1.Enviroment installations:
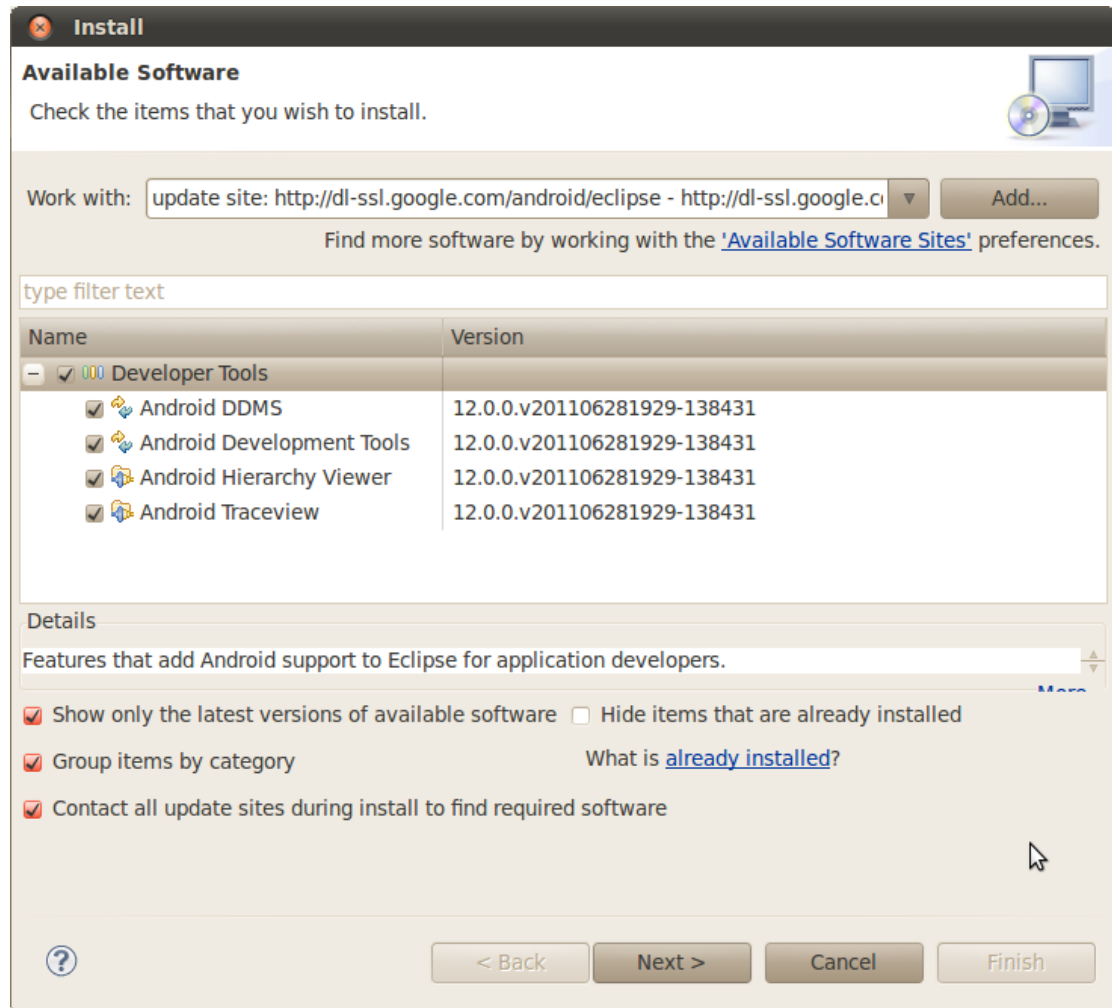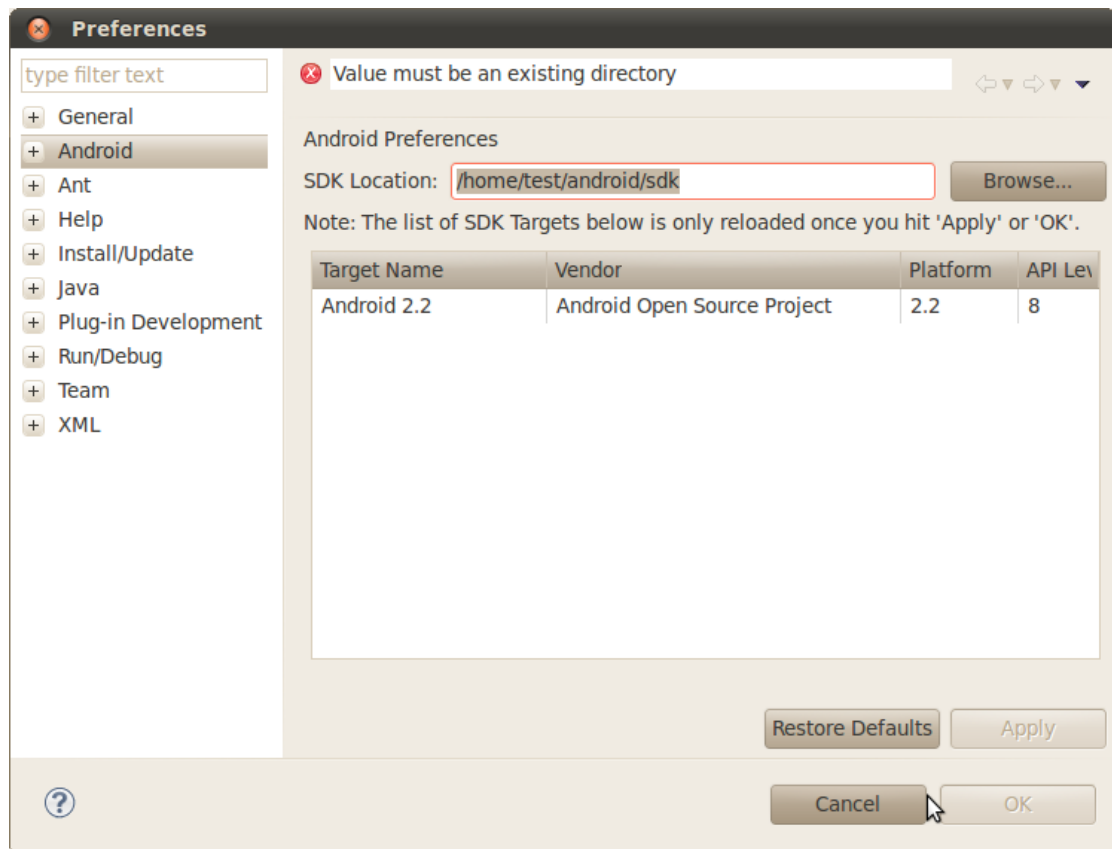
First you should install "Android SDK" and "Eclipse"

Then you should install ADT(Android Development Kit) for eclipse, with fllowing steps:
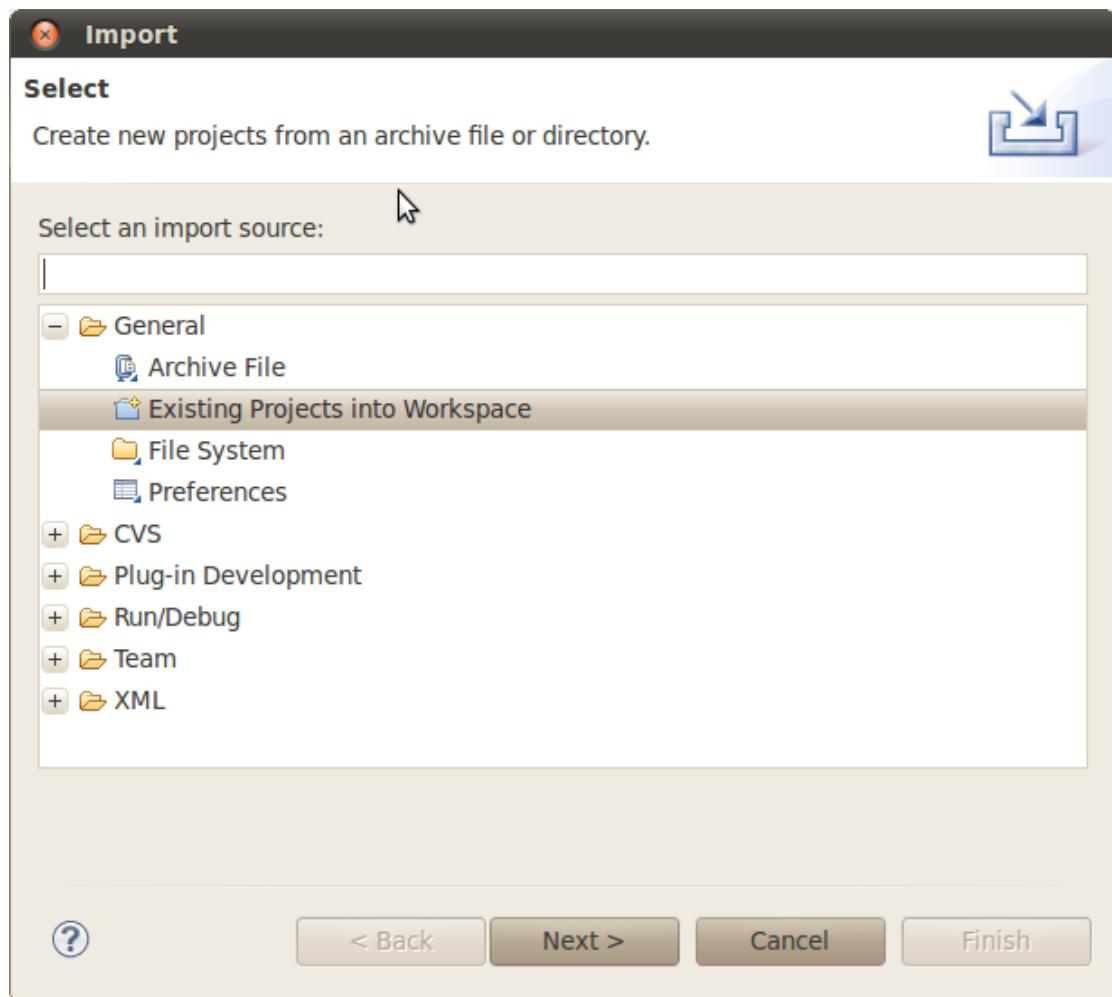
1. open eclipse and clip the menu item "Help"->"Install New Software"
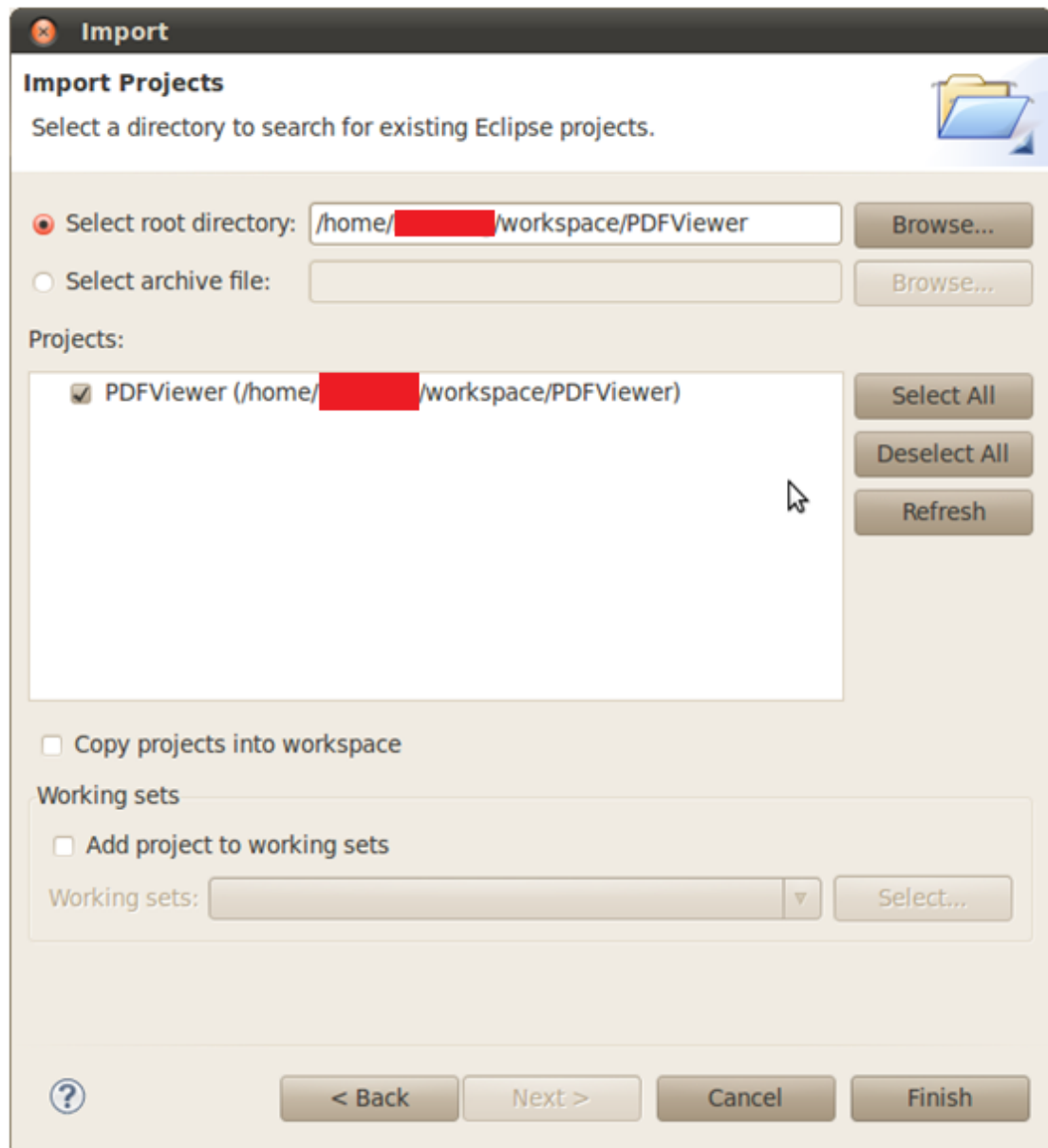2. type url "http://dl-ssl.google.com/android/eclipse" in editbox just like this:

3. then check all items to download and finish installation.
4. After install ADT, click menu item "Windows"->"Preference"->"Android", and type installation path of Android SDK to complete config:

5. Extract PDFViewer project to a dictionary. Then click "File"->"Import" like this:

Click "Next" and type project path, show this dialog:

Then click "Finish" to import the project.

2.How To Start

There are 7 classes in project that are the interfaces for PDF JNI library.

"com.radaee.pdf.Global" is the global environment for PDF parser programs.

"com.radaee.pdf.Document" is Document objects for PDF Files.

"com.radaee.pdf.Page" is Page objects for each page.

"com.radaee.pdf.Matrix" is the matrix setting to render pages.

"com.radaee.pdf.HWriting" is the ink visual for writing

"com.radaee.pdf.PDFV" is a C/C++ werapped View class.

"com.radaee.pdf.PDFVS" is setting for PDFV class.

Extra, an package named "com.radaee.pdfex" used in project for page layouts.
This package is close source, in jar format, placed in "libs" subdir

To open PDF Files, you should do these:
1. Call initialize methods:
        Global.Init (Activity act);
    This method need some resources placed in "assets" in project.
    You can find these resources in demo project.
    You can edit this function if needed, for some font setting and others.
2. Create a Document object, and invoke Document.open
3. Pass the object to PDFView interface to manage pages layouts.
4. Invoke PDFView.Close when exit your application.

3.Native API Reference

See java doc to get more information about API reference, which placed in "doc" subdir in demo
project.

4.Some Question

1. How to apply and active license?
    When you purchasing a license, you need input 3 fields, the first field is package name of
application, while others are defined by yourself.
    Package name must be same as package name defined in "AndroidManifest.xml"
    example:

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.radaee.reader"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <supports-screens android:largeScreens="true" android:anyDensity="true" andr
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".PDFReaderAct"
            android:configChanges="orientation"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Once you purchased a license, you should active license by invoke function Global.activeXXX() in Global.init() like this:

boolean succeeded = activeProfessional(context, "radaee", "radaee_com@yahoo.cn", "Z5A7JV-5WQAJY-9ZOU9E-OQ31K2-FADG6Z-XEBCAO");

if you did not active license, and run your application.

you should see watermarks on each page, and you can't click any annotation on page.

2. right to left text selecting?

Global.selRTOL = true;

Then open document.

3. dark mode support

Global.dark_mode = true;

4. set ink color or ink width?

Global.inkColor = 0xAARRGGBB;

Global.inkWidth = XX;

5. is there any other activity for testing?

Yes, there are 6 activities to testing:

--PDFReaderAct

This is the most complete test activity, this is default setting.

--ReaderActivity

This is the simplest sample for using PDFView classes.

--PDFCropAct

This activity clip page to blocks in size 100*100, and then spell them to whole screen.

--PDFSimpleAct

This is a simplest sample to display page.

--PDFViewerAct

This is PDF Viewer sample, based on C/C++ codes, not same as "pdfex" classes which written by pure java.

And this is a light view class for developer to customize UI.

--PDFInkAct

This is not PDF sample. But testing HWriting class.

To test these activity, you just need modify "AndroidManifest.xml"

6. how to change view style in demo?

Just invoke PDFReader.set_viewer():

7.how to customize text tool? Highlight/strikeout/underline

set a boolean variable to ture after user click button. when user long clicked page and moving and click up. PDFView will return a callback "PDFViewListener.onSelectEnd"

if you don't want long-pressed, you should invoke:

PDFView.viewSetSel(x1, y1, x2, y2);

x1, y1, x2, y2 are all View coordinates.

this function return callback "PDFViewListener.onSelectEnd" for each invoking.

So, in these functions you can do:

```java
    private float sel_x;
    private float sel_y;
    private boolean markup;
    public boolean onTouchEvent (MotionEvent event)
    {
        if( markup && m_viewer != null )
        {
            switch( event.getActionMasked() )
            {
            case MotionEvent.ACTION_DOWN:
                sel_x = event.getX();
                sel_y = event.getY();
                break;
            case MotionEvent.ACTION_MOVE:
                m_viewer.viewSetSel(sel_x, sel_y, event.getX(), event.getY());
                break;
            case MotionEvent.ACTION_UP:
                m_viewer.annotSetMarkup(0);
                markup = false;
                break;
            }
            return true;
        }
        if( m_viewer != null )
            return m_viewer.viewTouchEvent(event);
        else
            return true;
    }
    public void onSelectEnd(String text)
    {
        if( markup ) return;
        else ...
    }
```

8.how to customize reader based on PDFView classes?

See ReaderActivity, this shows the simplest Reader based on PDFView classed

If any question, please post issue on forum, we are online!
So good luck!