liberty\java\CommissionHome\HomeViewController\deploy\CommissionHome-HomeViewController-Shlib.war

[03:21:48 PM] Wrote Enterprise Application Module to C:\JDeveloper\mywork\repo\Liberty\java\CommissionHome\deploy\CommissionHome.ear

[03:21:48 PM] Elapsed time for deployment: 24 seconds

[03:21:48 PM] ---- Deployment finished. ----

</snip>

## Deploying to WebLogic

Currently deployments are manually but there has been some work done regarding the automation of this process. For more information speak to Robert.

## 1.4.1 Deploying to production

For production deployments, send the .EAR file to Robert and ask him to deploy. See section 1.3 on how to generate the .EAR file.

## 1.4.2 Deploying to non-production env

There are 4 non production env's to which developers needs to deploy to, they are:

DEV: <http://wlswctdevlz1:7001/console/login/LoginForm.jsp>

UAT: <http://wlswctuatlz1:7001/console/login/LoginForm.jsp>

TEST: <http://wlswctdevlz1:7003/console/login/LoginForm.jsp>

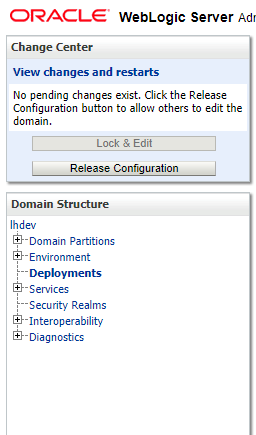
PRE-PROD: <http://wlswctuatlz1:7003/console/login/LoginForm.jsp>

The username and password for all non-production envs are:

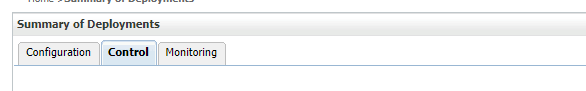
username: lhh\_deploy

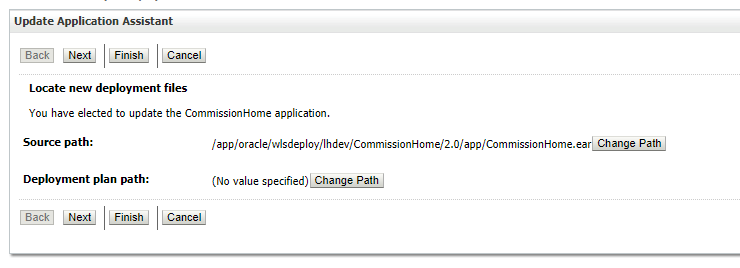
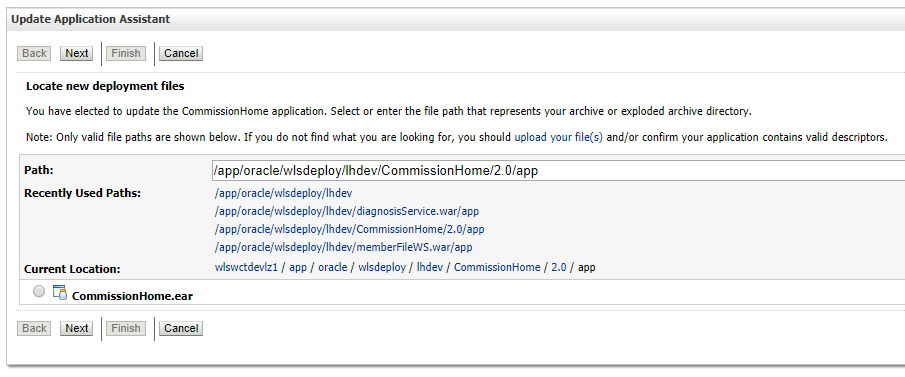
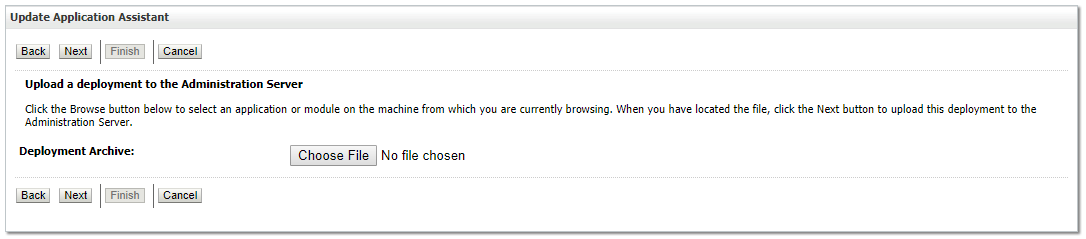
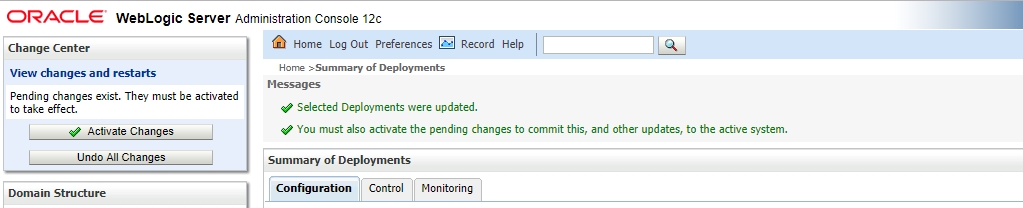
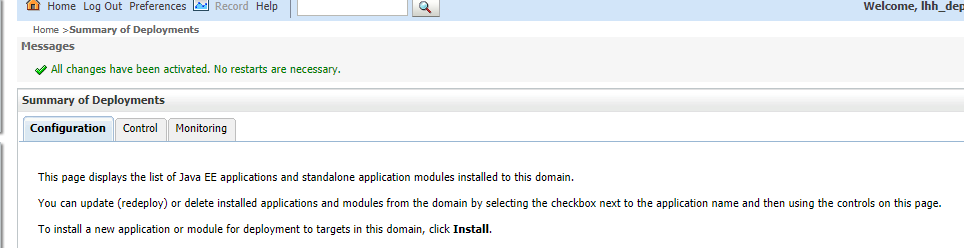
password: D3pl0y95!

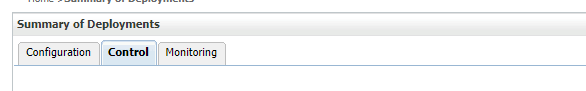
* Log onto the weblogic env you want to deploy the application, using the above details.
* Click "Lock and Edit" to lock the server for other users
* In the Domain Structure window, select Deployments



* Select the Control tab under Summary of Deployments



* Locate the Commissions Application and select the associated check box
* Stop the application by clicking on the Stop menu button and selecting the "Force stop now" option
* Once the application has stopped, select the configuration tab under Summary of Deployments
* Locate the Commissions Application and select the associated check box
* Click the Update button under the Deployments section. This will open the Update Application Assistant wizard.
* 
* Click the Change Path button under the Source Path option. Click the "upload your files(s) hyperlink.
* 
* Click the Choose file button. Navigate to where the .ear file is and select it.
* Click Next
* Click Next
* Click Finish
* Weblogic will navigate back to the Summary page. You need to activate the deployment.
* Click the "Active Changes" button as per below
* 
* Once completed you should receive confirmation as below. The application needs to be started again.
* 
* Select the Control tab under Summary of Deployments again



* Locate the Commissions Application and select the associated check box
* Start the application by clicking on the Start menu button and selecting the "Servicing all request"

## BitBucket - Promote code

The code has to be promoted through the different environments.

* The development branch needs to be deployed to the DEV environment
* The test-release branch needs to be deployed to TEST and UAT environments
* The master branch needs to be deployed to PRE-PROD and PROD.

In order to achieve this:

* Open SourceTree
* Double click the branch you would like to promote to. Ex. Double click test-release
* Right click the branch you want to promote from. Ex. Development
* Select the merge [from branch] into current branch
* Resolve any conflicts
* Check the changes in