Quick Start Guide

Foxit PDF SDK

For JAVA

Microsoft® Partner
Gold Independent Software Vendor (ISV)

TABLE OF CONTENTS

1 INTRODUCTION TO FOXIT PDF SDK

Have you ever thought about building your own application that can do everything you want with PDF files? If your answer is "Yes", congratulations! You just found the best solution in the industry that allows you to build stable, secure, efficient and full-featured PDF applications.

Foxit PDF SDK provides high-performance libraries to help any software developer add robust PDF functionality to their enterprise, mobile and cloud applications across all platforms (includes Windows, Mac, Linux, Web, Android, iOS, and UWP), using the most popular development languages and environments.

1.1 Why Foxit PDF SDK is your choice

Foxit is an Amazon-invested leading software provider of solutions for reading, editing, creating, organizing, and securing PDF documents. Foxit PDF SDK libraries have been used in many of today's leading apps, and they are proven, robust, and battle-tested to provide the quality, performance, and features that the industry's largest apps demand. Customers choose Foxit PDF SDK product for the following reasons:

- **Easy to integrate**

Developers can seamlessly integrate Foxit PDF SDK into their own applications.

- **Lightweight footprint**

Do not exhaust system resource and deploys quickly.

- **Cross-platform support**

Support current mainstream platforms, such as Windows, Mac, Linux, Web, Android, iOS, and UWP.

- **Powered by Foxit's high fidelity rendering PDF engine**

The core technology of the SDK is based on Foxit's PDF engine, which is trusted by a large number of the world's largest and well-known companies. Foxit's powerful engine makes the app fast on parsing, rendering, and makes document viewing consistent on a variety of devices.

- **Premium World-side Support**

Foxit offers premium support for its developer products because when you are developing mission critical products you need the best support. Foxit has one of the PDF industry's largest team of support engineers. Updates are released on a regular basis to improve user experience by adding new features and enhancements.

1.2 Foxit PDF SDK for Java

Application developers who use Foxit PDF SDK can leverage Foxit's powerful, standard-compliant PDF technology to securely display, create, edit, annotate, format, organize, print, share, secure, search documents as well as to fill PDF forms. Additionally, Foxit PDF SDK includes a built-in, embeddable PDF Viewer, making the development process easier and faster. For more detailed information, please visit the website https://developers.foxitsoftware.com/pdf-sdk/.

In this guide, we focus on the introduction of Foxit PDF SDK for Java API on Windows and Linux platforms.

Foxit PDF SDK for Java API ships with simple-to-use APIs that can help Java developers seamlessly integrate powerful PDF technology into their own projects on Windows and Linux platforms. It offers the most common features in PDF SDK, such as PDF viewing, bookmark navigating, text selecting/copying/searching, annotations, and signature.

1.3 Evaluation

Foxit PDF SDK allows users to download trial version to evaluate SDK. The trial version has no difference from a standard version except for the 30-day limitation trial period and the trail watermarks that will be generated on the PDF pages. After the evaluation period expires, customers should contact Foxit sales team and purchase licenses to continue using Foxit PDF SDK.

1.4 License

Developers should purchase licenses to use Foxit PDF SDK in their solutions. Licenses grant users permissions to release their applications based on PDF SDK libraries. However, users are prohibited to distribute any documents, sample codes, or source codes in the SDK released package to any third party without the permission from Foxit Software Incorporated.

1.5 About this guide

This guide is intended for developers who need to integrate Foxit PDF SDK for Java into their own applications. It aims at introducing installation package structure on desktop platform with Java, and the usage of SDK.

2 GETTING STARTED

It is very easy to setup Foxit PDF SDK and see it in action! This guide will provide you with a brief introduction about our SDK package. The following sections introduce the contents of system requirements, the installation package as well as how to run a demo, and create your own project.

2.1 System Requirements

Platform	System Requirement	JDK version	Memo
Windows	Windows XP, Vista, 7, 8 and 10 (32-bit and 64-bit) Windows Server 2003, 2008 and 2012 (32-bit and 64-bit)	At least 1.8	It only supports for Windows 8/10 classic style, but not for Store App or Universal App.
Linux	32-bit and 64-bit OS	At least 1.8	

2.2 What is in the Package

Package for Windows Java is named "foxitpdfsdk_6_2_1_win_java.zip" and package for Linux Java is named "foxitpdfsdk_6_2_1_linux_java.zip". They have the same structure, so in this guide mainly introduce "foxitpdfsdk_6_2_1_win_java.zip" as an example.

Download the package for Windows Java and extract it to a new directory like "foxitpdfsdk_6_2_1_win_java". The structure of the release package is shown in Figure 2-1.

***NOTE**: the highlighted rectangle in the figure is just the version of Foxit PDF SDK. Here the SDK version is 6.2.1, so it shows 6_2_1 in the package name. Other highlighted rectangles in other figures have the same meaning in this guide.*

This package contains the following folders:

docs:	API references, quick start guide
lib:	libraries and license files
samples:	sample projects and demos

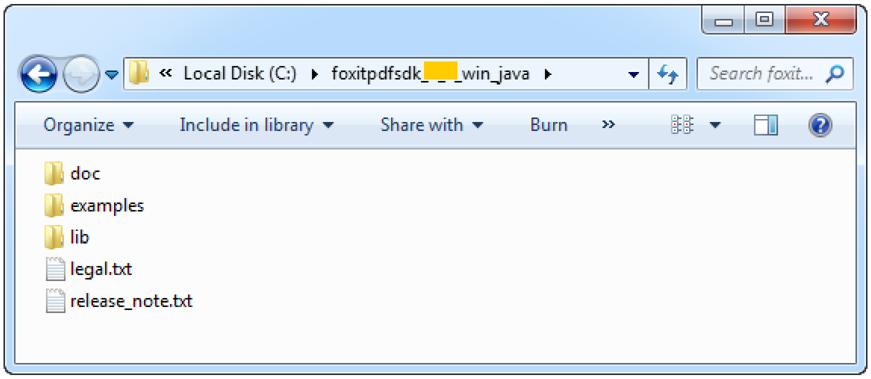


Figure 2-1

Foxit PDF SDK provides "fsdk.jar" file in directory "\lib". It contains 14 packages that are shown in Figure 2-2. Please refer to API Reference in directory "\doc" for more details about classes and APIs.

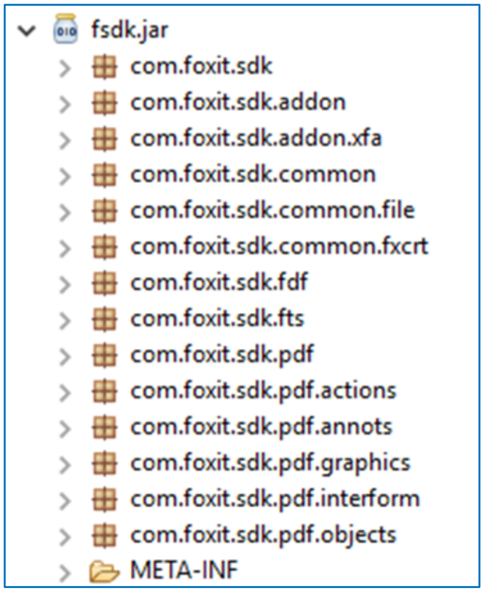

- fsdk.jar
  - com.foxit.sdk
  - com.foxit.sdk.addon
  - com.foxit.sdk.addon.xfa
  - com.foxit.sdk.common
  - com.foxit.sdk.common.file
  - com.foxit.sdk.common.fxcrt
  - com.foxit.sdk.fdf
  - com.foxit.sdk.fts
  - com.foxit.sdk.pdf
  - com.foxit.sdk.pdf.actions
  - com.foxit.sdk.pdf.annots
  - com.foxit.sdk.pdf.graphics
  - com.foxit.sdk.pdf.interform
  - com.foxit.sdk.pdf.objects
  - META-INF

Figure 2-2

2.3 How to initialize Foxit PDF SDK

It is necessary for applications to initialize Foxit PDF SDK with license before calling any APIs. The static function ***Library.initialize****(sn, key)* is provided in com.foxit.sdk.common.Library. The trial license files can be found in the "lib" folder.

Following is an example:

```
// The value of "sn" can be got from "gsdk_sn.txt" (the string after "SN=").
// The value of "key" can be got from "gsdk_key.txt" (the string after "Sign=").
int error_code = Library.initialize("sn", "key");
if (error_code != e_ErrSuccess) {
  return;
}
```

2.4 How to run a demo

Foxit PDF SDK provides several simple demos in directory "\examples". All these demos can be run directly with the ".bat" files or ".sh" files in directory "\examples\simple_demo":

a. Go to directory "\examples\simple_demo", and run all demos by "RunAllDemo.bat" for Windows or by "RunAllDemo.sh" for Linux.

b. If you want to run a specific single demo, please locate to the directory of the demo, for example locate to "\examples\simple_demo\annotation", and run this demo by "RunDemo.bat" for Windows or by "RunDemo.sh" for Linux.

"\examples\simple_demo\input_files" contains all the input files used among these demos. Some demos will generate output files (pdf, text or image files) to a folder named by the demo name under "example\simple_demo\output_files\".

2.5 How to create a simple project

In this section, we will show you how to use Foxit PDF SDK for Windows Java (Linux Java is similar to Windows Java) to create a simple project that renders the first page of a PDF to a bitmap and saves it as a JPG image. For better writing code, we use Eclipse IDE to create a Java project called "test". Then follow the steps below:

1) Copy "lib" folder from the download package to the project folder, and then refresh the project. The structure of the *test* project is shown in Figure 2-3.

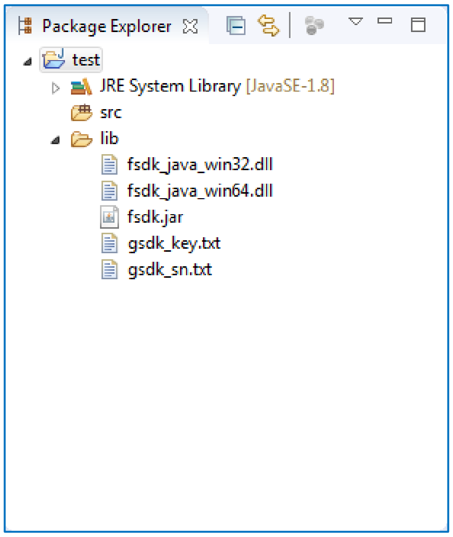


Figure 2-3

2) Add "fsdk.jar" to the project. Right click the *test* project, select "Build Path > Configure Build Path > Libraries > Add JARs", and choose the "fsdk.jar" in "test/lib" as shown in Figure 2-4.

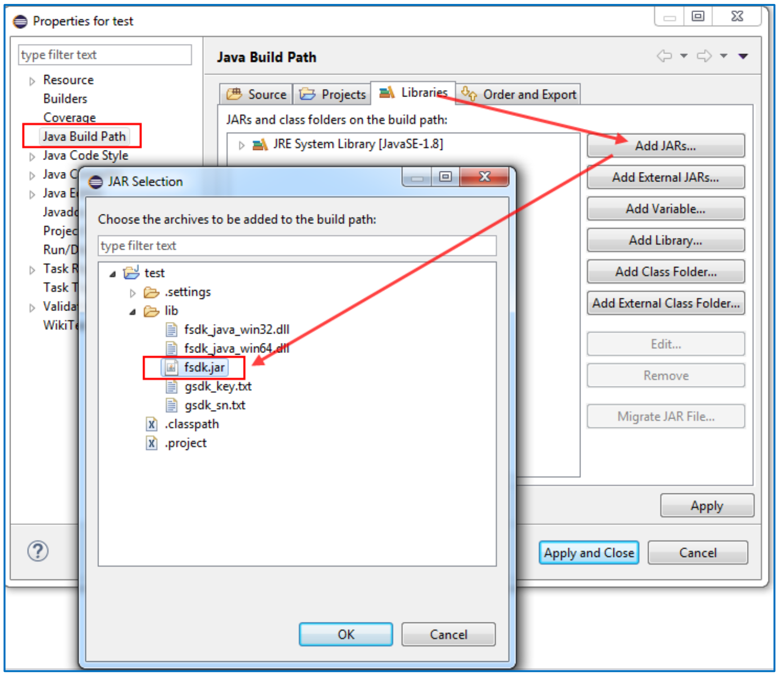
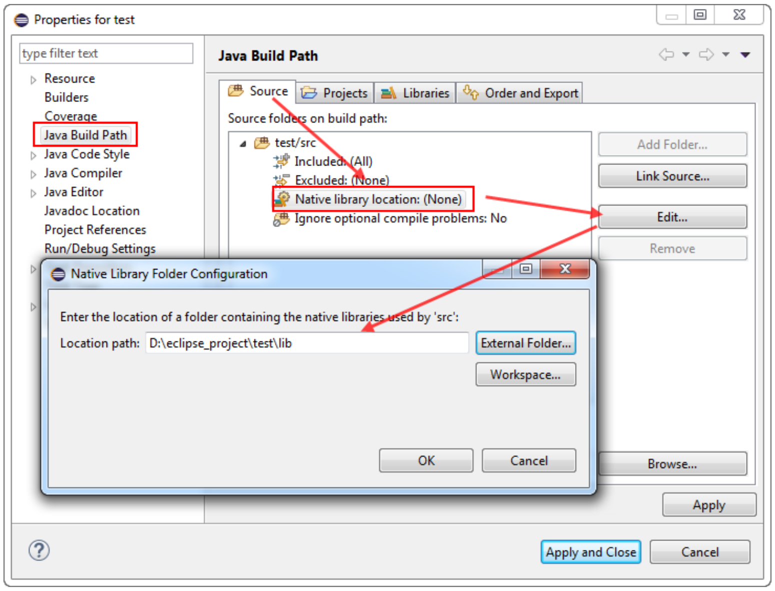


Figure 2-4

3) Configure build path for .dll (for Windows) and .so (for Linux). Right click the *test* project, select "Build Path > Configure Build Path > Source > Native library location: (None) > Edit", and locate to the "test/lib" folder as shown in Figure 2-5.



Enter the location of a folder containing the native libraries used by 'src':

Location path: D:\eclipse_project\test\lib

Figure 2-5

4) Create a new class file called test.java under "test/src/test" directory.

5) Open the "test.java" file, import the classes that you need to use in "fsdk.jar". Here, we just import the classes as follows:

```
import com.foxit.sdk.PDFException;
import com.foxit.sdk.common.Bitmap;
import com.foxit.sdk.common.Image;
import com.foxit.sdk.common.Library;
import com.foxit.sdk.common.Renderer;
import com.foxit.sdk.common.fxcrt.Matrix2D;
import com.foxit.sdk.pdf.PDFDoc;
import com.foxit.sdk.pdf.PDFPage;
```

6) Initialize Foxit PDF SDK. This should be done before calling any other APIs in Foxit PDF SDK. The trial license files can be found in the "lib" folder.

```
// The value of "sn" can be got from "gsdk_sn.txt" (the string after "SN=").
// The value of "key" can be got from "gsdk_key.txt" (the string after "Sign=").
String sn = "";
String key = "";
int error_code = Library.initialize(sn, key);
if (error_code != e_ErrSuccess) {
       return;
}
```

7) Load a PDF document, and parse the first page of the document. Assume that you have already put a "Sample.pdf" to the "*test*" folder.

```
// load a "Sample.pdf" document.
PDFDoc doc = new PDFDoc("Sample.pdf");
```

```
error_code = doc.load(null);
if (error_code != e_ErrSuccess) {
       return;
}

// Get the first page of the document.
PDFPage page = doc.getPage(0);

// Parse page.
page.startParse(e_ParsePageNormal, null, false);
```

8) Render the first page to a bitmap and save it as a JPG file.

```
int width = (int) page.getWidth();
int height = (int) page.getHeight();
Matrix2D matrix = page.getDisplayMatrix(0, 0, width, height, page.getRotation());

// Prepare a bitmap for rendering.
Bitmap bitmap = new Bitmap(width, height, e_DIBArgb, null, 0);
bitmap.fillRect(0xFFFFFFFF, null);

// Render page
Renderer render = new Renderer(bitmap, false);
render.startRender(page, matrix, null);

// Add the bitmap to image and save the image.
Image image = new Image();
image.addFrame(bitmap);
image.saveAs("testpage.jpg");
```

9) Build and Run the project. Right-click the *test* project in package explorer, and then choose "Run as > Java Application" to run it. The "testpage.jpg" will be generated in the "test" folder (See Figure 2-6).

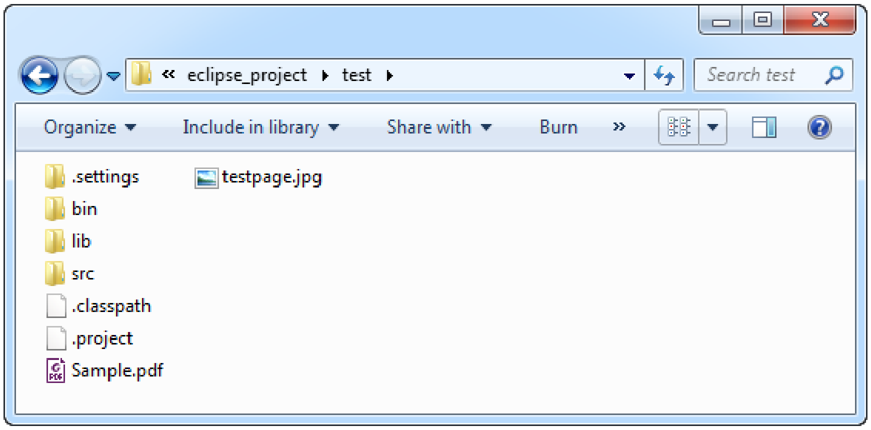


Figure 2-6

The final contents of "test.java" is as follow:

```java
package test;

import com.foxit.sdk.PDFException;
import com.foxit.sdk.common.Bitmap;
import com.foxit.sdk.common.Image;
import com.foxit.sdk.common.Library;
import com.foxit.sdk.common.Renderer;
import com.foxit.sdk.common.fxcrt.Matrix2D;
import com.foxit.sdk.pdf.PDFDoc;
import com.foxit.sdk.pdf.PDFPage;

import static com.foxit.sdk.common.Bitmap.e_DIBArgb;
import static com.foxit.sdk.common.Constants.e_ErrSuccess;
import static com.foxit.sdk.pdf.PDFPage.e_ParsePageNormal;

public class test {

	public static void main(String[] args) throws PDFException {

		// The value of "sn" can be got from "gsdk_sn.txt" (the string after "SN=").
		// The value of "key" can be got from "gsdk_key.txt" (the string after "Sign=").
		String sn = "";
		String key = "";
		int error_code = Library.initialize(sn, key);
		if (error_code != e_ErrSuccess) {
			return;
		}

		// load a "Sample.pdf" document.
		PDFDoc doc = new PDFDoc("Sample.pdf");
		error_code = doc.load(null);
		if (error_code != e_ErrSuccess) {
			return;
		}

		// Get the first page of the document.
		PDFPage page = doc.getPage(0);
```

```
            // Parse page.
            page.startParse(e_ParsePageNormal, null, false);

            int width = (int) page.getWidth();
            int height = (int) page.getHeight();
            Matrix2D matrix = page.getDisplayMatrix(0, 0, width, height,
page.getRotation());

            // Prepare a bitmap for rendering.
            Bitmap bitmap = new Bitmap(width, height, e_DIBArgb, null, 0);
            bitmap.fillRect(0xFFFFFFFF, null);

            // Render page.
            Renderer render = new Renderer(bitmap, false);
            render.startRender(page, matrix, null);

            // Add the bitmap to image and save the image.
            Image image = new Image();
            image.addFrame(bitmap);
            image.saveAs("testpage.jpg");
      }
}
```

Note: *If you have not installed Eclipse IDE in your machine, you can create a Java file, and then use command line to build and run it. For example, we use the test.java file created in the previous Eclipse IDE, and then follow the steps below:*

1. *Create a folder named "test_cm". Put the test.java into this folder. (Please note that you need to delete the first line "package test;" in the "test.java").*
2. *Copy "lib" folder to the same folder (test_cm) with "test.java".*
3. *Put a PDF document named "Sample.pdf" into "test_cm" folder.*
4. *Open command line window and navigate to "test_cm" folder, and then use the following commands to build and run the Java file.*

   For Windows,

   ```
   javac -cp .;./lib/fsdk.jar *.java
   java -Djava.library.path=lib -classpath .;lib/fsdk.jar test
   ```

   For Linux,

   ```
   javac -cp .:./lib/fsdk.jar *.java
   java -Djava.library.path=lib -classpath .:lib/fsdk.jar test
   ```

REFERENCES

[1] PDF reference 1.7
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51502

[2] PDF reference 2.0
https://www.iso.org/standard/63534.html

[3] Foxit PDF SDK API reference
sdk_folder/doc/Foxit PDF SDK Java API Reference.html

Note: sdk_folder is the directory of unzipped package.

SUPPORT

Foxit support home link:

http://www.foxitsoftware.com/support/

Sales contact phone number:

Phone: 1-866-680-3668

Email: sales@foxitsoftware.com

Support & General contact:

Phone: 1-866-MYFOXIT or 1-866-693-6948

Email: support@foxitsoftware.com