|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 1.0a | 25-05-2015 | Anurag Gautam,  Ashok Kasthuri | All | New Release  Features :Tagging, Logging, Secure Storage, App Identity & Service Discovery |

Mobile App Infrastructure library Integration

|  |  |
| --- | --- |
| Author | Anurag Gautam, Ashok Kasthuri |
| Approved by |  |
| Email Id | [anurag.gautam@philips.com](mailto:anurag.gautam@philips.com) ashok.kasturi@philips.com |

1. INTRODUCTION 3

2. INTEGRATION 3

2.1 Maven repository Integration 3

2.2 Library Integration 3

2.3 Library versioning 4

2.4 Root gradle changes 4

2.5 Gradle dependencies 4

2.6 Proxy dependencies 5

2.7 Configuration File 5

3. INITIALIZATION 6

6. Android Manifest Changes 7

6.1 Other User Permissions 7

7. **Design Document Reference 7**

# INTRODUCTION

This document provides an overview of integration procedure for Mobile App Infra library in android mobile applications.

# INTEGRATION

There are two ways to integrate “Mobile App Infrastructure” library with any Android app.

* + 1. **Maven repository based**: At compile time, machine has to be connected with Philips network. Do not follow section 2.2
    2. **Library Integration**: If unable to connect with Philips network then include libraries to your root application. Do not follow section 2.4, 2.5

## Maven repository (Artifactory based) Integration

The easiest and preferred way to use these components is using maven.

All dependent libraries should be downloaded from artifactory.

**Artifactory Path:**

<http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-android/com/philips/cdp/AppInfra/1.0.0/>

If you are inside Philips network then you can directly refer “**2.5 Gradle dependencies**” section. It will automatically download all nested dependencies from artifactory.

## Library Integration

Need to copy all aar files in libs folder; below are the libraries needed, Please make gradle changes

dependencies {  
 compile fileTree(**dir**: **'libs'**, **include**: [**'\*.jar'**])  
 androidTestCompile **'org.mockito:mockito-core:1.9.5'** androidTestCompile **'com.google.dexmaker:dexmaker:1.2'** androidTestCompile **'com.google.dexmaker:dexmaker-mockito:1.2'** compile **'com.android.support:appcompat-v7:23.4.0'** compile **'adobeMobileLibrary:adobeMobileLibrary:4.8.3'** compile(**'com.philips.cdp:prx:2.0.0@aar'**) {  
 exclude **group**: **'com.android.support'** }  
 compile(**'com.philips.cdp:localeMatch:2.0.0@aar'**) {  
 exclude **group**: **'com.android.support'** transitive = **true** }  
 compile **'com.android.volley:volley:1.0.0'**}

## Library versioning

Library version can be obtained by using below API

version = objcdp.getVersion()

## .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'

// NOTE: Do not place your application dependencies here; they belong

// in the individual module build.gradle files

}

}

allprojects {

repositories {

maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-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/ext-release-local'}

maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-android' }

maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-stage-local-android'}

}

}

## Gradle dependencies

Just by adding below gradle dependencies, digitalcare and nested possible libraries will be downloaded from artifcatory. But it has to be inside Philips network.

compile(group: **'com.philips.cdp'**, name: **'prx'**, version: **'2.0.0'**, ext: **'aar'**){  
 exclude group: **'com.android.support'** transitive=**true** }  
 compile(group: **'com.philips.cdp'**, name: **'localeMatch'**, version: **'2.0.0'**, ext: **'aar'**){  
 exclude group: **'com.android.support'** transitive=**true** }

## Proxy dependencies

Gradle dependencies can get some network/proxy related issues. 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 this proxy settings locally. But Eindhoven, does not use above proxy settings.

## Configuration File

1. **logging.properties**

Go to library’s assets folder and copy *logging.properties* file from it and place it in

integrating app’s assets folder. Once app is launched this file will be copied to device internal memory (eg data/data/com.cdp.philips/logging.properties) in the app root directory and thereafter app will only read this file and NOT the asset file.

Developer should perform log configuration change(*logging.properties*) here in internal memory via device or connected PC and to see reflected changes app needs to be restarted.

1. ADBMobileConfig.json

Keep this json file in Assets folder. Make sure SSL is “true” for secure HTTPS requests.

Change rsids tag accordingly to dev or release. batchlimit is another tag where one can define the count of requests.

{  
 **"version"** : **"1.0"**,  
 **"acquisition"**: {  
 **"server"**: **"c00.adobe.com"** },  
  
 **"analytics"** : {  
 **"referrerTimeout"**: 5,   
 **"rsids"** : **"philipsmobileappsdev"**,  
  
*// "server" : "localhost:50000",* **"server"** : **"philips.112.2o7.net"**,  
 **"charset"** : **"UTF-8"**,  
 **"ssl"** : **true**,  
 **"offlineEnabled"** : **false**,  
 **"lifecycleTimeout"** : 30,  
 **"batchLimit"** : 0,  
 **"privacyDefault"** : **"optunknown"**,  
 **"poi"** : [  
 ]  
 },  
 **"target"** : {  
 **"clientCode"** : **"amsdk"**,  
 **"timeout"** : 5  
 },  
 **"audienceManager"** : {  
 **"server"** : **""** }  
}

1. AppIdentity.json

developer needs to be crerate appidentity.json and add microsite, sector & AppState key value. Remaining AppVersion and AppName will be written from gradle & AppLocalName will be written manifest file.

{  
 "micrositeId" : "12345",  
 "sector" : "B2C",  
 "AppState" : "DEVELOPMENT"  
}

# INITIALIZATION

AppInfra object should be created in the class which extend Application.

**public class** FrameworkApplication **extends** Application {  
 **public static** AIAppTaggingInterface *mAIAppTaggingInterface*;  
 **public static** AppInfra *gAppInfra*;  
 @Override  
 **public void** onCreate() {  
 **super**.onCreate();  
  
 ***gAppInfra* = new AppInfra.Builder().build(getApplicationContext());**  
 *mAIAppTaggingInterface* = *gAppInfra*.getTagging().createInstanceForComponent(**"Component name"**,**"Component ID"**);  
  
  
 }  
  
}

While launching any component above created AppInfra instance should be passed as parameter to component constructor.

*e.g. Uikit uikit = new Uikit(****gAppInfra****); // Each component will have same single instance of App Infra*

Note: All component must have a constructor to recieve AppInfra as parameter

Single App Infra object will exist in entire App Framework.

# Android Manifest Changes

Add these permissions:

## Other User Permissions

<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>  
<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"**/>

# Supporting apps with Over 65K Methods

This is special case if app which has more than 65K methods do follow below link for reference.

<https://developer.android.com/tools/building/multidex.html>

According to this do following changes in gradle and application class

In Gradle file:

android {  
    compileSdkVersion 21  
    buildToolsVersion "21.1.0"  
  
    defaultConfig {  
        ...  
        minSdkVersion 14  
        targetSdkVersion 21  
        ...  
  
        // Enabling multidex support.  
        multiDexEnabled true  
    }  
    ...  
}  
  
dependencies {  
  compile 'com.android.support:multidex:1.0.1'  
}

In Application Class:

@Override  
public void onCreate() {  
   MultiDex.install(this);

  Super.onCreate();

}

# Supported Languages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| English | German | French (FR) | Simplified Chinese | Traditional Chinese | Portuguese (EU) |
| Russian | Arabic | Japanese | Dutch | Malay | Montenegrin |
| French-CA | Romanian | Spanish Mexican | Ukrainian | Portuguese (BR) | Serbian |
| Gaelic | Greek | Hebrew | Hindi | Finnish | Swedish |
| Italian | Polish | Spanish | Korean | Czech | English UK |
| Norwegian | Albanian | Bosnian | Bulgarian | Croatian | Estonian |
| Hungarian | Indonesian | Kazakh | Latvian | Lithuanian | Macedonian |
| Slovak | Slovenian | Spanish (AR) | Thai | Vietnamese | Danish |
| Farsi | Turkish | Hongkong-China |  |  |  |

**Design Document Reference:**

https://atlas.natlab.research.philips.com/confluence/download/attachments/7439362/Android%20\_Code\_Design\_AppInfra\_global\_v1%200a.docx?api=v2