

# **Foxit Reader SDK ActiveX Programming Guide**

Revision 1.0

(C) Foxit Software Company

2006.5

Overview  
Tutorials  
Reference

## Overview

**Foxit Reader SDK ActiveX** is a visual component that offers PDF displaying capability and it can be easily integrate into a wide range of applications. It is developed by Foxit Software Company for rendering PDF document with minimal resource demand and re-distribution size.

**Foxit Reader SDK ActiveX** uses the same rendering engine as [Foxit Reader](#). Therefore any PDF file that can be opened by Foxit Reader, can also be opened by this ActiveX, with same high quality and fast speed.

The ActiveX version of Foxit Reader SDK allows the application to display PDF much more easier than using the DLL version of our SDK, because the ActiveX not only displays pages, but also handles all the user interface, such as navigating, zooming, rotating, scrolling, and printing.

**Foxit Reader SDK ActiveX** runs on Windows 95/NT or later OS. For detailed information on runtime configuration, please refer to re-distribution section.

**UNLOCK Code:** If you have purchased Foxit Reader SDK ActiveX and received the full version of the ActiveX and the unlock code, you should call UnLockActiveX function once, before you call ANY functions of the ActiveX. The function is described in the Reference section. You don't need to call this function if you just want to evaluate the ActiveX.

## Tutorials

The Foxit Reader SDK ActiveX control comes in a single file, to install it, use "regsvr32 foxitreader\_ax.ocx" command. You may need to specify proper path if foxitreader\_ax.ocx is not stored in current directory.

The ActiveX control handles user interface for you. It also supply some properties and methods so that your application may control the ActiveX.

In the following example, suppose we have already created an ActiveX control called FoxitReaderSDK.

### **TUTORIAL 1: Open a PDF File**

We will open a PDF document named "testdoc.pdf" and go to the first page of the document.

```
// NOTE: If you are evaluating ActiveX, you don't need unlock the ActiveX,  
// then evaluation marks will be shown on all PDF pages.  
// Otherwise, for paid customers, please unlock the ActiveX first
```

```
FoxitReaderSDK.UnlockActiveX("license_id","unlock_code");  
FoxitReaderSDK.OpenFile("testdoc.pdf", "");
```

### **TUTORIAL 2: Go to a specified page**

We will go to the third page of the document, and keep the page size, rotate direction.

```
FoxitReaderSDK.GotoPage(2, -1,-1).
```

### **TUTORIAL 3: Zoom the page**

If you want to show the page with its original size, you can use following code:

```
FoxitReaderSDK.SetZoomLevel(0)  
or  
FoxitReaderSDK. SetZoomLevel(100);
```

If you want to show the page with its original size doubled, use following code:

```
FoxitReaderSDK. SetZoomLevel(200);
```

For more information, see [Function Reference](#).section of this guide.

### **TUTORIAL 4: Rotate the page**

PDF pages can be displayed upright like it is, or, it can be rotated in both direction (clockwise or counter-clockwise), or it can be displayed upside-down. To achieve different rotation display, you just need to call `SetRotateMode` .

If you want to rotate the page (90 degrees) clockwise, you can use following code

```
FoxitReaderSDK. SetRotateMode(1);
```

### **TUTORIAL 5: Print**

What you need to do is to call `PrintFile` method, and then a “print” dialog will be pop-up and user will be able to print from there  
or call `Print` method ,it supports silent printing

### **TUTORIAL 6: Hide or show some specified windows**

You can call `ShowToolBar` to show or hide the toolbar. Likewise, you may call `ShowBookmark` to show or hide the bookmark and call `ShowStatusBar` to show or hide the statusbar, call `ShowTitleBar` to show or hide the titlebar.

### **TUTORIAL 7: Two events used to notify a container that a PDF Document has been Loaded or UnLoaded.**

When a Document is Loaded, the event `DocLoad` fires. When a Document is UnLoaded, the event `DocUnLoad` fires, You can use these events when you want to do something when the Document is loaded or unloaded.

## Reference

This section describes all properties and methods exposed by ActiveX. Please note that the reference only shows everything in C syntax. If you use a programming language other than C/C++, you have to make necessary change.

### Properties:

FilePath (BSTR ,read-only):	Full path to the current opened PDF file. If no PDF file is opened, the property is empty;
Title (BSTR ,read-only):	The title of the current opened PDF file.
Author (BSTR ,read-only):	The author of the current opened PDF file.
PageCount (long ,read-only):	Total number of pages in the current opened PDF file.
CurPage (long ,read-only):	Current page index of the PDF file. Page index starts from zero for the first page.
Rotate (short ,read-only):	Current rotate orientation, the value can be 0 (normal), 1 (rotated 90 degrees clockwise), 2 (rotated 180 degrees), 3 (rotated 90 degrees counter-clockwise).
Zoomlevel (long ,read-only):	Percentage of current zoom level, The value can be between 10 and 1600.
CurrentTool :	Sets the currently selected tool. Can be set to one of the following tools that are already supported in FoxIt Viewer: <ul style="list-style-type: none"><li>- HandTool</li><li>- SelectText</li><li>- SnapShot</li></ul>

## **Methods:**

### **Function: SetZoomLevel**

Change the zoom level of the PDF.

#### **Prototype:**

```
void SetZoomLevel (long zoom_level);
```

#### **Parameters:**

zoom\_level - zoom factor ( percentage). Normally, this value can be between 10 and 1600.

You may also use special value for special purpose as following:

0=displaying the page in actual page size, this has the same effect as setting zoom level to 100%.

1=displaying the page with proper zoom level so that the whole page can be fit into the client window,

2=displaying the document with proper zoom level so that the width of the page fit to the client window.

#### **Return value:**

None

### **Function: PrintFile**

Popup a "Print" dialog so that users can print currently opened PDF file.

#### **Prototype:**

```
void PrintFile()
```

## Parameters

None

## Return value:

None

## Function: Print

Implements silent printing

## Prototype:

```
Void Print(long hDc, short nPageFrom, short nPageTo, short nCopies, bool bCollate, float fScaling, bool bAutoRotate, bool bAutoCenter)
```

## Parameters

hDC - Handle to printer DC

nPageFrom - Starting page number

nPageTo - ending page number

nCopies - Number of copies

bCollate - Collate

fScaling - number of scaling

The parameter range of fScaling is from 800 to 8.33, e.g. fScaling=50, it means the scaling is set as 50%.

bAutoRotate -set auto-rotate

1=auto-rotate

bAutoCenter -set auto-center

1=auto-center

## Function: OpenFile

Open a PDF file and show a specified page.

### Prototype:

BOOL OpenFile(BSTR file\_path, long password)

### Parameters

File\_path - Path to the PDF file (including file extension).

The path can be a file on an HTTP server,

Password - For password protected pdf documents, this is the user/owner password for been able to load the document.

### Return value:

Nonzero if the pdf file is successfully opened, otherwise it is zero.

### Remarks:

If the file\_path is pointing to a file on a HTTP server, the pdf file will be downloaded to the client machine's memory with a progress bar being displayed in the ActiveX's window, and open the pdf file from memory When the download is complete.

## Function: OpenStream

Open a pdf file from the IStream interface

### Prototype:

BOOL OpenStream (IUnknown\* Stream)

### Parameters

Stream - IStream interface.

### Return value:

Nonzero if the pdf file is successfully opened, otherwise it is zero.

## **Function: CloseFile**

Close a opened PDF file.

### **Prototype:**

```
void CloseFile()
```

### **Parameters**

None

### **Return value:**

None

## **Function: SetRotateMode**

Rotate the PDF page in different direction

### **Prototype:**

```
void SetRotateMode(short Rotate);
```

### **Parameters:**

Rotate - The parameter can be 0, 1, 2, or 3.  
0 = upright,  
1 = rotated (90 degrees) clockwise,  
2 = rotate upside-down,  
3 = rotated (90 degrees) counter-clockwise.

### **Return value:**

None

## Function: GotoPage

Go to a specified page with specified zoom factor and rotate orientation.

### Prototype:

```
void GotoPage(long page_index, long zoom_level, short Rotate);
```

### Parameters:

- page\_index - Index for the page to be loaded. 0 for the first page in document.
- zoom\_level - Zoom factor (percentage). This value can be between 10 and 1600.

You may also use special value for special purpose as following:

-1=keep the current zoom level

0=displaying the page in actual page size, this has the same effect as setting zoom level to 100%.

1=displaying the page with proper zoom level so that the whole page can be fit into the client window,

2=displaying the document with proper zoom level so that the width of the page fit to the client window.

- Rotate - The parameter can be -1,0, 1, 2, or 3.
- 1=keep current rotate direction
  - 0 = upright,
  - 1 = rotated (90 degrees) clockwise,
  - 2 = rotate upside-down,
  - 3 = rotated (90 degrees) counter-clockwise.

### Return value:

None

## **Function: ShowToolBar**

Show or hide the toolbar;

### **Prototype:**

```
void ShowToolBar(short show);
```

### **Parameters:**

Show - If this parameter is 0, the toolbar will be invisible. If this parameter is nonzero, the toolbar will be visible.

### **Return value:**

None

## **Function: ShowBookmark**

Show or hide the Bookmark

### **Prototype:**

:

```
void ShowBookmark(short show)
```

### **Parameters:**

show - If this parameter is 0, the Bookmark will be invisible. If this parameter is nonzero, the Bookmark will be visible.

### **Return value:**

None

## **Function: ShowStatusBar**

Show or hide the status bar

### **Prototype:**

```
void ShowStatusBar(short show);
```

### **Parameters:**

show - If this parameter is 0, the Status Bar will be invisible. If this parameter is nonzero, the Status Bar will be visible.

### **Return value:**

None

## **Function: ShowTitleBar**

Show or hide the title bar

### **Prototype:**

```
void ShowTitleBar(short show);
```

### **Parameters:**

show - If this parameter is 0, the Title Bar will be invisible. If this parameter is nonzero, the Title Bar will be visible.

### **Return value:**

None

## **Function: ShowProperties**

Pop up a Dialog to show some properties of the PDF file

**Prototype:**

void ShowProperties();

**Parameters:**

None

**Return value:**

None

**Function: FindNext**

Search the document for a string and returns true if it finds the search string, otherwise false

**Prototype:**

BOOL FindNext(BSTR FindWhat, BOOL bMatchCase, BOOL bMatchWholeWord, BOOL bSearchDown)

**Parameters:**

FindWhat - the string to find.

bMatchCase - determine whether the user wants to match the case of the find string exactly.

bMatchWholeWord - determine whether the user wants to match entire words only.

bSearchDown –determine whether the user wants the search to proceed in a downward direction

**Return value:**

Nonzero if the user wants to find the next occurrence of the search string; otherwise 0.

## **Function: UnLockActiveX**

Unlock the activex using license key info received from Foxit Company

### **Prototype:**

```
void UnLockActiveX(BSTR lisenca_id, BSTR unlock_code)
```

### **Parameters:**

license\_id - A string received from Foxit identifying the SDK license

unlock\_code - A string received from Foxit for unlocking the DLL

### **Return value:**

None

### **Remarks:**

For evaluating ActiveX , you don't need to call this function, then evaluation marks will be shown with all rendered pages.

For paid activex customers, this should be the first function to be call before you call any other ActiveX functions.

## **Events:**

### **Function: DocLoad**

When a Document is Loaded , the event fires

### **Prototype:**

```
void DocLoad (BSTR filepath)
```

### **Parameters:**

filepath - Path to the PDF file (including file extension).

**Return value:**

None

**Function: DocUnLoad**

When a Document is UnLoaded , the event fires

**Prototype:**

void DocUnLoad (BSTR filepath)

**Parameters:**

filepath - Path to the PDF file (including file extension).

**Return value:**

None

**Function: BeforeDraw**

Sent just before the painting of the viewer contents is about to begin.

**Prototype:**

void BeforeDraw (long dc)

**Parameters:**

dc - Handle to the DC on which painting is about to take place.

**Return value:**

None

## **Function: AfterDraw**

Sent just after the painting of the viewer contents is completed.

### **Prototype:**

```
void AfterDraw (long dc)
```

### **Parameters:**

dc - Handle to the DC on which painting is about to take place.

### **Return value:**

None

## **Function: LinkClicked**

Triggered when a hyperlink (URL etc) is clicked by user.

### **Prototype:**

```
void LinkClicked (BSTR link, BOOL * cancel)
```

### **Parameters:**

link - URL or URI of link.

cancel - if cancel variable is set to nonzero the control will not 'follow' the hyperlink, else execute link normally.

### **Return value:**

None