Oracle FLEXCUBE Direct Banking

Android Workspace Configuration
Release 12.0.3.0.0

Part No. E52543-01

April 2014

ORACLE®

Android workspace Configuration

April 2014

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax:+91 22 6718 3001

Contents

1. Preface

1.1 Intended Audience

Any interested party working on the delivery of Oracle FLEXCUBE Direct Banking may read this document. The following profile of users would find this document useful:

- Application Architects
- End to End Designers
- Business Service Detailed Designers and Developers
- Implementation Partners

Specifically, however, this document is targeted at:

- **Implementation Partners, Customization Development Teams or Vendors** providing customization, configuration and implementation services around the Oracle FLEXCUBE Direct Banking product.

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

1.3 Access to OFSS Support

https://support.us.oracle.com

1.4 Structure

This manual is organized into the following chapters:

Section 1 *Preface* gives information on the intended audience. It also lists the various chapters covered in this User Manual.

Section 2-4 *Provides introduction and abbreviations details along with prerequisites.*

Section 5 *Provides information for creating workspace for Android Phone.*

Section 6 *Provides information for creating workspace for Android Tablet.*

1.5 Related Information Sources

For more information please, refer to the following documents:

- *Oracle_FLEXCUBE_Direct_Banking_Mobile_iPhone_Client_Developer_Guide*
- *Oracle_FLEXCUBE_Direct_Banking_Mobile_J2ME_Clients_Developer_Guide*
- *Oracle_FLEXCUBE_Direct_Banking_Mobile_Banking_User_Interface_Guide*

2. Introduction

This document provides the guidance to create and configure workspace for Android phones for FLEXCUBE Direct Banking 12.0.3.

Mobile banking is a secure channel to access the banking accounts & financial transactions similar to the Internet banking channel.

FLEXCUBE Direct Banking provides various client applications that can be distributed by the Bank to their end customers. These client applications are to be installed on the mobile device of end customer and enable the customers to access banking facilities over the mobile device.

The mobile client communicates with FLEXCUBE Direct Banking application server in order to execute the Inquiry or payment transaction. The user has to enter his/her login credentials in order to access the functionalities.

The request is in the form of a HTTP request to the FLEXCUBE Direct Banking application server. And the response from the server is in the form of XML over HTTP. This XML is parsed on the client and rendered into appropriate screen display.

The client applications use network to communicate with FLEXCUBE Direct Banking application server. Hence network data charges may be applicable to the end user based on his/her network provider. In order to reduce the payload on the mobile, application is designed to transfer minimum data over the network. The response XML contains single character node and attribute names. These names are taken from the real HTML tags, e.g. H for hidden fields, T for table etc. This XML is in a pre-defined format which is produced by the mLEAP framework of FLEXCUBE Direct Banking.

The details of mLEAP Framework are available in Oracle_FLEXCUBE_Direct_Banking_mLEAP_Framework_DeveloperGuide.docx.

For details on the supported XML tags please refer to document Oracle_FLEXCUBE_Direct_Banking_Mobile_App_API.docx.

FLEXCUBE Direct Banking mobile clients are designed to understand the XML generated by mLEAP and convert it to appropriate screen definition to be rendered on mobile device screen.

3. Acronyms and Abbreviations

FCDB	FLEXCUBE Direct Banking
AVD	Android Virtual Device
IDE	Integrated Development Environment
SDK	Software Development Kit
ADT	Android Development Tools
UI	User Interface
APK	Android Package

4. Prerequisites

For successful completion of this project the prerequisites are:
- Eclipse (3.7 or higher)
- Android ADT plugin (revision 22 or above)
- Android SDK with platforms for Android-8(Froyo), Android 11(Honeycomb) and Android 17(JellyBean latest version).
- For more details on how to setup the Eclipse plugin, refer: http://developer.android.com/sdk/installing/installing-adt.html

5. Configuring Workspace for Android Phone

1. Checkout the Android Workspace from the svn onto your local machine, the files that must be checked out are:

   - FCDB_12.0.3_Libraries.zip

   - FCDB_ANDROID_PHONE_12.0.3.zip

   The location of files in the svn is: workspace\mobile\android

2. Extract the zip files into a folder onto the local machine.

3. Open Eclipse with ADT plugin. For more information on how to setup Eclipse with ADT plugin, please refer: http://developer.android.com/sdk/installing/installing-adt.html

4. Go to File->Import
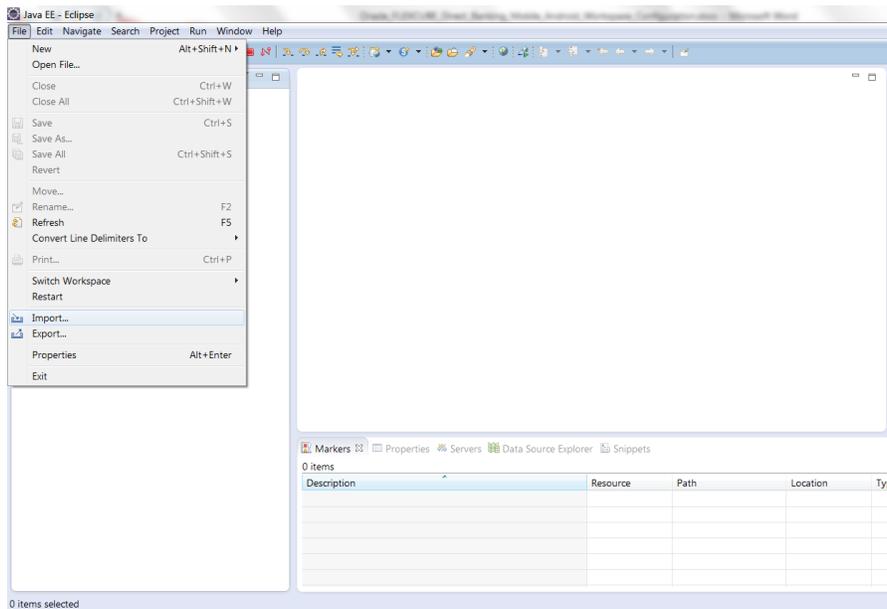
5. Select Android->Existing Project into workspace
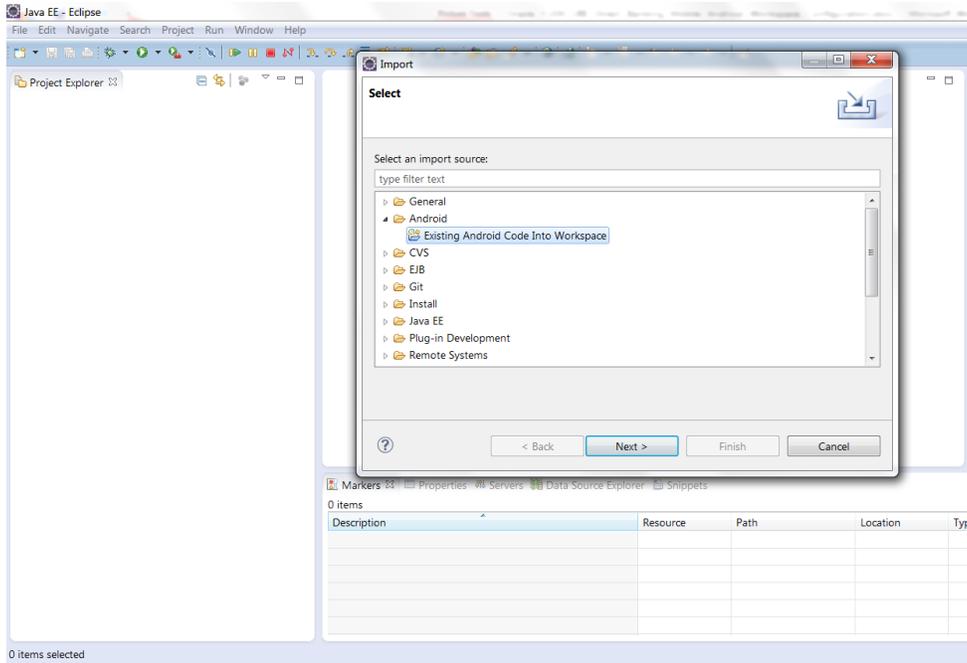
6. Browse and select the *FacebookSDK* folder from the location where the extracted files from the FCDB_12.0.3_Libraries.zip have been placed and click OK.
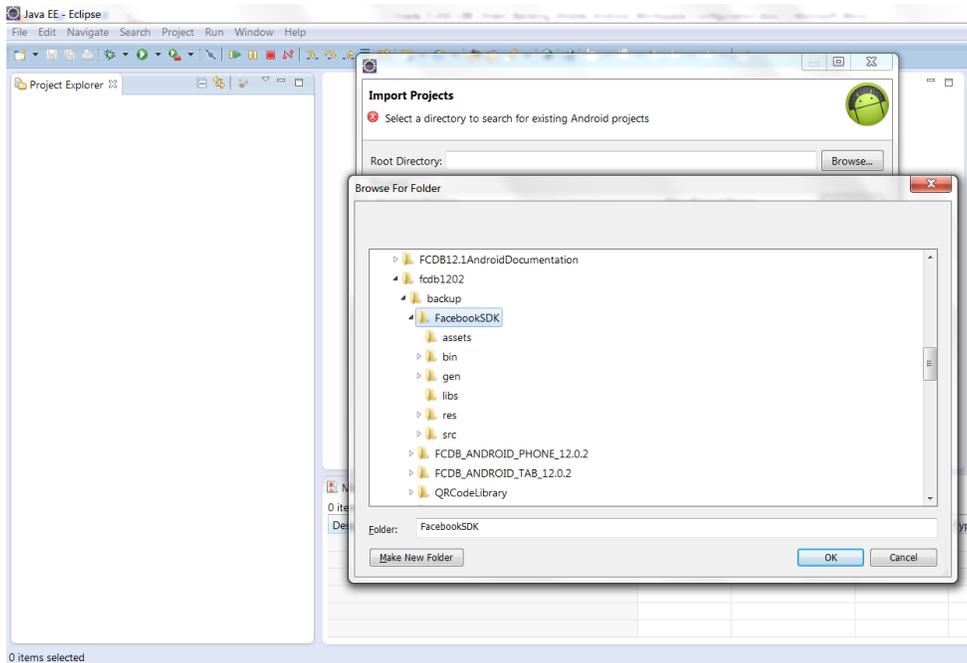
7. Press Finish.
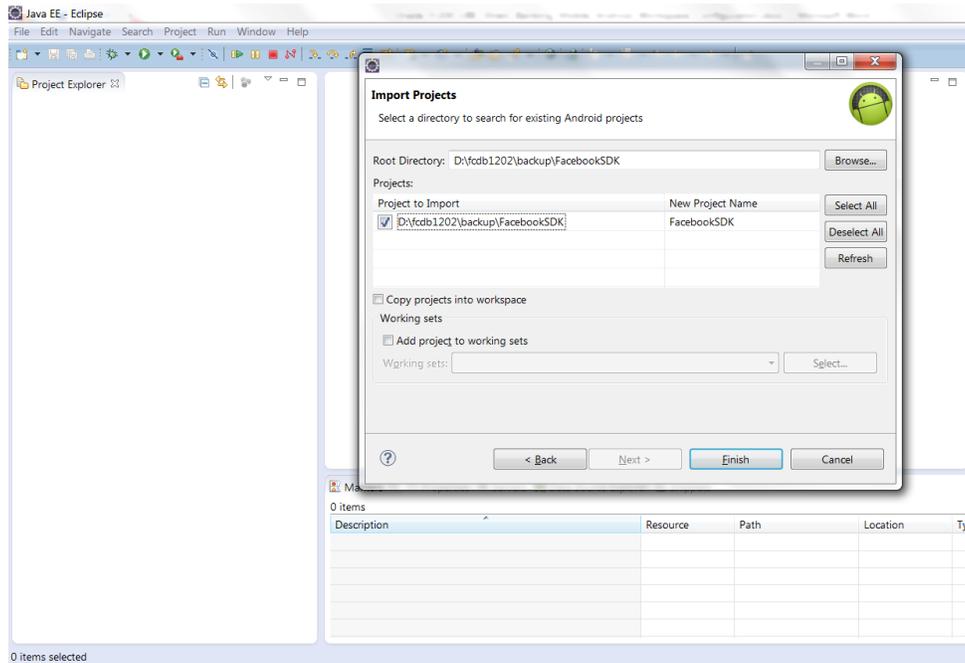
8. Repeat steps 5 – 7 and add four more projects from the Library, namely: **AndroidAugReality, google-play-services_lib, *RMSAndroidLibrary* and *QRCodeLibrary***. If the projects are imported into the workspace and have a different name, refractor the project and keep their names as highlighted above in bold.

9. To verify the successful import of the projects, expand all the projects. Open the file with the name "project.properties", it should contain a property "android.library=true".
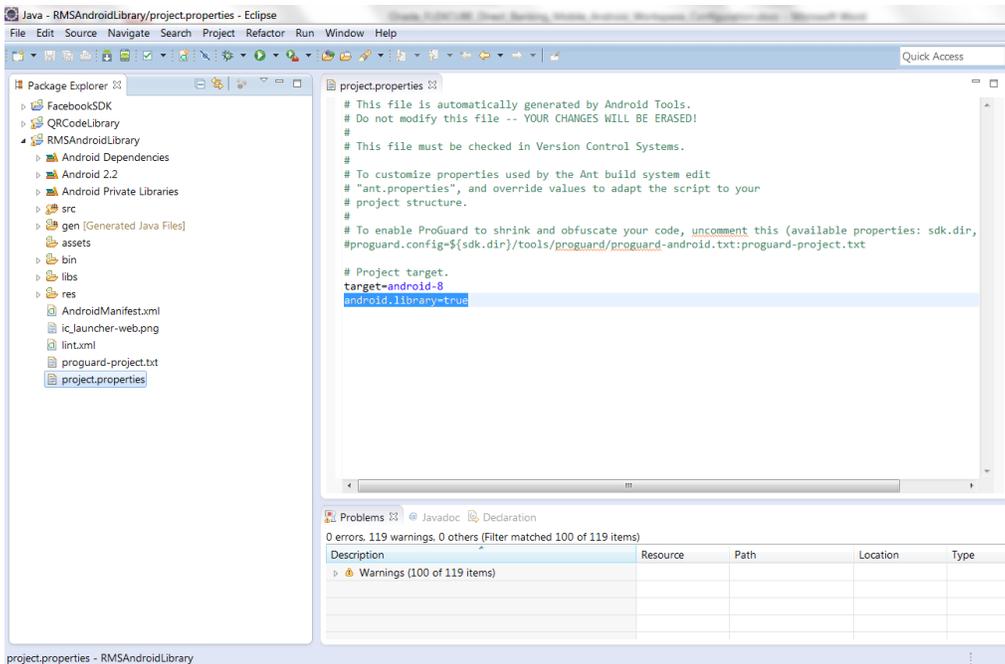
10. Repeat steps 5-7 and import the project FCDB_ANDROID_PHONE_12.0.3.

11. Verify the successful import of the project by expanding it in the project explorer. Open the project.properties file, it should contain three properties namely:

android.library.reference.1=..\\RMSAndroidLibrary

android.library.reference.2=..\\QRCodeLibrary

android.library.reference.3=../FacebookSDK

android.library.reference.4=..\\google-play-services_lib

android.library.reference.5=../AndroidAugReality

The sequencing may differ based on the Library project that was imported first. In case any of the three Libraries does not appear in the project.properties file:

- right-click the FCDB_ANDROID_PHONE_12.0.3 in the project explorer

- Goto -> Properties.

- Select Android from the left panel and scroll the right-panel to the bottom.

- Click the "Add" button in the library section
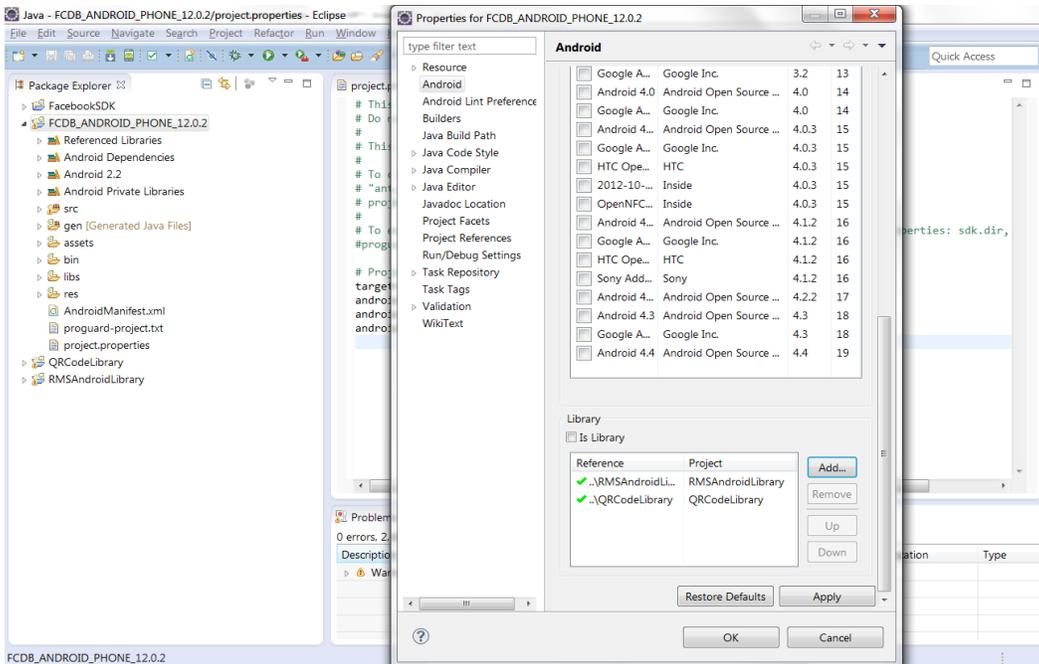
- Select the library project from the displayed options and press Ok. Click Apply and press Ok to apply the changes.

12. Android framework uniquely identifies different applications based on their package names. The default package name shipped with FCDB_ANDROID_PHONE_12.0.3 happens to be "com.ofss.fcdb.mobile.android.*kernel*.launcher". Refractor the package name and change the part "kernel" to the name of the bank i.e. "com.ofss.fcdb.mobile.android.*bankname*.launcher".

13. Open the AndroidManifest.xml from project explorer. The root tag <manifest> has an attribute "package", update the package name to the respective bankname and save the file.

14. After making the changes, certain number of files will show error in the <root>/res/layout, layout-large, layout-normal, layout-xlarge folders. The refactoring changes done above in step 12 need to be done here as well. In all the files showing error, the root xml node has an attribute: xmlns:app=http://schemas.android.com/apk/res/com.ofss.fcdb.mobile.android.phone. kernel.launcher. Update the attribute and replace the term "kernel" with the "bankname".

Below is the list of files that will show error and will need to be validated:

(Layout folder):

- activity_edit_favorites.xml

- branch_locator.xml

- editfavorites_listitems.xml

- footerpopuplistitems.xml

- foreground_dispatch_success.xml

- list_row.xml

- loginactivity.xml

- menu_footer_popup.xml

- options_popup.xml

- optionspopuplistitems.xml

- seekbar.xml

- sending_funds.xml

- simple_spinner_dropdown_item.xml

- simple_spinner_item.xml

- topmenu_layout.xml

- userdetail_layout.xml

- userwelcome_layout.xml

(Layout-large):

- foreground_dispatch_success.xml

- sending_funds.xml

(Layout-normal)

- foreground_dispatch_success.xml

- sending_funds.xml

(Layout-xlarge)

- foreground_dispatch_success.xml

- sending_funds.xml

15. Open the AppUrl.java file. It contains a variable by the name "APP_SERVER_LIST_URL". Enter the server url against this variable.

16. Create a relevant Emulator using the AVD Manager with Target Android 2.2 or above. The detailed method of creating an AVD can be found at the url: http://developer.android.com/tools/devices/managing-avds.html

17. Right-click the FCDB_ANDROID_PHONE_12.0.3 project, goto Run As-> Android Application to run it onto the AVD created.

.
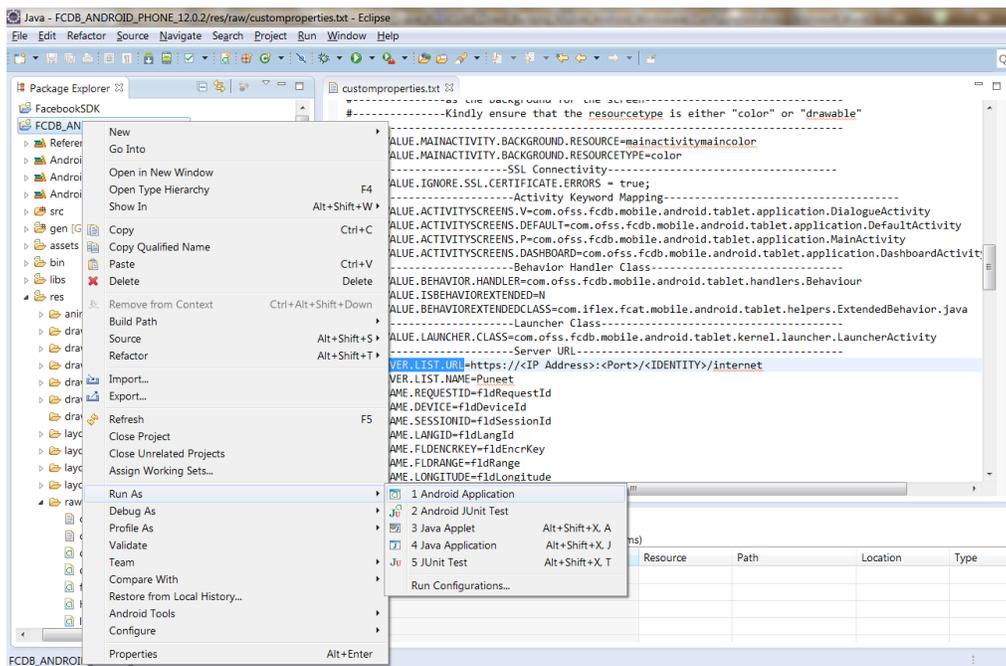
6. Configuring Workspace for Android Tablet

1. Setting up the project for Android Tablet is similar to setting up the project for Android Phone Project (See Section 5 of this document for more information).

2. Download the project FCDB_ANDROID_TAB_12.0.3.zip from svn workspace and proceed as mentioned for Phones.

3. If you have already setup the project for Android Phone, *you need not setup the Library projects again as they are common to both phone and tablets*.