App Framework Android Integration

Document History				
Version	**Date**	**Author**	**Section**	**Changes**
0.1	09-Aug-2016	Ritesh Jha, Rakesh Krishamurthy	All	Initial draft

Author	Ritesh Jha, Rakesh Krishnamurthy
Approved By	Depth Shivakumar
Email Id	Rakesh.krishnamurthy@philips.com, ritesh.jha@philips.com

Contents

1. Introduction

This document will provide step by step procedure to integrate various horizontal components within **AppFramework** application.

Source code Links:

http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/simple/libs-release-localandroid/com/philips/cdp/appFramework/0.1.0

BigBucket Link:

https://atlas.natlab.research.philips.com/bitbucket/projects/MAF/repos/appframework_android/browse

2. Integration

Integration can be done in following ways.

2.1 Artifactory

All dependent libraries should be downloaded from artifactory.

Artifactory path:

```
http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/simple/libs-
releaselocal-android/com/philips/cdp/appFramework/
```

2.2 Root gradle changes

```
buildscript {
    repositories {

        maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }

    }
    dependencies {         classpath
'com.android.tools.build:gradle:2.1.0'          }
}
allprojects {
repositories {
        maven                    {                    url                    'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/libsrelease-local-android' }
        maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }
maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/extrelease-local'}
        maven                    {                    url                    'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/libsrelease-local-android' }
        maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libsstage-
local-android'}

    }}
```

2.3 Dependencies

2.3.1 Gradle dependencies

```
dependencies {
    compile 'com.android.support:support-annotations:23.2.0'     compile
'com.android.support:multidex:1.0.1'     compile
'com.android.support:design:23.1.1'

    compile(group: 'com.philips.cdp', name: 'registrationApi', version: '7.0.1', ext: 'aar') {         exclude
group: 'com.android.support'         transitive = true
    }

    compile(group: 'com.philips.cdp', name: 'uikitLib', version: '3.2.3', ext: 'aar') {         transitive = true
    }
    compile(group: 'com.philips.cdp', name: 'product-registration-lib', version: '1.1.1', ext:
'aar') {
        exclude group: 'com.android.support'         transitive = true
    }

    compile(group: 'com.philips.cdp', name: 'digitalCare', version: '6.1.0', ext: 'aar'){         exclude
group: 'com.android.support'         transitive=true
    }

    compile(group: 'com.philips.cdp', name: 'iap', version: '4.1.0', ext: 'aar'){
```

```
                    exclude group: 'com.android.support'
            transitive=true
               }

               compile(group: 'com.philips.cdp', name: 'AppInfra', version: '1.1.1', ext: 'aar')

            }    }
               compile (name: 'AppInfraLib-release', ext:'aar')
}
```

2.3.2 Library dependencies

1. User Registration : 7.0.1
2. Uikit Library : 3.2.3
3. Product Registration : 1.1.1
4. Digital Care : 6.1.0
5. InAppPurchase : 4.1.0
6. AppInfra : 1.1.1

2.4 Package Options

```
   packagingOptions {        exclude 'META-
INF/DEPENDENCIES.txt'        exclude
'META-INF/LICENSE.txt'        exclude
'META-INF/NOTICE.txt'        exclude 'META-
INF/NOTICE'        exclude 'META-
INF/LICENSE'        exclude 'META-
INF/DEPENDENCIES'        exclude 'META-
INF/notice.txt'        exclude 'META-
INF/license.txt'        exclude 'META-
INF/dependencies.txt'        exclude 'META-
INF/LGPL2.1'        pickFirst
'lib/mips/librsjni.so'        pickFirst
'lib/mips/libblasV8.so'
```

```
      pickFirst 'lib/mips/libRSSupport.so'
pickFirst 'lib/x86/librsjni.so'        pickFirst
'lib/x86/libblasV8.so'        pickFirst
'lib/x86/libRSSupport.so'        pickFirst
'lib/armeabi-v7a/librsjni.so'        pickFirst
'lib/armeabi-v7a/libblasV8.so'        pickFirst
'lib/armeabi-v7a/libRSSupport.so'        pickFirst
'lib/arm64-v8a/libRSSupport.so'        pickFirst
'lib/arm64-v8a/librsjni.so'        exclude 'META-
INF/INDEX.LIST'
   }}
```

2.5 Some more dependencies

```
   minSdkVersion 19
targetSdkVersion 23
compileSdkVersion 23
buildToolsVersion "23.0.3"
```

2.6 Proxy dependencies

Gradle dependencies can get some network/proxy related issues with Philips. In order to fix this issue, we are using below proxy settings in gradle.properties of root folder.

```
                              systemProp.https.proxyHost=42.99.164.34
systemProp.https.proxyPort=10015
```

We are using these proxy settings locally. But Eindhoven, does not use above proxy settings.

2.7.Prerequisites

NA

3.Steps for API integration of different configuration

Any invocation in the application context of the app can be done by creating an application class and adding them there. Following are some sample codes to initialize elements for ActivityLife cycle callbacks and User registration .

To remove any common component , we need to remove library from build.gradle and remove its corresponding state file .

The imports for User registration and Inapp Purchase in present in 2 files and it needs to be removed from both the files in order to remove the component completely

Eg: Following is the Application class used to initialize User Registration component :

```
public class AppFrameworkApplication extends
Application{     public static AppInfraInterface
gAppInfra;     public static LoggingInterface
loggingInterface;

    @Override
    public void onCreate() {
         MultiDex.install(this);
        super.onCreate();
        AppInfraSingleton.setInstance( new
      AppInfra.Builder().build(getApplicationContext()));
gAppInfra = AppInfraSingleton.getInstance();
        loggingInterface =
      gAppInfra.getLogging().createInstanceForComponent(BuildConfig.APPLICATION_ID,
      BuildConfig.VERSION_NAME);
        loggingInterface.enableConsoleLog(true);
loggingInterface.enableFileLog(true);

        initializeUserRegistrationLibrary();
   }

    private void initializeUserRegistrationLibrary() {

  AppTaggingInterface aiAppTaggingInterface = gAppInfra.getTagging();
        aiAppTaggingInterface.createInstanceForComponent("User Registration",
                RegistrationHelper.getRegistrationApiVersion());
aiAppTaggingInterface.setPreviousPage("demoapp:home");
        aiAppTaggingInterface.setPrivacyConsent(AppTaggingInterface.PrivacyStatus.OPTIN);
        RegistrationConfiguration.getInstance().
setPrioritisedFunction(RegistrationFunction.Registration);
        RLog.init(this);
        String languageCode = Locale.getDefault().getLanguage();
        String countryCode = Locale.getDefault().getCountry();
        PILLocaleManager localeManager = new PILLocaleManager(this);
localeManager.setInputLocale(languageCode, countryCode);

RegistrationHelper.getInstance().initializeUserRegistration(this);     } }
```

4. Manifest Details Kindly find manifest details in source folder to find the various Android Components added and used.

5. Notes

1. Please refer interface Spec Doc or Java documents for more details on APIs for each individual component.
2. Please refer demo app for implementation details of various CoCo.