

SAP How-to Guide

Mobility

SAP Mobile Platform

How To... Enable User On-boarding using MAF Logon (Android)

Applicable Releases:

SAP Mobile Platform 3.0 SP04

SAP Mobile SDK 3.0 SP05

Version 1.0



The Best-Run Businesses Run SAP™

© Copyright 2014 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

SAP "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

| Document Version | Description |
|------------------|--------------------------------------|
| 1.00 | First official release of this guide |

Typographic Conventions

| Type Style | Description |
|-----------------------------|--|
| Example Text | Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation |
| Example text | Emphasized words or phrases in body text, graphic titles, and table titles |
| Example text | File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools. |
| Example text | User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation. |
| <Example text> | Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system. |
| EXAMPLE TEXT | Keys on the keyboard, for example, F2 or ENTER. |

Icons





| Icon | Description |
|---|-----------------------|
|  | Caution |
|  | Note or Important |
|  | Example |
|  | Recommendation or Tip |

Table of Contents

| | | |
|-----|----------------------------------|----|
| 1. | Business Scenario | 1 |
| 2. | Background Information | 1 |
| 3. | Prerequisites | 1 |
| 4. | Step-by-Step Procedure | 2 |
| 4.1 | Download Project | 2 |
| 4.2 | Android Project | 4 |
| 4.3 | Set up dependencies | 6 |
| 4.4 | Review AndroidManifest.xml | 12 |
| 4.5 | Review Application ID | 13 |
| 4.6 | Implement MAFLogonActivity | 13 |
| 4.7 | Get User information | 16 |
| 4.8 | Run the Application | 16 |
| 4.9 | Verifying the registration | 19 |
| 5. | Appendix | 20 |



1. Business Scenario

A travel agency franchise would like to build an online mobile application to manage its. The SAP Mobile Platform provides a means for them to securely and efficiently access backend location exposed by their SAP NetWeaver Gateway system via OData-based REST services.

Before any communication with the SAP backend systems can take place at all, the app needs to on-board users onto the SAP Mobile Platform. The Mobile Application Framework (MAF) contains the MAF logon component that provides easy integration for applications that use logon UI behavior.

2. Background Information

The goal of this exercise is not to show how to create a project from scratch and dissect every line of code. Instead, it shows the key pieces of code and information, along with a starter project template, so that developers understand how to leverage the MAF Logon component to on-board users in their own apps.

3. Prerequisites

This exercise has the following prerequisites:

- Java Standard Edition 7
- Eclipse Kepler
- Android SDK with API level 8 and 19 downloaded
- Android ATD Eclipse plugin installed
- To get the most out of this exercise, some experience with Java is recommended.
- SAP Mobile Platform 3.0 SP04

 This sample code assumes you have configured an application in SMP 3.0 called `com.sap.flight`.

For more information on how to create an application configuration, please visit http://help.sap.com/saphelp_smp304svr/helpdata/en/7c/2a3c6070061014bc14ea56b0ab4883/content.htm

- SAP Mobile SDK 3.0 SP05

To access the sample projects you need to install GIT

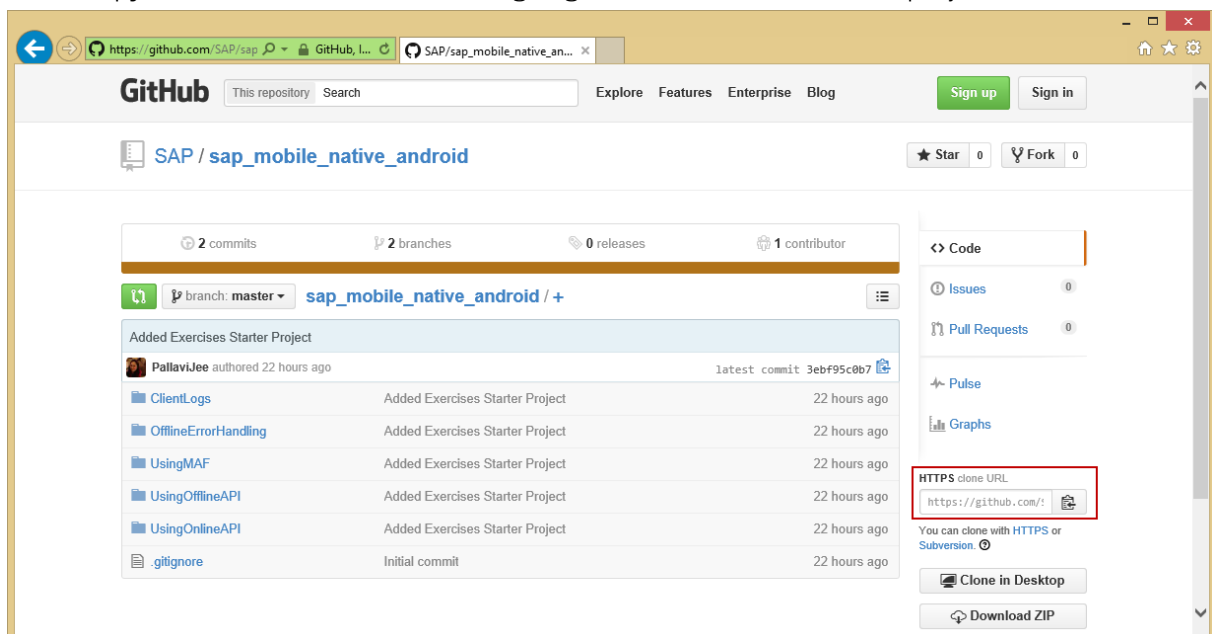
- GIT installed: <http://msysgit.github.io>

4. Step-by-Step Procedure

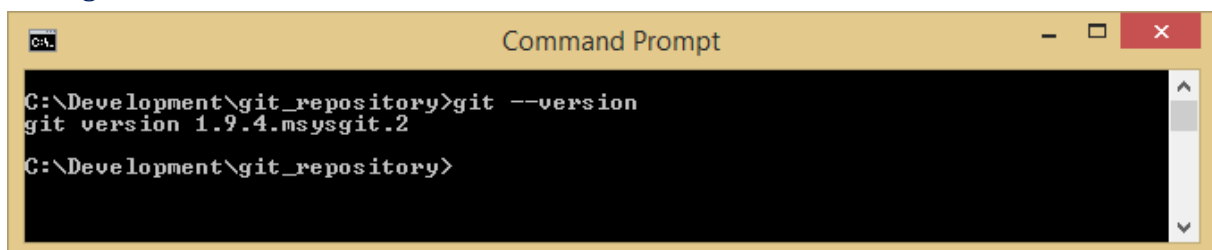
The following sections provide a detailed step-by-step procedure on how to code an Android app to on-board/register a user onto the SAP Mobile Platform using the MAFLogon.

4.1 Download Project

1. The sample project used in this guide (UsingMAF) is available in the following Git repository:
https://github.com/SAP/sap_mobile_native_android
2. Copy the HTTPS clone URL, we are going to need it to download the projects



3. Verify you have Git installed. Open a Command Prompt and run the following command:
`git --version`

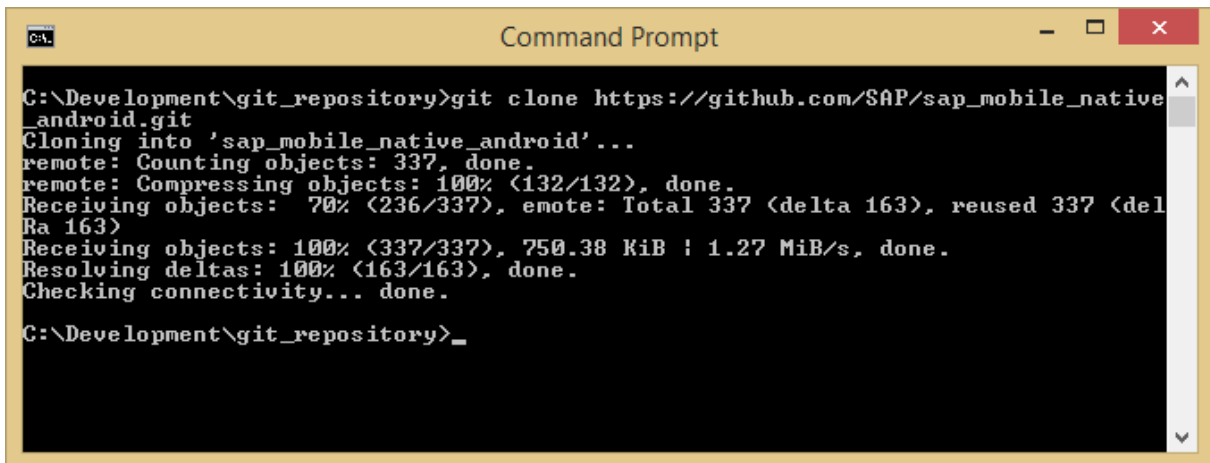


You should see the git version you have installed.

Otherwise, download it from <http://msysgit.github.io> and install it


4. From the folder where you want to clone the projects, run the following command:

```
git clone https://github.com/SAP/sap_mobile_native_android.git
```



```
C:\Development\git_repository>git clone https://github.com/SAP/sap_mobile_native_android.git
Cloning into 'sap_mobile_native_android'...
remote: Counting objects: 337, done.
remote: Compressing objects: 100% (132/132), done.
Receiving objects: 70% (236/337), remote: Total 337 (delta 163), reused 337 (delta 163)
Receiving objects: 100% (337/337), 750.38 KiB | 1.27 MiB/s, done.
Resolving deltas: 100% (163/163), done.
Checking connectivity... done.

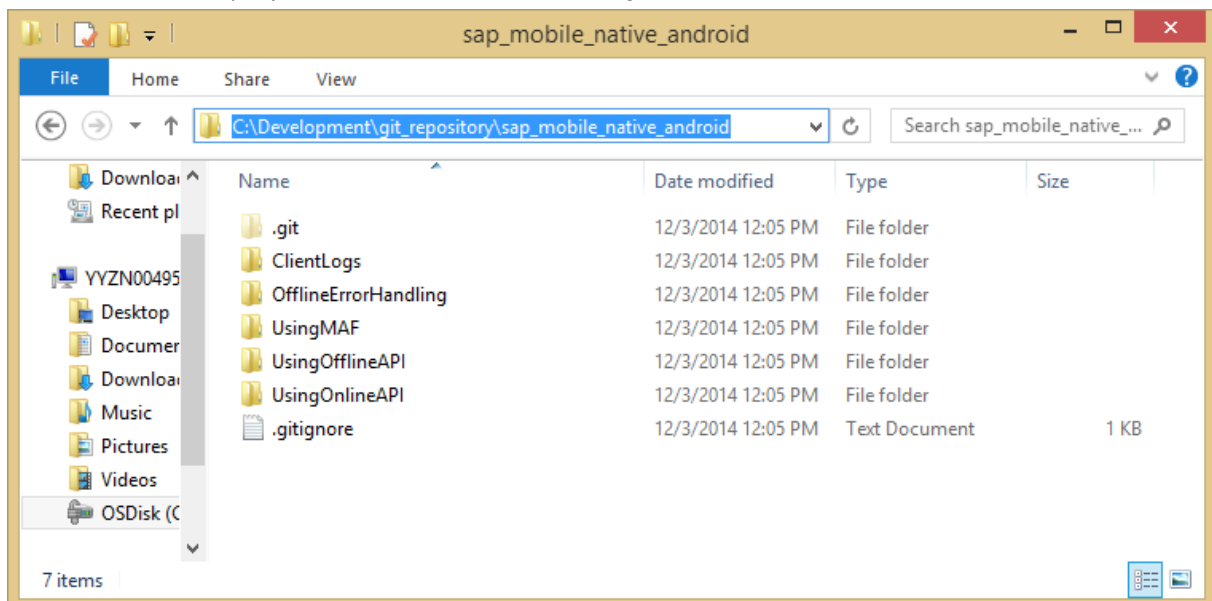
C:\Development\git_repository>
```

 If you are on a network behind a firewall, you need to configure Git to use your http proxy. Use the following commands:

```
git config --global http.proxy http://<proxyhost>:<proxyport>
```

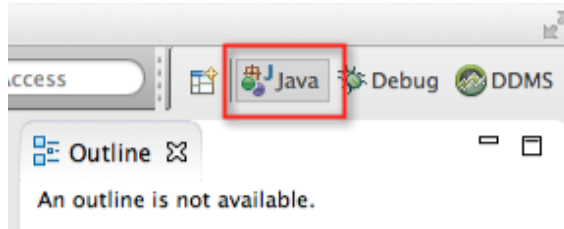
```
git config --global https.proxy http://<proxyhost>:<proxyport>
```

5. Check if the projects downloaded successfully

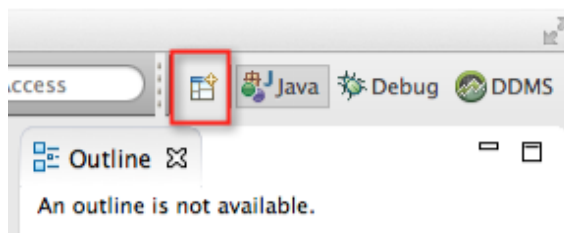


4.2 Android Project

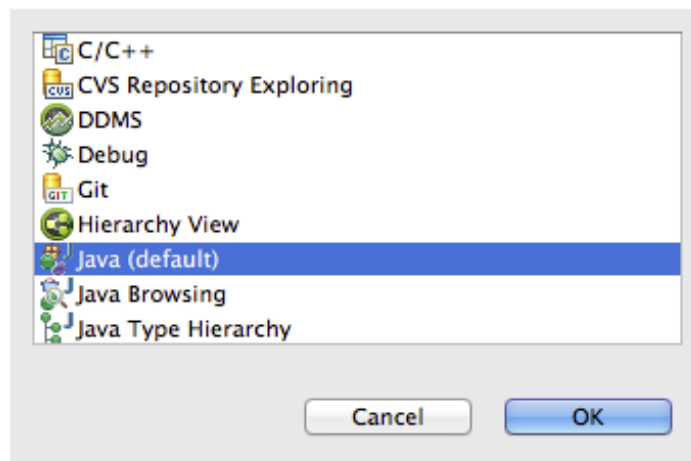
1. Open eclipse and verify you have the Java perspective selected.



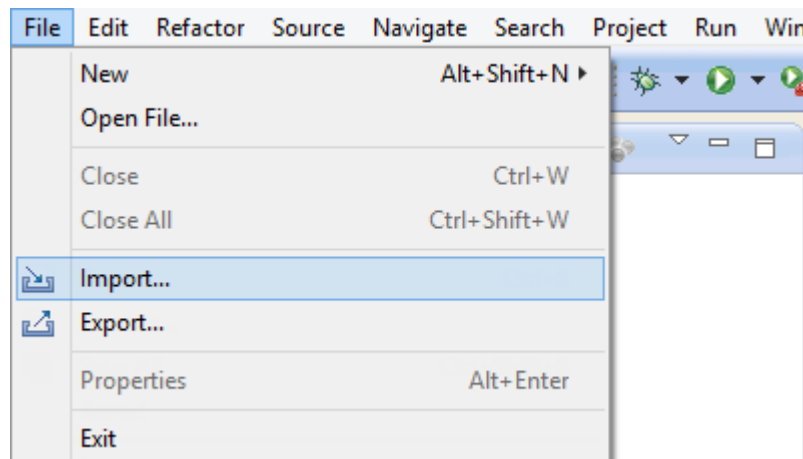
2. If the Java perspective is NOT selected. Select the Open Perspective icon



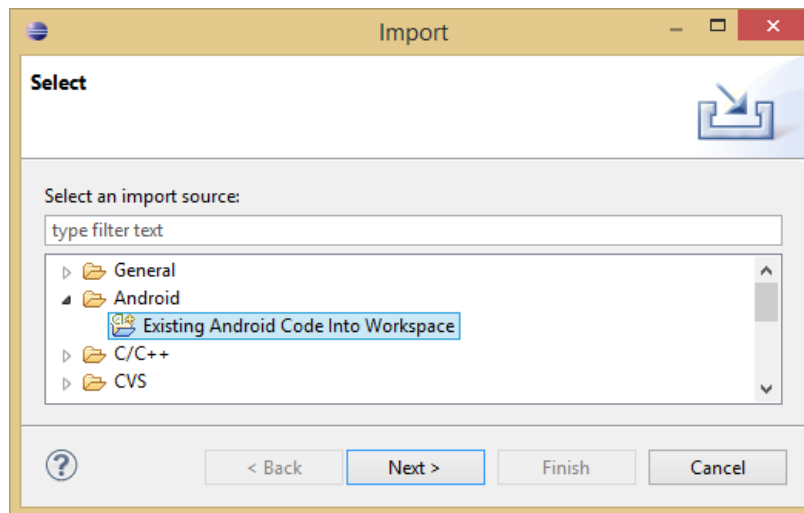
3. Select the Java perspective and click OK



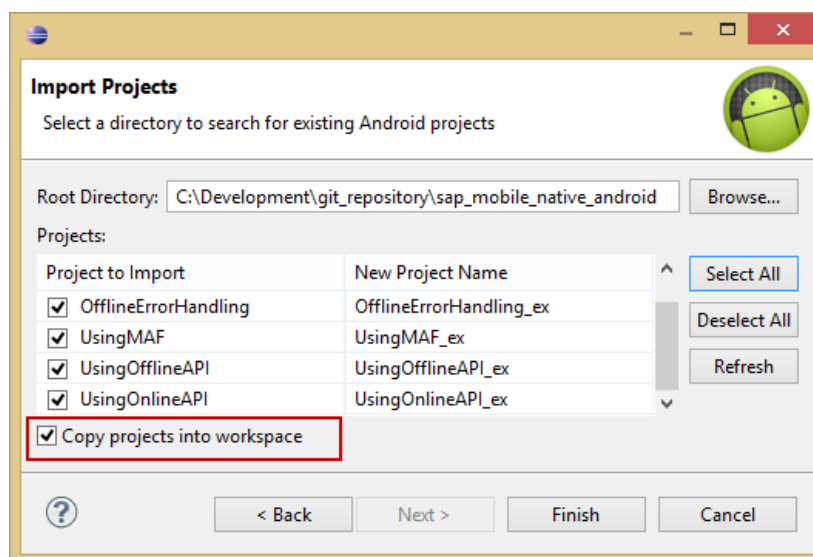
4. Go to the File menu and choose Import.



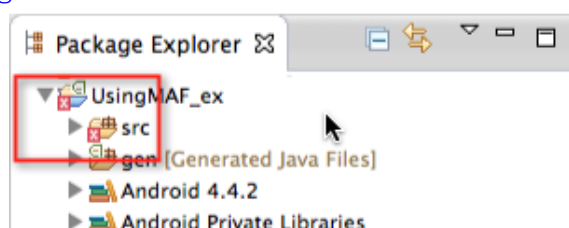
5. The Import window will open, select Existing Android Code into Workspace and click Next.



6. Click the Browse button to locate the local folder where you clone the projects.
 7. Select the projects you want to import to eclipse. You must include UsingMAF to complete this guide
 8. Make sure you check Copy projects into workspace to create a copy of your cloned projects, and click Finish



9. If your project contains errors, go to the libs folder and import the libraries as explained in [Appendix A - Importing OData Libraries](#)



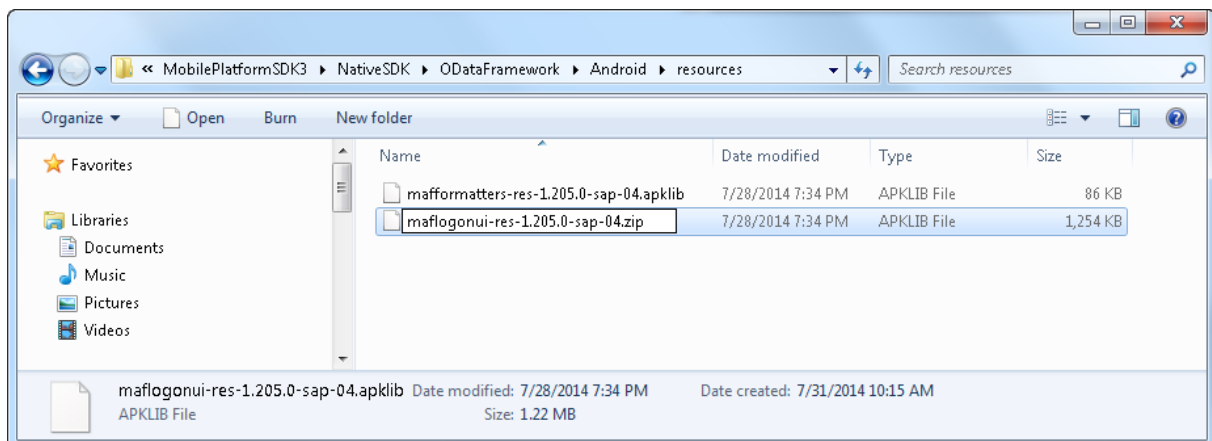
10. You will also need to import MAF resources (maflogonui, mafsettingscreen and mafuicomponents) as indicating in [Step 4.3 Set up MAF dependencies](#)

4.3 Set up dependencies

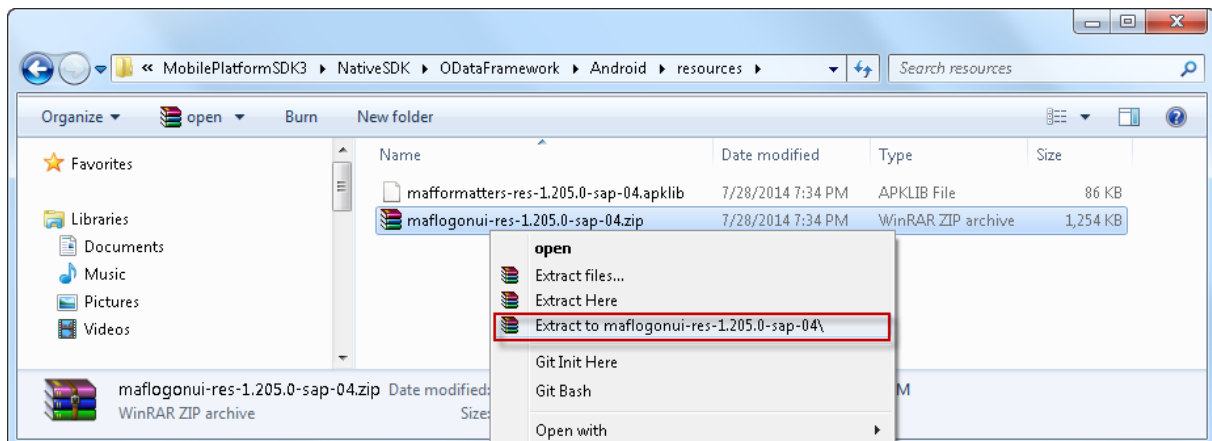
To use the MAF Login Component you must import a number of Libraries and Resources. MAF Libraries were installed with SMP Client SDK installer into the folder specified by you when you executed the installer. This libraries and resources are already included in the project. Please follow the steps to confirm the project was set up correctly.

MAF is shipped with various resources. These projects have Android nature, but don't ship any source code. Their only purpose is to ship all Android resource files like: images, layouts, localization, XMLs

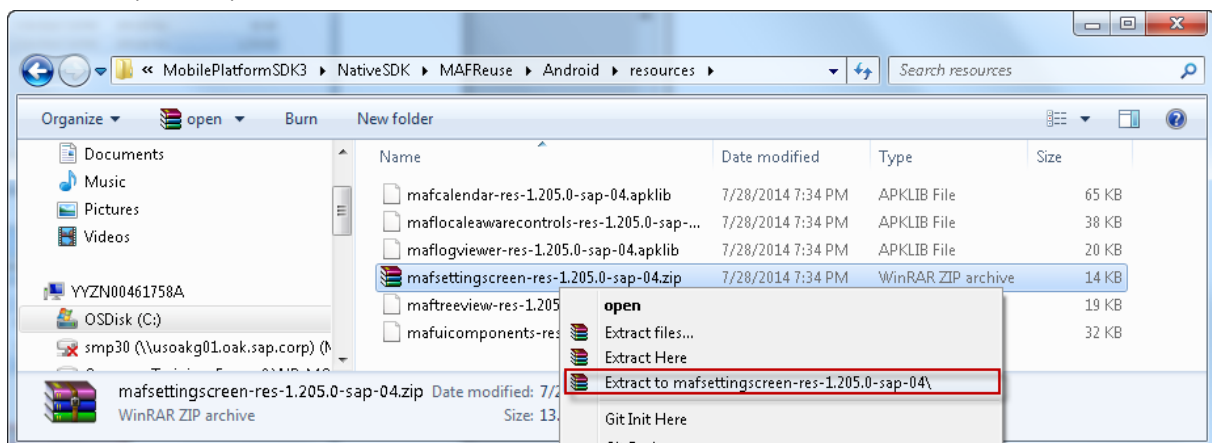
1. As a result of the SDK installer you will find the following .apklib files in the installation folder:
 - a. <Client SDK
dir>\NativeSDK\ODataFramework\Android\Resources\maflogonui.apklib
 - b. <Client SDK
dir>\NativeSDK\MAFReuse\Android\Resources\mafsettingscreen.apklib
 - c. <Client SDK
dir>\NativeSDK\MAFReuse\Android\Resources\mafuicomponents.apklib
2. Open folder <Client SDK dir>\NativeSDK\ODataFramework\Android\Resources\
3. The resource archives have .apklib extension (i.e maflogonui-res-xxx.apklib). You will need to change this extension to .zip



4. Unzip the .zip file into a local folder

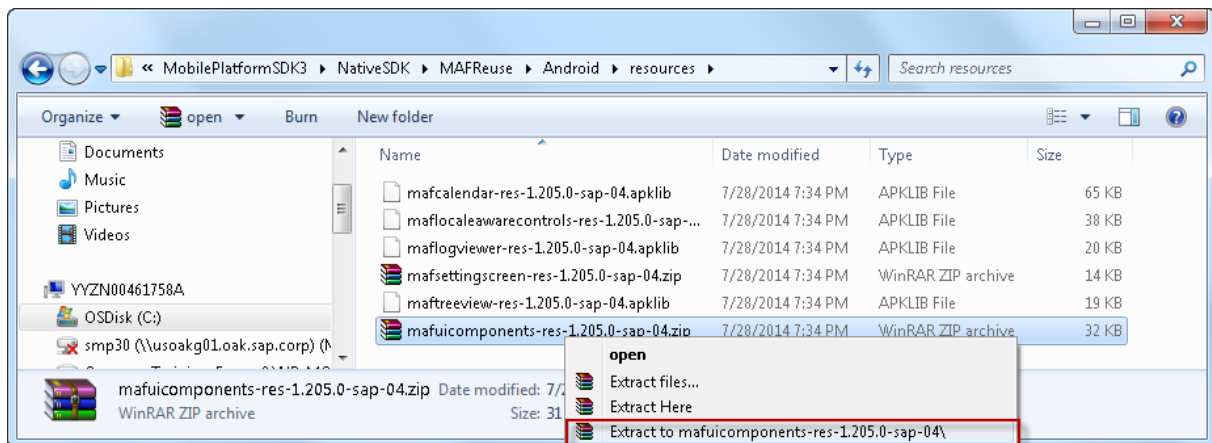


5. Open folder <Client SDK dir>\NativeSDK\MAFReuse\Android\resources\
6. The resource archives have .apklib extension (i.e mafsettingscreen-res-xxx.akplib). You will need to change this extension to .zip
7. Unzip the .zip file into a local folder

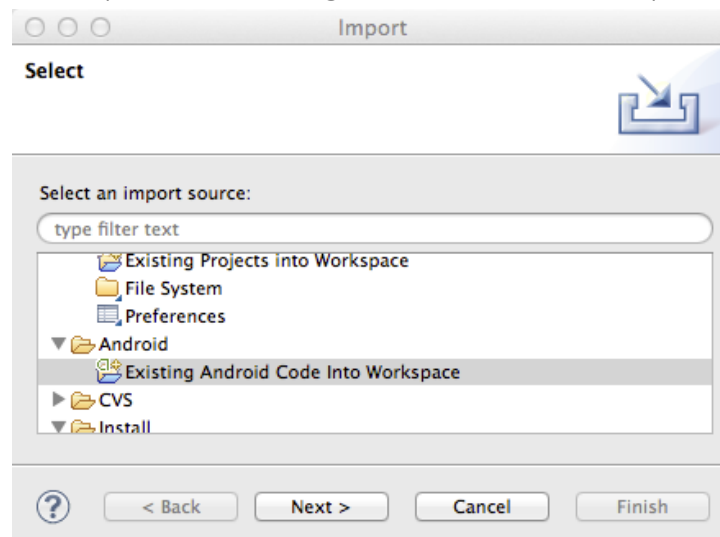


8. In the same folder <Client SDK dir>\NativeSDK\MAFReuse\Android\resources\
9. The resource archives have .apklib extension (i.e mafuicomponents-res-xxx.akplib). You will need to change this extension to .zip

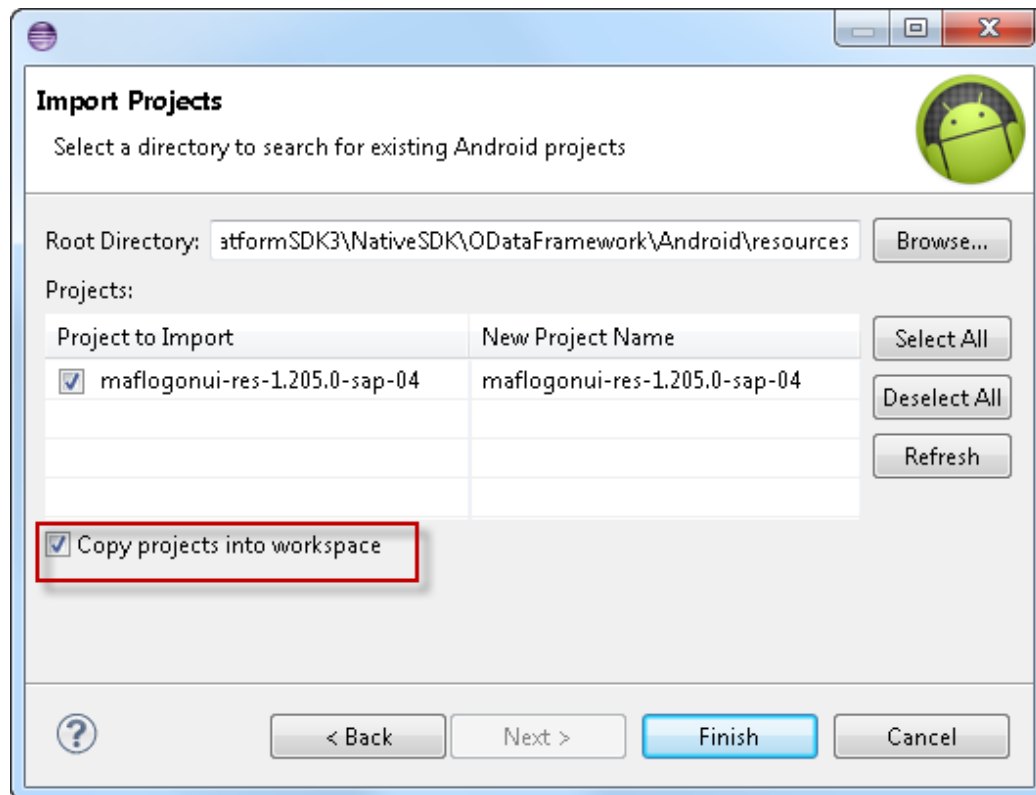
10. Unzip the .zip file into a local folder



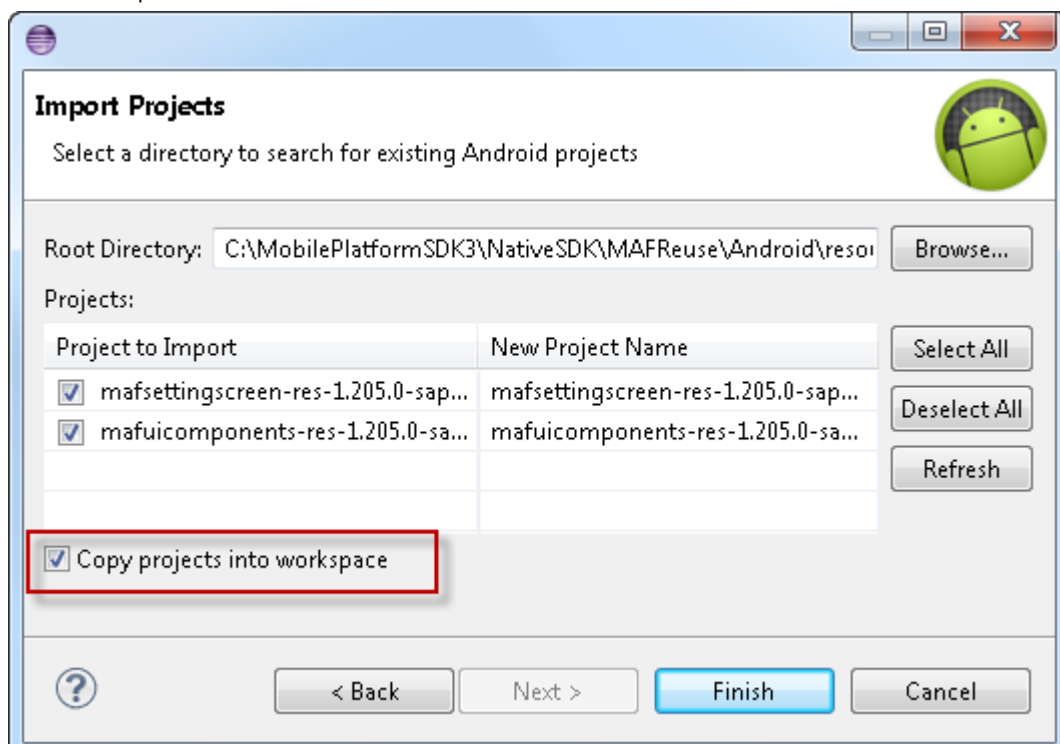
11. Open eclipse, import the projects by selecting File -> Import menu.
12. The Import window will open, select Existing Android Code into Workspace and click Next.



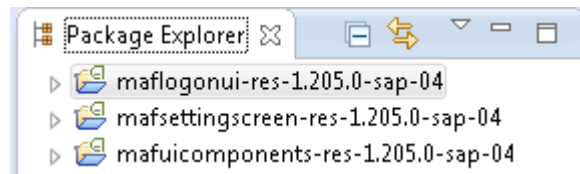
13. In the Root Directory field, click the Browse button to locate the .zip files in your local folder (i.e. <Client_SDK_dir>\NativeSDK\ODataFramework\Android\resources\maflogonui-res-xxx.apklib) and click Finish.



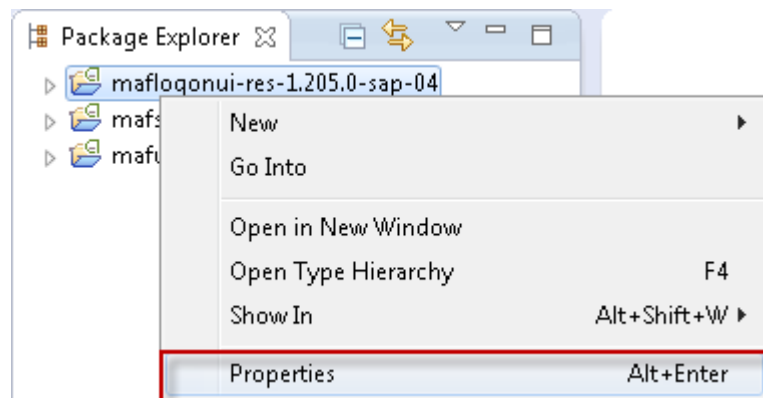
14. Repeats steps 12-13 to import the other MAF project resources: mafsettingscreen and mafuicomponents




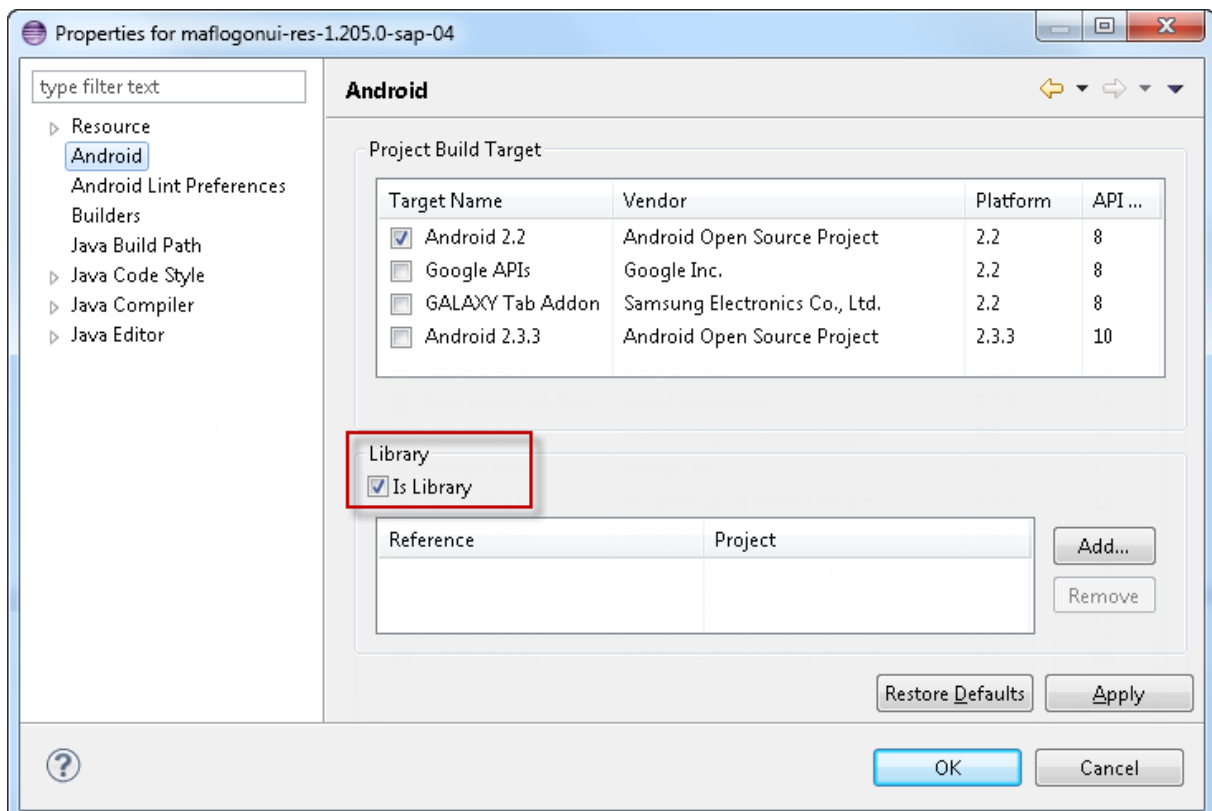
15. After importing the three MAF resources, in the Package Explorer view, check your project, it should look like the image below.



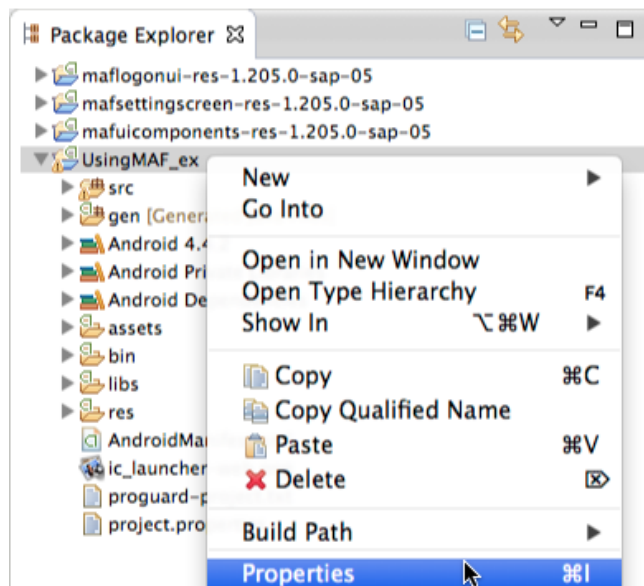
16. Select maflogonui-res-xxx project. Right click and select the project properties from the context menu. These MAF resources projects need to be marked as Android library project.



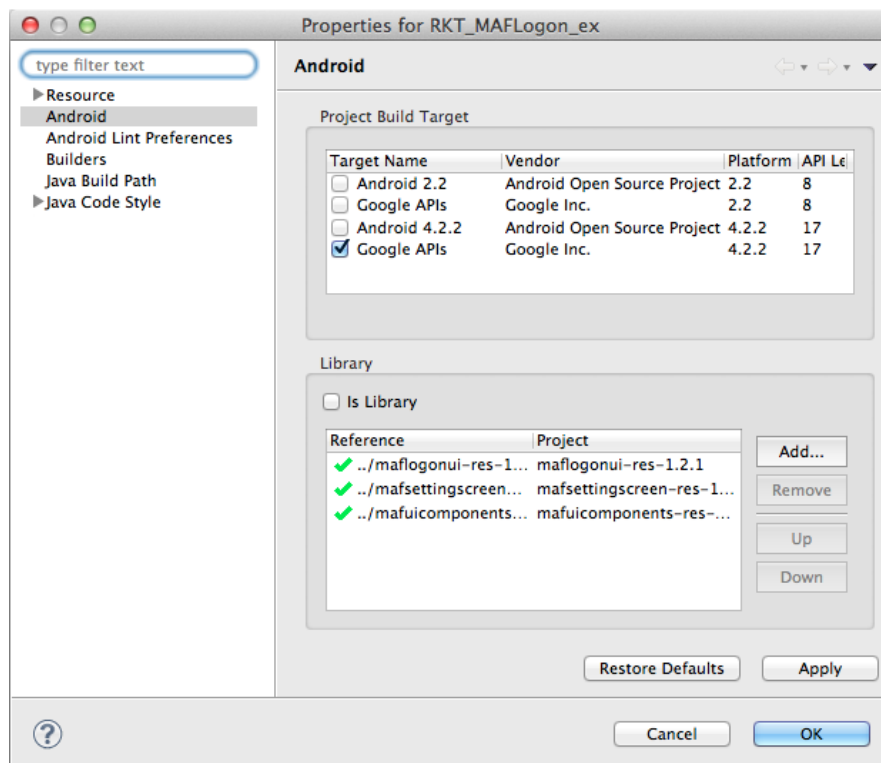
17. In the project properties you need to select Android and then mark the project as Library.
 Remember to repeat this step for all three of the resource projects: maflogonui, mafsettingscreen and mafuicomponents.



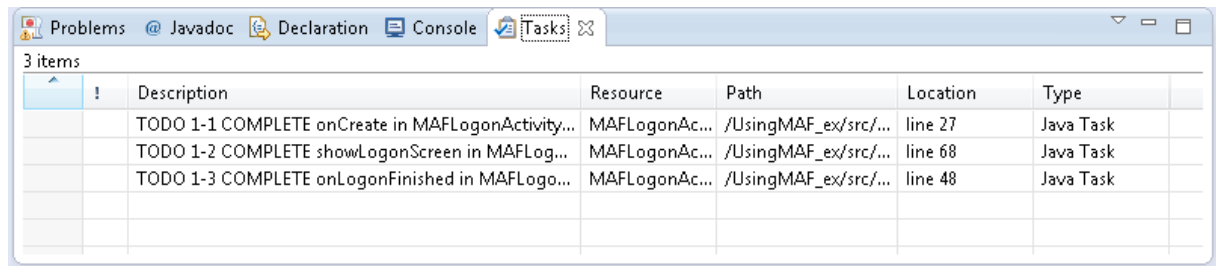
18. Select UsingMAF_ex and open the properties to set up the reference to the MAF resources



19. Select Android. In the Library section you need to click the Add button. The Project Selection modal view will be shown.



20. (optional) you can open the Task view and check the TODO items you will complete in the following section



| | Description | Resource | Path | Location | Type |
|---|---|---------------|----------------------|----------|-----------|
| ! | TODO 1-1 COMPLETE onCreate in MAFLogonActivity... | MAFLogonAc... | /UsingMAF_ex/src/... | line 27 | Java Task |
| | TODO 1-2 COMPLETE showLogonScreen in MAFLog... | MAFLogonAc... | /UsingMAF_ex/src/... | line 68 | Java Task |
| | TODO 1-3 COMPLETE onLogonFinished in MAFLogo... | MAFLogonAc... | /UsingMAF_ex/src/... | line 48 | Java Task |

4.4 Review AndroidManifest.xml

1. Open the AndroidManifest.xml



2. Check if the required permissions are present. In order to give access to some Android resources make sure the following permissions are in the AndroidManifest.xml

```

<!-- allow connections to Internet Services. -->
<uses-permission android:name="android.permission.INTERNET" />
<!-- Network State Permissions to detect Internet status -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!-- External storage permission required by MAF Logon -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

MAF Logon uses various capabilities of your Android device like:

- Network and Internet communication, which was already included in the AndroidManifest.xml
- External Storage

3. In the AndroidManifest.xml file we also define the screens or activities the user will see. Check the MAFLogonActivity is defined in the manifest file.

```
<activity
    android:name=".ui.main.MAFLogonActivity"
    android:configChanges="orientation|screenSize"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
</activity>
```

4.5 Review Application ID

Previously you should have created an application configuration in the SAP Management Cockpit with a specific name (i.e com.sap.flight). This name must match the application id this app uses.

1. Go to src -> com.sap.dcode.usingmaf.services and open AppSettings.java
2. If you have used a different name, please change it here

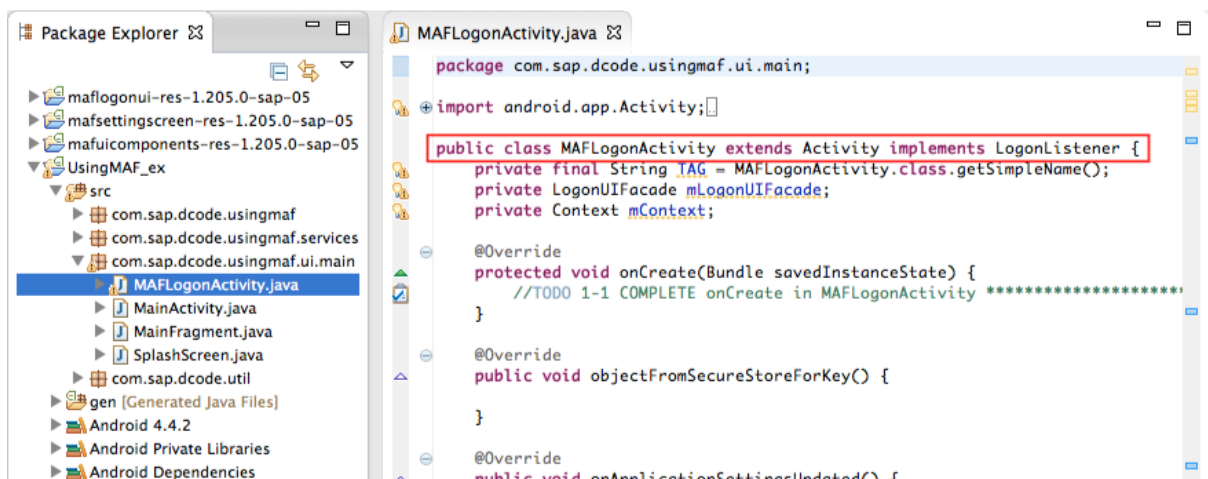
```
package com.sap.dcode.usingmaf.services;

public final class AppSettings {
    public final static String APP_ID = "com.sap.flight";
}
```

4.6 Implement MAFLogonActivity

Now that we have successfully set up the Android project, we can use the MAF Logon resources in our project.

3. Go to src -> com.sap.dcode.usingmaf.ui.main and open MAFLogonActivity.java
4. Check the definition of this class. MAFLogonActivity implements LogonListener



5. Go to the onCreate method and after the **//TODO 1-1** marker initialize the LogonUIFacade with the following code

```
protected void onCreate(Bundle savedInstanceState) {
    //TODO 1-1 COMPLETE onCreate in MAFLogonActivity *****
    LogonCore lgCore = LogonCore.getInstance();
    //check if the store has been initialized
    if (lgCore!=null && lgCore.isStoreAvailable()) {
        LogonCoreContext lgCtx = lgCore.getLogonContext();
        //Check if the secure store is open
        if (lgCtx != null && lgCtx.isSecureStoreOpen()){
            try {
                //Check if the device is registered by
                //verifying the connection id
                if (!TextUtils.isEmpty(lgCtx.getConnId())) {
                    //If the device is registered it opens the Main screen
                    Intent intent = new Intent();
                    intent.setClass(this, MainActivity.class);
                    startActivity(intent);
                    finish();
                }
            } catch (LogonCoreException e){
                TraceLog.e(TAG + ":", "onCreate", e);
            }
        }
    }
    super.onCreate(savedInstanceState);
    // get an instance of the LogonUIFacade
    mLogonUIFacade = LogonUIFacade.getInstance();
    // set context reference
    mContext = this;
    this.showLogonScreen();
}
```

6. Go to the showLogonScreen method and enter the following code after the **//TODO 1-2** marker

```
private void showLogonScreen() {
    //TODO 1-2 COMPLETE showLogonScreen in MAFLogonActivity *****
    // initialize the LogonUIFacade with context and application id
    mLogonUIFacade.init(this, mContext, AppSettings.APP_ID);

    // ask LogonUIFacade to present the logon screen
    // set the resulting view as the content view for this activity
    setContentView(mLogonUIFacade.logon());
}
```



setContentView method is used to present the logon screen from the MAFLogonActivity

7. Now implement the `onLogonFinished` method. Enter the following code after the **//TODO: 1-3** marker

```
public void onLogonFinished(String message, boolean isSuccess,
    LogonContext lgContext) {
    //TODO 1-3 COMPLETE onLogonFinished in MAFLogonActivity *****
    TraceLog.d("onLogonFinished: "+message);
    //Check if it finished successfully
    if (isSuccess) {
        try {
            //For debugging purposes will log the app connection id and
            // the end point url.
            // In a productive app, remember to remove these logs
            String appConnID = LogonCore.getInstance().getLogonContext()
                .getConnId();
            TraceLog.d("onLogonFinished: appcid:"+ appConnID);
            TraceLog.d("onLogonFinished: endpointurl:"+
lgContext.getEndPointUrl());

        } catch (LogonManagerException e) {
            Log.e(TAG, e.getLocalizedMessage(), e);
        } catch (LogonCoreException e) {
            Log.e(TAG, e.getLocalizedMessage(), e);
        }

        // Navigate to the Main menu screen
        Intent goToNextActivity = new Intent(this, MainActivity.class);
        startActivity(goToNextActivity);
        finish();
    }
}
```

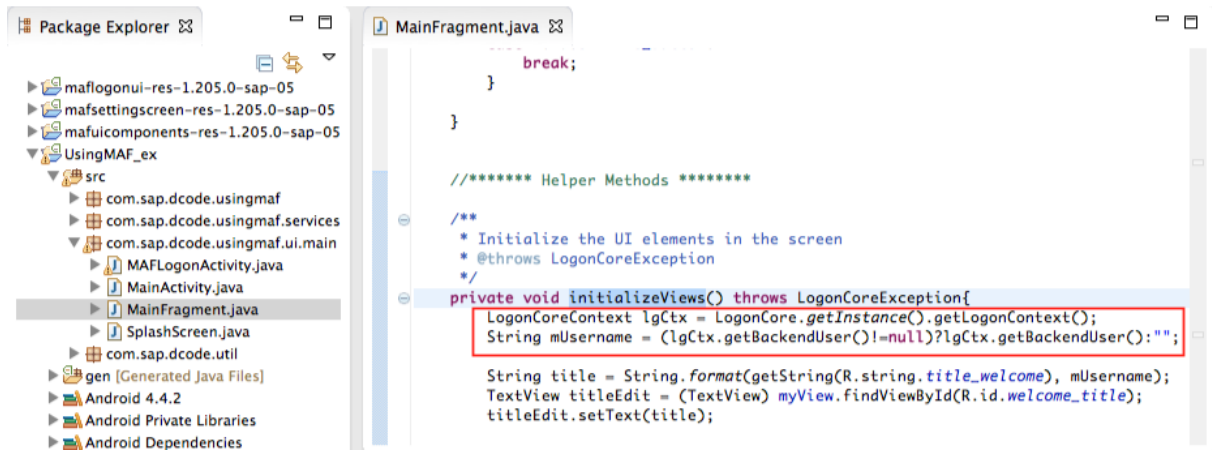
This callback method handles BOTH success and error cases.

7. Save the changes.

4.7 Get User information

In the main screen we would like to display the name of the user.

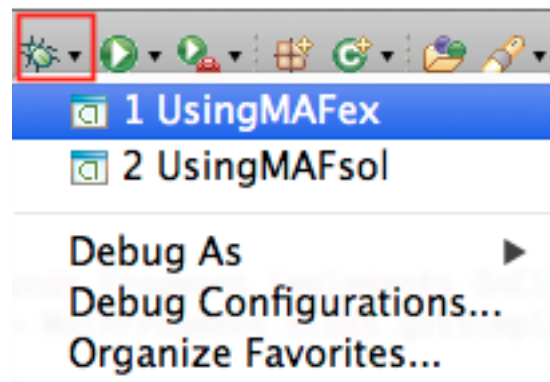
1. Go to src -> com.sap.dcode.usingmaf.ui.main
2. Open MainFragment.java
3. Go to the InitializeViews method and check how you can get user information from the MAF logon core context object



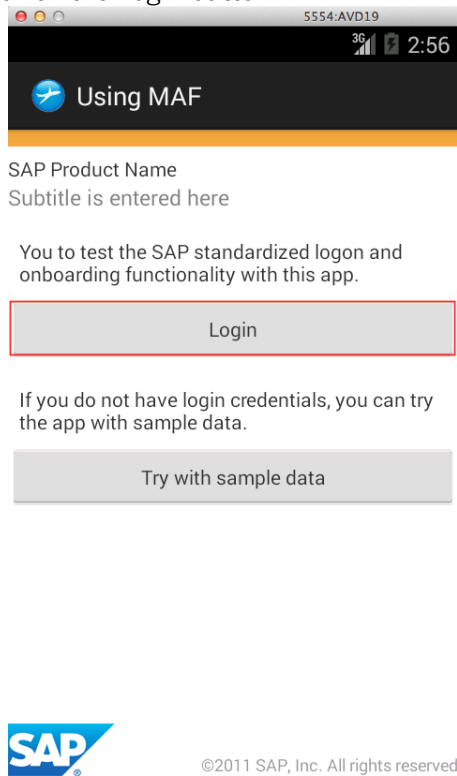
4.8 Run the Application

If you haven't created the Debug Configuration, please review [Appendix B – Creating Debug Configuration](#)

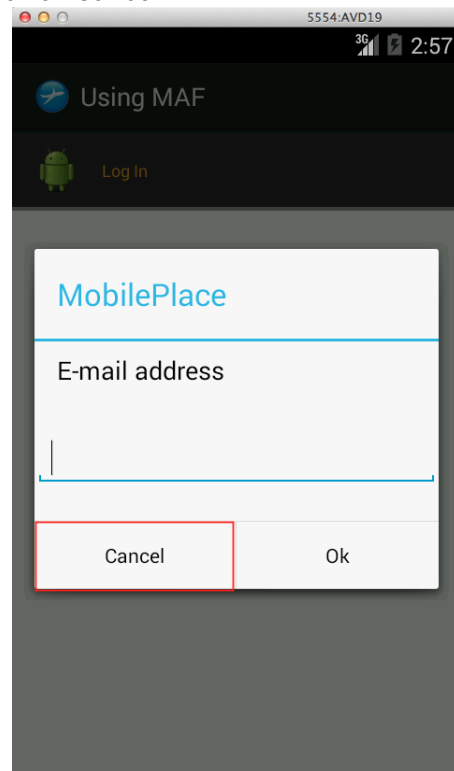
1. In Eclipse, go to the Debug icon and select the corresponding UsingMAF_ex configuration



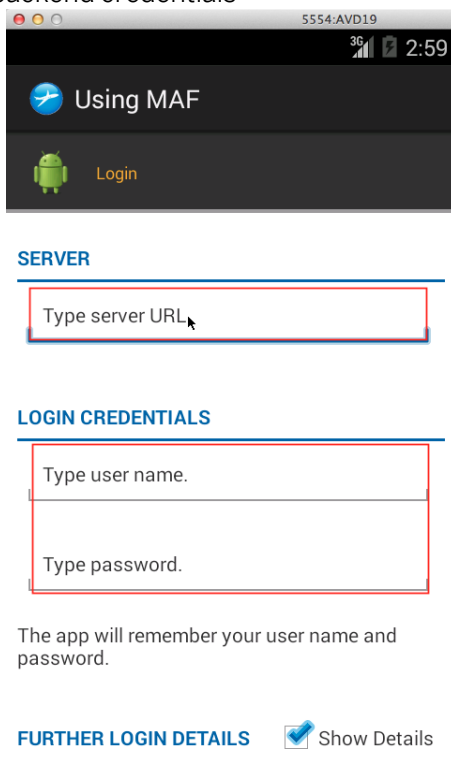
8. Click on the Login button



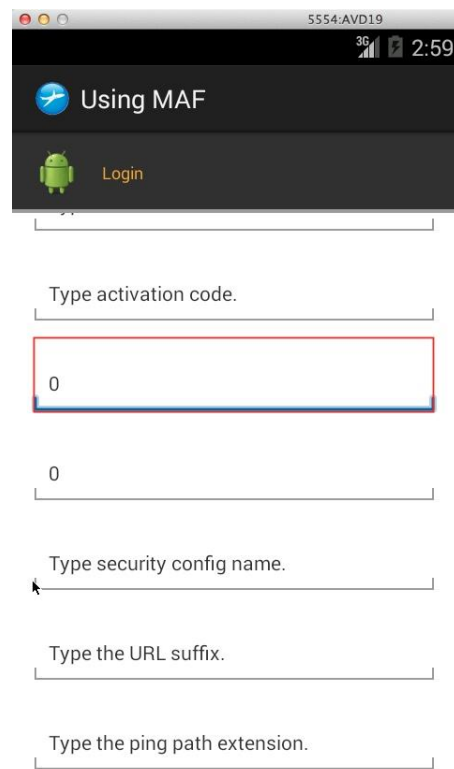
9. Click on Cancel



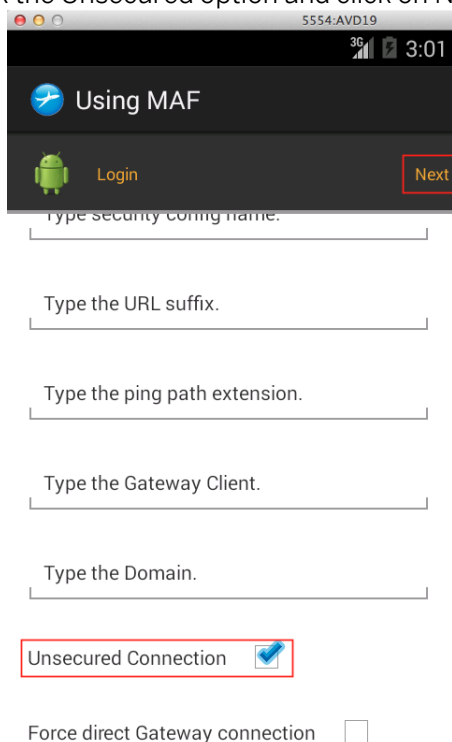
10. Enter the IP address of the SMP 3.0 server and backend credentials



11. Scroll down to enter port (8080 is default)



12. Keep scrolling until the end of the screen to check the Unsecured option and click on Next



5554:AVD19 3G 3:01

Using MAF

Login Next

Type security config name.

Type the URL suffix.

Type the ping path extension.

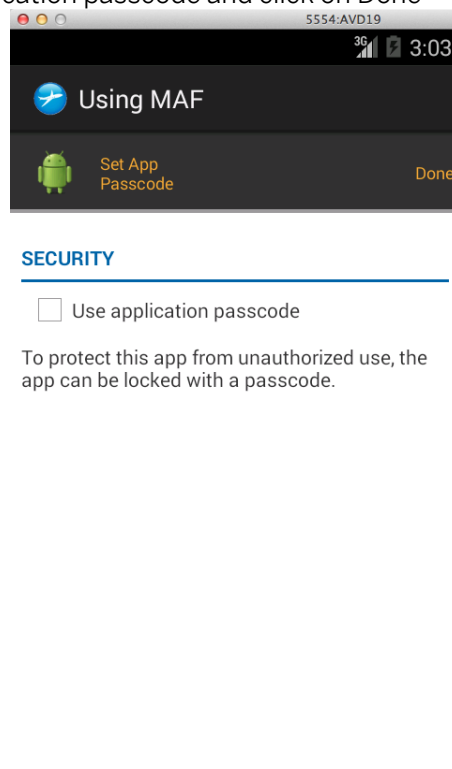
Type the Gateway Client.

Type the Domain.

Unsecured Connection ☒

Force direct Gateway connection ☐

13. Enter a passcode or uncheck the Use application passcode and click on Done



5554:AVD19 3G 3:03

Using MAF

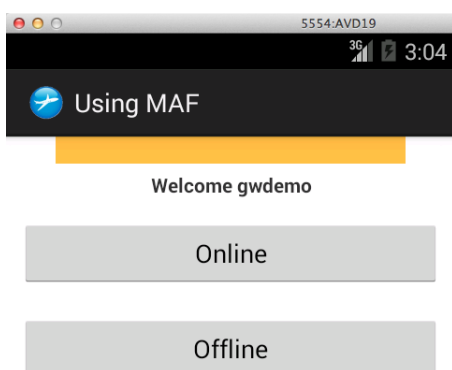
Set App Passcode Done

SECURITY

☐ Use application passcode

To protect this app from unauthorized use, the app can be locked with a passcode.

14. The main screen will be displayed



5554:AVD19 3G 3:04

Using MAF

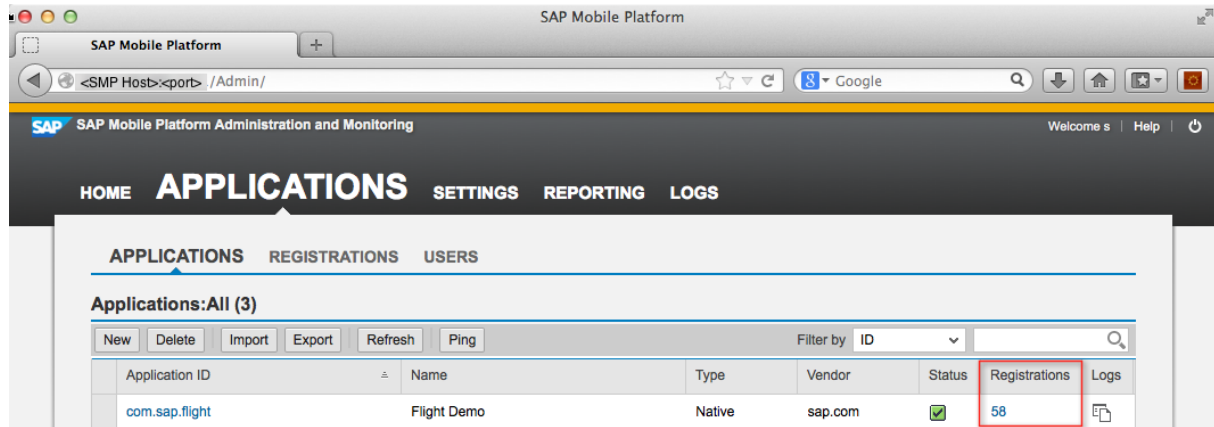
Welcome gwdemo

Online

Offline

4.9 Verifying the registration

To verify that the user was registered successful, you can open the SAP Management Cockpit and check the number if the number of registrations increases when you register your device



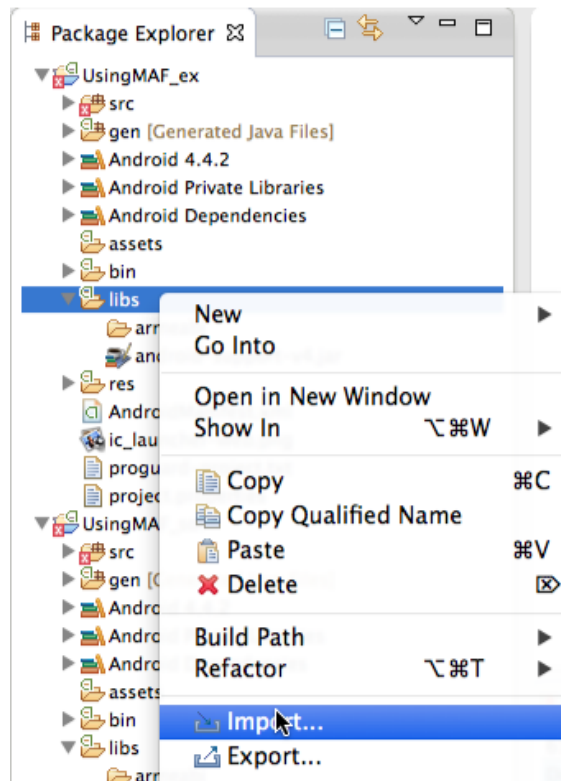
The screenshot shows the SAP Mobile Platform Administration and Monitoring web interface. The 'APPLICATIONS' tab is selected, displaying a table of registered applications. The table has columns for Application ID, Name, Type, Vendor, Status, Registrations, and Logs. A red box highlights the 'Registrations' column for the application 'com.sap.flight', which shows a value of 58.

| Application ID | Name | Type | Vendor | Status | Registrations | Logs |
|----------------|-------------|--------|---------|--------|---------------|------|
| com.sap.flight | Flight Demo | Native | sap.com | ✓ | 58 | |

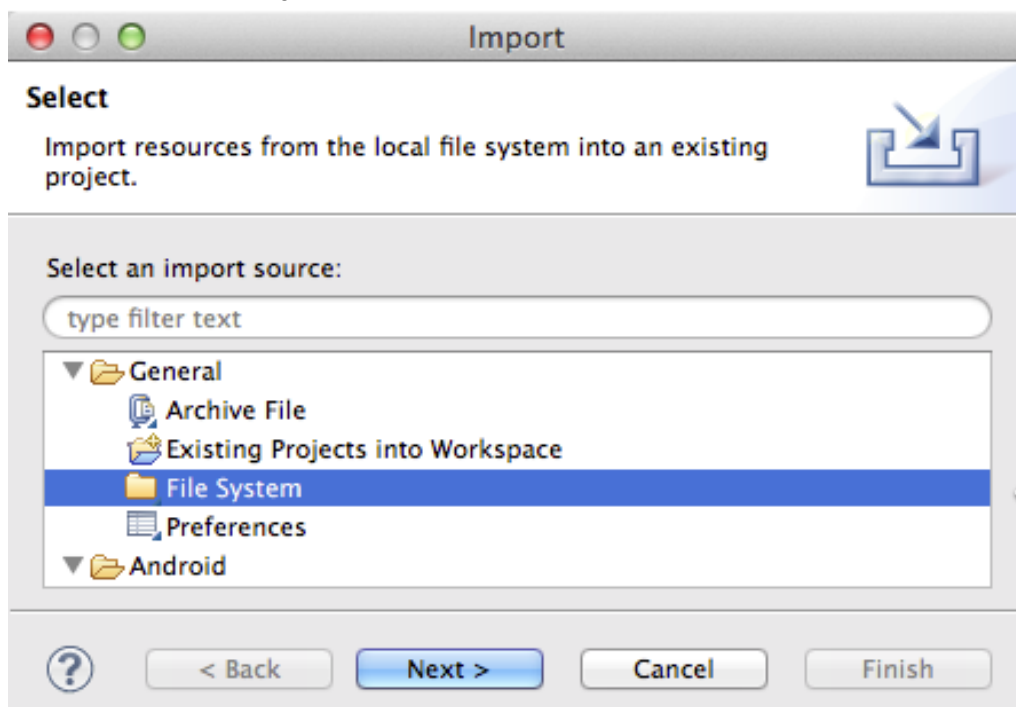
5. Appendix

Appendix A – Importing OData Libraries

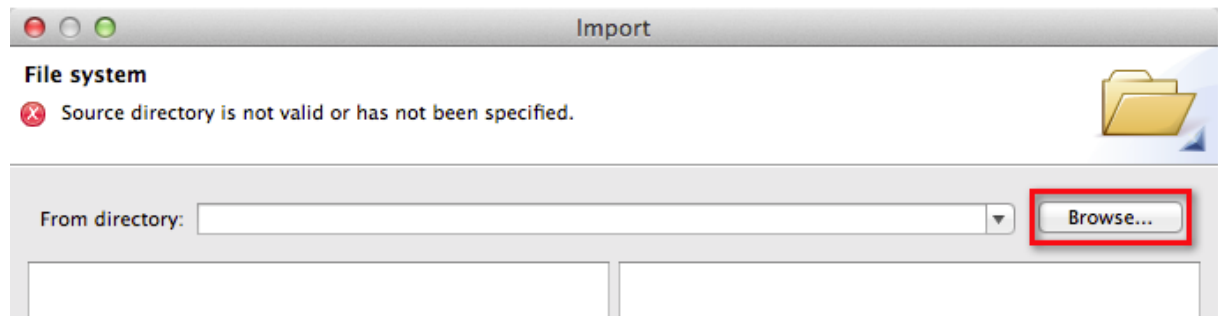
1. Select the libs folder and right-click to open the context menu and select Import



2. Select General -> File System and click Next



3. In the Import window, click Browse

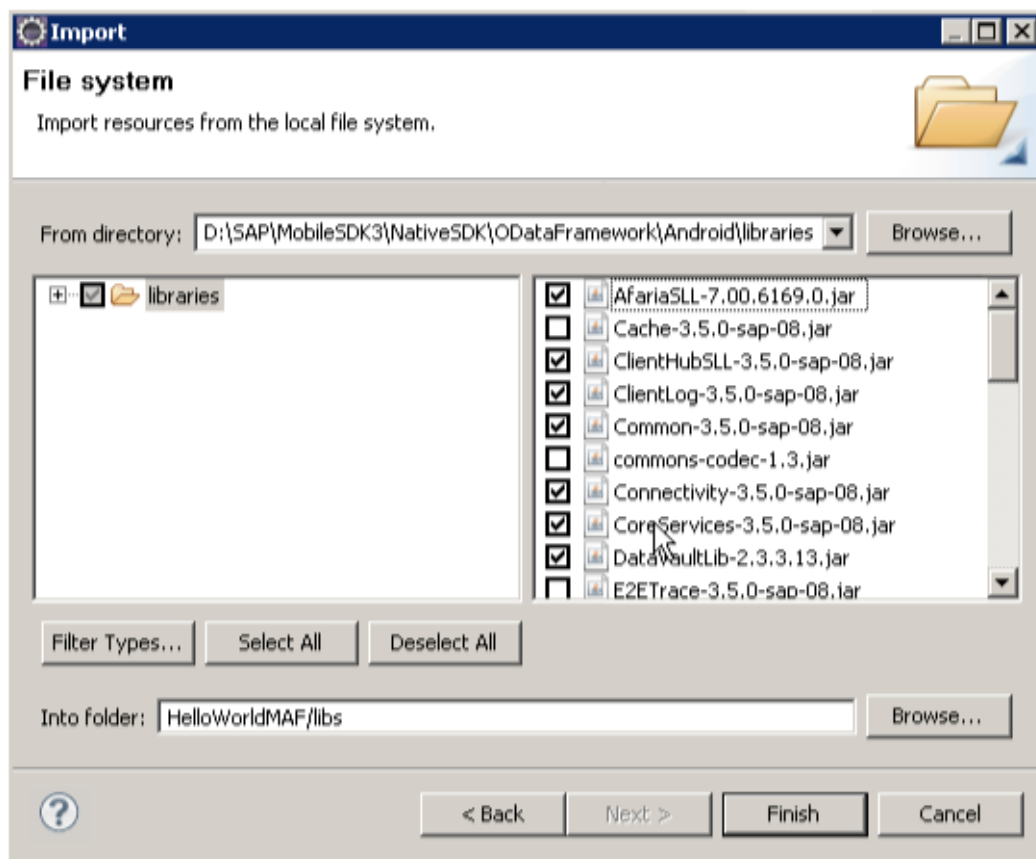


4. Locate your SMP 3.0 SDK installation folder. You will find the jar files in the following folders:

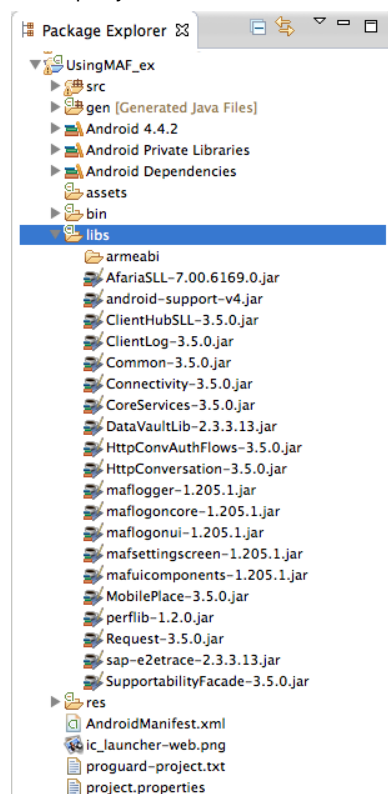
- a. <Client SDK dir>\NativeSDK\ODataFramework\Android\libraries
- b. <Client SDK dir>\NativeSDK\MAFReuse\Android\libraries

5. Select the following jar and click Finish

- | | | |
|--------------------|-------------------------|----------------------------|
| • AfariaSLL.jar | • HttpConvAuthFlows.jar | • MobilePlace.jar |
| • ClientHubSLL | • HttpConversation.jar | • perflib.jar |
| • ClientLog.jar | • maflogoncore.jar | • Request.jar |
| • Common.jar | • maflogger.jar | • sap-e2etrace.jar |
| • Connectivity.jar | • maflogonui.jar | • SupportabilityFacade.jar |
| • CoreServices.jar | • mafuicomponents.jar | |
| • DataVaultLib.jar | • mafsettingscreen.jar | |



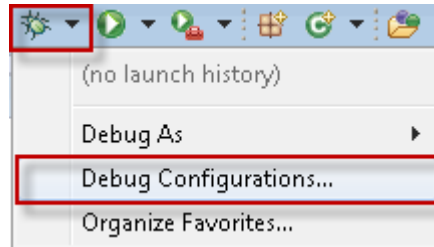
6. Your project should look like the image below



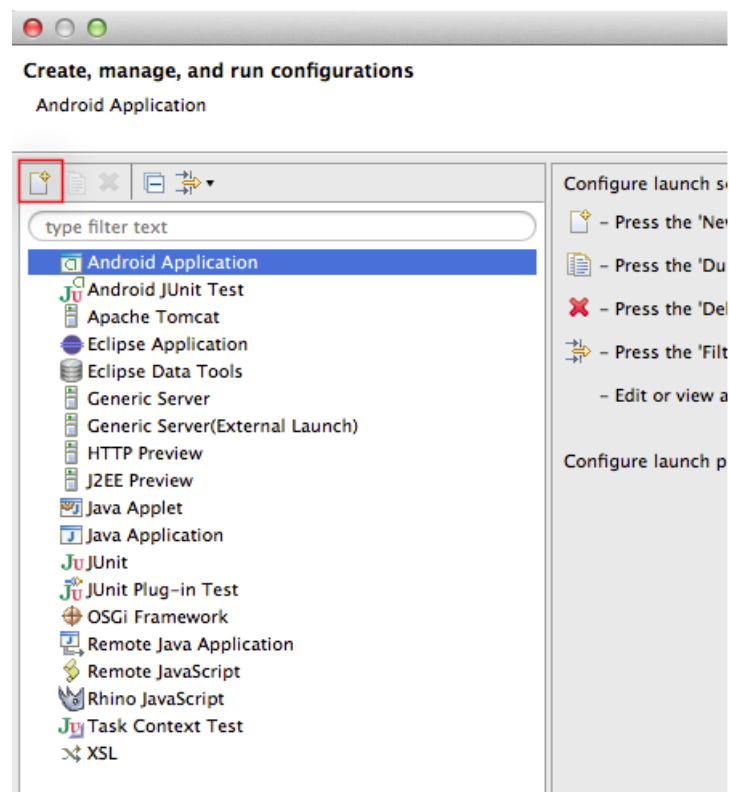
7. Save your changes

Appendix B – Creating Debug Configuration

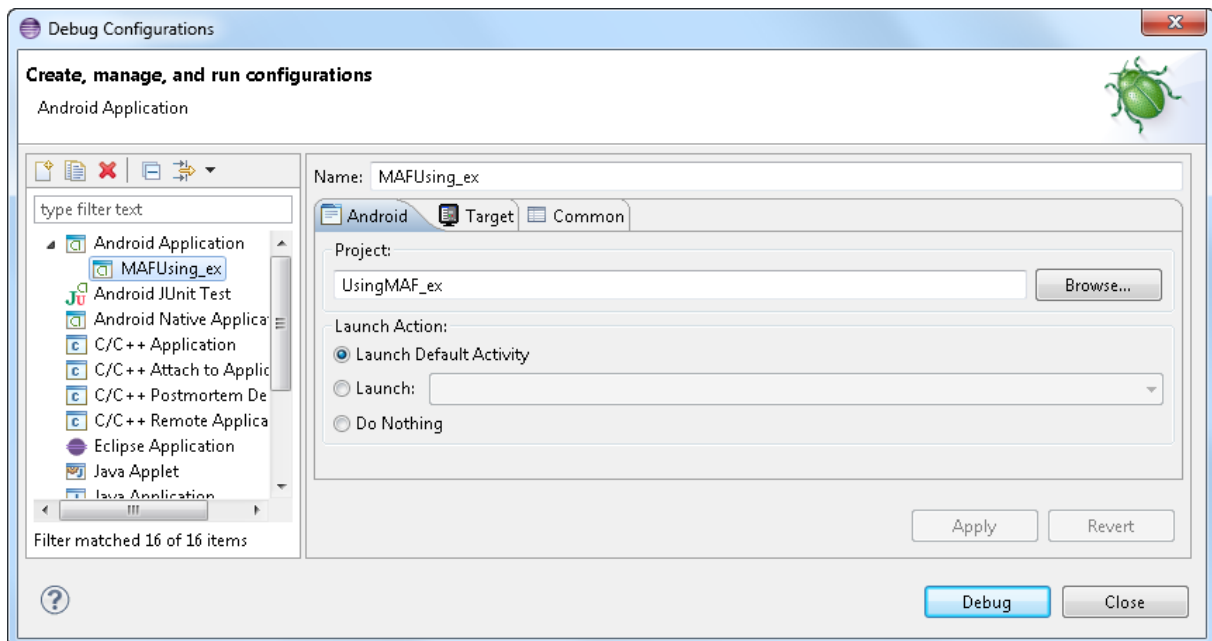
1. In Eclipse, go to the Debug icon and select Debug Configurations



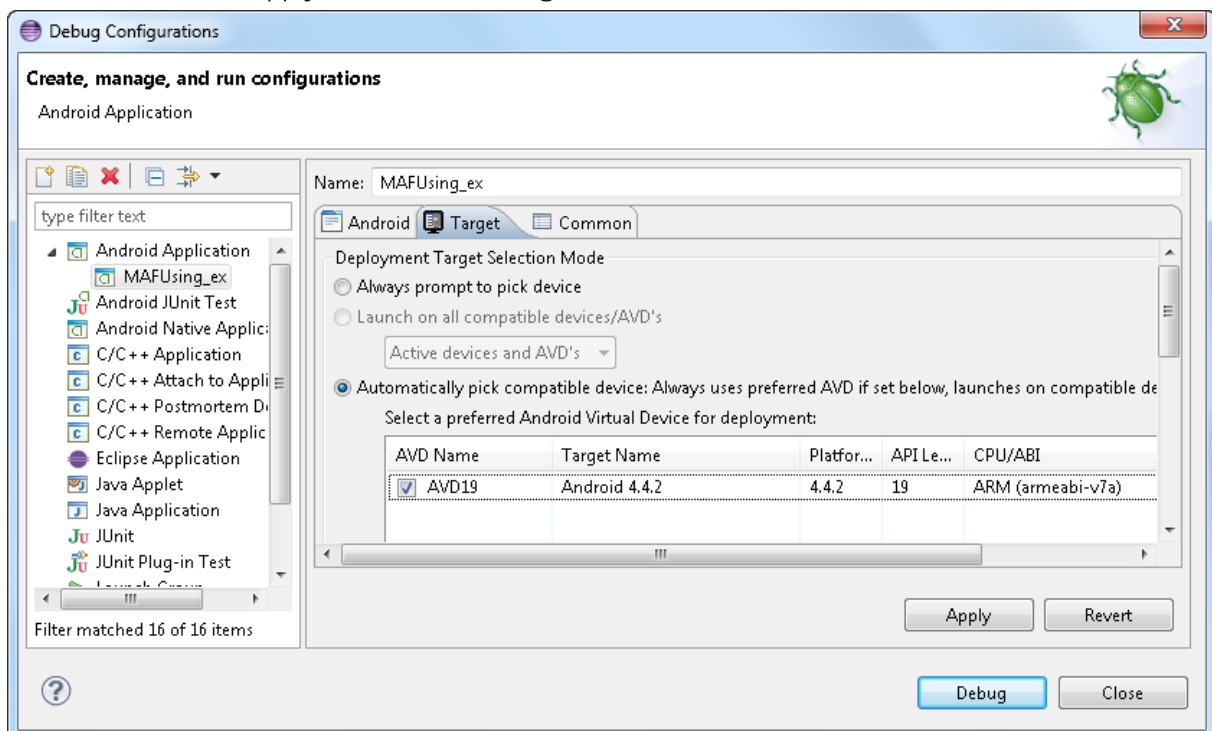
2. Select Android Application and click on the Create icon

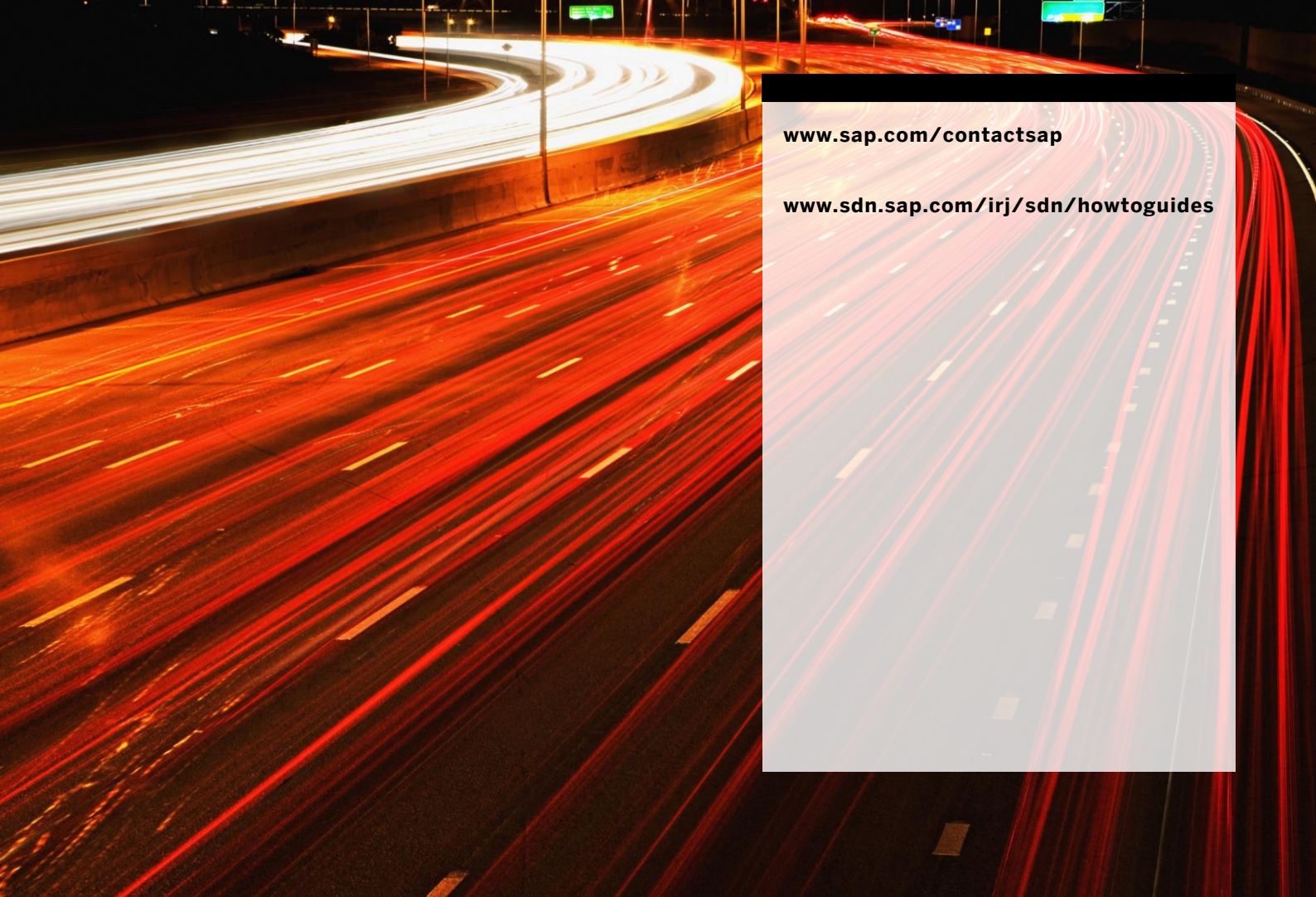


3. Select the Android tab and select the corresponding MAFUsing_ex.



4. Select the Target tab and make sure you have one Android Virtual Device (AVD) selected. Then click on Apply and then on Debug.





www.sap.com/contactsap

www.sdn.sap.com/irj/sdn/howtoguides

© 2013 SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Please see

<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>

for additional trademark information and notices.



The Best-Run Businesses Run SAP™