

TÀI LIỆU PLUGIN-X CHO FACEBOOK.

1. Giới thiệu

- Đây là tài liệu hướng dẫn viết plugin-x cho facebook nói riêng và plugin-x cho plugin-x nói chung cho nền tảng android.

2. Các thư viện và bản cài đặt cần có.

2.1 Thư viện :

- Facebook sdk. Bản mới nhất là 3.8
- libFacebook.jar (thư viện facebook).
- libProtocol.jar (thư viện protocols).

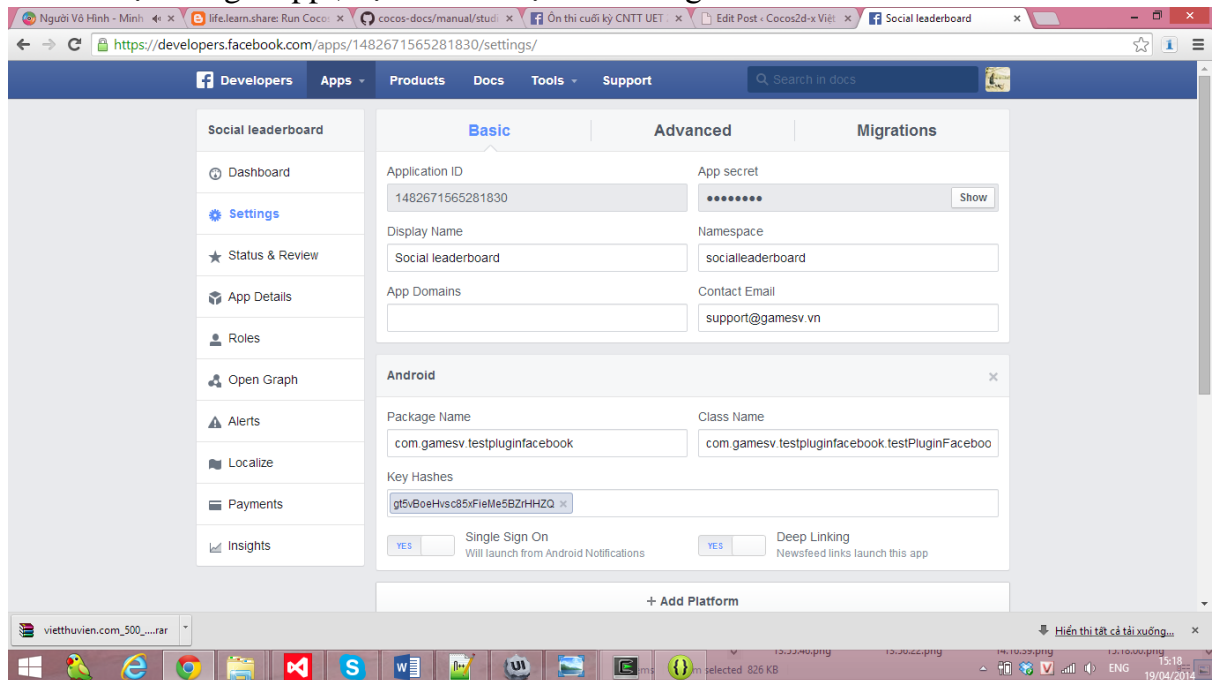
2.2 Các phần mềm cần có.

- Eclipse.
- Android-Sdk.
- Android –ndk.
- Ant bản (1.93 trở lên).

3. Hướng dẫn viết.

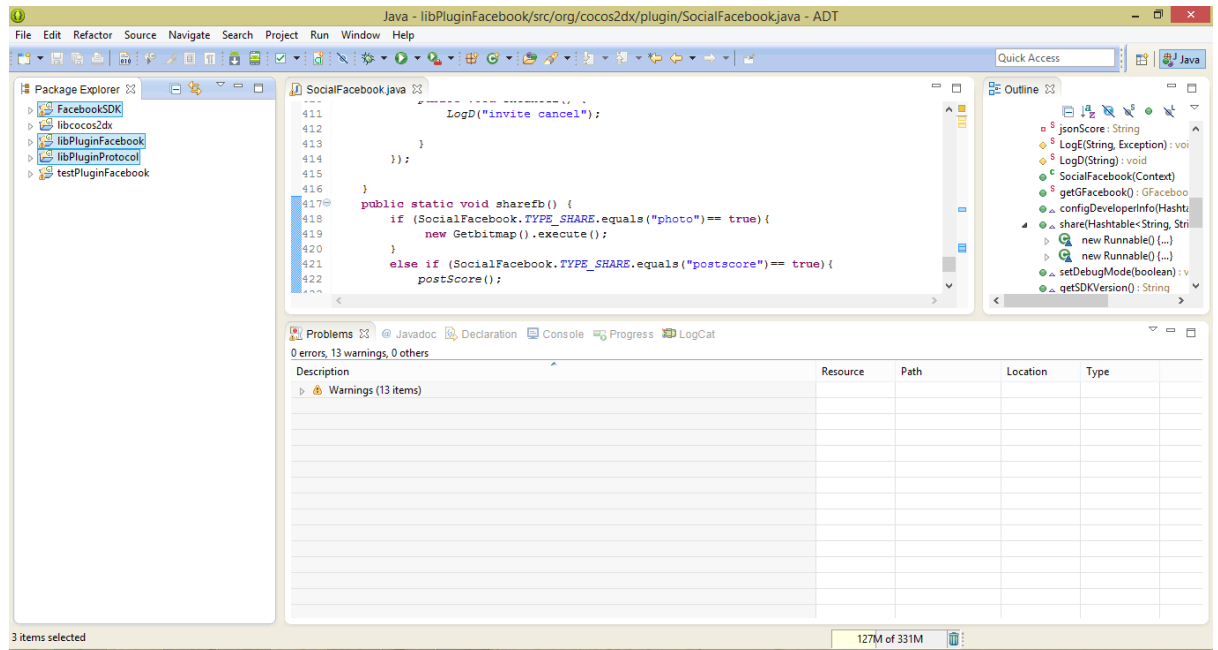
3.1 Tạo app trên trang developer của facebook.

- Vào trang <https://developers.facebook.com/> , chọn vào tab App->creat App.
- Lưu ý : namespace và displayname các chữ cái phải giống nhau , không nên cho thêm các kí tự đặc biệt vào trong namespace .
- Sau khi tạo xong 1 app , bạn sẽ có được các thông tin như sau



3.2 Import các thư viện vào eclipse.

Import các thư viện vào eclipse , bao gồm facebook sdk , libFacebookPlugin (Nếu chưa có cái này , copy lấy proj.android của twitter sau đó về sửa để tạo thành project của libFacebookPlugin) , libProtocols.



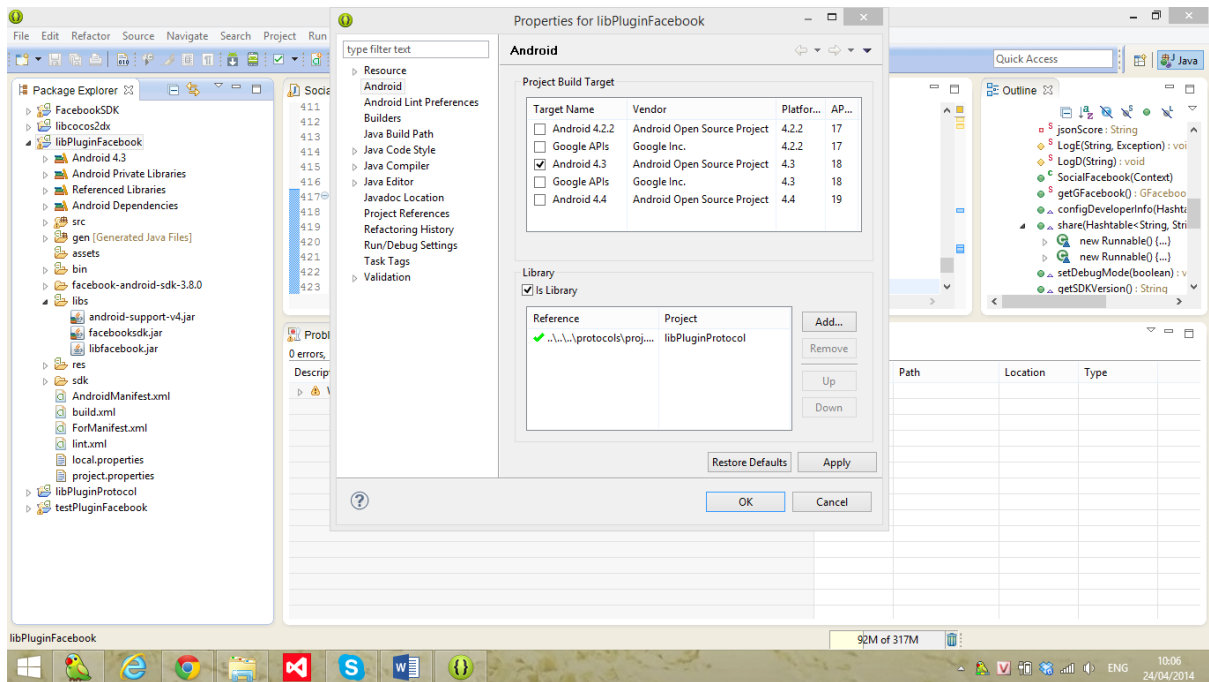
Như vậy 3 thư viện facebook sdk, libPluginFacebook , libPluginProtocols đã được import vào eclipse.

3.3 Cấu hình thư cho thư viện libPluginFacebook.

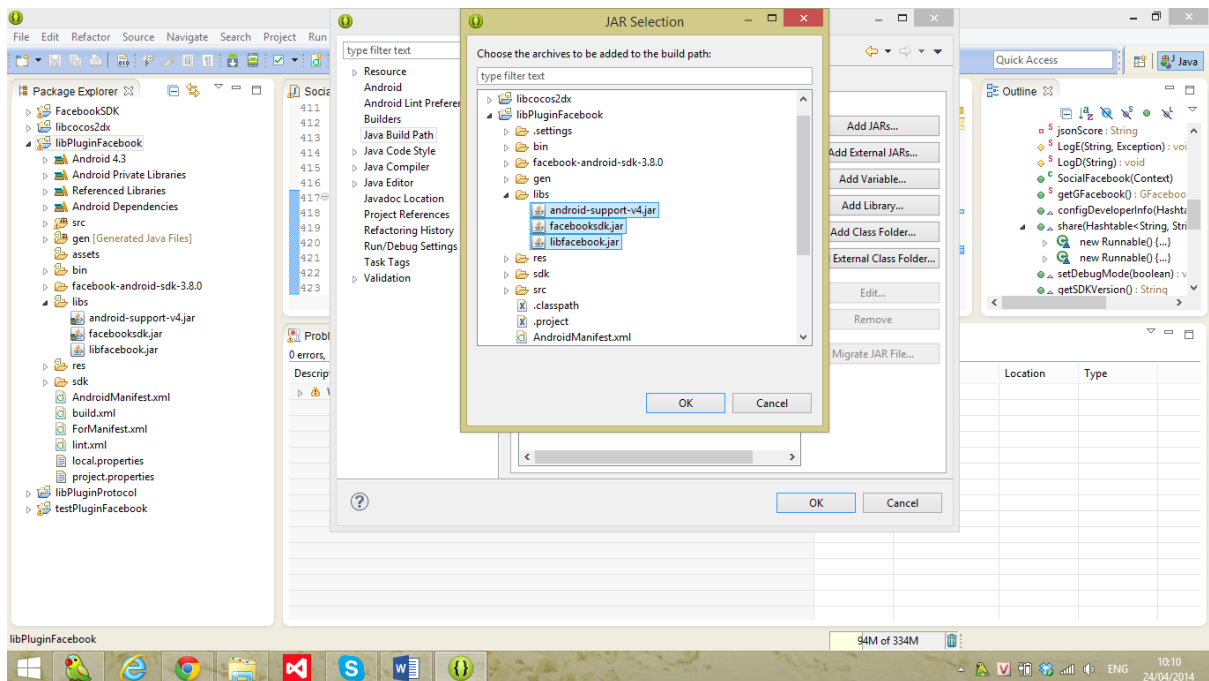
Sau khi import thư viện vào rồi , bắt đầu cấu hình các thư viện liên quan cho thư viện facebook. Đầu tiên bạn copy tất cả các file .jar của thư viện facebook , là facebook.sdk, thư viện libFacebook.jar (đây là thư viện của facebook đã định nghĩa các hàm có sẵn cho facebook).vào thư mục libs của libPluginFacebook. Sau đó chuột phải vào project, chọn properties-> Android ->Add (ở phần library) thư viện libPluginProtocols .

Tick vào ô isLibrary . Phần chọn Project Build Target tick vào bản Android mà bạn muốn build nhé.

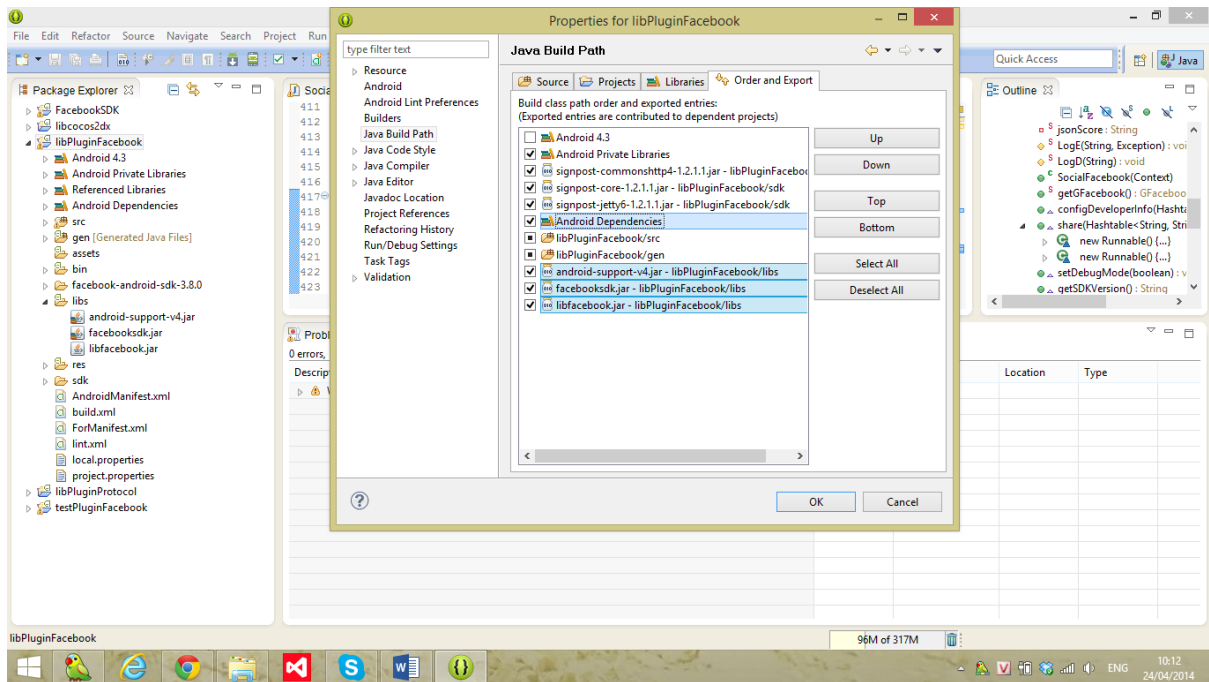
- Lưu ý : Bạn có thể copy thư viện libPluginProtocols.jar vào thư mục libs của project . Nhưng vì ở đây mình làm cho trường hợp bạn có sửa đến thư viện libPluginProtocols.



Tiếp đó chọn đến Java Build Path, chọn Add JARs , vào thư mục libs của libPluginFacebook , add tất cả các thư viện vào project -> chọn OK



Chuyển sang tab Order and Export , tick vào tất cả các ô của thư viện vừa mới add vào.



Như vậy , đã cấu hình xong cho project libPluginFacebook.

3.4 Xây dựng plugin libPluginFacebook.

Sau khi đã cấu hình xong cho project , bạn tạo 1 class trong thư project .

Ở đây mình tạo class SocialFacebook.java .

Với mục đích không cần phải viết các class để gọi từ c++ sang java . Thì mình cho class này implement 1 class có sẵn trong thư mục protocols đó là InterfaceSocial.

Khi implement InterfaceSocial thì mình định lại 4 class có sẵn từ nó :

```
@Override
public void configDeveloperInfo (Hashtable<String,
String> cpInfo);

@Override
public void share(Hashtable<String, String> info);

@Override
public void setDebugMode (boolean debug);

@Override
public String getSDKVersion()
```

ở đây mình sẽ nói rõ mục đích của 4 hàm , và giải thích lại 2 method quan trọng là hàm configDeveloperInfo và method share .

- Phương thức configDeveloperInfo là phương thức để mình viết các cấu hình cho các sdk , ví dụ app_id của nó , quyền truy cập của mạng xã hội nào đó.

- Phương thức Share là phương thức để mình gọi các hàm chức năng của plugin , ví dụ ở trong phương thức share mình có thể định nghĩa để gọi đến chức năng postScore , login , share link... của facebook.
- Phương thức setDebugMode là set debug cho plugin.
- Phương thức getSDKVersion() là để trả lại xem version của phiên bản sdk.

3.4.1 Phương thức ConfigDeveloperInfo.

```

@Override
-   public void configDeveloperInfo(Hashtable<String, String>
cpInfo) {
-       LogD("initDeveloperInfo invoked " +
cpInfo.toString());
-       try {
-           SocialFacebook.CONSUMER_KEY =
cpInfo.get("FacebookKey");
-           SocialFacebook.CONSUMER_SECRET =
cpInfo.get("FacebookSecret");
-           LogD("key : " + SocialFacebook.CONSUMER_KEY);
-           LogD("secret : " +
SocialFacebook.CONSUMER_SECRET);
-           // Phần này để mình set quyền permission cho ứng dụng của
//facebook
-           GPermissions[] permissions = new
GPermissions[] {
-               GPermissions.BASIC_INFO,
-               GPermissions.PUBLISH_ACTION,
-               GPermissions.PUBLISH_STREAM,
-           };
-           GFacebookConfiguration gFacebookConfiguration
= new GFacebookConfiguration.Builder()
-
-               .setAppId(SocialFacebook.CONSUMER_KEY) // set app_id của
facebook
-
-               .setPermissions(permissions).build();
-
-           GFacebook.setConfiguration(gFacebookConfiguration);
-
-           // Nếu như khởi tạo plugin bị lỗi return ;
-           if(isInitialized) {
-               return;
-           }
-           isInitialized = true;
-           // truyền activity cho class của mình,
-           mContext = (Activity) PluginWrapper.getContext();
-           // Khởi tạo plugin
-           gFacebook = GFacebook.getInstance(mContext);
-
-           } catch (Exception e) {
-               LogE("Developer info is wrong!", e);
-           }
-       }

```

```
-
- }
```

3.4.2 Phương thức Share.

```
@Override
public void share(Hashtable<String, String> info) {
    // biến info ở đây là các nội dung bạn truyền từ c++ sang java.
    LogD("share invoked " + info.toString());

    // Đây là mình get thông tin để truyền lên facebook
    SocialFacebook.TYPE_SHARE = info.get("typeShare");
    if ( SocialFacebook.TYPE_SHARE.equals("photo") == true) {
        SocialFacebook.DESCRPTION_SHARE =
            info.get("Description");
        SocialFacebook.DIR_PHOTO = info.get("dirPhoto");
        shareFile = SocialFacebook.DIR_PHOTO;
    }
    else {
        SocialFacebook.LINK_SHARE = info.get("LinkShare");
        SocialFacebook.NAME_SHARE = info.get("Name");
        SocialFacebook.CAPTION_SHARE = info.get("Caption");
        SocialFacebook.PICTURE_SHARE = info.get("LinkPicture");
        SocialFacebook.DESCRPTION_SHARE = info.get("Description");
    }

    if (! networkReachable()) {
        shareResult(SocialWrapper.SHARERESULT_FAIL, "Network error!");
        return;
    }

    if (! isInitialized) {
        shareResult(SocialWrapper.SHARERESULT_FAIL, "Initialize
        failed!");
        return;
    }

    // Kiểm tra login hay chưa , nếu chưa thì login
    if (gFacebook.isLogin() == false ){
        PluginWrapper.runOnMainThread(new Runnable() {

            @Override
            public void run() {
                gFacebook.login(mLoginListener);
            }

        });

        return;
    }

    // Nếu đã login rồi thì thực hiện các chức năng share , post ..
    PluginWrapper.runOnMainThread(new Runnable() {
        @Override
        public void run() {
            managerFacebook();
        }
    });
}
```

Tiếp đến managerFacebook() method , đây là phương thức để mình viết quản lý để gọi các chức năng của facebook.

```
public static void managerFacebook() {  
    // Nếu là post photo thì gọi phương thức postphoto  
    if (SocialFacebook.TYPE_SHARE.equals("photo")== true) {  
        new Getbitmap().execute();  
    }  
    // post diem  
    else if (SocialFacebook.TYPE_SHARE.equals("postscore")== true) {  
        postScore();  
    }  
    // share link  
    else if (SocialFacebook.TYPE_SHARE.equals("sharelink")== true) {  
        publishFeedDialog();  
    }  
    // lấy danh sách điểm của bạn bè.  
    else if (SocialFacebook.TYPE_SHARE.equals("getscore")== true) {  
        getScore();  
    }  
    // invite bạn bè  
    else if ( SocialFacebook.TYPE_SHARE.equals("invite")== true) {  
        functionInvite();  
    }  
}
```

Tiếp đến là 1 phương thức lấy biến gFacebook để gọi sang cho bên class. Java của project test.

```
public static GFacebook getGFacebook() {  
    return gFacebook;  
}
```

Tiếp đến là các phương thức trả về kết quả từ java sang bên c++ . ở đây mình có viết 2 hàm onShareResult() và onScoreResult();

Mục đích của hàm onShareResult() để trả về việc kết quả thực hiện của 1 chức năng nào đó có thể là failed , thành công hay exception..

Còn onScoreResult() để trả về kết quả điểm khi mình lấy từ facebook về . (Bạn có thể dùng với mục đích khác).

Vì hàm onShareResult() đã có sẵn ở trong class của protocols , còn hàm onScoreResult() là do mình viết thêm , để gọi được sang c++ thì mình phải định nghĩa hàm đó ở trong thư mục protocols.

Mở file ProtocolsSocial.cpp ở trong thư mục plattform/android. Thêm vào 2 hàm sau .

```
JNIEXPORT void JNICALL  
Java_org_cocos2dx_plugin_SocialWrapper_nativeOnScoreResult (JNIEnv* env,  
jobject thiz, jstring className, jint ret, jstring msg)  
{
```

```

        std::string strMsg = PluginJniHelper::jstring2string(msg);
        std::string strClassName =
PluginJniHelper::jstring2string(className);
        PluginProtocol* pPlugin =
PluginUtils::getPluginPtr(strClassName);
        PluginUtils::outputLog("ProtocolSocial",
"nativeOnScoreResult(), Get plugin ptr : %p", pPlugin);
        if (pPlugin != NULL)
        {
            PluginUtils::outputLog("ProtocolSocial",
"nativeOnScoreResult(), Get plugin name : %s", pPlugin->getPluginName());
            ProtocolSocial* pSocial =
dynamic_cast<ProtocolSocial*>(pPlugin);
            if (pSocial != NULL)
            {
                pSocial->onScoreResult((ShareResultCode) ret,
strMsg.c_str());
            }
        }
    }
}

```

```

void ProtocolSocial::onScoreResult(ShareResultCode ret, const char*
msg)
{
    if (m_pListener)
    {
        m_pListener->onShareResult(ret, msg);
    }
    else
    {
        PluginUtils::outputLog("ProtocolSocial", "Result listener
is null!");
    }
    PluginUtils::outputLog("ProtocolSocial", "Score result is : %d(%s)",
(int) ret, msg);
}

```

Tiếp đến , bạn sửa cả trong ProtocolsSocial.h ở trong thư mục include , thêm 2 proptype

```

class ShareResultListener
{
public:
virtual void onShareResult(ShareResultCode ret, const char* msg) = 0;
virtual void onScoreResult(ShareResultCode ret, const char* msg) = 0;
};

class ProtocolSocial : public PluginProtocol
{
    ...

    void onScoreResult(ShareResultCode ret, const char* msg);
}

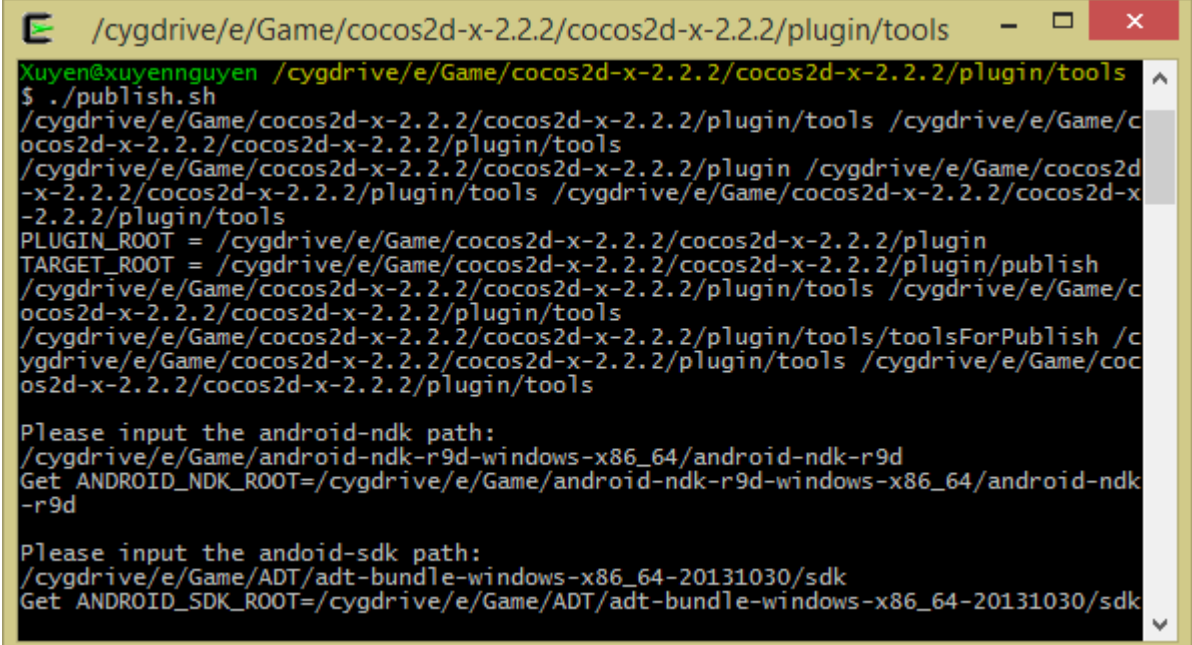
```

3.5 Buid project libPluginFacebook và libPluginProtocols bằng Cygwin.

- Trước tiên , mở file config.sh trong thư mục tools của plugin. Thêm thư viện mà bạn muốn build .
Ví dụ như ở đây , mình chỉ muốn build thư viện facebook . Nên chỉ cần viết là :

```
#define plugins array
export ALL_PLUGINS=( "facebook")
```

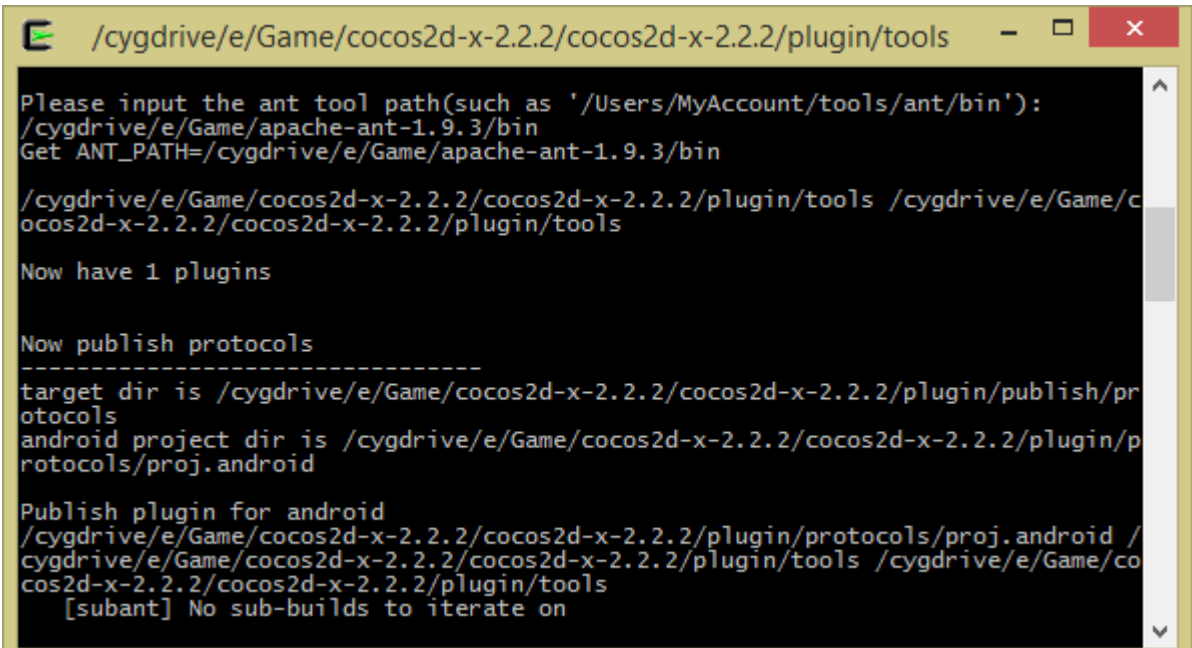
- Tiếp đến mở Cygwin , cd đến thư mục tools ở trong plugin. Chạy lệnh ./publish.sh
Chương trình sẽ yêu cầu nhập vào đường dẫn của android-ndk , android-sdk, ant-bin . Bạn sẽ nhập các đường dẫn cho chuẩn đường dẫn đến các thư mục đó.



```
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
Xuyen@xuyennguyen /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
$ ./publish.sh
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/c
ocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin /cygdrive/e/Game/cocos2d
-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x
-2.2.2/plugin/tools
PLUGIN_ROOT = /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin
TARGET_ROOT = /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/publish
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/c
ocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools/toolsForPublish /c
ygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/coc
os2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

Please input the android-ndk path:
/cygdrive/e/Game/android-ndk-r9d-windows-x86_64/android-ndk-r9d
Get ANDROID_NDK_ROOT=/cygdrive/e/Game/android-ndk-r9d-windows-x86_64/android-ndk
-r9d

Please input the android-sdk path:
/cygdrive/e/Game/ADT/adt-bundle-windows-x86_64-20131030/sdk
Get ANDROID_SDK_ROOT=/cygdrive/e/Game/ADT/adt-bundle-windows-x86_64-20131030/sdk
```



```
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

Please input the ant tool path(such as '/Users/MyAccount/tools/ant/bin') :
/cygdrive/e/Game/apache-ant-1.9.3/bin
Get ANT_PATH=/cygdrive/e/Game/apache-ant-1.9.3/bin

/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/c
ocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

Now have 1 plugins

Now publish protocols
-----
target dir is /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/publish/pr
otocols
android project dir is /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/p
rotocols/proj.android

Publish plugin for android
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/protocols/proj.android /
cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/co
cos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
[subant] No sub-builds to iterate on
```

Đến đây là chương trình sẽ dịch thư viện protocols và 1 thư viện plugin facebook. Khi lần đầu tiên dịch thư mục protocols . Nếu gặp trường hợp báo lỗi này.

```
Build C++ code
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools/toolsForPublish /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/protocols/proj.android /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/protocols/proj.android /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
make: Entering directory '/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/protocols/proj.android'
Android NDK: WARNING:jni/Android.mk:PluginProtocolStatic: LOCAL_LDLIBS is always ignored for static libraries
[armeabi] Cygwin      : Generating dependency file converter script
[armeabi] Compile++ thumb: PluginProtocolStatic <= PluginFactory.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= PluginJniHelper.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= PluginUtils.cpp
jni/../../platform/android/PluginUtils.cpp: In static member function 'static void cocos2d::plugin::PluginUtils::outputLog(char const*, char const*, ...)':
jni/../../platform/android/PluginUtils.cpp:156:53: error: format not a string literal and no format arguments [-Werror=format-security]
cc1plus.exe: some warnings being treated as errors

/cygdrive/e/Game/android-ndk-r9d-windows-x86_64/android-ndk-r9d/build/core/build
```

Thì chỉ cần vào phần protocols/ platform/android . Mở file PluginUtils.cpp xóa đi dòng 156 vì dòng đó chỉ là log của android , xóa đi sẽ không ảnh hưởng gì đến project của mình cả.

Sau khi xóa xong , nếu dịch thành công protocols và facebook thì chương trình sẽ hiện ra kiểu này.

```
jni/../../platform/android/PluginProtocol.cpp:181:5: instantiated from here
jni/../../platform/android/PluginUtils.h:104:13: warning: NULL used in arithmetic [-Wpointer-arith]
[armeabi] Compile++ thumb: PluginProtocolStatic <= ProtocolAnalytics.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= ProtocolIAP.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= ProtocolAds.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= ProtocolSocial.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= PluginManager.cpp
[armeabi] Compile++ thumb: PluginProtocolStatic <= PluginParam.cpp
[armeabi] StaticLibrary : libPluginProtocolStatic.a
make: Leaving directory '/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/protocols/proj.android'

Native build action success.
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
-----
Now publish facebook
-----
target dir is /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/publish/plugins/facebook
android project dir is /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/p
```

```
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

-----
target dir is /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/publish/pl
ugins/facebook
android project dir is /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/p
lugins/facebook/proj.android

Publish plugin for android
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/plugins/facebook/proj.an
droid /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/
Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

BUILD SUCCESSFUL
Total time: 2 seconds

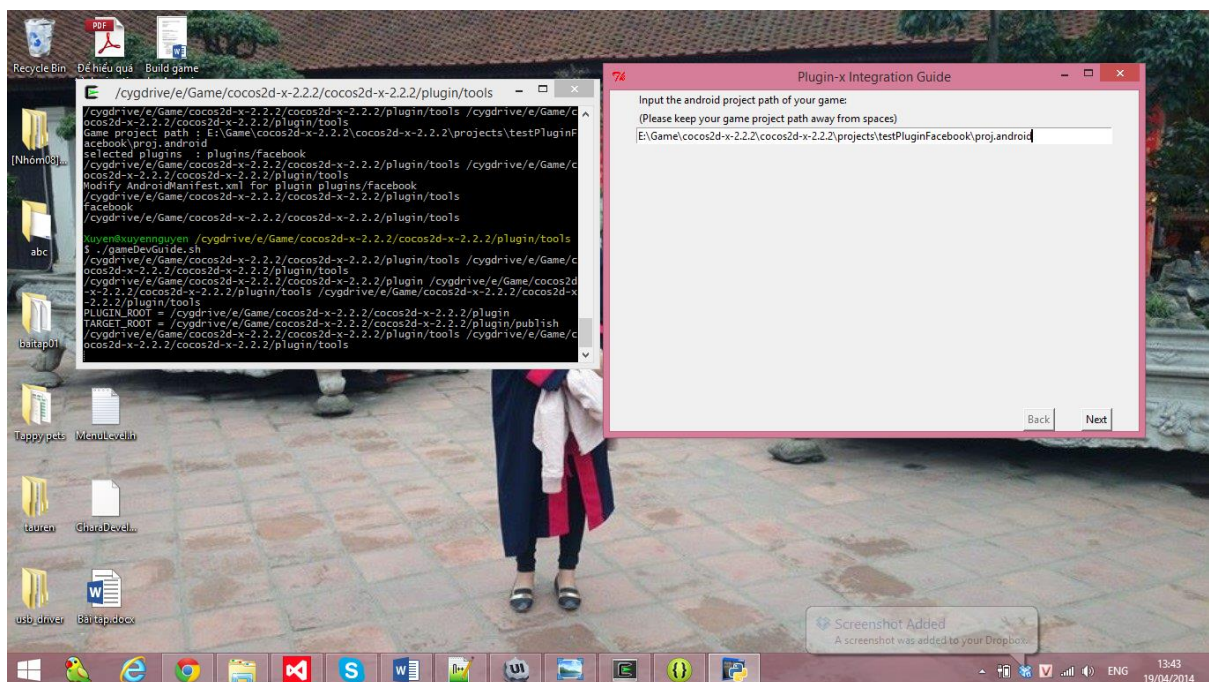
BUILD SUCCESSFUL
Total time: 8 seconds
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools /cygdrive/e/Game/c
ocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

-----
/cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools

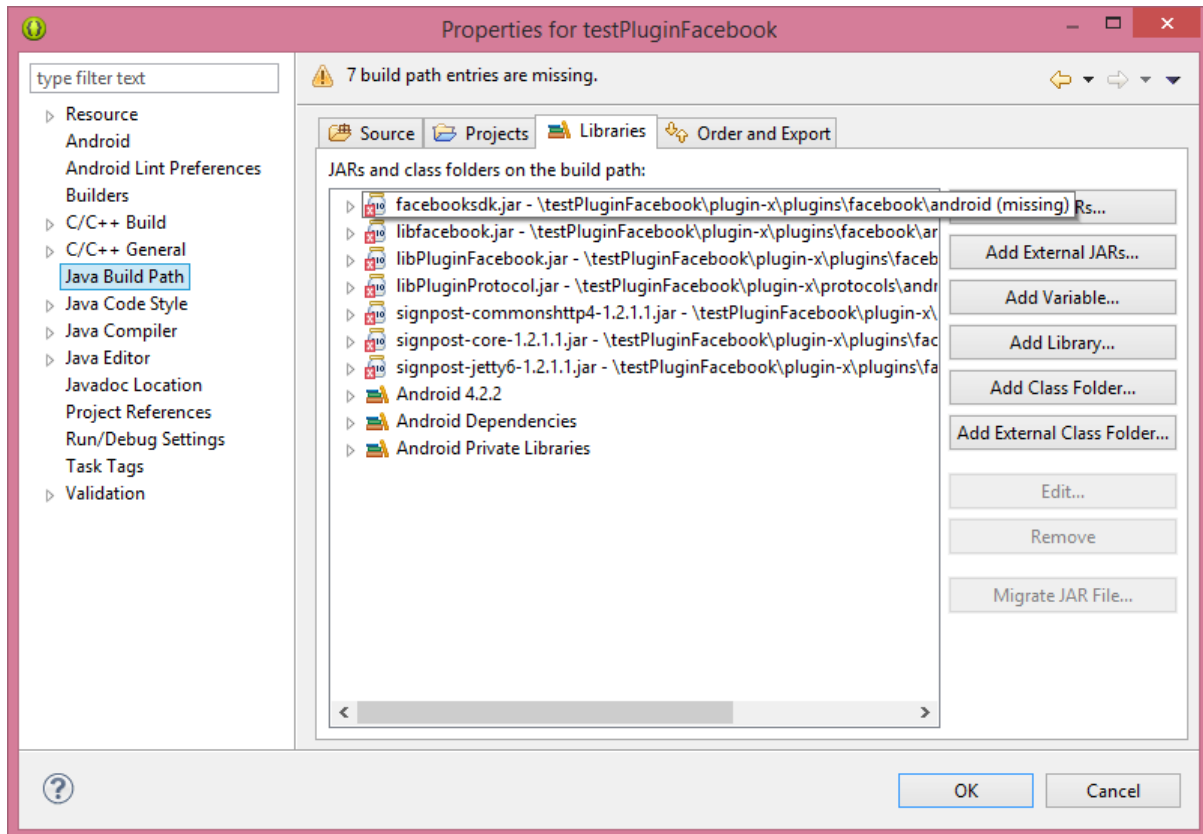
Xuyen@xuyennguyen /cygdrive/e/Game/cocos2d-x-2.2.2/cocos2d-x-2.2.2/plugin/tools
$ |
```

Sau khi dịch thành công thì quyền ng dùng thư mục publish . thêm lệnh chmod.exe 777 -R ./publish/

Sau khi dịch thành công facebook và protocols thì bạn chạy file ./gameDevGuide.sh , sau khi cửa sổ hiện ra thì yêu cầu chỉ đường dẫn project mà bạn muốn tích hợp plugin thì bạn điền vào nhé , rồi chọn next , tiếp theo tick vào ô facebook -> chọn finish . thì lúc đó là hoàn thành việc cấu hình plugin cho dự án project của bạn .



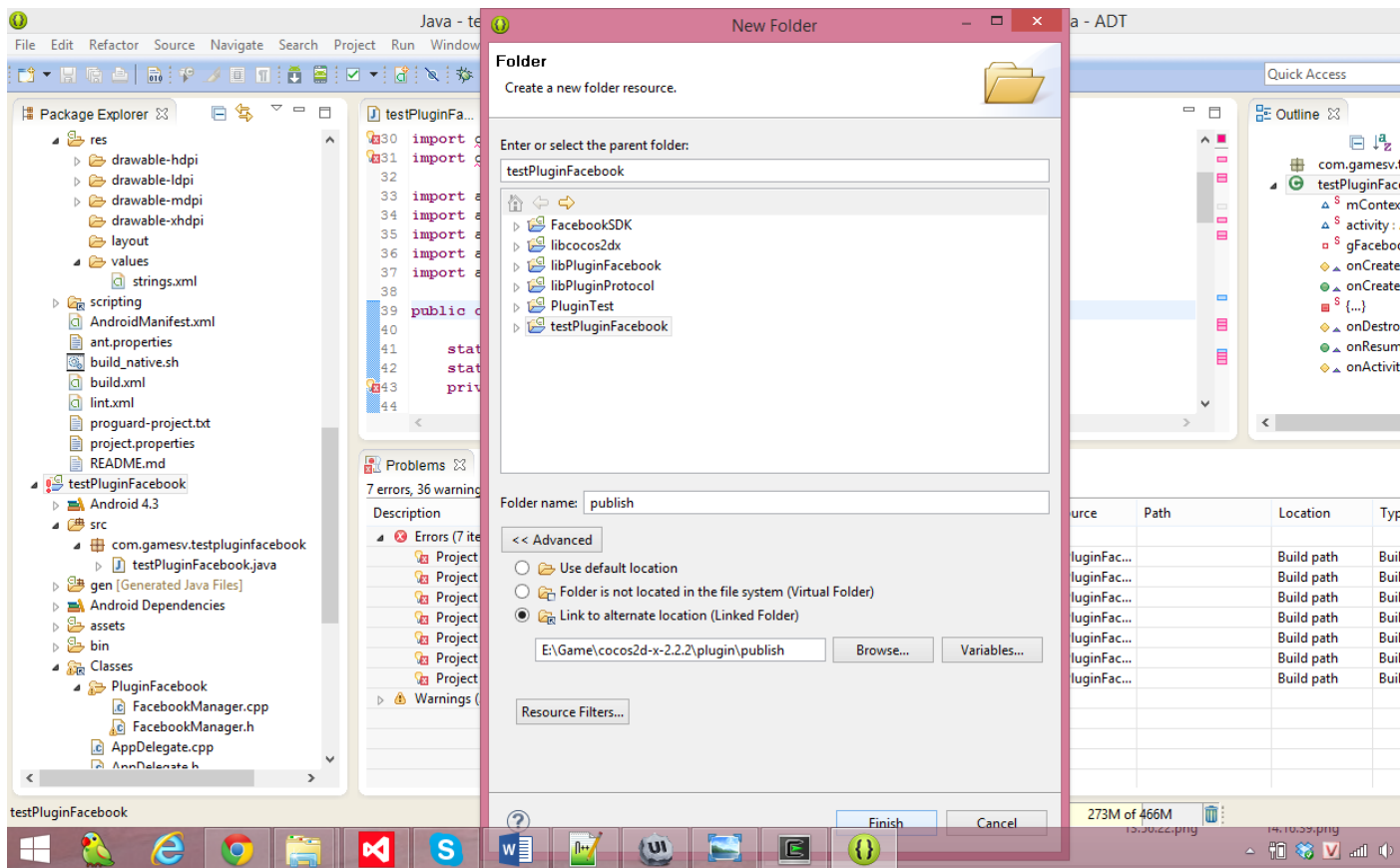
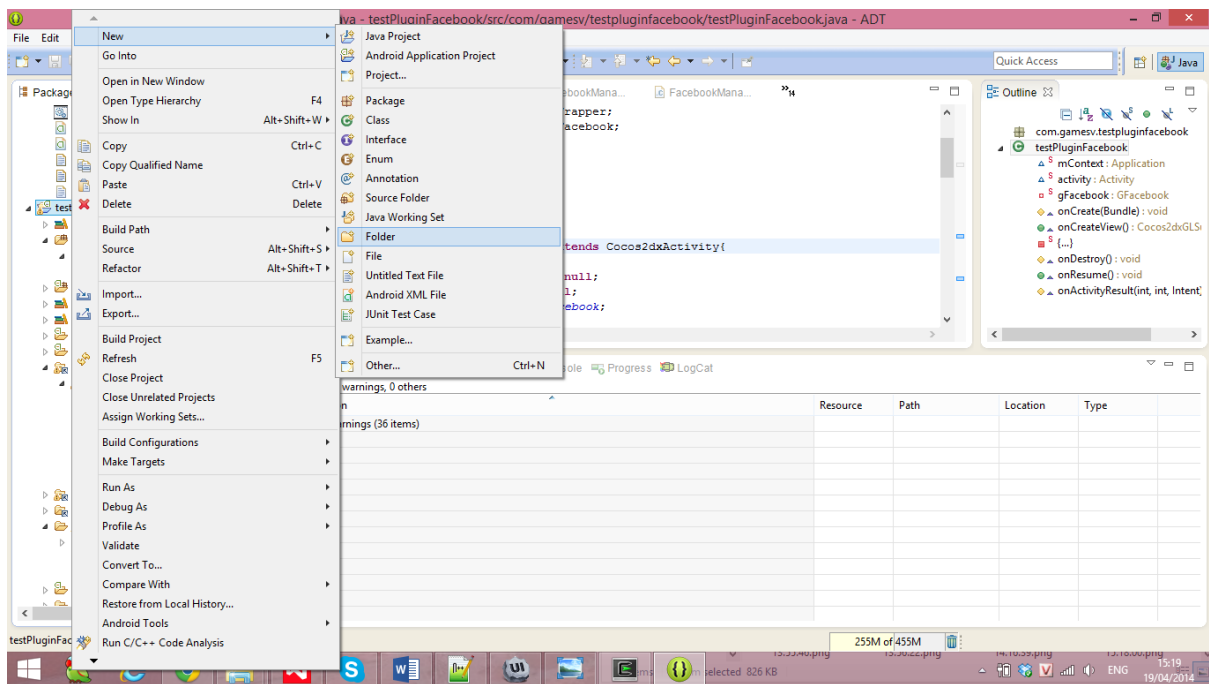
Khi bạn hoàn thành xong bước này thì plugin facebook đã được tích hợp cho project của bạn . Bạn chọn chuột phải vào project game rồi chọn properties , chọn đến cái Java build Path . bạn sẽ thấy , đường dẫn của thư viện bị sai , do tên folder chứa thư viện là plugin-x nhưng chúng ta chưa có.



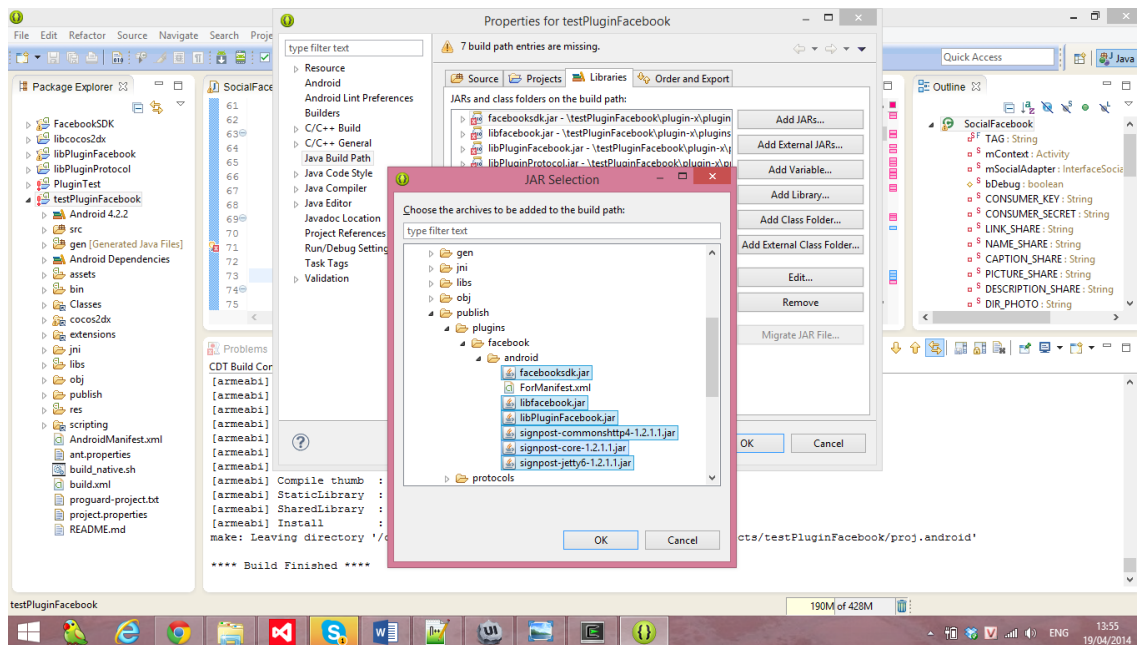
Vì vậy chỉnh sửa lại đường dẫn của các thư viện kia nhé . Để làm cái này đầu tiên bạn phải tạo 1 folder tham chiếu đến folder publish của plugin .

Chúng ta làm theo 2 bước :

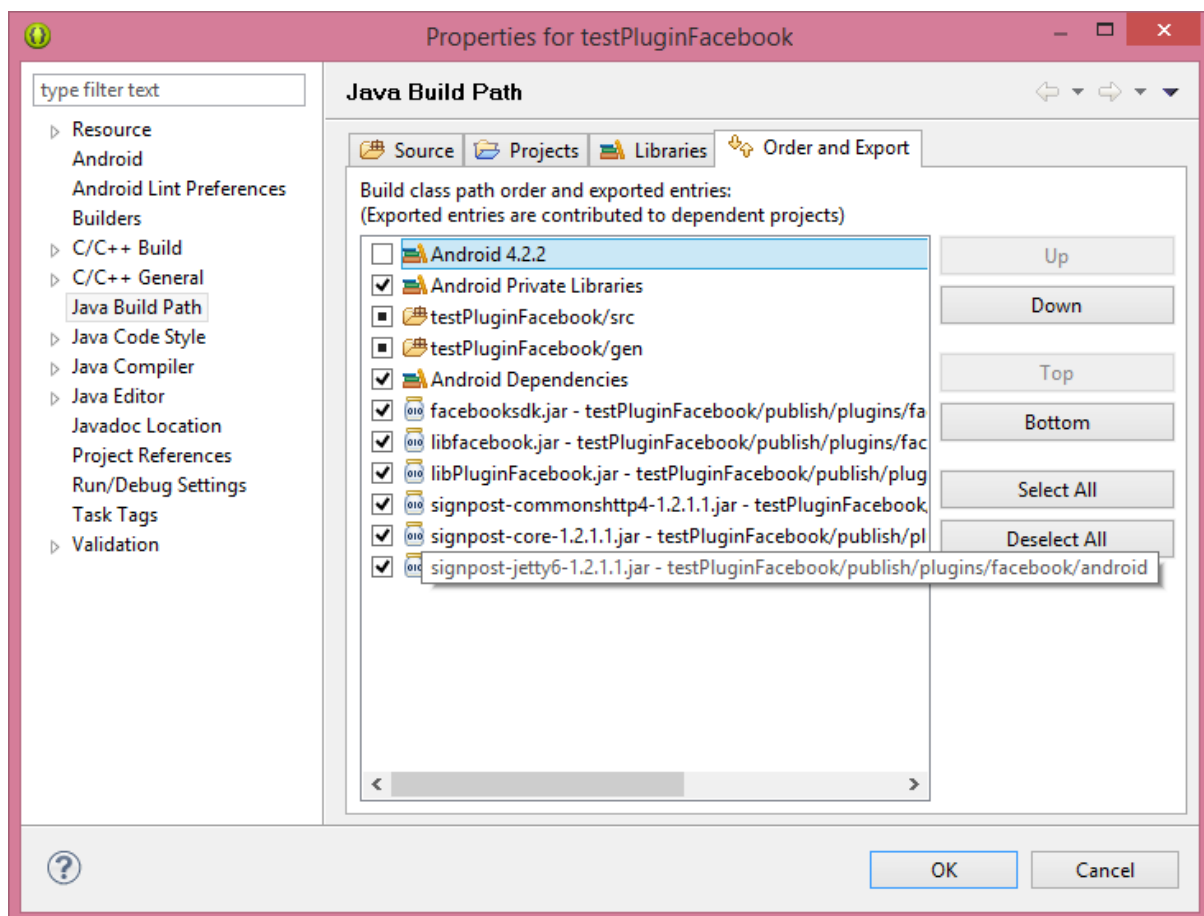
Chuột phải vào project -> chọn new -> folder



Như vậy ta đã tạo folder tham chiếu đến folder publish của plugin , sau này mỗi lần bạn dịch lại plugin thì thư viện .jar sẽ tự động được load lại .
 Tiếp theo bạn sửa lại đường dẫn của các file thư viện của facebook và protocols nhé .
 Bạn chọn chuột phải vào project chọn properties , sau đó chọn đến Java build Path , tiếp đến chọn add JARs -> tìm đến thư mục publis trong project và add tất cả các thư viện của facebook và protocols vào nhé .



Sau đó xoá hết các thư viện cũ báo lỗi missing kia đi nhé , sau đó chọn sang tab order and export , bạn tick vào các thư viện đã add nhé .



Khi đã cấu hình xong plugin vào project game của bạn rồi thì đến phần viết FacebookManager.cpp , đây là class giúp bạn tương tác giữa game và plugin facebook của bạn .

Trong class này bạn kế thừa class public cocos2d::plugin::ShareResultListener có sẵn ở trong thư viện protocols . Mình sẽ giải thích chức năng của các hàm dưới đây .

```
// Đây là hàm để load plugin facebook
void FacebookManager::loadSocialPlugin()
{
    if (s_pRetListener == NULL)
    {
        s_pRetListener = new MyShareResult();
    }
    // init Facebook plugin
    // ở đây là nếu plugin facebook chưa được khởi tạo thì mình
    // sẽ gọi hàm để khởi tạo plugin facebook . Tên "SocialFacebook" chính
    // là class mà bạn đặt ở trong project plugin của nhé . Khi bạn gọi hàm
    // này chương trình sẽ tìm đến và khởi tạo class SocialFacebook.java
    if (!s_pFacebook) {
        s_pFacebook =
dynamic_cast<ProtocolSocial*>(PluginManager::getInstance()-
>loadPlugin("SocialFacebook"));
    }
    if (NULL != s_pFacebook)
    {
        // Các thông tin cấu hình của facebook
        TSocialDeveloperInfo pFacebookInfo;
        pFacebookInfo["FacebookSecret"] =
"5f9ea40176c5f9fdf2b9ee7f78f68288"; // cái này key secret , bạn lấy
ở trong app trên trang developer nhé
        pFacebookInfo["FacebookKey"] = "1482671565281830"; // cái này
chính là app_id
        pFacebookInfo["NameSpace"] = "socialleaderboard"; // namespace
của app
        // pFacebookInfo["DisplayName"] = "Killing F. Birds"; // Hiện
thị lên facebook khi bạn share link lên hay up load ảnh...
        if (pFacebookInfo.empty())
        {
            char msg[256] = { 0 };
            sprintf(msg, "Developer info is empty. PLZ fill your facebook
info in %s(nearby line %d)", __FILE__, __LINE__);
            CCMessagesBox(msg, "Facebook Warning");
        }
        s_pFacebook->setDebugMode(true); // set chế độ debug cho
plugin của bạn
        s_pFacebook->configDeveloperInfo(pFacebookInfo); // gửi các
thông tin cấu hình cho app của bạn
        s_pFacebook->setResultListener(s_pRetListener); // lắng nghe
kết quả trả về
    }
}
```

- Hàm hủy plugin

```

// hàm huỷ load plugin khi bạn ko muốn sử dụng nó nữa
void FacebookManager::unloadSocialPlugin()
{
    if (s_pFacebook)
    {
        PluginManager::getInstance()-
>unloadPlugin("SocialFacebook");
        s_pFacebook = NULL;
    }
}

// Hàm để bạn thực hiện login , share , upload ảnh , post score ..
void FacebookManager::shareByMode(TShareInfo info)
{
    // Khởi tạo phần nghe kết quả trả về .
    if (s_pRetListener == NULL)
    {
        s_pRetListener = new MyShareResult();
    }
    ProtocolSocial* pShare = NULL;
    pShare = s_pFacebook;
    if (pShare) {
        pShare->share(info); // gọi đến hàm share .
    }
}

// Hàm nghe sự kiện trả về từ java về c++ chức năng được thực hiện thành
công hay không ?
void MyShareResult::onShareResult(ShareResultCode ret, const char* msg)
{
    CCLog("Goi den onShareResult");
    char shareStatus[1024] = { 0 };
    sprintf(shareStatus, "Share %s", (ret == kShareSuccess)?
    "Successted" : "Failed");
    m_pLister->onShareResult(ret, msg );
}

// hàm lấy kết quả trả về bên java sang c++ của chức năng lấy điểm ,
// kết quả trả về là 1 chuỗi string json.
void MyShareResult::onScoreResult(cocos2d::plugin::ShareResultCode
ret, const char* msg){
    m_pLister->onScoreResult(ret,msg);
    std::string msg_(msg);
    CCUserDefault::sharedUserDefault()-
>setStringForKey("score",msg_);
    CCUserDefault::sharedUserDefault()->flush();
}

```

Trên đó là class quản lý plugin ở bên c++ giúp bạn thao tác hơn khi .
giờ đến phần bạn tích hợp plugin bên c++ .
giải sử ở trong class helloworldscene.cpp bạn muốn tích hợp plugin vào thì bạn làm
như sau .


```

bool HelloWorld::init()
{
    //////////////////////////////////////
    // 1. super init first
    if ( !CCLayer::init() )
    {
        return false;
    }
    // Tích hợp plugin facebook
    cocos2d::plugin::TShareInfo pInfo;
    pInfo["typeShare"]="sharelink"; // đây là mình muốn báo cho plugin
    là mình muốn share link lên facebook
    pInfo["LinkShare"] = "http://gamesv.vn/product/pikachu-hd-2014-
    7.html#&slider1=2"; // link mình muốn share
    pInfo["Name"] = "Game Store"; // tên mình muốn share
    pInfo["Caption"] = "Game hot 2014"; // caption cho game của bạn
    pInfo["LinkPicture"] =
    "http://gamesv.vn/upload/b30202/poster1024x500.png"; // link ảnh
    mình muốn share
    CCString *des = CCString::create("Ha Ha . Tó đã qua level với chiến
    thắng %d sao nhá .Hí hí "); // mô tả game
    pInfo["Description"] = des->getCString();
    FacebookManager *mFace ;
    mFace = FacebookManager::sharedSocialManager(); // khởi tạo cho
    class FacebookManager
    mFace->loadSocialPlugin(); // load plugin
    mFace->shareByMode(pInfo); // hàm share
    mFace->setListener(this); // lắng nghe sự kiện trả về
    return true;
}

//Tiếp theo đến hàm nghe sự kiện trả về từ java về c++. Bạn muốn có hàm này
nghe sự kiện thực hiện thao tác nào đó thì bạn tự định nghĩa trong hàm này
, dựa vào các mã code trả về là fail , success , hay thinking , exception...
void HelloWorld::onShareResult(ShareResultCode ret, const char* msg)
{
}

void HelloWorld::onLoginResult(ShareResultCode ret, const char* msg)
{
}

void HelloWorld::onScoreResult(ShareResultCode ret, const char* msg)
{
    //std::string usr = CCUserDefault::sharedUserDefault()-
    >getStringForKey("score");
    CCLog("%s ", msg);
}

```

Tiếp theo , vì các function ở trong facebook nó cần cần nhận các kết quả từ activity ở trong hàm onActivityResult() . nhưng ở trong class SocailFacebook.java không được kế thừa từ activity, nên chúng ta phải gọi qua class chính của game có kế thừa từ activity . Vậy ở trong class java của game . Mình phải thêm 1 số lệnh sau .

```

package com.gamesv.testpluginfacebook;

import lib.facebook.connection.GFacebook;
import org.cocos2dx.lib.Cocos2dxActivity;
import org.cocos2dx.lib.Cocos2dxGLSurfaceView;
import org.cocos2dx.plugin.PluginWrapper;
import org.cocos2dx.plugin.SocialFacebook;
import android.app.Activity;
import android.app.Application;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;

public class testPluginFacebook extends Cocos2dxActivity {
    static Application mContext = null;

    static Activity activity = null;

    private static GFacebook gFacebook;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // cái này là khởi tạo 1 activity để truyền sang class
        SocialFacebook.java
        PluginWrapper.init(this);
        PluginWrapper.setGLSurfaceView(Cocos2dxGLSurfaceView.getInstance());
        mContext = this.getApplication();
        activity = this;
    }

    public Cocos2dxGLSurfaceView onCreateView() {
        Cocos2dxGLSurfaceView glSurfaceView = new
        Cocos2dxGLSurfaceView(this);
        // testPluginFacebook should create stencil buffer
        glSurfaceView.setEGLConfigChooser(5, 6, 5, 0, 16, 8);
        return glSurfaceView;
    }

    static {
        System.loadLibrary("cocos2dcpp");
    }

    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
    }

    @Override
    public void onResume() {

```

```

// gọi hàm resume cho facebook
gFacebook = SocialFacebook.getGFacebook();
if (gFacebook != null) {
gFacebook = GFacebook.getInstance(this);
}
super.onResume();
}
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
// TODO Auto-generated method stub
// để truyền cho gFacebook nhận kết quả trả về từ facebook .
gFacebook = SocialFacebook.getGFacebook();
if (gFacebook != null) {
gFacebook.onActivityResult(this, requestCode, resultCode, data);
Log.d("code result", "Goi vao cai dialog");
}
super.onActivityResult(requestCode, resultCode, data);
}

```

Sau khi xong phần này , thì ở project game bạn cũng phải thêm 2 dòng này vào file AndroidManifest.xml nhé .

```

<application android:icon="@drawable/icon"
android:label="@string/app_name">
<meta-data
android:name="com.facebook.sdk.ApplicationId"
android:value="@string/app_id" />
<activity
android:name="com.facebook.LoginActivity"
android:label="@string/app_name" />

```

Tiếp đến , chúng ta sửa file build_native.sh , android.mk và main.cpp ở ở trong thư mục hellocpp của project test . Xem mẫu ở trong project đính kèm theo. Như vậy việc tích hợp plugin cho dự án trên nền tảng android đã xong.