Oracle FLEXCUBE Direct Banking

Android Tab Client Installation Guide
Release 12.0.3.0.0

Part No. E52543-01

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

www.oracle.com/financialservices/

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.
The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

1. Preface

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.

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.

Access to OFSS Support

https://support.us.oracle.com

2. Introduction

FLEXCUBE Direct Banking mobile client applications are supposed to be designed in order to understand this XML structure and render it on the mobile device screen appropriately. This stands as the basic requirement for Android Tab client as well. The Android Tab client is a client application specifically targeted for Google's Android OS platform. This document is a generic guide for development of Android Tab client.

FLEXCUBE Direct Banking's mLEAP framework is the entity responsible for generating the content for mobile clients. This content is represented in a pre-defined XML format. For details on the XML structure, please refer to the document Oracle_FLEXCUBE_Direct_Banking_Mobile_App_XML_structure.docx.

Some key points about developing an application on Android platform are highlighted below:

- Android applications are written in the Java programming language.
- The compiled Java code along with any data and resource files required by the application — is bundled by the Android Asset Packaging Tool into an Android package, an archive file marked by an .apk suffix.
- This file is the vehicle for distributing the application and installing it on mobile devices; it's the file users download to their devices.
- All the code in a single APK file is considered to be one application.
- The APK file is signed with a certificate before deployment.
- The client receives the response from the server in XML and parses it to render the screen.
- The screen rendering logic is written in client who majorly uses components of the Android SDK framework.

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. Scope

This document speaks about setting up the workspace for the development of ORACLE FLEXCUBE Direct Banking Android Tablet client development. This document further explains how to export the final APK file and install the same on android tablet.

5. Eclipse IDE Setup

5.1 ANDROID SDK ECLIPSE PLUGIN

Android Development Tools (ADT) is a plugin for the Eclipse IDE which is an integrated environment to build Android applications. Before you can install or use ADT, you must have a compatible version of Eclipse installed on your development computer. For this purpose, Eclipse 3.4 (Ganymede) or greater is required.

5.2 DOWNLOADING THE SDK STARTER PACKAGE

The SDK starter package is not a full development environment—it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform). The starter package can be obtained from:
http://developer.android.com/sdk/index.html

Make a note of the name and location of the SDK directory on your system—you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from command line.

5.3 INSTALLING THE ADT PLUGIN FOR ECLIPSE

Use Update Manager feature of your Eclipse installation to install the latest revision of ADT on your development computer. The ADT Plugin URL location is:
https://dl-ssl.google.com/android/eclipse/

Note: If you have trouble acquiring the plugin, try using "http" in the Location URL, instead of "https" (https is preferred for security reasons).

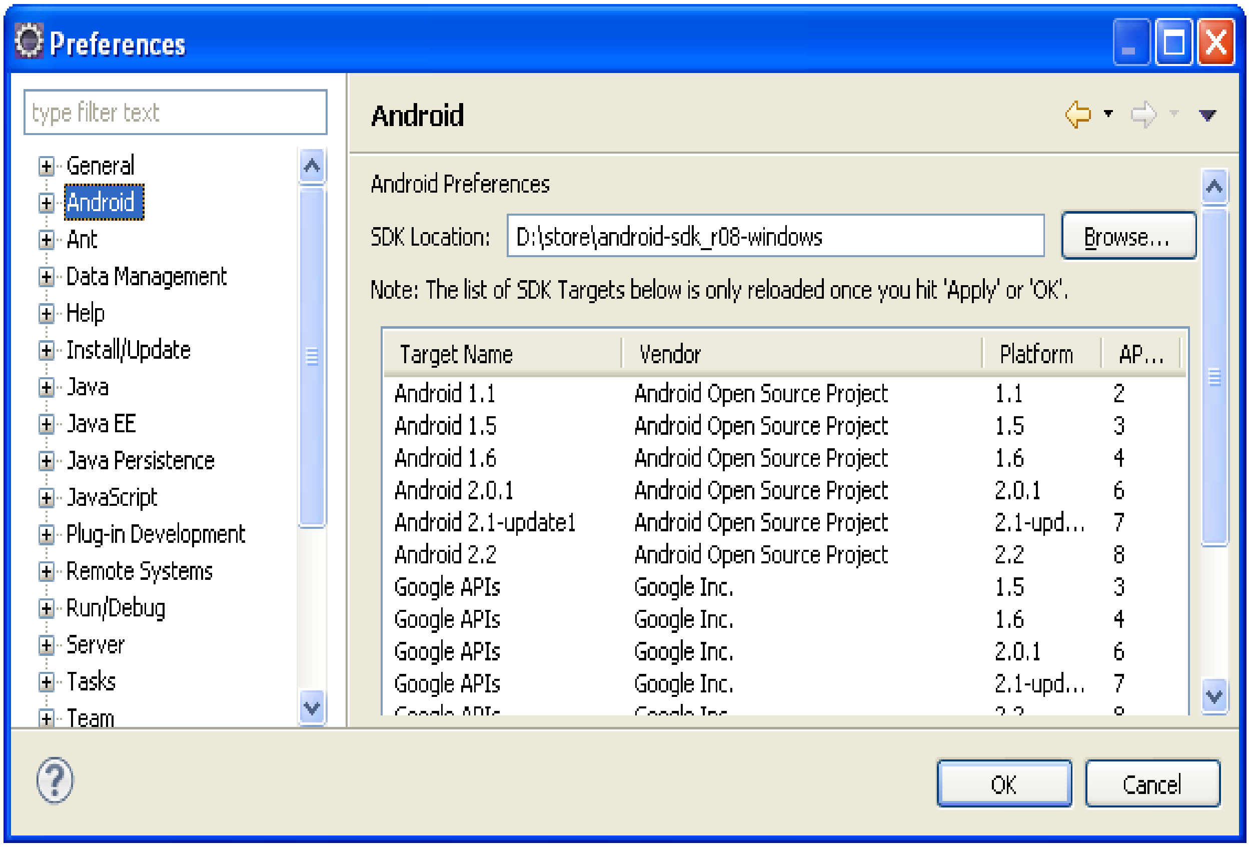
If you are having trouble downloading the ADT plugin because of the network firewall, you can configure proxy information from the main Eclipse menu:
Window > Preferences > General > Network Connections

For further information kindly use the following link:
http://developer.android.com/sdk/eclipse-adt.html#installing

5.4 CONFIGURING THE ADT PLUGIN IN ECLIPSE

Once you've successfully downloaded ADT as described above, the next step is to modify your ADT preferences in Eclipse to point to the Android SDK directory:

- In Eclipse menu bar, select **Window** > **Preferences** to open the Preferences panel.
- Select Android from the left panel.
- For the SDK Location in the main panel, click the Browse button and locate your downloaded SDK directory.
- Click the OK button.
- Eclipse restart is required.



Android Preferences

SDK Location: D:\store\android-sdk_r08-windows

Note: The list of SDK Targets below is only reloaded once you hit 'Apply' or 'OK'.

Target Name	Vendor	Platform	AP...
Android 1.1	Android Open Source Project	1.1	2
Android 1.5	Android Open Source Project	1.5	3
Android 1.6	Android Open Source Project	1.6	4
Android 2.0.1	Android Open Source Project	2.0.1	6
Android 2.1-update1	Android Open Source Project	2.1-upd...	7
Android 2.2	Android Open Source Project	2.2	8
Google APIs	Google Inc.	1.5	3
Google APIs	Google Inc.	1.6	4
Google APIs	Google Inc.	2.0.1	6
Google APIs	Google Inc.	2.1-upd...	7

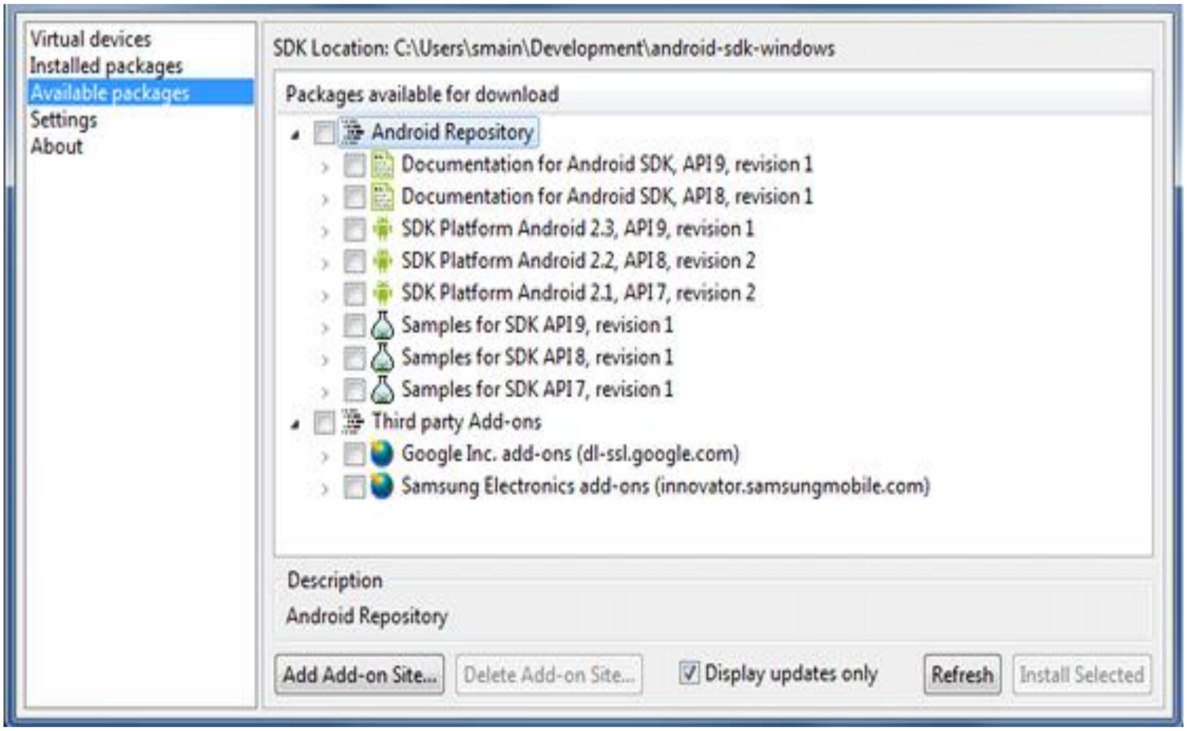
5.5 ADDING PLATFORMS AND OTHER IN ECLIPSE

The last step in setting up your SDK is using the **Android SDK and AVD Manager** (a tool included in the SDK starter package) to download essential SDK components into your development environment.

You can start the Android SDK and AVD Manager in one of the following ways:

- From within Eclipse menu, select **Window > Android SDK and AVD Manager**.
- Run **android.bat** in **<Android SDK>\tools** folder.

To download components, use the graphical UI of the Android SDK and AVD Manager to browse the SDK repository and select new or updated components.



SDK Location: C:\Users\smain\Development\android-sdk-windows

Packages available for download

- □ Android Repository
  - □ Documentation for Android SDK, API 9, revision 1
  - □ Documentation for Android SDK, API 8, revision 1
  - □ SDK Platform Android 2.3, API 9, revision 1
  - □ SDK Platform Android 2.2, API 8, revision 2
  - □ SDK Platform Android 2.1, API 7, revision 2
  - □ Samples for SDK API 9, revision 1
  - □ Samples for SDK API 8, revision 1
  - □ Samples for SDK API 7, revision 1
- □ Third party Add-ons
  - □ Google Inc. add-ons (dl-ssl.google.com)
  - □ Samsung Electronics add-ons (innovator.samsungmobile.com)

Description

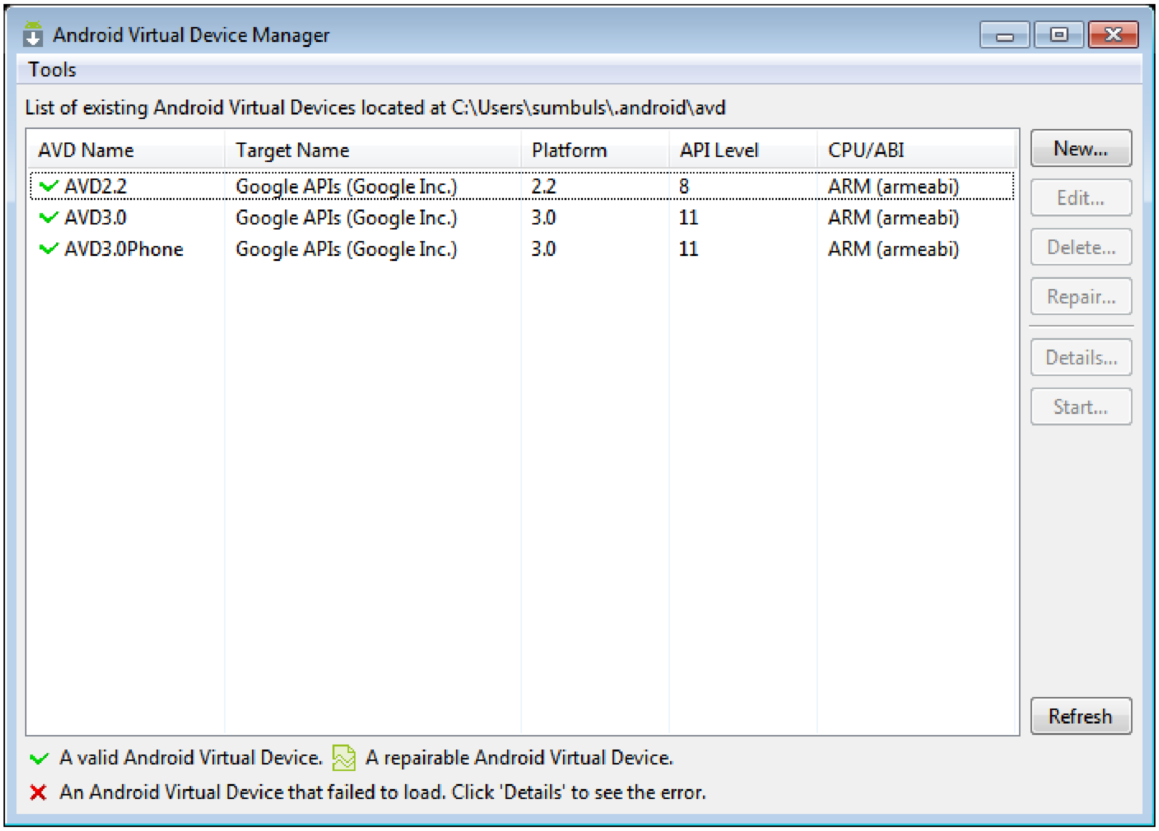
Android Repository

☑ Display updates only

Note: Kindly note that the **Android Repository** node must have an API level 3 and above.
Also, **Third party Add-ons** node should have a Google Inc. component, as this will be required for Google Map support.
If either of the above is missing, then it must be downloaded using **section 1.3** of this document.

5.6 CREATING A NEW AVD

- Goto the “tools” folder in Android SDK
  e.g.   D:\store\android-sdk_r08-windows\tools
- Run “android.bat”.
- “Android SDK and AVD Manager” will open in a new window



List of existing Android Virtual Devices located at C:\Users\sumbuls\.android\avd

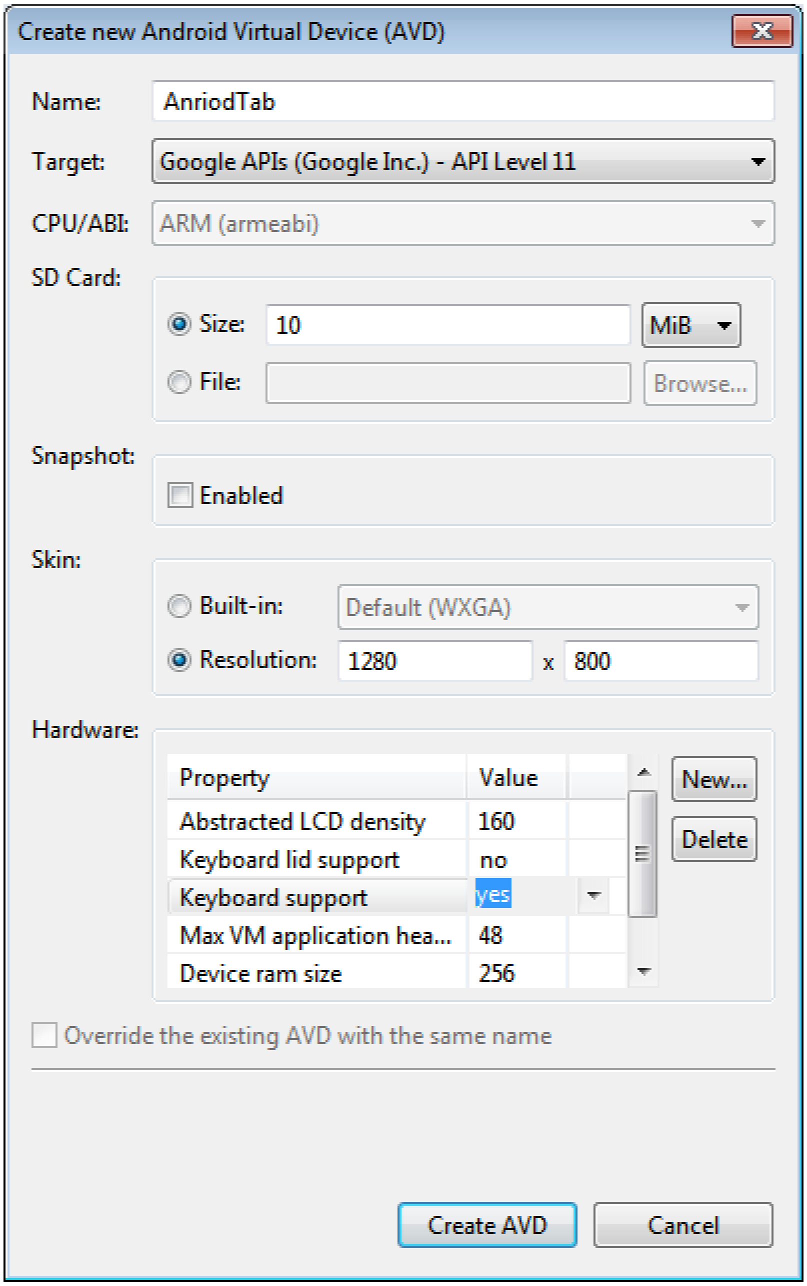
AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ AVD2.2	Google APIs (Google Inc.)	2.2	8	ARM (armeabi)
✓ AVD3.0	Google APIs (Google Inc.)	3.0	11	ARM (armeabi)
✓ AVD3.0Phone	Google APIs (Google Inc.)	3.0	11	ARM (armeabi)

✓ A valid Android Virtual Device. A repairable Android Virtual Device.

✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.

- From the panel on the left side, select “Virtual Devices”.
- A list of previously created AVDs will be displayed.
- From the right side of the window, click on “New” button.
- A new window will appear.
- Enter the name for the new AVD, e.g. “SampleAVD”.
- Select the “Target” as “Google APIs (Google Inc.) – API Level 11”

  Note: Kindly note that the API level can be 3 and above, but the API should be of Google and not any other, as this will be required for Google Map support



Name: AnriodTab

Target: Google APIs (Google Inc.) - API Level 11

CPU/ABI: ARM (armeabi)

SD Card:

● Size: 10 MiB

○ File:

Snapshot:

□ Enabled

Skin:

○ Built-in: Default (WXGA)

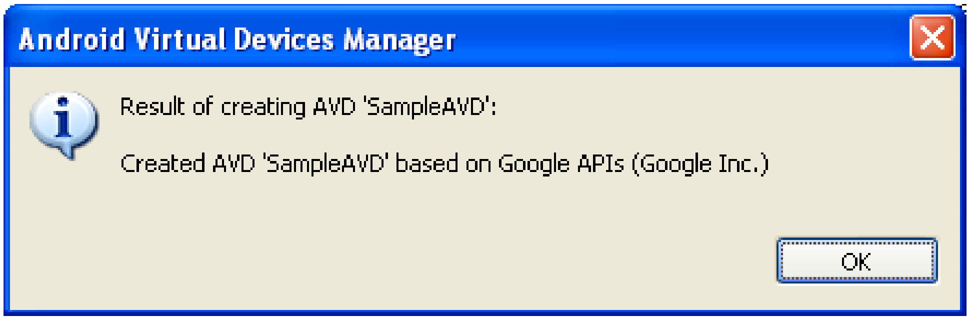
● Resolution: 1280 x 800

Hardware:

Property	Value
Abstracted LCD density	160
Keyboard lid support	no
Keyboard support	yes
Max VM application hea...	48
Device ram size	256

□ Override the existing AVD with the same name

- Click on “Create AVD” button. A message box will be shown confirming the successful creation of the AVD.



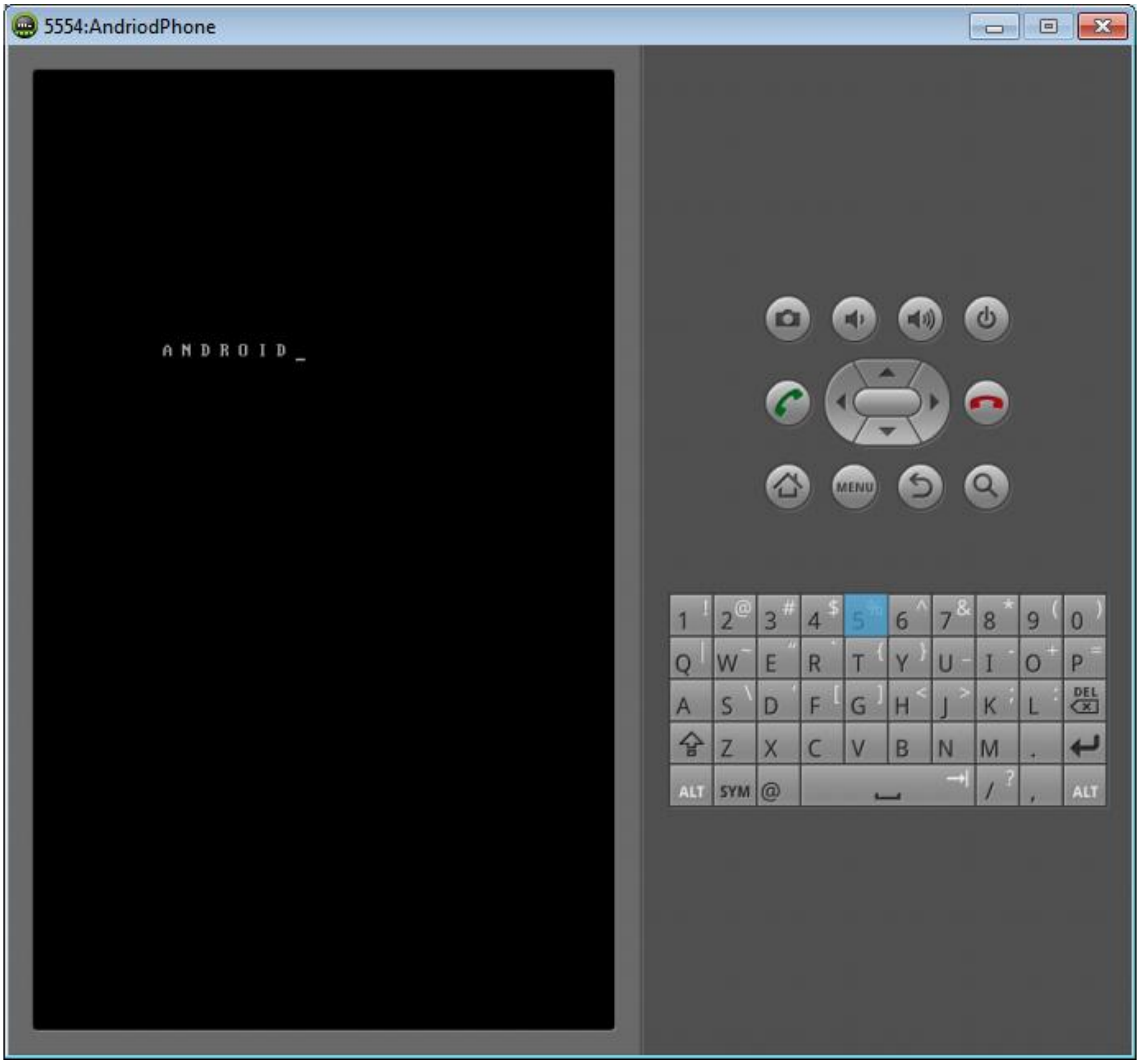
Result of creating AVD 'SampleAVD':

Created AVD 'SampleAVD' based on Google APIs (Google Inc.)

- Close the “Android SDK and AVD Manager”.

5.7 RUNNING AVD

- Open a command prompt and goto “tools” folder in Android SDK
  e.g. D:\store\android-sdk_r08-windows\tools
- Run the following command “**emulator -avd SampleAVD”**
  where “SampleAVD” is the name of the AVD created.
- This will start the AVD:



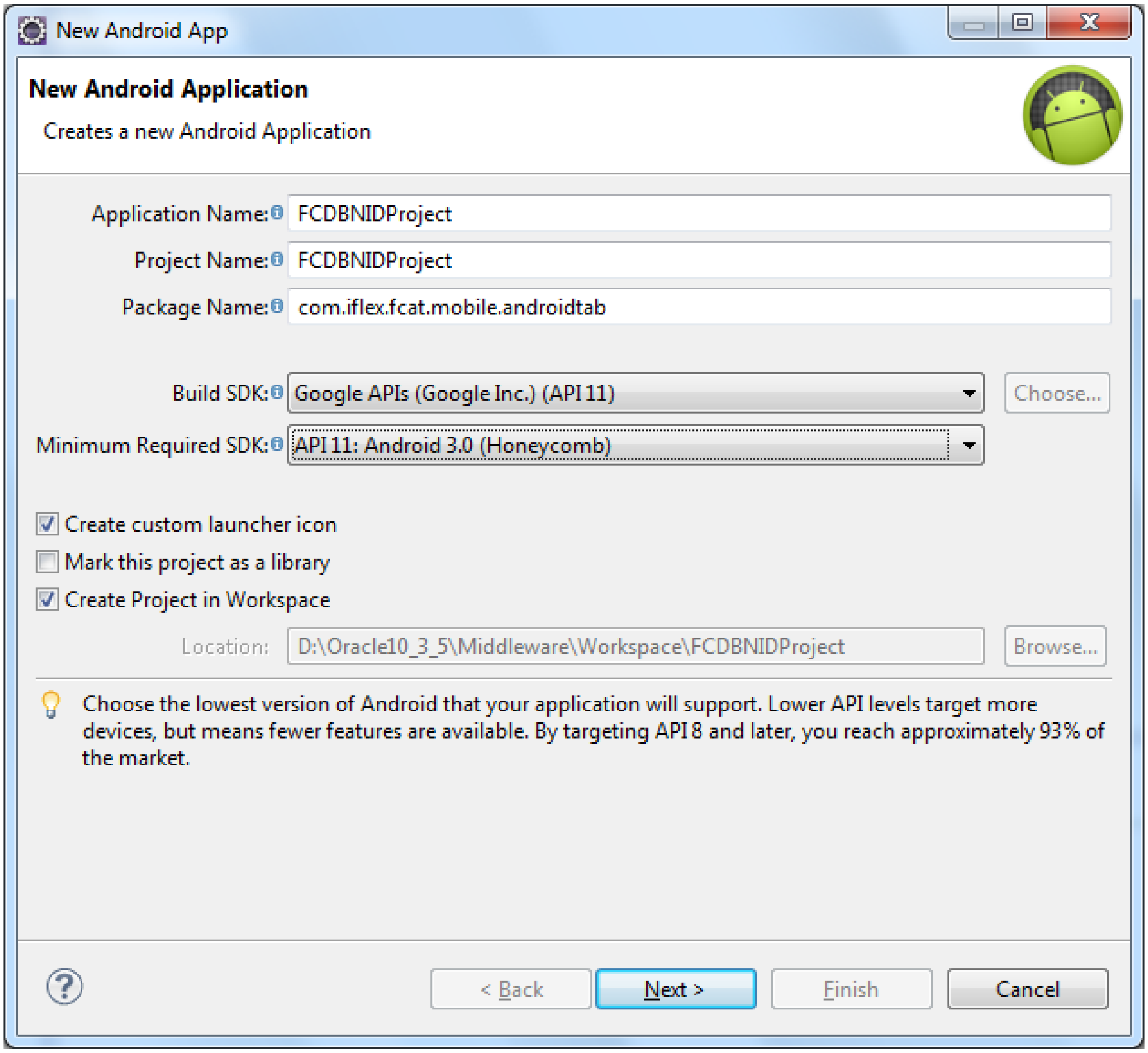
5.8 INSTALL APPLICATION ON AVD

- Open AVD and run the emulator.
- Open the browser application in the virtual phone.
- Access the URL where **.apk** file is present – for example access http://10.180.58.229:9000/download.html URL from the browser
- Click on the APK and it should get installed on the AVD.

5 Importing FCDB Android Tablet client jar into workspace

5.1. CREATING A PROJECT IN ECLIPSE

- Create a new android application in eclipse.



New Android Application

Creates a new Android Application

Application Name: FCDBNIDProject

Project Name: FCDBNIDProject

Package Name: com.iflex.fcat.mobile.androidtab

Build SDK: Google APIs (Google Inc.) (API 11)

Minimum Required SDK: API 11: Android 3.0 (Honeycomb)

☑ Create custom launcher icon

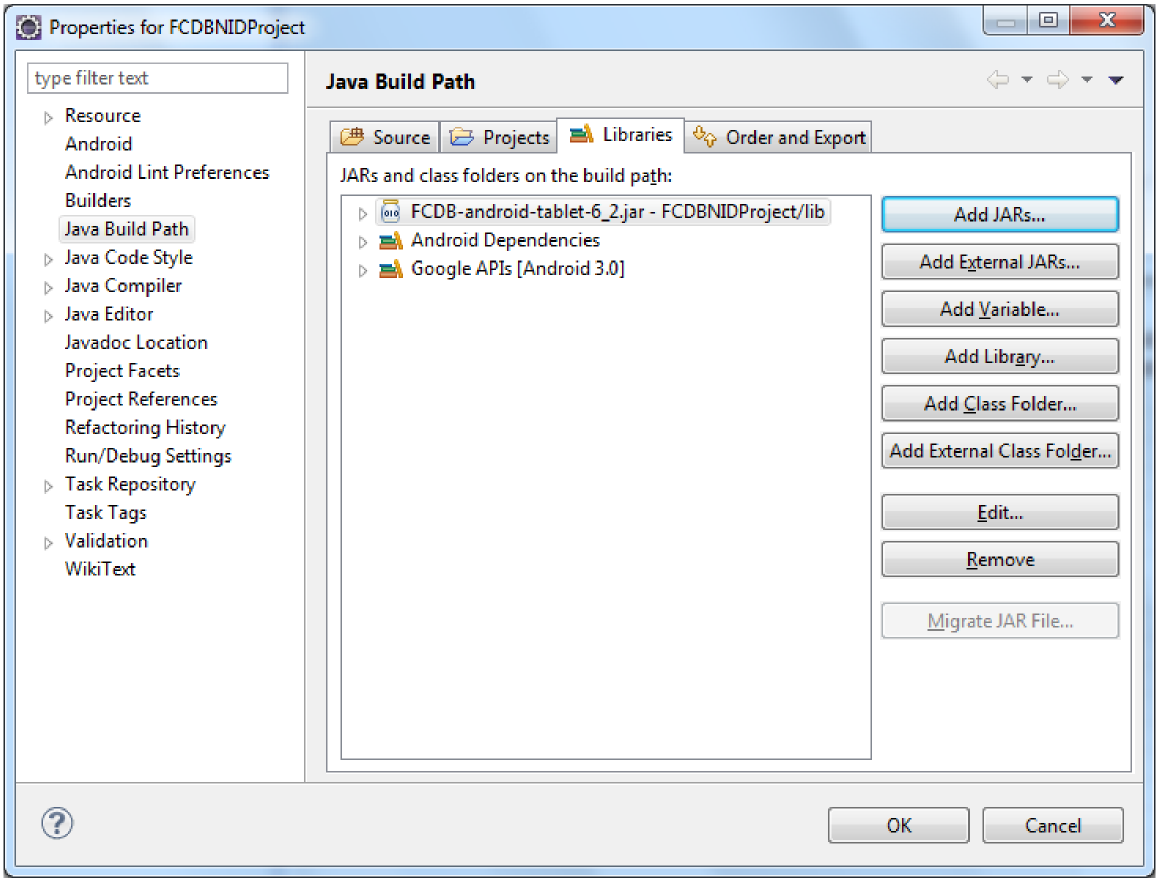
□ Mark this project as a library

☑ Create Project in Workspace

Location: D:\Oracle10_3_5\Middleware\Workspace\FCDBNIDProject

Choose the lowest version of Android that your application will support. Lower API levels target more devices, but means fewer features are available. By targeting API 8 and later, you reach approximately 93% of the market.

- Import following files into the workspace at below locations –
  - Lib\FCDB-android-tablet-6_2.jar
  - AndroidManifest.xml
  - Res folder
- Add the FCDB-android-tablet-6_2.jar into the Java build path.



JARs and class folders on the build path:

- FCDB-android-tablet-6_2.jar - FCDBNIDProject/lib
- Android Dependencies
- Google APIs [Android 3.0]

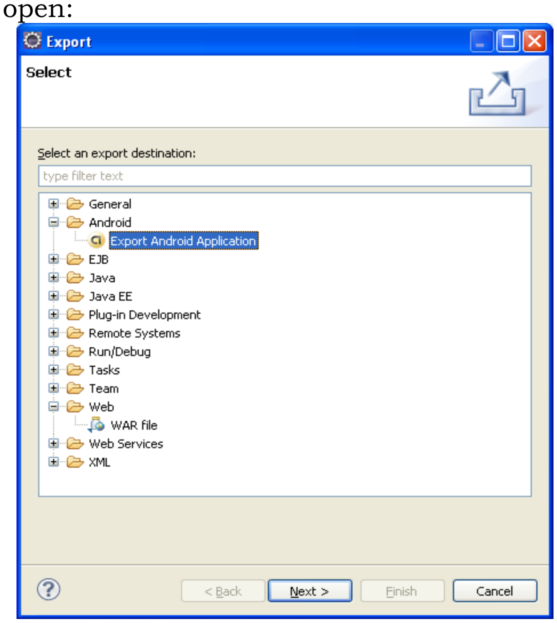
- Also add it in the Order and Export tab.
- Build and Launch the application.

In NID mode you can add your contribution classes, activities into the client or make customizations as allowed. Make sure that the classes would be overridden in the final APK.

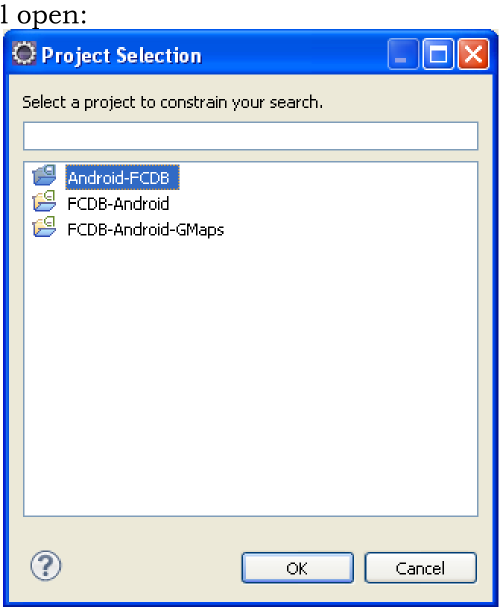
6 Exporting Final Output APK File

6.1 EXPORT WIZARD

- In Eclipse menu, go to **File** > **Export**.
- A new window will open:

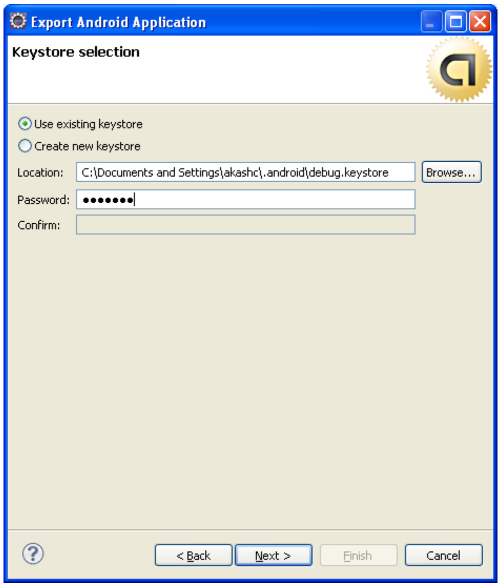


- Select **Android** > **Export Android Application**.
- Click on 'Next >'.
- In the next screen, click on 'Browse'.
- A new window will open:



- Select your project and click on 'OK'.
- Click 'Next' in the parent window.
- Select 'Use existing keystore'.
- Give the location of your keystore and the keystore password.

- You can create a new keystore by selectin ‘Create new keystore’.
- For development purpose, you can also use the default keystore. The default keystore password is ‘android’.


Export Android Application

Keystore selection

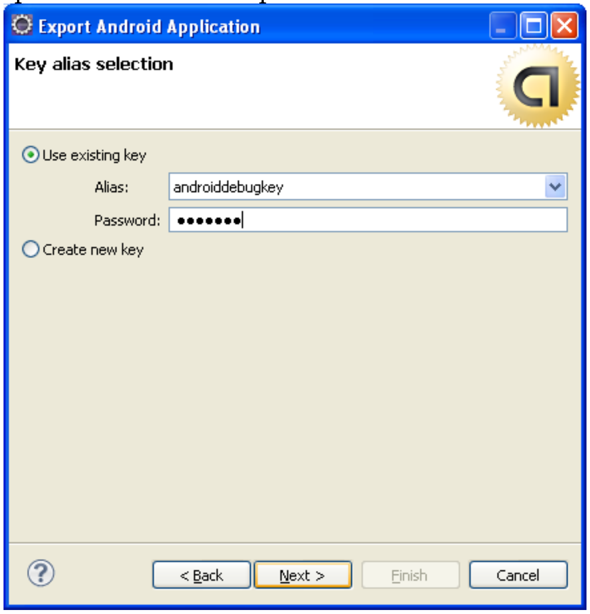
◉ Use existing keystore
○ Create new keystore

Location: C:\Documents and Settings\akashc\.android\debug.keystore

Password: •••••••

Confirm:

- Click on ‘Next >’.
- Select ‘Use existing key’.
- Select your Alias from the dropdown. Default alias for default keystore is ‘androiddebugkey’.
- Enter the alias password. Default password for ‘androiddebugkey’ is ‘android’.


Export Android Application

Key alias selection

◉ Use existing key

Alias: androiddebugkey

Password: •••••••

○ Create new key

- Click on ‘Next’.
- Give the complete path of the destination APK file. E.g.
  D:\store\Android-FCDB\output\FCDB_ANDROID_6.2.1.0.0.apk

- Click on ‘Finish’.
- The wizard will create the above mentioned APK.

Reference Documents

Sr.No	*Name of Document*
1	*Oracle_FLEXCUBE_Direct_Banking_Mobile_J2ME_Clients_Developer_Guide*
2	*Oracle_FLEXCUBE_Direct_Banking_Mobile_iPhone_Client_Developer_Guide*
3	*Oracle_FLEXCUBE_Direct_Banking_Mobile_App_XML_structure*
4	*Oracle_FLEXCUBE_Direct_Banking_Mobile_Banking_User_Interface_Guide*
5	*Oracle_FLEXCUBE_Direct_Banking_Parameter_Sheet.xls*