



# QUICK GUIDE

Foxit<sup>®</sup> PDF SDK

*(PDFium)*

**Microsoft<sup>®</sup> Partner**  
Gold Independent Software Vendor (ISV)

## TABLE OF CONTENTS

1	Introduction to PDFium version of Foxit PDF SDK.....	1
1.1	PDFium .....	1
1.2	Foxit PDF SDK .....	1
1.3	PDFium version of Foxit PDF SDK.....	1
1.4	Features.....	2
1.5	Evaluation.....	2
1.6	License.....	2
1.7	About this guide .....	3
2	Getting Started .....	4
2.1	System Requirements .....	4
2.2	What is in the Package .....	4
2.2.1	For Windows .....	4
2.2.2	For Linux.....	6
2.2.3	For Mac .....	6
2.3	How to run a demo .....	7
2.3.1	For Windows .....	8
2.3.2	For Linux.....	10
2.3.3	For Mac .....	11
3	Changes Compared with PDFium Open-source Library .....	13
3.1	Initialize library.....	13
3.2	Destroy library.....	13
3.3	Add macro definitions.....	14
3.4	Rename some functions and macro definitions .....	14
3.5	Remove some unused definitions and enumerations .....	14
3.6	Other Notices .....	15
4	FAQ.....	16

5 Support.....17

## 1 INTRODUCTION TO PDFIUM VERSION OF FOXIT PDF SDK

---

### 1.1 PDFium

PDFium is an open-source project which was developed by Foxit and Google. Google is responsible for some of the most important open-source projects of our time and Foxit is honored to be chosen as the PDF providers for the PDFium project. PDFium provides developers the opportunity to leverage a high performance standards-compliant PDF open-source software library to view, search, print PDF documents, and fill form in PDF documents. It also has been wrapped into Google's Chrome browser, which replaced codes that was previously closed source.

### 1.2 Foxit PDF SDK

Foxit PDF SDK is a commercial SDK which shares the underlying technology that produced the PDFium rendering engine. It is easy to integrate and provides a robust set of platform independent PDF libraries, allowing developers to "develop once, port many times". It helps application developers focus on functionality that they really need regardless of the platform and reduce the development cost. Foxit PDF SDK includes many additional PDF functionalities that are not contained in PDFium, such as, accessing and editing PDF layers, annotating PDF documents, creating bookmarks for PDF documents, incorporating pressure sensitive digital ink capabilities into PDF solutions, and etc.. For more details, please visit Foxit website at <https://www.foxitsoftware.com/products/sdk/pdf-sdk/>.

### 1.3 PDFium version of Foxit PDF SDK

Foxit PDF SDK (PDFium) is an integrated, commercial PDF SDK that includes the APIs in both PDFium and Foxit commercial PDF SDK. It is provided to help developers who have already integrated PDFium APIs into their projects to transfer from PDFium open-source SDK to Foxit commercial PDF SDK easily and conveniently.

PDFium version of Foxit PDF SDK provides almost all SDK APIs that are included in PDFium open-source library. The PDF projects that use PDFium APIs do not need to make drastic changes if customers require additional PDF functionalities that are only provided in Foxit commercial PDF SDK. PDFium version of Foxit PDF SDK is available on Windows, Linux and Mac platforms.

PDFium open-source project has been developed for different versions in the open-source community by the developers who are interested in it. Foxit PDF SDK (PDFium) wraps the PDFium APIs based on Google PDFium committed version "c3e4ae5fe5067723b58a2029a95c6411c92bed15".

## 1.4 Features

PDFium version of Foxit PDF SDK provides all the PDF features included in Foxit commercial PDF SDK. It has a standard package and 9 optional packages, each of which contains several features as listed in Table 1-1. Users can choose the packages and features based on their needs, and then purchase the corresponding function packages of PDF SDK to use the PDFium version of Foxit PDF SDK.

**Table 1-1**

Package name	Features included
<b>Standard</b>	PDF rendering, document navigation, get page information, font information, text extraction and search, access to PDF objects, asynchronous PDF, text reflow, and access to layer's information.
<b>Edit</b>	Edit document, pages and PDF objects.
<b>Image Conversion</b>	Convert between PDF files and images (bmp, tif, jpx, jpg, gif, etc.).
<b>Form</b>	Access form information, import a FDF file into a form and export data to a FDF file.
<b>Annotation</b>	Create, edit and remove annotations. Create watermarks.
<b>Security</b>	Support password, certificate, DRM and custom encryption.
<b>Pressure Sensitive Ink</b>	Generate PSI and convert PSI to annotation.
<b>Barcode</b>	Generate a barcode bitmap from a given string and barcode type.
<b>Signature</b>	Sign a PDF document, verify a signature, add or delete a signature field.
<b>RMS</b>	Support Microsoft RMS encryption and decryption.

## 1.5 Evaluation

PDFium version of Foxit PDF SDK allows users to download a 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 PDFium version of Foxit PDF SDK.

## 1.6 License

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

## 1.7 About this guide

This guide is intended for the developers who require additional PDF functionalities that are not included in PDFium open-source library and need to transfer from PDFium to Foxit commercial PDF SDK. It aims at introducing license, installation package structure on Windows, Linux and Mac platforms, and customization sections.

## 2 GETTING STARTED

---

It is very easy to setup PDFium version of Foxit PDF SDK and see it in action! It takes just a few minutes and we will show you how to use it. PDFium version of Foxit PDF SDK is a cross platform commercial PDF SDK product. It supports the same interfaces for desktop system of Windows, Linux, and Mac. The following sections introduce the structure of installation package, how to apply a license, and how to run a demo.

### 2.1 System Requirements

- **Windows:**

Windows 7 and 8 (32-bit, 64-bit)

Windows Server 2003, 2008 and 2012 (32-bit and 64-bit)

The release package includes a 32 bit version and native 64 bit version DLL library for windows 32/64.

Note: it only supports for Windows 8 classic style not for Store App.

- **Linux:**

Linux 32-bit and Linux 64-bit OS

All Linux samples have been tested on Centos 6.3 32/64 bit.

The release package includes both 32-bit and 64-bit version Linux libraries (.so files).

- **Mac OS X:**

Mac OS X 10.6 to 10.10

### 2.2 What is in the Package

In this guide, one thing to note is that the highlighted rectangles in the figures are the version of the SDK. Here the SDK version is 5.2, so it shows 5\_2.

#### 2.2.1 For Windows

Download PDFium version of Foxit PDF SDK zip for Windows package and extract it to a new directory like "foxitpdfium\_5\_2\_win". The structure of the release package is shown in Figure 2-3. This package contains the following folders:

**docs:** API references, Quick Guide, Developer Guide, and Demo Tutorial

- include:** header files for Foxit PDF SDK API and PDFium API
- lib:** libraries and license files
- samples:** sample projects and demos

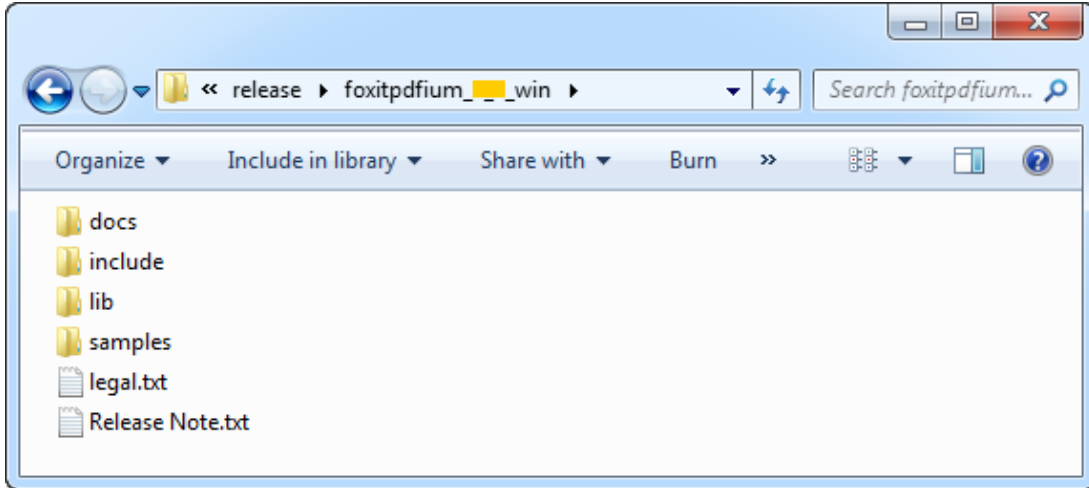


Figure 2-1

In “samples\pdfium\_simple\_sample”, there are more than 20 demos that cover a wide range of PDF applications as shown in Figure 2-2.

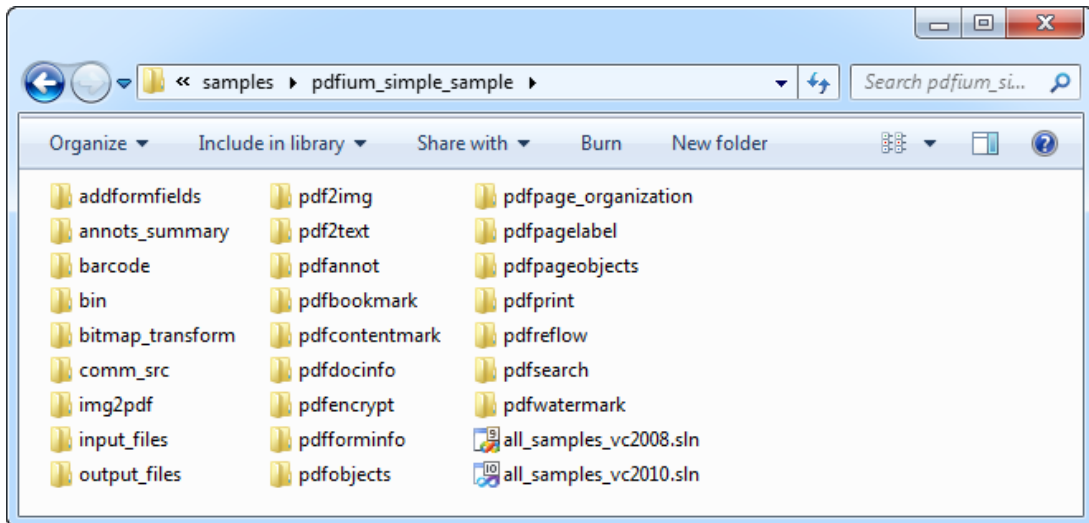


Figure 2-2



### 2.2.2 For Linux

Download PDFium version of Foxit PDF SDK zip for Linux package and extract it to a new directory like “foxitpdfium\_5\_2\_linux”. The structure of the release package is shown in Figure 2-3. This package contains the following folders:

- docs:** API references, Quick Guide, Developer Guide, and Demo Tutorial
- include:** header files for Foxit PDF SDK API and PDFium API
- lib:** libraries and license files
- samples:** sample projects and demos

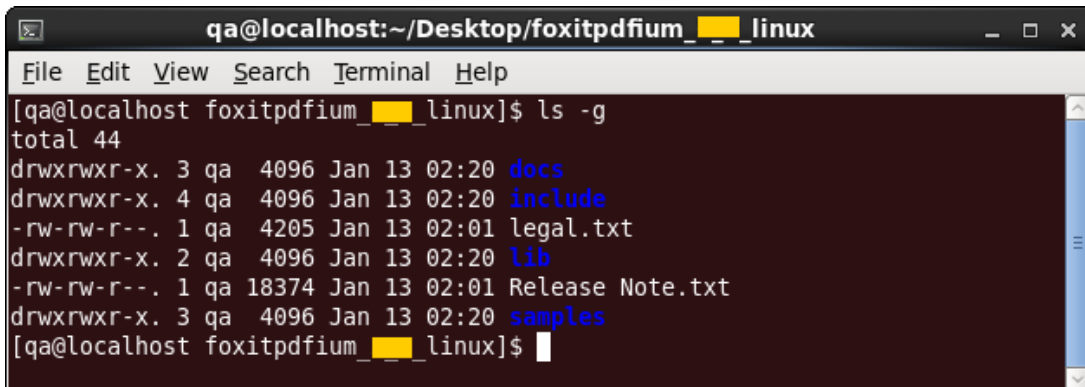


Figure 2-3

In “samples\pdfium\_simple\_sample”, there are more than 20 demos that cover a wide range of PDF applications as shown in Figure 2-4.

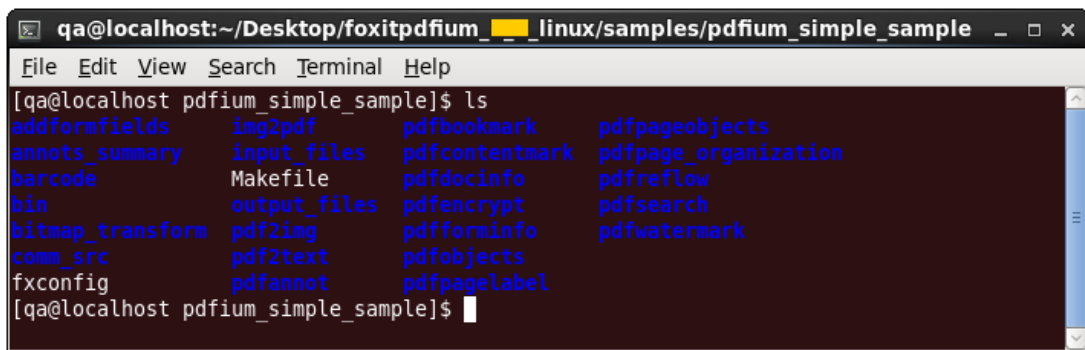


Figure 2-4

### 2.2.3 For Mac

Download PDFium version of Foxit PDF SDK zip for mac package and extract it to a new directory like “foxitpdfium\_5\_2\_mac”. The structure of the release package is shown in Figure 2-5. This package contains the following folders:

- docs:** API references, Quick Guide, Developer Guide, and Demo Tutorial
- include:** header files for Foxit PDF SDK API and PDFium API
- lib:** libraries and license files
- samples:** sample projects and demos

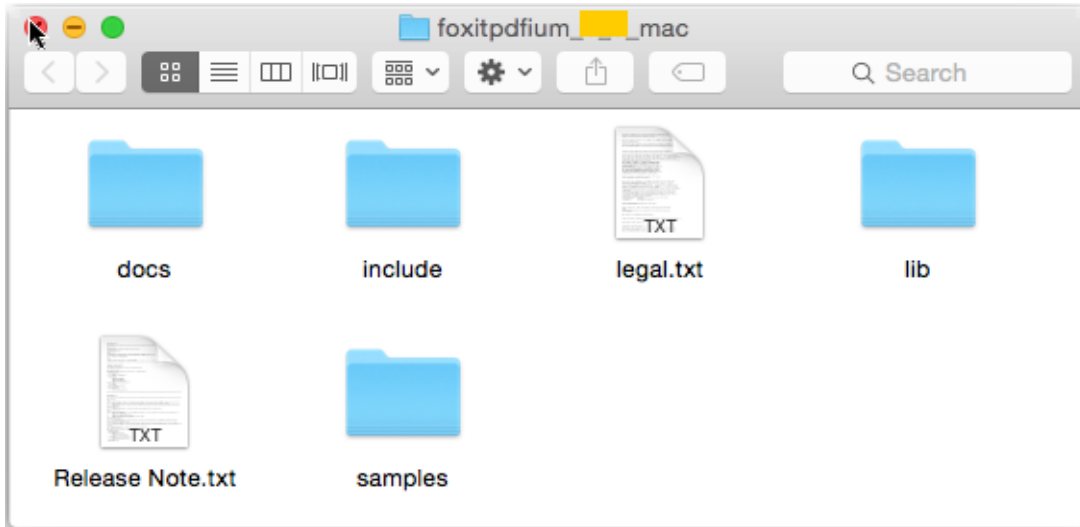


Figure 2-5

In “samples\pdfium\_simple\_sample”, there are more than 20 demos that cover a wide range of PDF applications as shown in Figure 2-6.

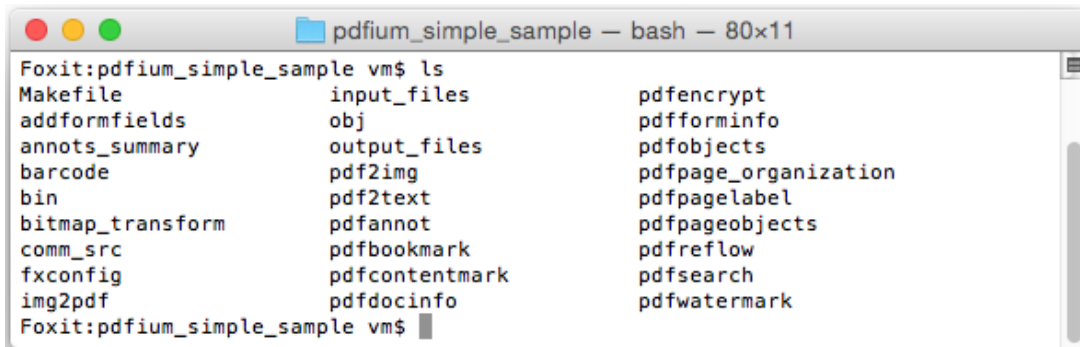


Figure 2-6

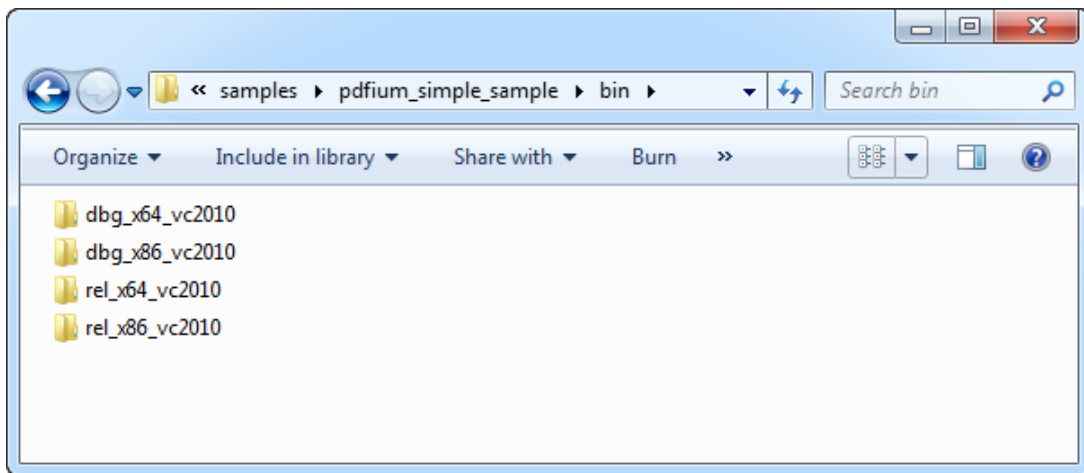
### 2.3 How to run a demo

PDFium version of Foxit PDF SDK provides many simple samples that demonstrate the power of the PDF SDK for developers on how to effectively integrate Foxit PDF SDK (PDFium) API into their own applications. All these simple samples are also provided in Foxit PDF SDK C API packages, but the difference is that the samples in PDFium version of Foxit PDF SDK use APIs in PDFium and Foxit PDF SDK, while those in the Foxit PDF SDK version use only Foxit PDF SDK.

### 2.3.1 For Windows

To run a demo in Visual Studio, load the visual studio solution files “all\_samples\_vc2008.sln” or “all\_samples\_vc2010.sln” depending on your Visual Studio version. Another way is to load the .vcxproj file in the folder of a specific demo project.

In this guide, “pdf2text” demo will be used as the example to show how to run a demo in Visual Studio 2010. Open “pdf2text\pdf2text\_vc2010.vcxproj” with Visual Studio 2010 and build it. The executable file “pdf2text.exe” is generated in one of the following four folders as shown in Figure 2-7, which depends on the build configuration.



**Figure 2-7**

To run the executable file, in this example, find it at “bin\dbg\_x86\_vc2010\pdf2text.exe”, there are two options: in command line or in Visual Studio. When running in command line, start “cmd.exe”, navigate to “bin\dbg\_x86\_vc2010” and run “pdf2text”. The terminal output is shown in Figure 2-8. When running in Visual Studio, click on “Debug->Start Debugging” or “Debug->Start Without Debugging” on the menu bar to run pdf2text.exe. This is shown in Figure 2-9.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Administrator>cd C:\release\foxitpdfium_..._win\samples\pdfium_simple_s
ample\bin\dbg_x86_vc2010

C:\release\foxitpdfium_..._win\samples\pdfium_simple_sample\bin\dbg_x86_vc2010>p
df2text.exe
Foxit PDF SDK example: pdf2text

Success: Create the default Foxit PDF SDK manager.
Success: Unlock the library.
Success: Initialize the PDF module.
Using extension implementation to open the file.
Success: Use <wb> mode to open the file gsdkLog.txt and associate it with a stre
am.
Success: Create the file object.
Input from folder: C:\release\foxitpdfium_..._win\samples\pdfium_simple_sample\b
in\dbg_x86_vc2010/../../input_files
Output to folder: C:\release\foxitpdfium_..._win\samples\pdfium_simple_sample\bi
n\dbg_x86_vc2010/../../output_files/pdf2text

===== Convert AboutFoxit.pdf to TXT =====
Success: Load PDF document successfully.
Using extension implementation to open the file.
Success: Use <wb+> mode to open the file AboutFoxit.txt and associate it with a
stream.
Success: Create the file object.
Success: Open file successfully .

Success: Load the PDF page with index 0.
Success: Load the text page.
Success: Convert to TXT form page with index 0.
        finish page count: 1
        elapsed time: 310ms

Finalize PDF module.
Finalize Foxit PDF SDK library.

C:\release\foxitpdfium_..._win\samples\pdfium_simple_sample\bin\dbg_x86_vc2010>
```

Figure 2-8

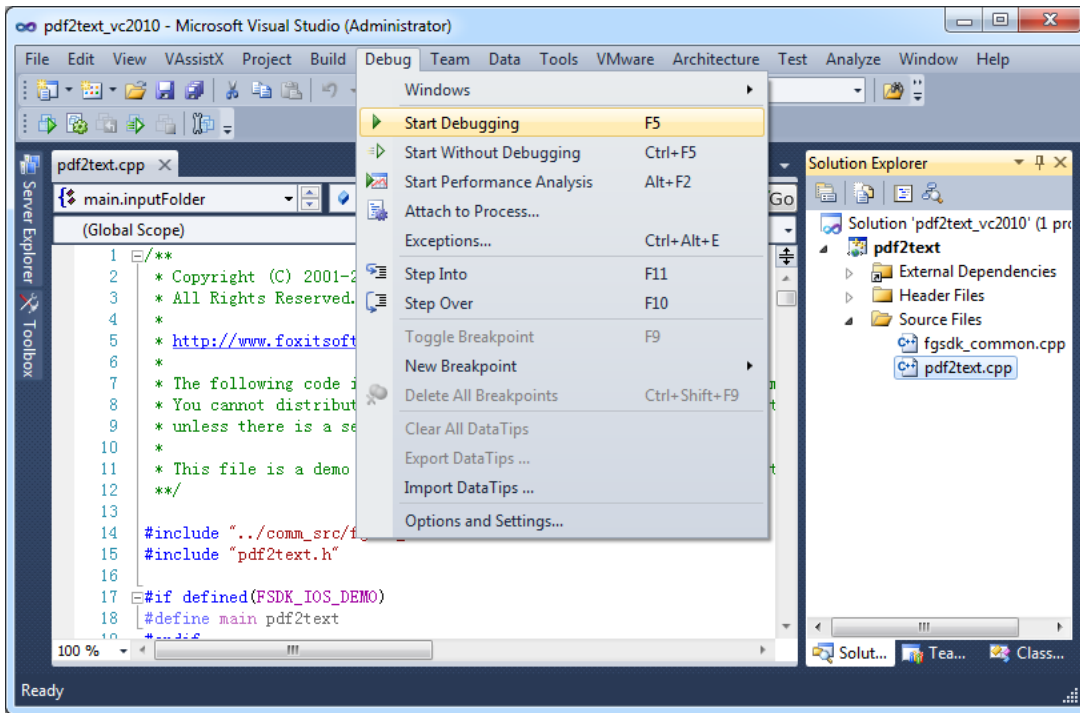


Figure 2-9

Some demos will generate output files (pdf, text or image files) to a folder named by the project name under “samples/pdfium\_simple\_sample/output\_files/”. In this example, output files are generated to “samples/pdfium\_simple\_sample/output\_files/pdf2text”.

### 2.3.2 For Linux

In a terminal window, run “make” or “make ver=debug” to build all demos or run “make project\_name” to build a demo named “project\_name”.

In this guide, “pdf2text” demo will be used as the example to show how to run a demo on Linux platform. Figure 2-10 shows the build process for pdf2text demo.

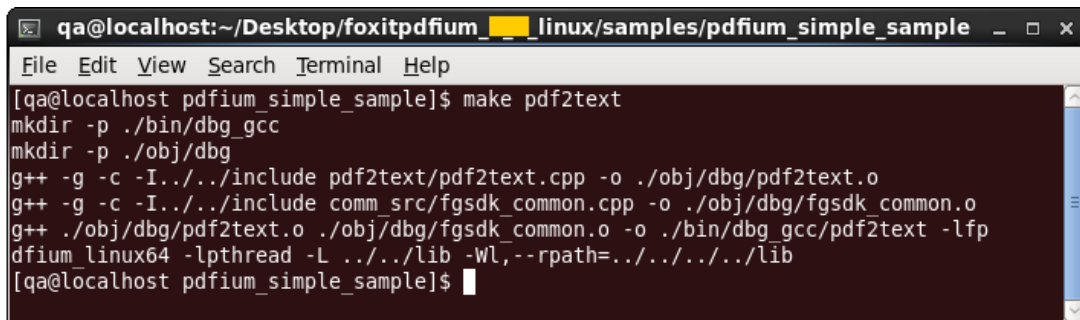
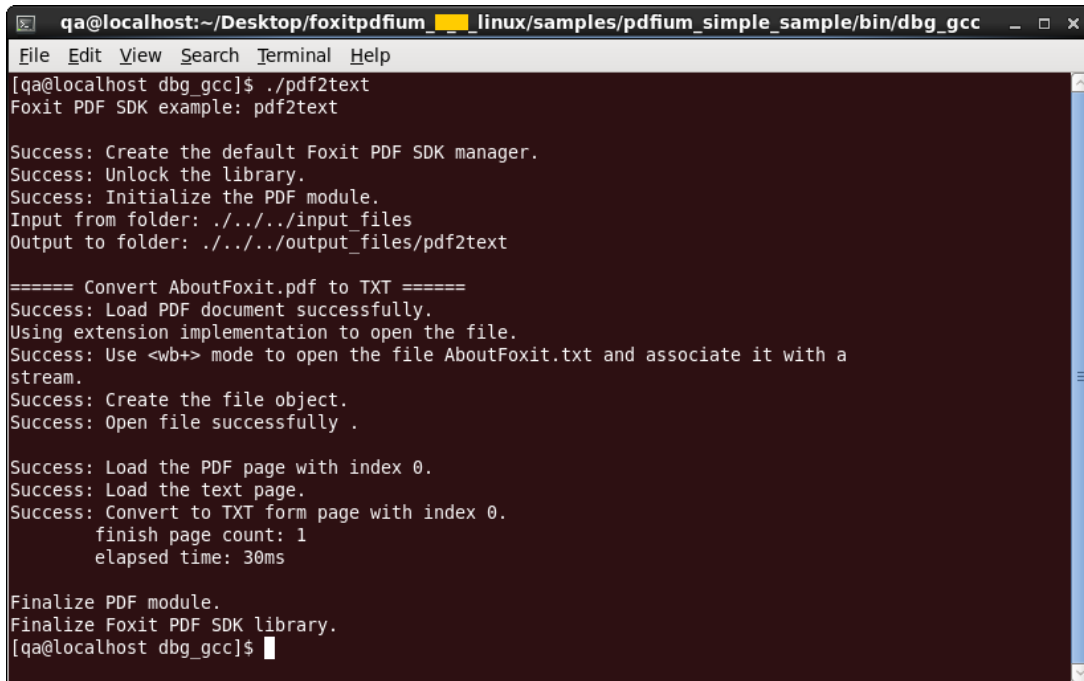


Figure 2-10

After building, the binary files are generated in folder “samples/pdfium\_simple\_sample/bin/rel\_gcc” or “samples/pdfium\_simple\_sample/bin/dbg\_gcc” depending on the build option. Navigate to the folder with the terminal, and run the binary file to get the demo running. Figure 2-11 shows a screenshot when running pdf2text demo.



```
qa@localhost:~/Desktop/foxitpdfium_linux/samples/pdfium_simple_sample/bin/dbg_gcc
File Edit View Search Terminal Help
[qa@localhost dbg_gcc]$ ./pdf2text
Foxit PDF SDK example: pdf2text

Success: Create the default Foxit PDF SDK manager.
Success: Unlock the library.
Success: Initialize the PDF module.
Input from folder: ../../input_files
Output to folder: ../../output_files/pdf2text

===== Convert AboutFoxit.pdf to TXT =====
Success: Load PDF document successfully.
Using extension implementation to open the file.
Success: Use <wb+> mode to open the file AboutFoxit.txt and associate it with a
stream.
Success: Create the file object.
Success: Open file successfully .

Success: Load the PDF page with index 0.
Success: Load the text page.
Success: Convert to TXT form page with index 0.
      finish page count: 1
      elapsed time: 30ms

Finalize PDF module.
Finalize Foxit PDF SDK library.
[qa@localhost dbg_gcc]$
```

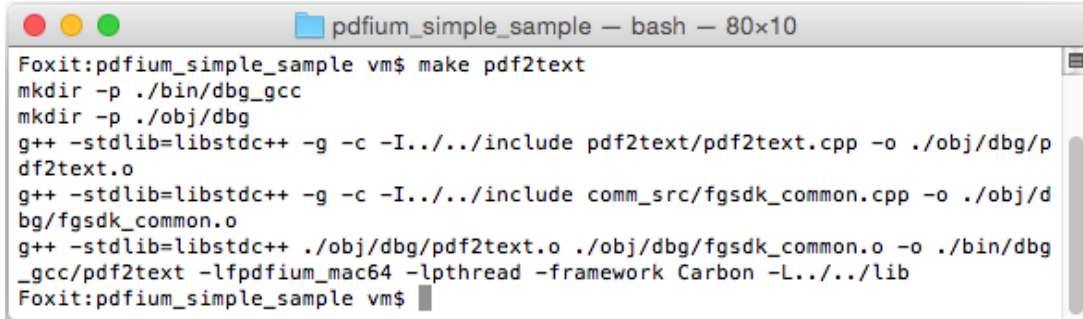
Figure 2-11

Some demos will generate output files (pdf, text or image files) to a folder named by the project name under “samples/pdfium\_simple\_sample/output\_files/”. In this example, output files are generated to “samples/pdfium\_simple\_sample/output\_files/pdf2text”.

### 2.3.3 For Mac

In a terminal window, run “make” or “make ver=debug” to build all demos or run “make project\_name” to build a demo named “project\_name”.

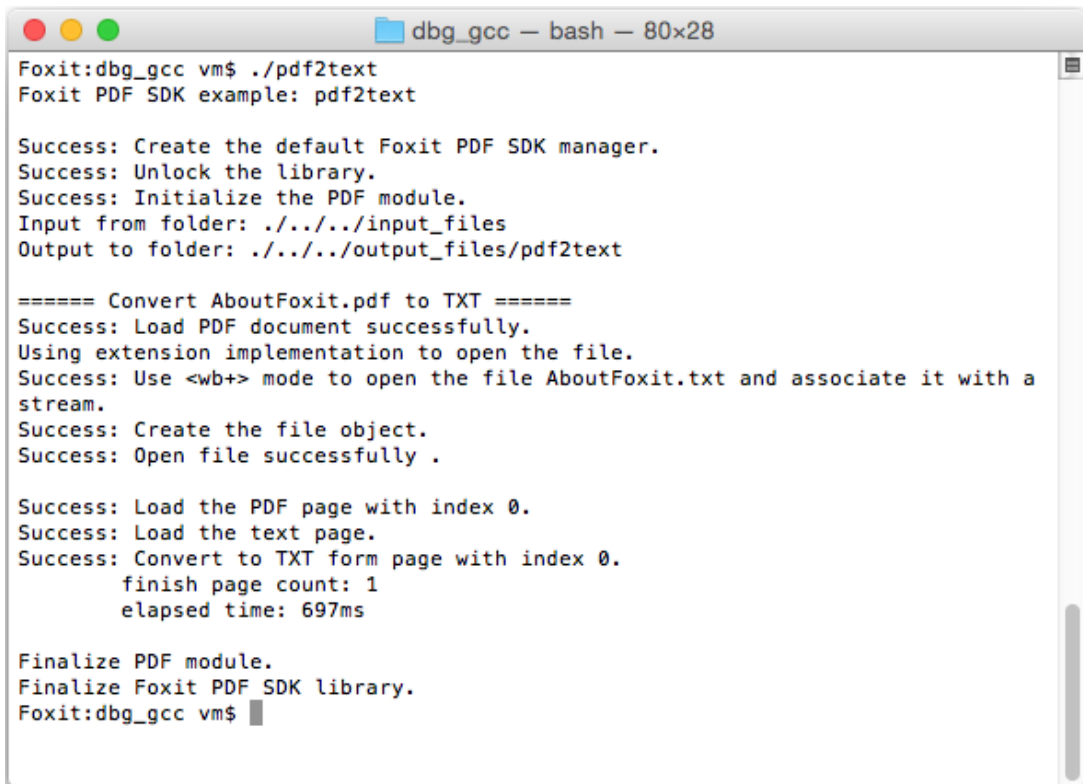
In this guide, “pdf2text” demo will be used as the example to show how to run a demo on Mac platform. Figure 2-12 shows the build process for pdf2text demo.



```
Foxit:pdfium_simple_sample vm$ make pdf2text
mkdir -p ./bin/dbg_gcc
mkdir -p ./obj/dbg
g++ -stdlib=libstdc++ -g -c -I../include pdf2text/pdf2text.cpp -o ./obj/dbg/p
df2text.o
g++ -stdlib=libstdc++ -g -c -I../include comm_src/fgsdk_common.cpp -o ./obj/d
bg/fgsdk_common.o
g++ -stdlib=libstdc++ ./obj/dbg/pdf2text.o ./obj/dbg/fgsdk_common.o -o ./bin/dbg
_gcc/pdf2text -lpdfium_mac64 -lpthread -framework Carbon -L../lib
Foxit:pdfium_simple_sample vm$
```

Figure 2-12

After building, the binary files are generated in “samples/simple samples/bin/rel\_gcc” or “samples/simple samples/bin/dbg\_gcc” depending on the build option. Navigate to the folder with the terminal, and run the binary file to get the demo running. Figure 2-13 shows the screen output when running pdf2text demo.



```
Foxit:dbg_gcc vm$ ./pdf2text
Foxit PDF SDK example: pdf2text

Success: Create the default Foxit PDF SDK manager.
Success: Unlock the library.
Success: Initialize the PDF module.
Input from folder: ../../input_files
Output to folder: ../../output_files/pdf2text

===== Convert AboutFoxit.pdf to TXT =====
Success: Load PDF document successfully.
Using extension implementation to open the file.
Success: Use <wb+> mode to open the file AboutFoxit.txt and associate it with a
stream.
Success: Create the file object.
Success: Open file successfully .

Success: Load the PDF page with index 0.
Success: Load the text page.
Success: Convert to TXT form page with index 0.
      finish page count: 1
      elapsed time: 697ms

Finalize PDF module.
Finalize Foxit PDF SDK library.
Foxit:dbg_gcc vm$
```

Figure 2-13

Some demos will generate output files (pdf, text or image files) to a folder named by the project name under “samples/pdfium\_simple\_sample/output\_files/”. In this example, output files are generated to “samples/pdfium\_simple\_sample/output\_files/pdf2text”.

### 3 CHANGES COMPARED WITH PDFIUM OPEN-SOURCE LIBRARY

PDFium version of Foxit PDF SDK is a commercial PDFium which wraps almost all SDK APIs in PDFium open-source library. It can help developers to transfer from PDFium open-source SDK to Foxit commercial PDF SDK conveniently.

Some changes have been made in PDFium version of Foxit PDF SDK when wrapping the SDK APIs of PDFium open-source library. This section will introduce these changes.

#### 3.1 Initialize library

In PDFium open-source library, developers need to call the following function to initialize the SDK library:

```
FPDF_InitLibrary();
```

In PDFium version of Foxit PDF SDK, developers should call the following three functions in turn to initialize the SDK library:

```
FPDF_InitLibrary();  
FSCRT_License_UnlockLibrary(licenseKey, unlockCode);  
FSCRT_PDFModule_Initialize();    // Initialize PDF module.
```

**Note:** The function **FSCRT\_License\_UnlockLibrary** is provided to unlock Foxit PDF SDK license. An example of applying a license is shown below. The string “**licenseKey**” can be found in the “gsdk\_sn.txt” (the string after “SN=”) and the string “**unlockCode**” can be found in the “gsdk\_key.txt” (the string after “Sign=”).

```
// Apply a license to unlock Foxit PDF SDK(PDFium)  
FSCRT_BSTR license_id;  
FSCRT_BStr_Init(&license_id);  
license_id.str = (FS_LPSTR)"licenseKey";  
license_id.len = strlen(license_id.str);  
FSCRT_BSTR unlockCode;  
unlockCode.str = (FS_LPSTR)"unlockCode";  
unlockCode.len = strlen(unlockCode.str);  
ret = FSCRT_License_UnlockLibrary(&license_id, &unlockCode);  
if (FSCRT_ERRCODE_SUCCESS == ret)  
    return TRUE;  
else  
{  
    return FALSE;  
}
```

#### 3.2 Destroy library

In PDFium open-source library, Developers need call the following function to destroy the SDK library:

```
FPDF_DestroyLibrary();
```



In PDFium version of Foxit PDF SDK, Developers should call the following two functions in turn to destroy the SDK library:

```
FSCRT_PDFModule_Finalize(); // Finalize PDF module.  
FPDF_DestroyLibrary();
```

### 3.3 Add macro definitions

PDFium version of Foxit PDF SDK adds a macro definition as follows:

**fpdfview.h:**

```
#define FPDF_ERR_INVALIDLICENSE 6 // Error code: License authorization error.
```

### 3.4 Rename some functions and macro definitions

Some functions and macro definitions in PDFium open-source library have been renamed in PDFium version of Foxit PDF SDK as shown in Table 3-1.

**Table 3-1**

PDFium open-source library	PDFium version of Foxit PDF SDK
<code>int (*Field_browse)(struct _IPDF_JsPlatform* pThis,void* filePath, int length);</code>	<code>int (*File_browse)(struct _IPDF_JsPlatform* pThis,void* filePath, int length);</code>
<code>#define FPDF_RENDER_READER 0</code>	<code>#define FPDF_RENDER_READY 0</code>

### 3.5 Remove some unused definitions and enumerations

Some unused definitions and enumerations in PDFium open-source library have been removed in PDFium version of Foxit PDF SDK. Following are the removed definitions:

**fpdfview.h:**

```
typedef void* FPDF_MODULEMGR;  
typedef void* FPDF_HMODULE;  
typedef void* FPDF_DOCSCHHANDLE;  
typedef int FPDF_ERROR;  
#define FPDF_ERR_PAGE 6 // Page not found or content error.  
#define FPDF_DEBUG_INFO 0x80 // Set if you want to get some debug info.  
// Please discuss with Foxit first if you need to collect debug info.  
#define FPDF_NO_CATCH 0x100 // Set if you don't want to catch exception.
```

**fpdf\_fwlevent.h:**

```
FWL_EVENTTYPE;  
FWL_EVENT_MOUSECMD;  
FWL_EVENT_MOUSE;  
FWL_EVENT_MOUSEWHEEL;  
FWL_EVENTKEYCMD;  
FWL_EVENT_KEY;  
FWL_EVENT ;  
typedef int FPDF_INT32;  
typedef unsigned int FPDF_UINT32;  
typedef float FPDF_FLOAT;
```

#### **fpdf\_transformpage.h**

```
typedef void* FPDF_PAGEARCSAVER;  
typedef void* FPDF_PAGEARCLoader;
```

### 3.6 Other Notices

The **FPDF\_InitLibraryWithConfig** interface in PDFium open-source Library has not been wrapped in PDFium version of Foxit PDF SDK. Therefore, if your applications have already called this interface, and when you use our PDFium version of Foxit PDF SDK (commercial PDFium), the applications will prompt that the interface is not defined.

## 4 FAQ

---

**1. What is the price of PDFium version of Foxit PDF SDK?**

To receive a price quotation, please send a request to [sales@foxitsoftware.com](mailto:sales@foxitsoftware.com) or call Foxit sales at 1-866-680-3668.

**2. How can I activate it after purchasing PDFium version of Foxit PDF SDK?**

There are detailed descriptions on how to apply a license in the section 2.4. You can refer to the descriptions to activate a license.

**3. How can I look for technical support when I try PDFium version of Foxit PDF SDK?**

You can send email to [support@foxitsoftware.com](mailto:support@foxitsoftware.com) for any questions or comments or call our support at 1-866-693-6948.

## 5 SUPPORT

---

**Foxit support home link:**

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

**Sales contact phone number:**

Phone: 1-866-680-3668

Email: [sales@foxitsoftware.com](mailto:sales@foxitsoftware.com)

**Support & General contact:**

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

Email: [support@foxitsoftware.com](mailto:support@foxitsoftware.com)