1 Building, Deploying and Testing DPES application

This chapter provides updated instructions for accessing the sources code, developing, building and deploying the DPES application in the user environment. DPES development environment is split into two part 1. Backend DPES webserver – Maven Java Project and 2. DPES Frontend Adobe Flash UI project using flex 3 SDK. The following instruction covers both development environments.

1.1 Development Environment

This section is useful for developers who wants to develop the DPES tools by extending some features which are already exists in the project or create some additional new features based users need. Source codes are freely available to the developer community at TIMBUS open source repository[i]. It is assumed that the developer would have sufficient knowledge on Java, Maven, Eclipse, Version controls such as Git and eGit and flex development. Hence the scope of this document is restricted to detailing DPES application only rather than providing information for setting up the above said technologies.

1.1.1 DPES Webserver System Requirements

- Java Development Kit (JDK) (version 1.7_17 or above)[ii]
- Apache Maven plugin for Eclipse (version – 3.2.3)[iii]
- Eclipse IDE for Java EE Developers (version – Juno or later)[iv]
- Tomcat Web Servlet Container (version – 7 or above)[v]
- EGit plugin for eclipse[vi]
- H2 data base for DPES registry (version – 1.3.173)[vii]

1.1.2 DPES UI application System Requirements

Adobe Flex Development Tools

- Flex builder 3 standalone or plugin to eclipse
- Flex SDK 3.2

Supported Operating Systems[viii]

- Microsoft® Windows® 7/8 (32/64 bit), Windows Vista® (32 bit), or Windows XP Professional with Service Pack 2 or 3
- Mac OS X v10.5.6–10.6 (Intel® only)

Java Virtual Machine

- Java™ Virtual Machine (32 bit), IBM® JRE 1.5, IBM JRE 1.6, Sun™ JRE 1.5, Sun JRE 1.6

Supported Runtimes

- Adobe® Flash® Player 11 – Web browser plugin

- Adobe AIR® 3 – standalone UI application

1.2 Requirement for Deploying DPES application

The following software packages, services and resources are needed for successfully deploying and testing of DPES application. DPES is integrated with other TIMBUS tools such as PI Preservation Identifier, iERM/Risk Management tools, Ontologies and Context Model Instances, In order for DPES to function properly the following resources and services must be available on the service end point described in the API implementation section (reference-add all the ontology lo curls, PI service url).

1.2.1 TIMBUS Services requirements

- iERM Service or Risk Management Tools - to Invoke preservation action by sending Risk docs

- PI – Preservation Identifier Service to provide alternate preservation recommendation.

- Ontologies for business process, DIO, DSO, CUDF, etc.

- Context Model Instances for the specific business process that are to be preserved.

1.2.2 DPES Runtime Environment Requirement

- Java Run Time JRE-JVM (version - 1.7 or above)

- Apache Tomcat Server (version – 7 or above)

1.2.3 DPES Application Bundles

- DPES Webserver War bundles file

- DPES Flash UI Deployment package

1.2.4 DPES Infrastructures Services

- H2 Database embedded server

- Amazon AWS S3 account to configure storage credentials

- Network connectivity to access to TIMBUS Services and Ontology

- KVM+QEMU Virtualisation Server (PROXMOX)

1.3 Building of DPES Application, Installation and Testing

Follow the below instruction to download DPES source code, compile and deploy them into you own test environment.

1.3.1 Building of DPES UI Flex 3 Project

Check out project source using following git command from your preferred command line interface
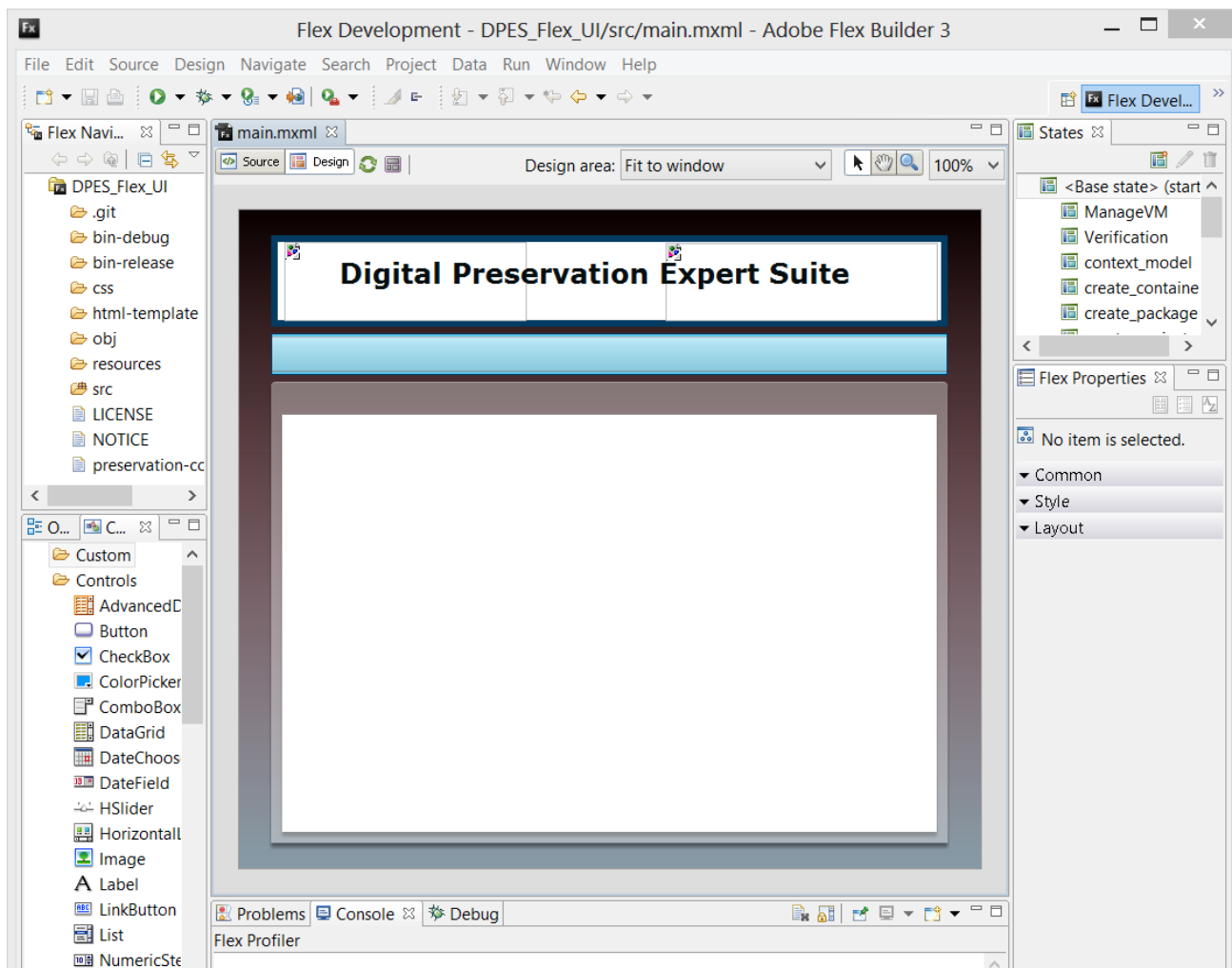
git clone https://@opensourceprojects.eu/git/p/timbus/dpes/preservation-console DPES_Flex_UI

Figure 1 : Flex Builder with DPES UI Project

Launch the flex builder or the Eclipse IDE with Flex builder plug-in installed. Import the project into the IDE from the folder (DPES_Flex_UI) location where project was cloned. Once the project is imported into the IDE, developers can be able to browse the code, make modification and do testing within the IDE. Flex development environment shown in **Error! Reference source not found.** displaying the main.mxml file in design view, flex builders provides graphical design environment to drag and drop the widget and tools to

build the rich user interface and the source code will be generated automatically that can be viewed by switching to source tab. Developers needs to make additional code changes at the source code to make new UI widgets to work. This particular UI development need programming skills in Action Script 3 and MXML development. The scope of the document is limited to provide detailed tutorials on above said technologies. However, Developers can learn more about flex development respective official website.

Once the project is loaded into the IDE, we needs to build the source code and export as release build for deploying at DPES server.
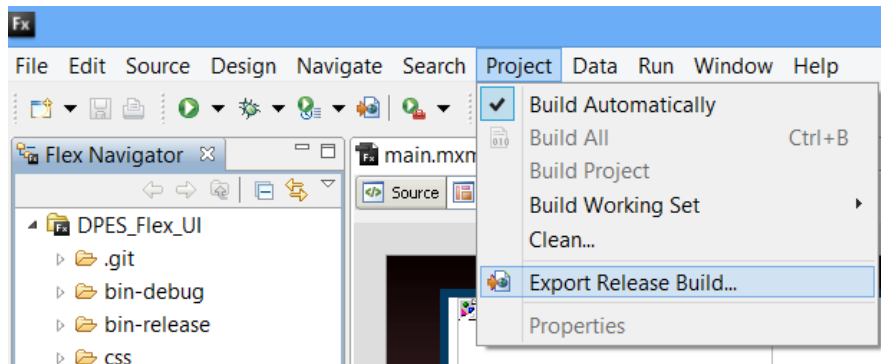
Figure 2 : DPES UI Project Export

In order export the release build of this UI project navigate to Project menu, enable build automatically check box and click Export Release Build option as shown in the screen capture **Error! Reference source not found.**. Follow the export release build dialog box as shown in the Figure 3 and specify main.mxml as main application and folder location where release build will be saved.
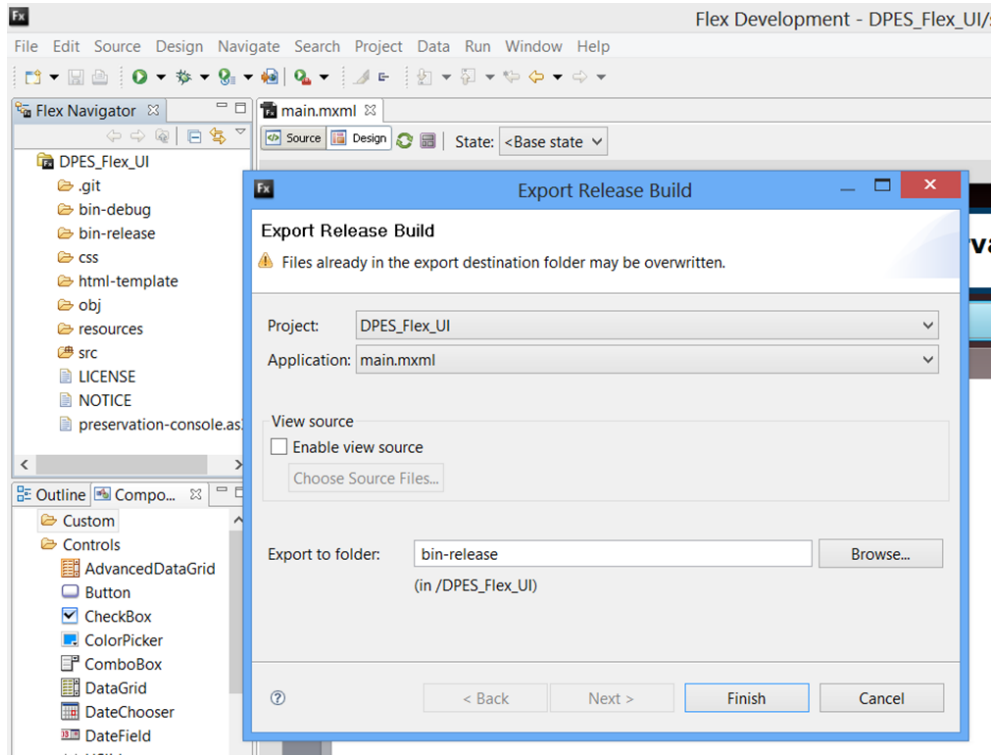
Figure 3 : Export DPES UI Release Build

1.3.2 Building of DPES Webserver application project

Check out the maven java project from TIMBUS open source repository into eclipse either by using git command line interface or using eGit plugin with Eclipse. The following git command shows cloning of project using command line interface.
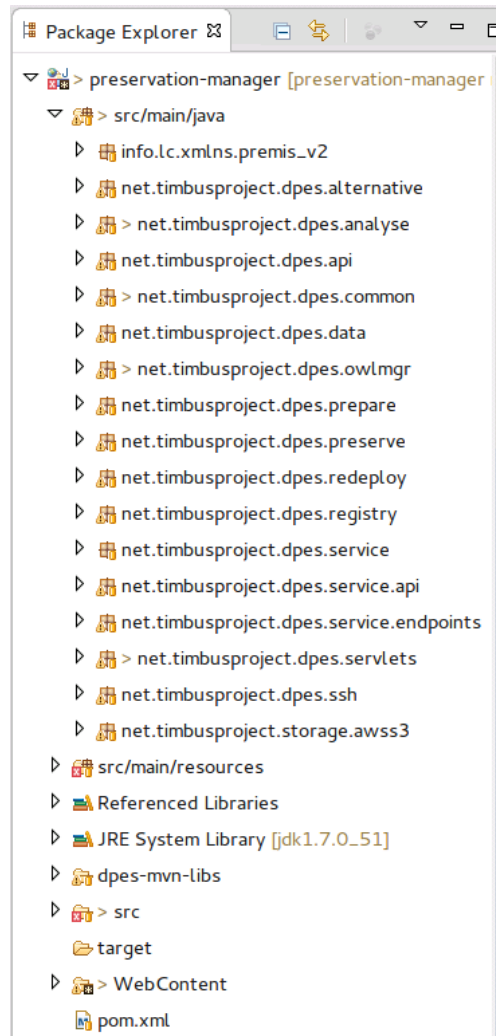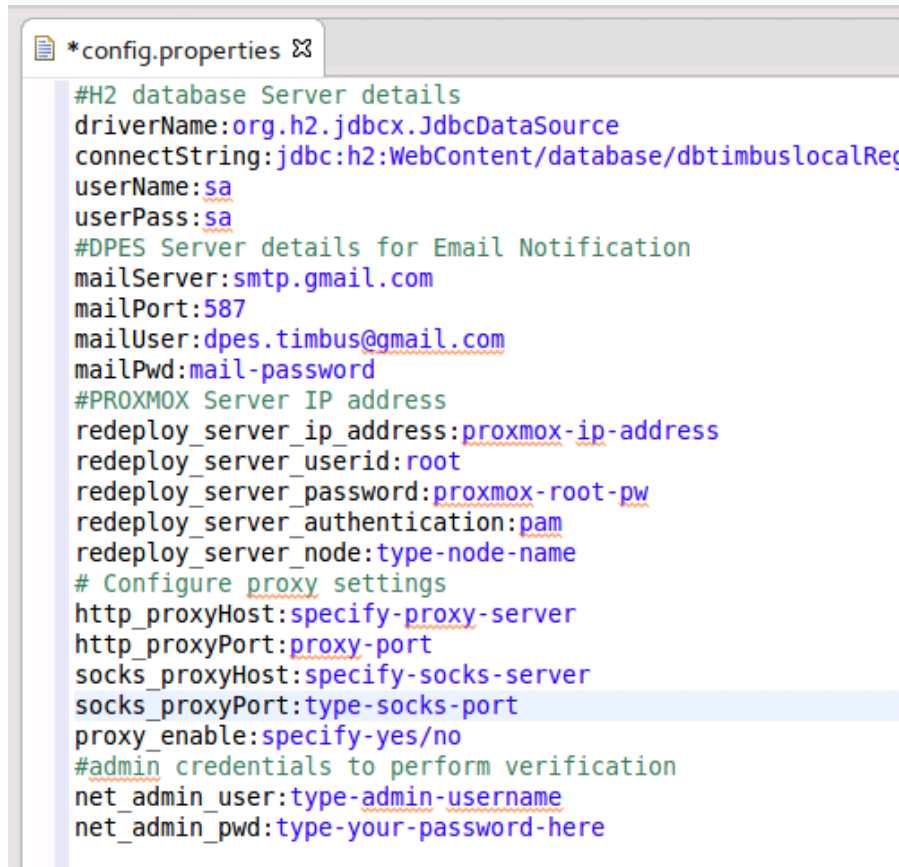
Git clone https://@opensourceprojects.eu/git/p/timbus/dpes/preservation-manager preservation-manager

Figure 4 : DPES Server Package lists from Eclipse

After cloning the complete source code developers can import the code into eclipse using project -> import option and selecting project type as maven. Eclipse will scan and automatically pick up the pom.xml file which is located at the root of the project as shown in the Figure 4, this figure also shows existing the package list in the eclipse package explorer window. There is a config.properties file located at **preservation-manager/src/main/resources/config.properties** location that needs to be updated to reflect your existing deployment environment. A screen capture shown in the Figure 5 details the list of parameters to be updated in the config.properties file. Almost all the values highlighted in the blue text have to be replaced by current live information to successfully run all the features of the DPES application.

Figure 5 : DPES Configuration Properties

Go to the command line interface, go to the project root directory and type the *mvn package* command to download all third party libraries and dependencies jars from the maven repositories and perform packaging. Finally from Eclipse, choose export and select type as *WAR* file and then type file name folder location to save the DPES web application file as shown in **Error! Reference source not found.**.

NOTE: Several alternate methods are available for all or some of the installation steps. For example any command executed through a GUI interface may also be possible to be executed through a command line interface. We have outlined above the methods popularly utilised by TIMBUS developers.
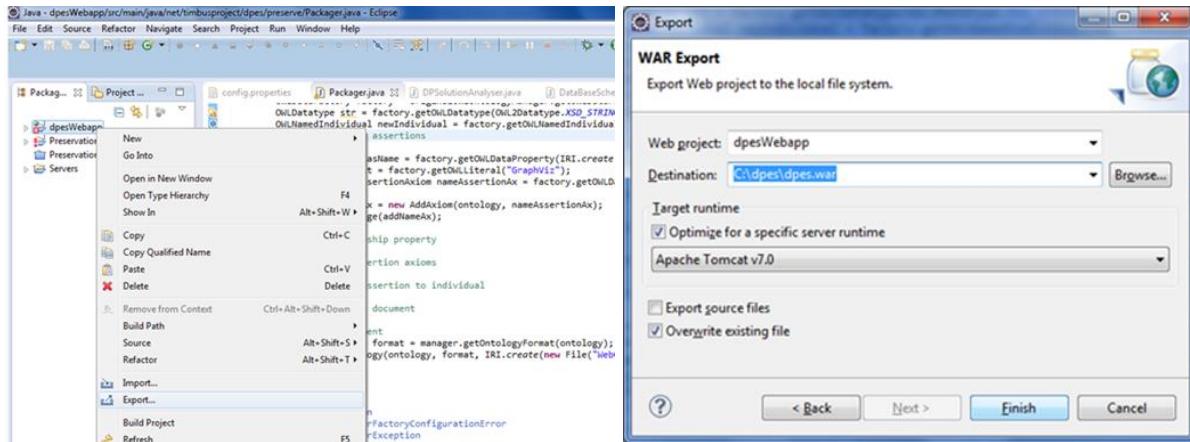
Figure 6 : Export DPES WAR file

1.3.3   Deploying Web application War file

To install the DPES web application into the Tomcat server, go the Tomcat admin URL (TIMBUS DPES Test Bed URL: http://134.191.240.68:16080) and click *manager app* button as shown by red arrow in the **Error! Reference source not found.** and then click choose file under Deploy ->WAR file to deploy.

Use the application tab from the Tomcat admin console to provision (stop, start, reload and undeploy) the DPES app and monitor the status. Use the following TIMBUS Test Bed URL to browse the application.
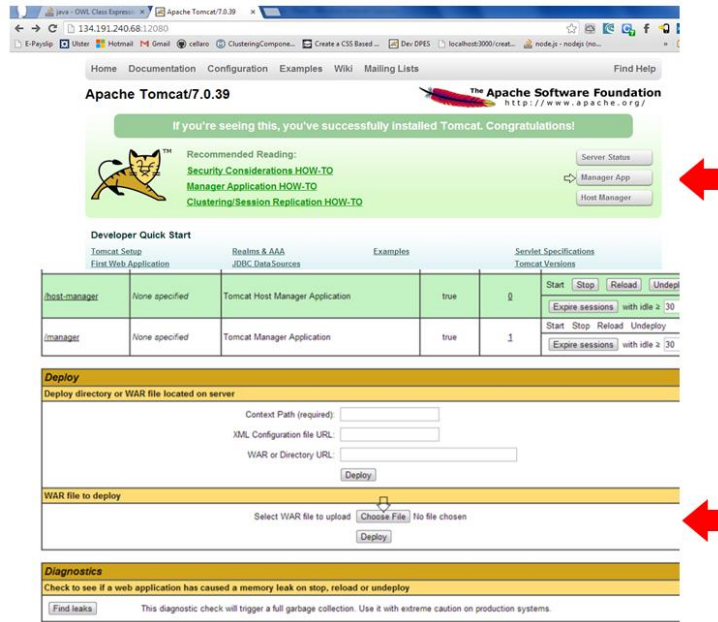
o   DPES Development site: http://134.191.240.68:16080/dpesui/

Figure 7 : Deploy DPES WAR file into Tomcat Server

1.3.4   Deploying DPES Flash UI to the Server

To deploy the DPES flash UI project copy the contents under bin-release folder and contents under resources from flex build IDE to tomcat directory …/webapps/dpesui/ and …/webapps/resources/ respectively. Directory locations are shown in the figure 8 and 8. Once the files are copies stop the tomcat server and restart, then goto the web browser anf type http://host-ip:8080/dpesui you will be served with dpes startup page.
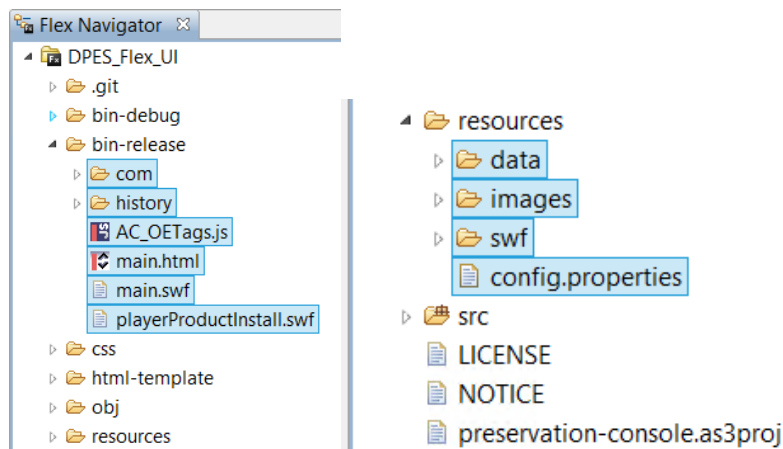

Figure 8 : Deploy DPES WAR file into Tomcat Server
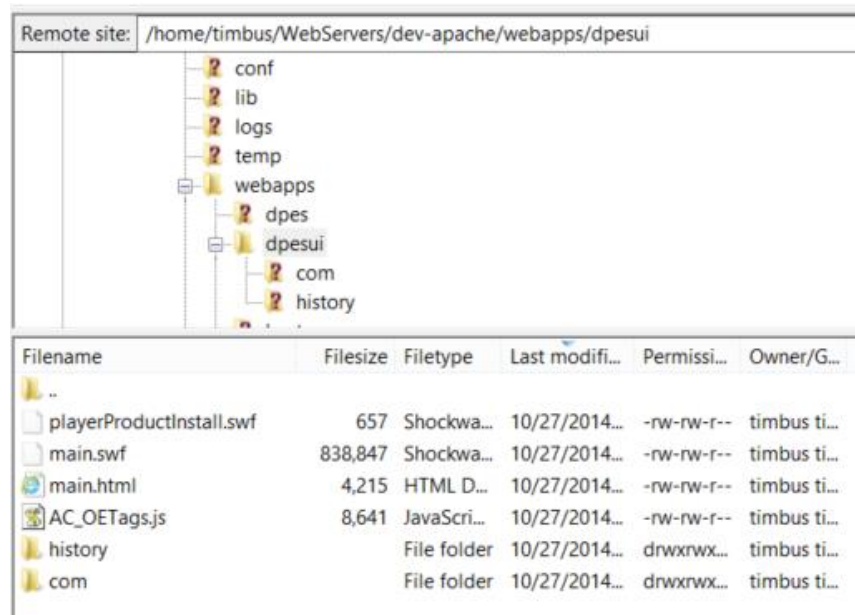
Server directory locations for flash UI deployment

Figure 9 : DPES UI Binary Deployment

i TIMBUS open source project repository https://opensourceprojects.eu/p/timbus/wiki/Home/

ii Java   Development   Kit   -   http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

iii Apache Maven Project - http://maven.apache.org/

iv Eclipse IDE download - https://www.eclipse.org/downloads/

v Tomcat Java Serverlet - http://tomcat.apache.org/

vi EGit Eclipse Team Provider - http://www.eclipse.org/egit/

vii H2 Relational database engine - http://www.h2database.com/html/main.html

viii Flex builder requirement - http://www.adobe.com/products/flex/tech-specs.html