



ATID Co.,Ltd

Barcode Scanner Programming Guide

Android Developer Guide

SDK Team

2017-02-22

		Barcode Scanner Programming Guide					
All That Identification		Android Developer Guide			회사		ATID Co.,Ltd
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0

v개정 이력

버전	개정일자	개정사유 ¹	개정내역 ²	작성자
V0.1	2014-05-27	초안	신규 생성	문흥기
V0.2	2014-09-25	초안	문서 보완 및 수정	문흥기
V1.0	2014-10-23	개시	1차 문서 개시	박영호
V2.0	2017-02-22	수정	수정 내역 적용	조영진

¹ 개정사유 : 제정 또는 개정 내용이 이전 문서에 대해 추가/수정/삭제인지 선택 기입

² 개정내역 : 개정이 발생하는 페이지 번호와 변경 내용을 기술

		Barcode Scanner Programming Guide					
Android Developer Guide					회사	ATID Co.,Ltd	
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0

목차

목차	3
1. Intro	4
2. Getting Started	5
Create Project for Android.....	5
Development Environment.....	12
3. Programming Guide.....	20
3.1. Crete and Activity.....	20
3.2 Event Handler.....	20
3.3.1 BarcodeEventLister	21
3.3 Action Operation	21

		Barcode Scanner Programming Guide					
Android Developer Guide					회사	ATID Co.,Ltd	
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0

1. Intro

본 문서는 Barcode Scanner 응용 프로그램을 개발하고자 하는 Android개발자들을 위하여 SDK Library를 사용할 수 있는 개발환경 구축과 SDK Library 사용 방법을 기술하는데 그 목적이 있다.

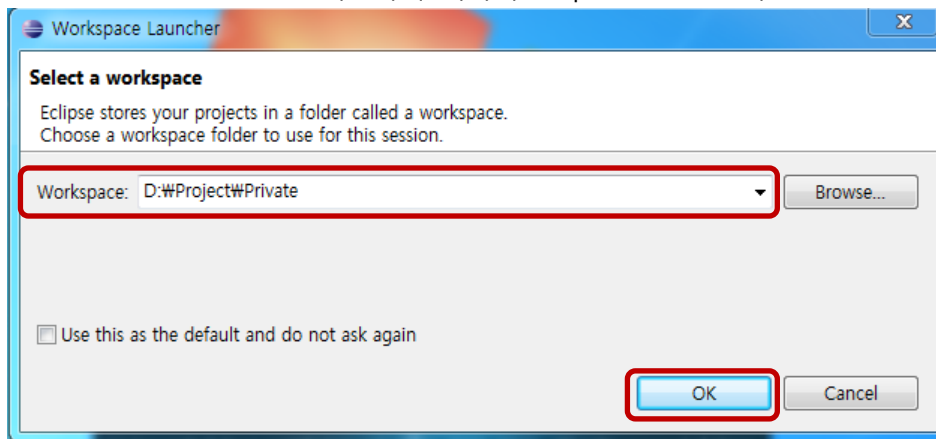
		Barcode Scanner Programming Guide					
All That Identification		Android Developer Guide			회사	ATID Co.,Ltd	
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0

2. Getting Started

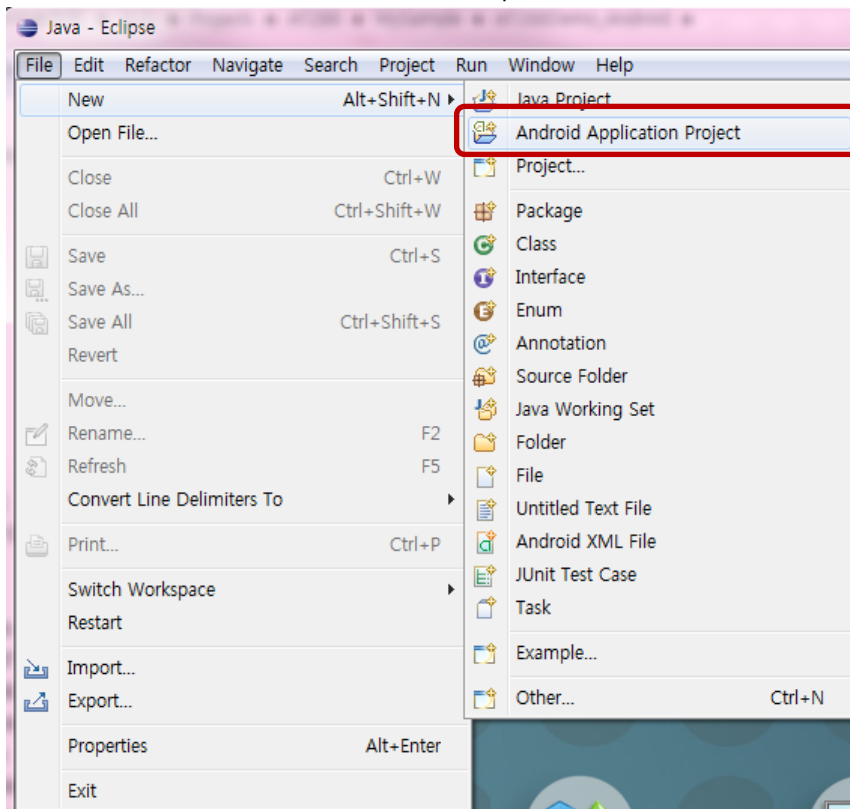
Getting Started 에서는 SDK Library 를 사용하여 간단하게 Android Sample 을 작성해 봄으로써, Barcode Scanner 응용프로그램을 개발하는 방법에 대하여 설명한다.

Create Project for Android

Android 용 응용프로그램을 개발하기 위하여 Eclipse 를 실행한다..

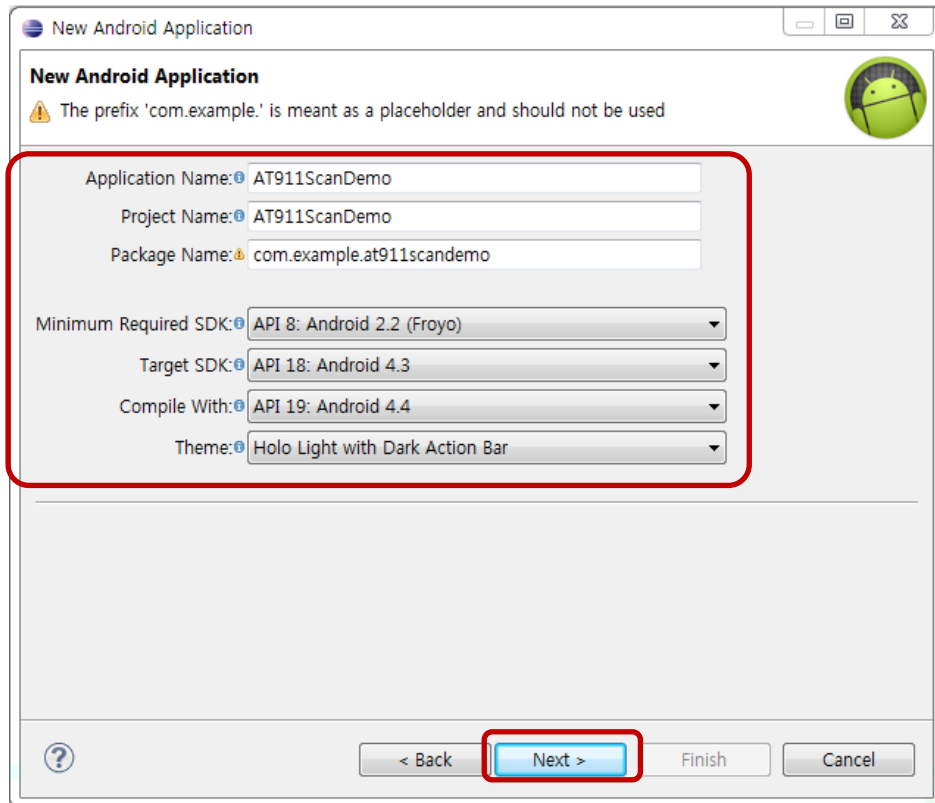


Eclipse를 실행하면 프로젝트를 작업 할 폴더를 선택하기 위한 대화상자가 나타난다. 프로젝트를 작업을 할 폴더를 입력하고 "OK" 버튼을 클릭하여 Eclipse를 진행한다.

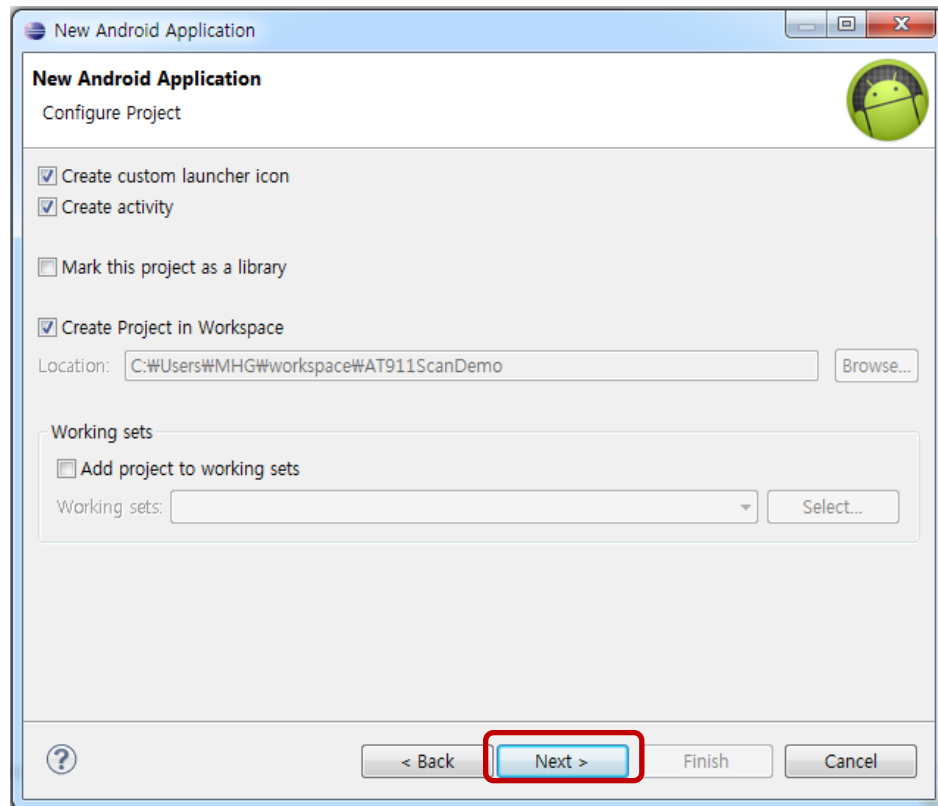


 All That Identification		Barcode Scanner Programming Guide					
Android Developer Guide					회사	ATID Co.,Ltd	
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0

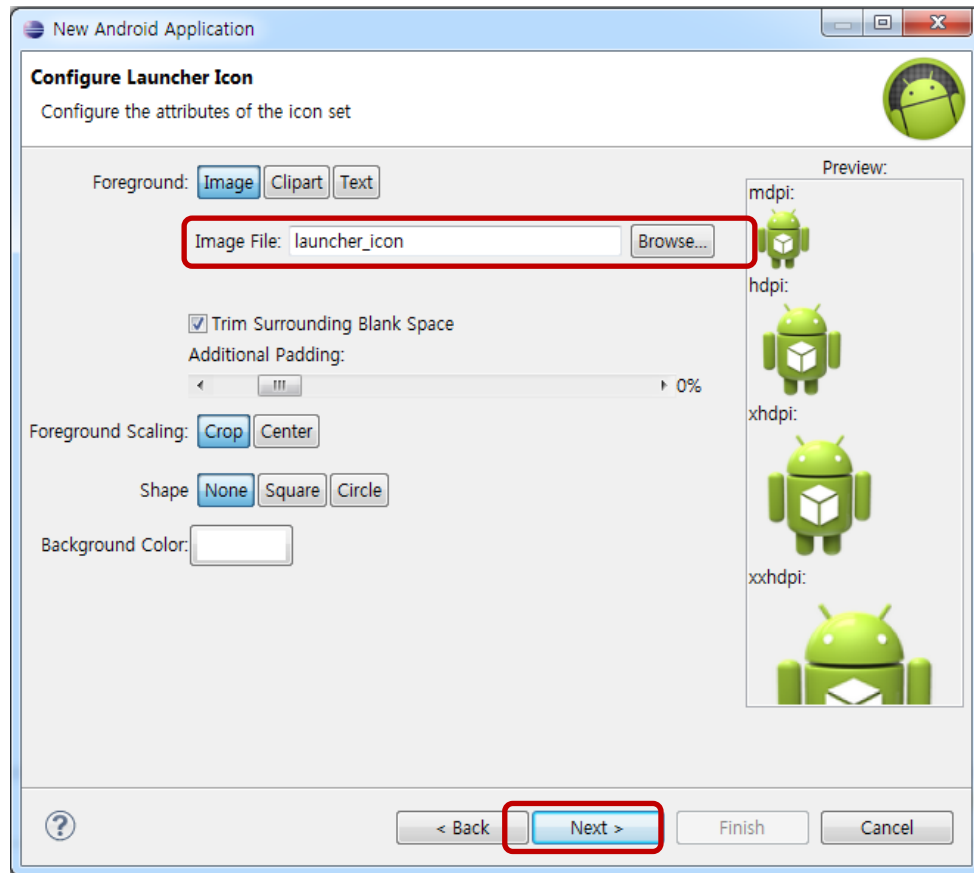
Eclipse가 실행되면 메뉴에서 File > New > Android Application Project를 선택하여 프로젝트를 생성하기 위한 대화상자를 나타나게 한다.



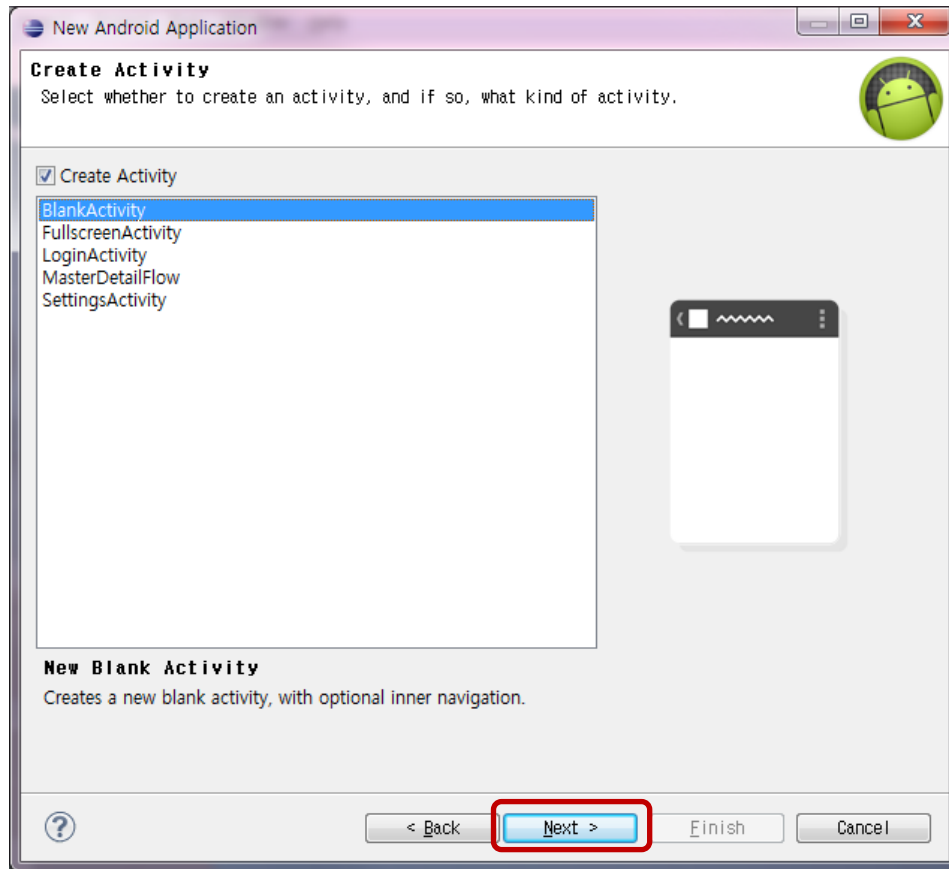
New Android Application 대화상자가 나타나면 Application Name과 Project Name을 입력하고 Package Name을 입력하고, Minimum Required SDK, Target SDK, Compile With 그리고 Theme를 선택하고 "Next" 버튼을 클릭하여 프로젝트 생성을 진행한다.



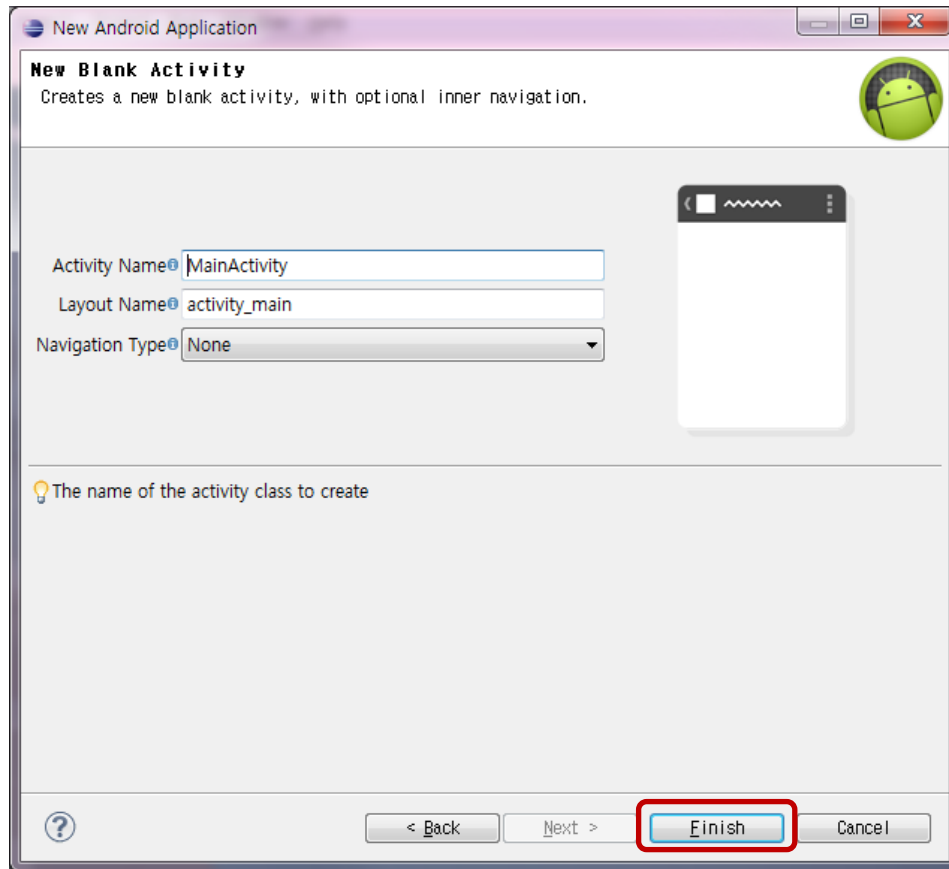
특별하게 수정할 것이 없으므로 "Next"버튼을 클릭하여 다음으로 넘어간다.



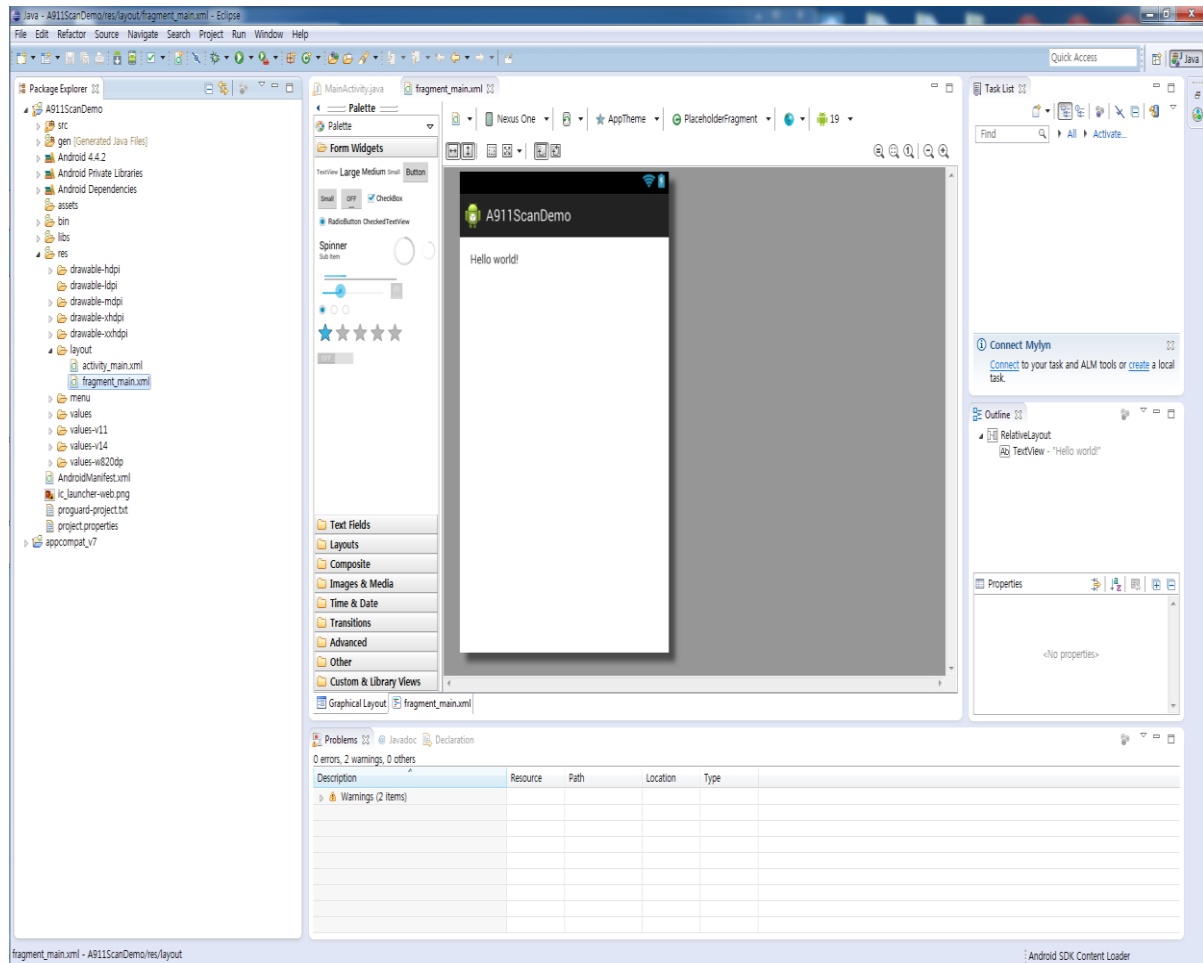
Application에서 사용할 Icon이 될 이미지를 입력하고 "Next"버튼을 클릭하여 다음으로 넘어간다.



Application의 Theme를 설정하는 화면이다. Theme를 선택한 후 "Next"버튼을 클릭하여 다음으로 넘어간다.



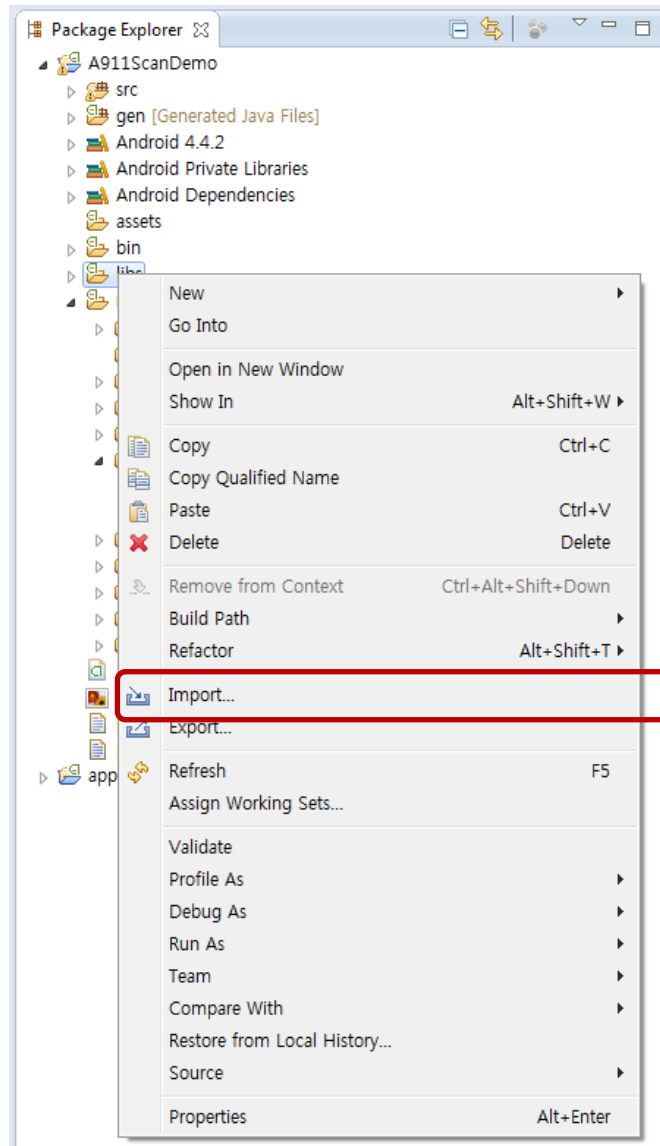
Activity Name에 Main Activity의 이름을 입력하고 Layout Name에도 개발에 사용할 Main Layout의 이름을 입력한 후 "Finish" 버튼을 클릭하여 프로젝트의 생성을 종료한다.



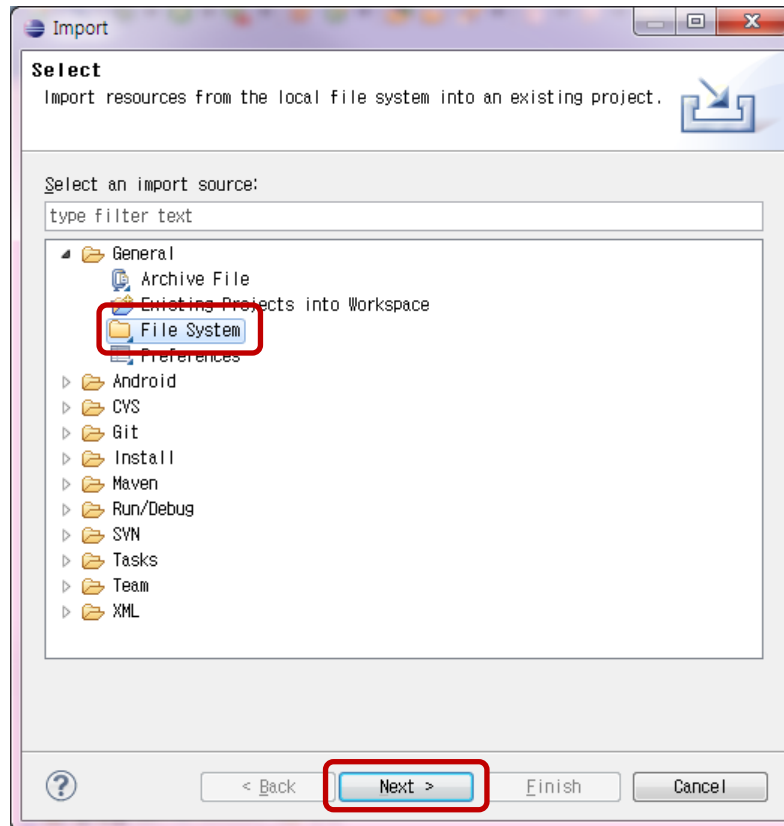
프로젝트가 생성된 Eclipse의 화면이다.

Development Environment

Android 응용 프로그램을 개발하기 위해서는 Barcode Scanner SDK Jar파일이 필요하다.

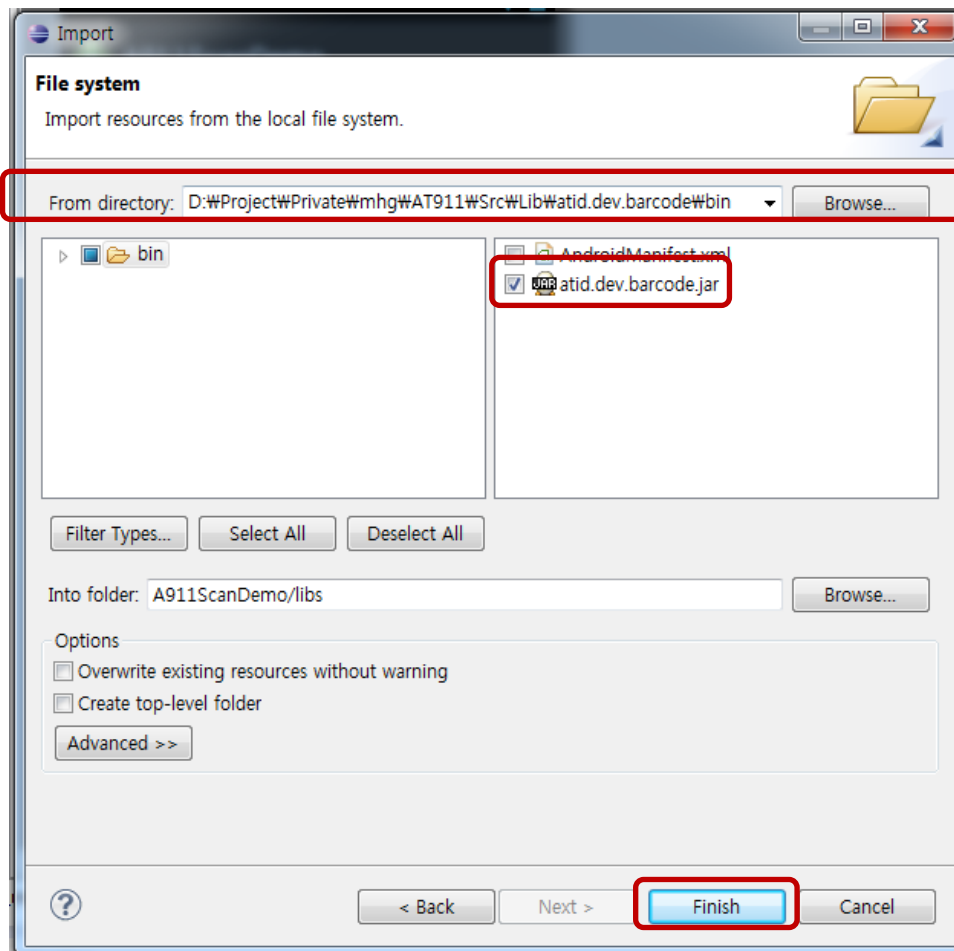


Barcode Scanner SDK jar파일을 Import하기 위해 Package Explorer에서 libs폴더를 선택하고 마우스 오른쪽 버튼을 클릭하여 팝업 메뉴가 나타나도록 한다. 팝업 메뉴에서 "Import..."메뉴를 클릭한다.

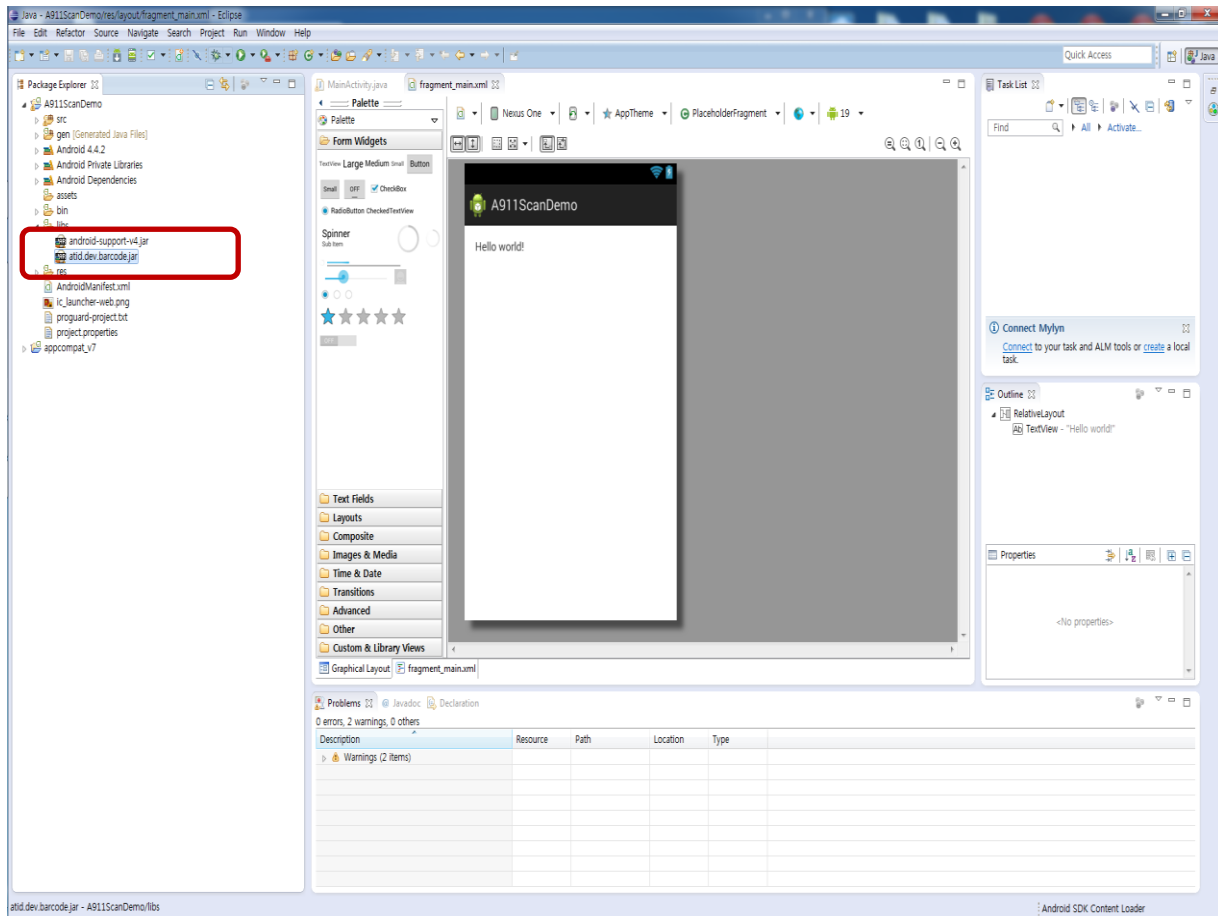


Import 대화상자가 나타나면 Select an import source에서 General > File System을 선택하고 "Next"버튼을 클릭한다.

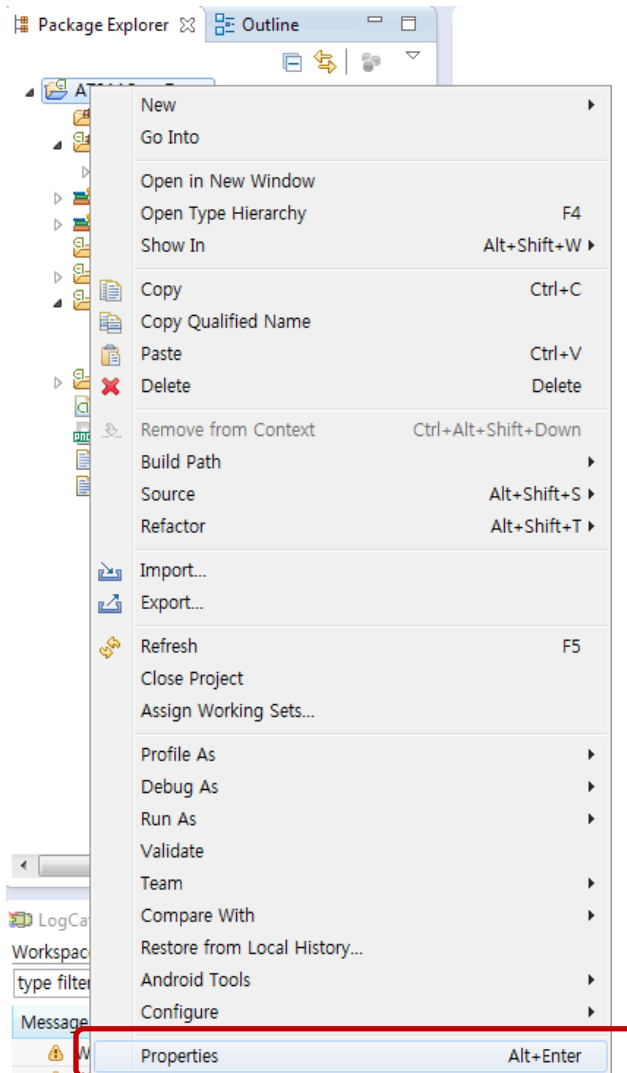
		Barcode Scanner Programming Guide					
All That Identification		Android Developer Guide			회사	ATID Co.,Ltd	
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0



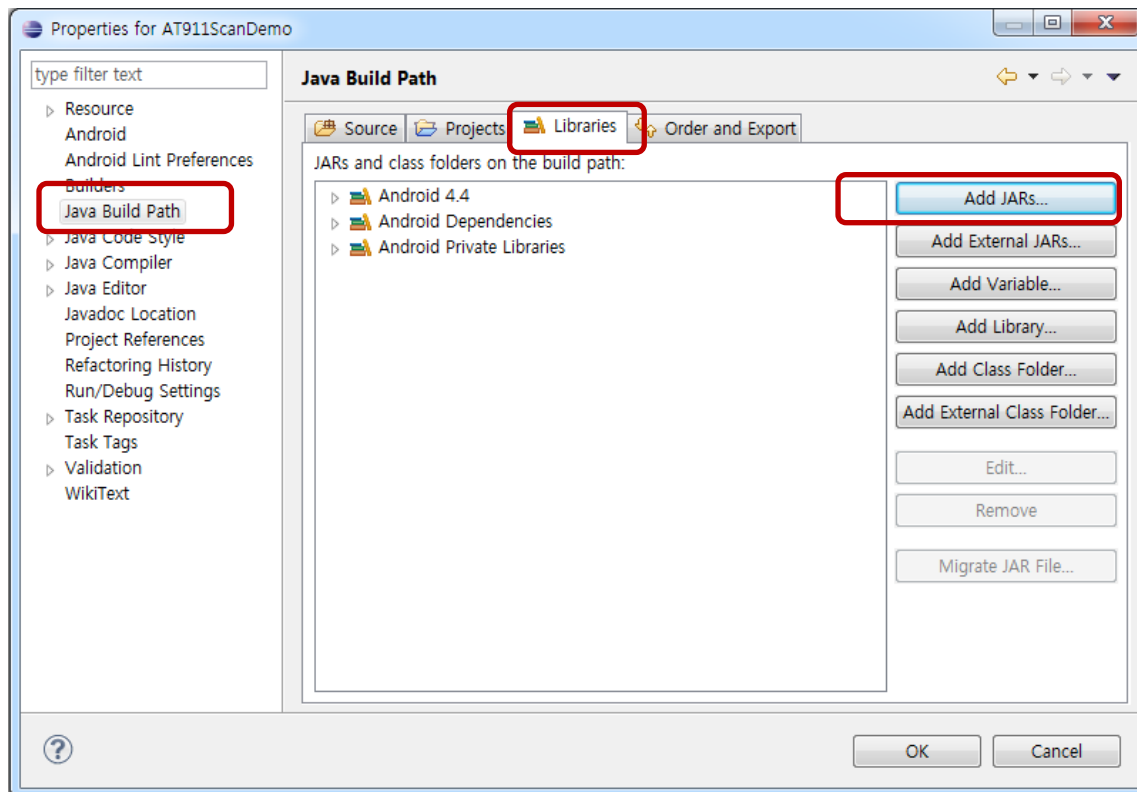
From directory에 SDK Library가 설치된 폴더에 있는 Library폴더의 하위 폴더의 Android폴더를 선택하면 atid.dev.barcode.jar파일(스캐너 라이브러리)이 보일 것이다. 체크박스에 체크를 하고 "Finish"버튼을 클릭하면 atid.dev.barcode.jar 파일을 프로젝트가 생성한 폴더에 있는 libs폴더로 복사해온다.



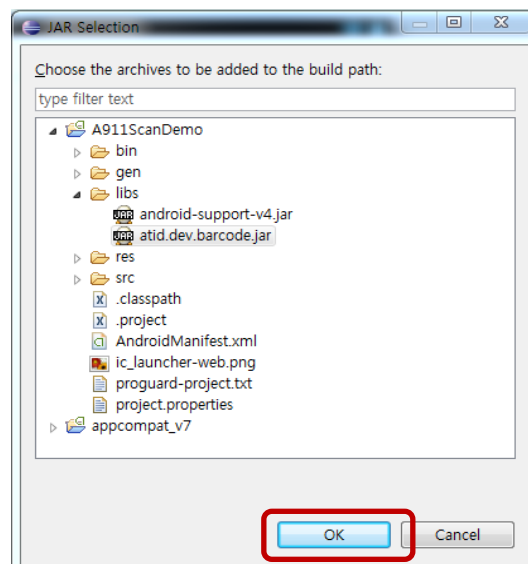
jar파일이 Package Explorer로 복사되어 온 것을 확인할 수 있다.



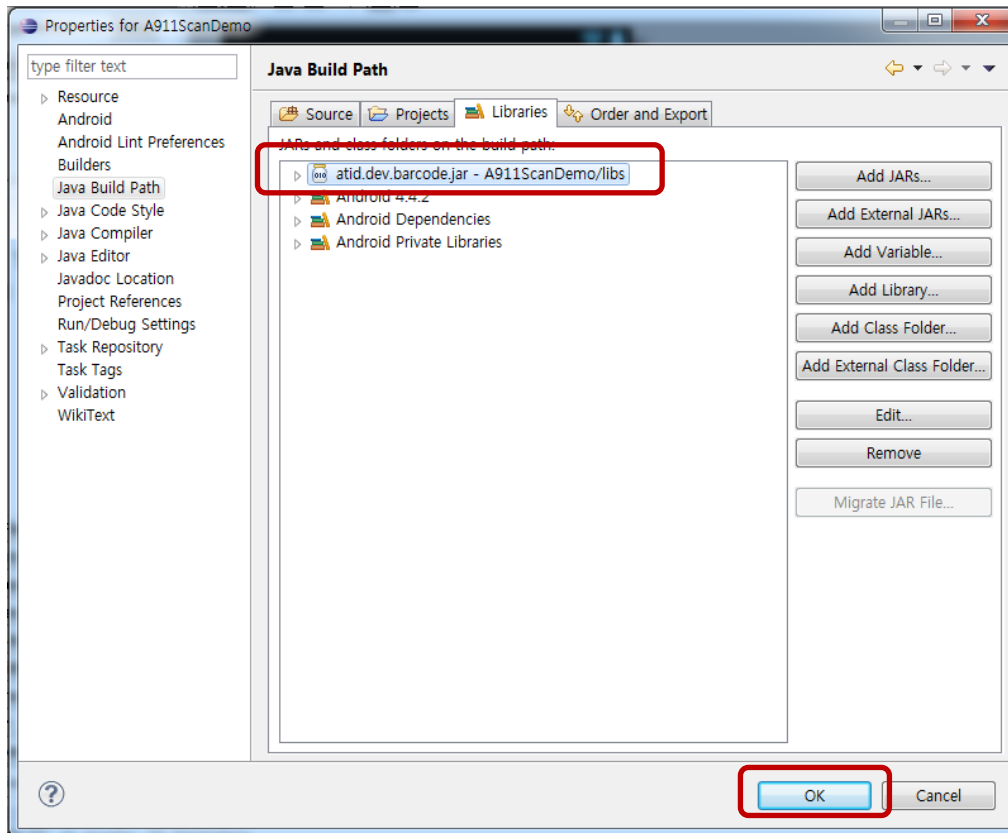
복사하여온 SDK jar파일을 프로젝트 포함하기 위하여 Package Explorer의 프로젝트 루트에서 마우스 오른쪽 버튼을 클릭하여 팝업 메뉴가 나타나면 Properties 메뉴를 선택한다.



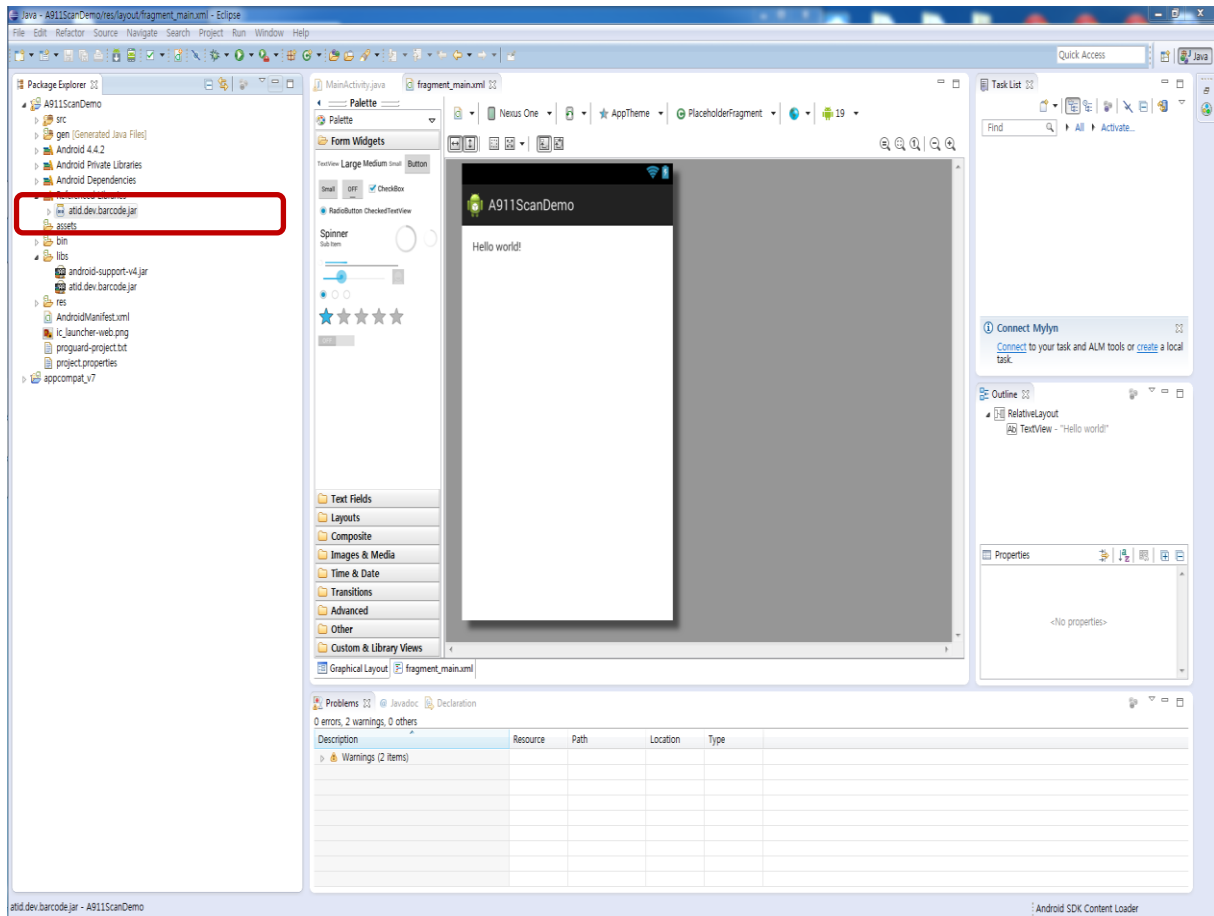
Properties 대화 상자가 나타나면 왼쪽 리스트에서 Java Build Path를 선택하고, 오른쪽 위의 탭에서 Libraries를 선택한다. 그리고 "Add JARs.."버튼을 클릭한다.



JAR Selection 대화상자가 나타나면 libs폴더에 있는 atid.dev.barcode.jar 파일을 선택하고 "OK"버튼을 클릭한다.



atid.dev.barcode.jar 파일이 Build Path로 설정된 것을 확인하고 "OK"버튼을 클릭하여 프로젝트 설정을 마친다.



Package Explorer 에 Referenced Libraries 가 생성되고 그 밑에 atid.dev.barcode.jar 파일이 추가된 것을 확인하자. 이제 SDK 를 사용하여 Android 응용프로그램을 개발할 준비가 끝났다.

		Barcode Scanner Programming Guide					
All That Identification		Android Developer Guide			회사		ATID Co.,Ltd
문서이름		작성자	SDK Team	날자	2017-02-22	버전	V2.0

3. Programming Guide

3.1. Create and Activity

스캐너 디바이스를 사용하기 위해서는 아래 기재되어 있는 것과 같이 Scanner 인스턴스를 생성한다. 이때 PDA 와 Scanner Device 를 Connect 하기 위해 ATScanManger 클래스의 GetInstance()함수를 호출한다. ATScanManger 내부에서 연결을 시도, 클래스가 소멸시 연결을 해지한다. 연결이 완료되면 activity 내에 이벤트를 받기 위해 event listener 설정한다.

```
private ATScanner scanner = null;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Initialize Scanner Instance
    if ((this.mScanner =
        ATScanManager.getInstance(getApplicationContext())) == null) {
        // Error occurred!
        showDeviceDialog();
        return;
    }

    this.mScanner.setEventListener(this);
    String revision = this.mScanner.getVersion();
}
```

3.2 Event Handler

Scanner Device로부터 스캐너에 대한 장비 상태, 설정, 디코딩 데이터에 관련하여 이벤트를 얻기 위해 ATScanner으로 개체를 생성 후 파라미터로 입력 받을 수 있도록 인터페이스를 추가한다. Scanner Device에 대한 인터페이스로는 ChangedStateListener와 DecodeEventListener로 나뉘어 진다. 각각의 인터페이스에서는 스캐너 동작 상태와 디코딩 데이터를 수신할 수 있다.

3.3.1 BarcodeEventListener

```
import com.atid.lib.dev.event.BarcodeEventListener;

public class MainActivity extends Activity implements
Button.OnClickListener, BarcodeEventListener
{
    @Override
    public void onStateChanged(EventType state) {
    }

    @Override
    public void onDecodeEvent(BarcodeType type, String barcode) {

        Log.d(TAG, "onDecodeEvent(" + type + ", [" + barcode
            + "])");

        if (type != BarcodeType.NoRead) {
            int position = this.adapterBarcode.addItem(type, barcode);
            this.lstBarcodeList.setSelection(position);
            beep();
        }
        enableScanButton(true);
    }
}
```

3.3 Action Operation

Scanner Device 연결 이후 scanner를 활용하여 데이터를 읽기 위해 빔을 출력하기 위해 연결 상태를 확인한 후 스캐너를 동작 또는 해지한다.

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.scan_action:
            enableScanButton(false);

            if (this.scanner.isDecode()) {
                this.scanner.stopDecode();
            } else {
                this.scanner.startDecode();
            }

            enableScanButton(true);
            break;
        case R.id.clear:
            this.adapterBarcode.clear();
            break;
    }
}
```