

# 用户手册



Foxit PDF SDK (ActiveX) 3.0

**Foxit**

©2010 福昕软件. 保留所有权

**Microsoft**  
GOLD CERTIFIED  
Partner

©福昕软件版权所有。保留所有权利。

未经福昕公司授权，任何人或组织不得以任何手段与形式对本文档内容进行形式复制、转移、分发或存储。

Anti-Grain Geometry - Version 2.3, Copyright (C) 2002-2005 Maxim Shemanarev

(<http://www.antigrain.com/>). FreeType2 (freetype2.2.1), Copyright (C) 1996-2001, 2002, 2003, 2004 |

David Turner, Robert Wilhelm, and Werner Lemberg. LibJPEG (jpeg V6b 27- Mar-1998),

Copyright (C) 1991-1998 Independent JPEG Group. ZLib (zlib 1.2.2),

Copyright (C) 1995-2003 Jean-loup Gailly and Mark Adler. Little CMS, Copyright (C) 1998-2004 Marti Maria. Kakadu,

Copyright (C) 2001, David Taubman, The University of New South Wales (UNSW). PNG, Copyright (C)

1998-2009 Glenn Randers-Pehrson. LibTIFF, Copyright (C) 1988-1997 Sam Leffler and Copyright (C)

1991-1997 Silicon Graphics, Inc.

在所有副本中的版权提示允许的情况下，您可以复制、使用、修改、出售和散布该软件。此软件不含任何隐藏条款，对于使用本软件所产生的任何后果，均由用户本人承担。

# 目录

概述 .....	5
教程 .....	6
1) 打开一个 PDF 文件 .....	6
2) 跳转到指定页面 .....	6
3) 缩放页面 .....	6
4) 旋转页面 .....	6
5) 打印 PDF 文档 .....	6
6) 隐藏或者显示界面组件 .....	6
7) 迭代整个大纲树 .....	7
8) 在 PDF 文档中查找文本 .....	7
9) * 注释 .....	7
10) #表单应用 .....	7
FoxitReaderSDK 控件 .....	8
属性 .....	8
FilePath .....	8
Password .....	8
PageCount .....	8
CurPage .....	8
Rotate .....	8
Zoomlevel .....	9
CurrentTool .....	9
Printer .....	10
*bHasFormFields .....	10
DocumentInfo .....	11
*bHighlightFormFields .....	11
*FormFieldsHighlightAlpha .....	11
*FormFieldsHighlightColor .....	11
ActiveXVersion .....	12
方法 .....	13
1) 打开和关闭 PDF 文件 .....	13
2) 视图 .....	15
3) 导航 .....	19
4) 查找 .....	23
5) 大纲 .....	25
6) 保存 .....	26
7) *注释 .....	26
8) 表单 .....	27
9) 绘画 .....	29
10) *运行 Javascript .....	30
11) *超级链接 .....	30

---

12) 其他 .....	32
事件 .....	40
#PDFForm .....	44
方法 .....	44
PDFFormField .....	47
属性 .....	47
方法 .....	52
PDFPrinter .....	56
属性 .....	56
方法 .....	58
PDFOutline .....	60
方法 .....	60
PDFDocumentInfo .....	61
属性 .....	61
FindResult .....	63
方法 .....	63

## 概述

**Foxit Reader SDK ActiveX** 是一款可视程序组件，可以让您的应用程序实现高质高效的 PDF 文档显示效果，其体积小巧，系统资源消耗少，可迅速集成于其它应用程序中。

Foxit Reader SDK ActiveX 与福昕阅读器（Foxit Reader）使用相同的解析和显示引擎。所以，只要福昕阅读器能够打开的 PDF 文件， ActiveX 3.0 就能够正常的显示，并且显示质量和显示速度与福昕阅读器完全一致，毫无差别。

相对于 Foxit Reader SDK (DLL 版本) 而言，ActiveX 版的使用更加方便，并且它还内嵌了丰富的界面和编程特性。程序设计人员可通过简单的拖放组件，迅速为应用程序添加 PDF 显示功能。该 ActiveX 还允许用户对 PDF 文档进行各种操作，如页面导航、改变缩放比例、页面旋转、页面滚动及打印，等等。

3.0 版本拥有许多高级的 PDF 功能。3.0 版本支持注释功能，并且允许用户填写、导入和导出 PDF 表单数据。3.0 版本拥有更多的功能和事件，让程序员能够灵活地控制组件，可以对 PDF 进行更多的操作。

ActiveX 3.0 有两个版本：标准版和专业版。这两个版本提供不同的 GUID 数字，您可以在同一台电脑上注册两个版本，而且还可以分别获取这两个 GUID。相对于专业版本，标准版本不具有创建或者编辑注释、导入或者到处表单数据、运行 JavaScript、转换 PDF 为文本等功能。您可以根据您的应用程序来选择 ActiveX 的版本。您可以在线购买标准版本和专业版本。如果您想要 3.0 版本中的表单模块，请联系 [sales@foxitsoftware.com](mailto:sales@foxitsoftware.com) 了解授权信息。在本开发手册中，所有带\*号的属性和功能只适用于专业版本。

Foxit Reader SDK ActiveX 可以在 Windows 95/NT 或者 Windows 95/NT 之后的环境中运行。Foxit Reader SDK ActiveX 是一个单机的组件，不需要其他的 PDF 软件来安装。请注意，在 Windows 系统下，单个用户需要在管理员的允许下才能成功注册 ActiveX。

我们有不同语言版本的完整的演示程序，包括 Visual Basic、 Visual C++、 Delphi，通过这些程序您可以了解到操作属性和方法的具体操作。请到 [www.foxitsoftware.com](http://www.foxitsoftware.com) 下载演示程序。

在专业版本的基础上，Foxit Reader SDK ActiveX 3.0 添加了几个接口来支持 PDF 表单域，允许用户在 PDF 文件中添加、删除、导入或者导出表单数据。表单模块需要单独的授权。如果您需要表单模块，请联系 [sales@foxitsoftware.com](mailto:sales@foxitsoftware.com) 。

**UNLOCK Code:** 如果您已经成功购买 Foxit Reader SDK ActiveX 并且已经成功获得 ActiveX 的函数和解码信息，那么在程序里调用 ActiveX 其他函数之前，您必须先调用 [UnLockActiveX](#) 或者 [UnLockActiveXEx](#)。这个函数在参考目录里有介绍。如果您只是想试用 ActiveX，您则无需调用这个函数。

# 教程

Foxit Reader SDK ActiveX 带有一个 OCX 文件。请用“regsvr32 foxitreader\_ax.ocx”命令来安装 Foxit Reader SDK ActiveX。如果当前的目录没有“foxitreader\_ax.ocx”，您可能需要指定一个恰当的路径来存储。

ActiveX 控件不仅帮助您操作接口，而且也提供了许多属性和方法来控制 ActiveX。

以下是一个例子，假设我们已经创建了一个叫做 FoxitReaderSDK 的 ActiveX 控件。

## 1) 打开一个 PDF 文件

我们打开一个名为“testdoc.pdf”的 PDF 文件。

```
//注意：如果您正在试用 ActiveX，您则无需解锁它，  
//PDF 的所有页面都将显示水印标记  
//如果您是购买用户，请先解锁 ActiveX。
```

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

```

## 2) 跳转到指定页面

我们将跳转到文档的第三页。

```
FoxitReaderSDK.GoToPage(2). //第一页的索引是 0
```

## 3) 缩放页面

如果您想显示 PDF 原始大小页面，您可以在 VC 上使用以下代码：

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

或

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

如果您想把缩放值定为 200%，请用以下代码：

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

欲了解更多信息，请查看本手册函数简介。

## 4) 旋转页面

PDF 页面可以在四个方向显示：垂直、90 度的旋转、180 度的旋转和 270 度的旋转。您只需调用“SetRotate()”就可以让它在不同方向显示。

如果您想顺时针 90 度旋转页面，您可以使用 VC 的以下代码：

```
FoxitReaderSDK. SetRotate(1);
```

## 5) 打印 PDF 文档

调用“PrintWithDialog”，然后 PDF 文档将弹出一个打印对话框，您设置完参数即可打印 PDF 文档。如果您不想打印时弹出对话框，您需使用 PDFPrinter 接口。

## 6) 隐藏或者显示界面组件

您可以调用“ShowToolBar”来隐藏或者显示工具栏。同样地，如果您想显示或者隐藏书签面板，您需调用“ShowBookmark”；如果您想显示或者隐藏状态栏，您需调用

“ShowStatusBar”。如果您想创建您自己的工具栏，您需隐藏内置 ActiveX 工具栏，然后在 ActiveX 外创建您自己的工具栏。

## 7) 迭代整个大纲树

您能通过调用“GetOutlineFirstChild”和“GetOutlineNextSibling”来迭代整个大纲树，允许您从 **PDFOutline** 接口查看大纲信息。

## 8) 在 PDF 文档中查找文本

您可调用“FindFirst”在整个文档中查找第一例指定的文本。如果找不到，那么则返回为 0。如果有找到，则返回到一个不为 0 的值，并且高亮它。再次选择“FindNext”，查找下一个。

如果您想在不打开和显示文本的情况下在 PDF 文件里查找文本，您可以调用“FindFileFirst”或者“FindFileNext”。

## 9) \* 注释

终端用户可以用不同的标注工具在 PDF 文档中画线条、圆圈或者其他形状的图形。您的应用程序也可通过编程改变“CurrentTool”的属性，例如设置成“Line Tool”。

## 10) #表单应用

在 3.0 版本中，Foxit ActiveX 给 PDF 表单提供了一个接口。您可以通过调用“GetCurrentForm”来获得当前 PDF 文档的 **IForm** 接口。接着，您可以调用 **IPDFForm** 和 **IPDFFormField** 接口属性和方法执行 PDF 表单的操作。给 PDF 页面添加表单按钮，代码如下：

```
CPDFformField button1 =  
form1.AddField("button1","button",m_nCurPage,0,0,55,30);  
button1.SetButtonCaption("N","Normal");  
button1.SetButtonCaption("R","Rollover");  
button1.SetButtonCaption("D","Down");  
button1.SetBehavior("push"); // push;Invert; None; Outline;  
button1.SetToolTip("reset all form");  
button1.SetTextFont("Courier");  
button1.SetTextSize(15);  
button1.SetJavaScriptAction("down", "app.alert(\"Mouse Down!\")");
```

## FoxitReaderSDK 控件

本节讲述了 ActiveX 公开的所有属性和方法。请注意本节的参考事例都是 C 语法的。如果您的编程语言不是 C/C++, 您必须遵循其他语言的语法

注意：标有\*的功能只适用于专业版本。

## 属性

### FilePath

类型:

BSTR, 只读

描述:

当前打开 PDF 文件的完整路径。如果没有打开 PDF 文件，那么这个属性值就是一个空的字符串。您可以通过 GetFilePath 来获得 FilePath 的值（这个函数只能在 VC 中使用）。

### Password

类型:

BSTR, 只读

描述:

PDF 密码。调用 GetPassword 您可获得这份文档的密码（这个函数只能在 VC 中使用）。

### PageCount

类型:

long, 只读

描述:

当前打开的 PDF 文件总页数。调用 GetPageCount 方法您可获得当前这份文档的总页数（这个函数只能在 VC 中使用）。

### CurPage

类型:

long, 只读

描述:

PDF 文件的当前页面的索引。页面索引从第一页开始，索引值为 0。通过调用 GetCurPage 可以获得当前页的索引值（这个函数只能在 VC 中使用）。

### Rotate

类型

**short**, 可读可写

描述:

当前旋转方向, 值可以为以下任意一种:

0 (正常)

1 (顺时针旋转 90 度)

2 (旋转 180 度)

3 (逆时针旋转 90 度)

您可通过调用 **SetRotate** 或者 **GetRotate** 来设定和获得旋转值 (这些函数只能在 VC 中使用)。

## **Zoomlevel**

类型:

**long**, 可读可写

描述:

通常情况下, 缩放的值在 10 到 1600 之间。您也可以使用以下特殊的值:

0=显示页面为实际页面大小, 这个值相当于缩放尺度的 100%。

1=显示页面为适当的缩放尺度, 让整个页面适合于客户的窗口。

2=显示文档为适当的缩放尺度, 让页面的宽度适合于客户的窗口。

调用 **SetZoomlevel** 或者 **GetZoomlevel** 您可以设置或者获得 **Zoomlevel** 的值 (这些函数只能在 VC 中使用)。

## **CurrentTool**

类型:

**BSTR**, 可读可写

描述

读取和设置当前工具。值可以是以下任何一个:

“Hand Tool”

“ZoomOut Tool”

“ZoomIn Tool”

“Select Text Tool”

“Find Text Tool”

“Snapshot Tool”

\*“Typewriter”

\*“Loupe Tool”

\*“Magnifier”

\*“Annot Tool”

\*“Rectangle Link Tool”

\*“Quadrilateral Link Tool”

\*“Arrow Tool”

\*“Line Tool”

\*“Dimension Tool”

\*“Square Tool”

- \*“Rectangle Tool”
- \*“Circle Tool”
- \*“Ellipse Tool”
- \*“Polygon Tool”
- \*“Cloudy Tool”
- \*“Polyline Tool”
- \*“Pencil Tool”
- \*“Rubber Tool”
- \*“Highlight Tool”
- \*“Underline Tool”
- \*“Strikeout Tool”
- \*“Squiggly Tool”
- \*“Replace Tool”
- \*“Note Tool”
- \*“Push Button Tool”
- \*“Check Box Tool”
- \*“Radio Button Tool”
- \*“Combo Box Tool”
- \*“List Box Tool”
- \*“Text Field Tool”
- \*“Distance Tool”
- \*“Perimeter Tool”
- \*“Area Tool”
- \*“Image Tool”
- \*“FileAttachment Tool”
- \*“Attach a file”
- \*“CallOut”
- \*“Sound Tool”
- \*“Movie Tool”

等等。

通过调用 `CountTools`，您可以知道在当前 ActiveX 版本中有多少工具，然后通过调用 `GetToolByIndex`，您可以获得这些工具的名称。您还可以通过调用 `GetCurrentTool` 或者 `SetCurrentTool` 来获取或者设定当前工具的名称（这些函数只能在 VC 中使用）。

## Printer

类型:

`IPDFPrinter`,只读

描述:

打印属性返回 `IPDFPrinter` 接口，您可以用来管理打印机和发送打印输出。在您的应用程序里，您可以调用 `GetPrinter` 来获得这个属性（这个函数只能在 VC 中使用）

## \*bHasFormFields

类型:

BOOL,只读

描述:

如果当前的文档里有表单域, `bHasFormFields` 则为 `True`, 反之, 则为 `False`。您可以在您的应用程序里调用 `GetBHasFormFields` 来获得 `bHasFormFields` 的值(这个函数只能在 VC 中使用)。

## DocumentInfo

类型:

`IPDFDocumentInfo*`,只读

描述:

`GetDocumentInfo` 返回 `PDFDocumentInfo` 接口, 您可以用这个接口检索文档里的作者、创建者、创建日期、关键字、修改日期、制作主题和标题 (这个函数只能在 VC 中使用)。

## \*bHighlightFormFields

类型:

BOOL,可读可写

描述:

设置 `bHighlightFormFields` 为 `True`, 将高亮所有互动表单域, 更有视觉效果。通过调用 `SetBHighlightFormFields` 或者 `GetBHighlightFormFields` 您可以设置或者获得是否要高亮所有交互式表单的值 (这个函数只能在 VC 中使用)。

## \*FormFieldsHighlightAlpha

类型:

`short`,可读可写

描述:

代表 256 级的表单高亮颜色的透明度

0=透明; 255=不透明

您可以通过 `SetFormFieldsHighlightAlpha` 或者 `GetFormFieldsHighlightAlpha` 来设置或者取得表单域高亮的 alpha 值 (这些函数只能在 VC 中使用)。

## \*FormFieldsHighlightColor

类型:

`OLE_COLOR`,可读可写

描述:

代表表单域的高亮颜色。您可以调用 `SetFormFieldsHighlightColor` 或者 `GetFormFieldsHighlightColor` 功能来设置或者获取表单域的高亮颜色 (这些函数只能在 VC 中使用)。

## ActiveXVersion

类型:

BSTR,只读

描述:

获取 ActiveX 控件版本信息。调用 `GetActiveXVersion` 函数您可获取当前注册的 ActiveX 的版本信息（这个函数只能在 VC 中使用）。

注意: 在 VC 中, 您需用以上方法来设置或者获取相关的属性; 在其他语言中, 您可以直接获得这些属性。例如:

VC 编码

```
FoxitReaderSDK.GetFilePath();
```

```
// the above statement returns the file path of current open PDF file. But you can directly
//以上语句返回当前打开 PDF 文件的路径, 但是在其他语言中您可以直接
//使用 FilePath 来获取当前打开的 PDF 文件的路径
```

VB 编码, javascript, C# etc:

```
FoxitReaderSDK.FilePath
```

## 方法

### 1) 打开和关闭 PDF 文件

#### OpenFile

从本地磁盘或者 http 服务器打开一个 PDF 文件。

原型:

BOOL OpenFile (BSTR FilePath, BSTR Password)

参数:

FilePath - PDF 文件路径或者 HTTP 服务器的 URL。

Password - PDF 文件密码。如果没有密码, 请指定一个空的字符串。

返回值:

如果 PDF 文件成功打开则值不为 0, 反之则为 0。

标注:

如果用此方法打开文件, 文件将不会被锁上。可以被其他程序打开文件。

#### OpenMemFile

从存储器中打开一个 PDF 文件。

原型:

BOOL OpenMemFile(long pBuffer, long Size, BSTR Password)

参数:

pBuffer - 调用者提供的一个指向 PDF 数据内容的指针。

Size - 指向 pBuffer 的缓冲区大小。

Password - PDF 密码。如果无密码, 请指定一个空的字符串。

返回值:

如果 PDF 文件成功打开则值不为 0, 反之则为 0。

#### OpenBuffer

从缓冲器上打开一个 PDF 文件。

原型:

BOOL OpenBuffer(VARIANT Buffer, long size, BSTR password);

参数:

Buffer - 包含 PDF 的字节数组

Size - 字节数组的大小

Password - 打开 PDF 文档的密码

返回值:

如果 PDF 文件成功打开则值不为 0, 反之则为 0。

#### OpenStream

从 IStream 接口打开一个 PDF 文件。

原型:

BOOL OpenStream (IStream\* Stream, BSTR Password)

参数:

Stream - 一个 IStream 接口。

Password - PDF 密码。如果无密码, 请指定一个空的字符串。

返回值:

如果 PDF 文件成功打开则值不为 0, 反之则为 0。

## OpenCustomFile

从一个自定义访问说明符打开一个 PDF 文件。当您的程序调用此方法时, ActiveX 将触发 CustomFileGetSize 和 CustomFileGetBlock 事件。在事件处理器中, 您的程序将以自定义格式打开 PDF 文档; 返回文件大小并锁定数据。欲了解更多信息, 请查看有关 CustomFileGetSize 和 CustomFileGetBlock 的描述。

原型:

BOOL OpenCustomFile(BSTR Password)

参数:

Password - PDF 密码。如果无密码, 请指定一个空的字符串。

返回值:

如果 PDF 文件成功打开则值不为 0, 反之则为 0。

## OpenFtpFile

从 FTP 服务器打开一个 PDF 文件。

原型:

BOOL OpenFtpFile(BSTR ftpName, BSTR username, BSTR userPassword,  
long port, BSTR filePath, BSTR filePassword, boolean Passive);

参数:

ftpName - FTP 服务器名称

userName - FTP 用户名

userPassword - FTP 密码

port - FTP 服务器端口

filePath - FTP 文件路径

filePassword - PDF 文件密码

Passive - 被动的或者主动的连接

返回值:

如果 PDF 文件成功打开则为(1), 反之则为(0)。

## \*UploadCurFileToFTP

上传当前 PDF 文件到一个 FTP 服务器。

原型:

```
BOOL UploadCurFileToFTP(BSTR ftpName, BSTR userName, BSTR userPassword,
                        long port, BSTR FilePath);
```

参数:

ftpName	-	FTP 服务器名称
userName	-	FTP 用户名
userPassword	-	FTP 密码
port	-	FTP 服务器端口
filePath	-	FTP 文件路径

返回值:

如果 PDF 文件成功打开则为(1), 反之则为(0)。

## CloseFile

关闭当前载入的 PDF 文件。

原型:

```
Void CloseFile()
```

参数:

[无]

返回值:

[无]

## SetFileStreamOption

设置打开文件时的文件数据流选项。

原型:

```
Void SetFileStreamOption(BOOL bFileStream);
```

参数:

bFileStream - BOOL 值指示文件数据流选项。

返回值:

[无]

标注

如果经常使用此文件, 您可以把数据流上的内容装入存储器里以此提高性能, 但是这样会占用更多的存储器容量。

## 2) 视图

### ShowTitleBar

显示或者隐藏标题栏。

原型:

```
void ShowTitleBar(BOOL bShow);
```

参数:

bShow - 如果参数值为 FALSE, 则隐藏标题栏。  
如果参数值为 TRUE, 则显示标题栏。

返回值:

[无]

## ShowToolBar

显示或者隐藏工具栏;

原型:

void ShowToolBar(BOOL bShow);

参数:

bShow - 如果参数值为 FALSE, 则隐藏工具栏。  
如果参数值为 TRUE, 则显示工具栏。

返回值:

[无]

## ShowToolBarButton

显示或者隐藏工具栏按钮

原型:

void ShowToolBarButton (short nIndex, BOOL bShow)

参数:

nIndex - 按钮的索引  
bShow - 如果参数值为 FALSE, 则隐藏工具栏按钮  
如果参数值为 TRUE, 则显示工具栏按钮

返回值:

[无]

## ShowBookmark

显示或者隐藏书签（大纲）面板。在这个文档中，书签和大纲属于同一个概念，它们可以交换使用。

注意: 在下个版本中这个函数将被删除。

原型:

void ShowBookmark(BOOL bShow)

参数:

bShow - 如果参数值为 FALSE, 则隐藏书签（大纲）面板  
如果参数值为 TRUE, 则显示书签（大纲）面板

返回值:

[无]

## ShowStatusBar

显示或者隐藏状态栏

原型:

void ShowStatusBar(BOOL bShow);

参数:

bShow - 如果参数值为 FALSE, 则隐藏状态栏如果参数值为 TRUE, 则显示状态栏

返回值:

[无]

## ShowNavPanelByString

通过名称来显示导航面板

原型:

BOOL ShowNavPanelByString(LPCTSTR lpszPanelName)

参数:

lpszPanelName - 面板名称包括Bookmark panel, Pages panel, Layer panel, and Attachments panel。

返回值:

如果成功则返回为 True, 反之则 False。

## \*ShowFormFieldsMessageBar

显示或者隐藏 FormFieldsMessageBar

原型:

void ShowFormFieldsMessageBar(BOOL bShow)

参数:

bShow - 如果参数值为 FALSE, 则隐藏 FormFieldsMessageBar  
如果参数值为 TRUE, 则显示 FormFieldsMessageBar

返回值:

[无]

## SetLayoutShowMode

设置页面布局。一份 PDF 文档可以显示成 n 纵列乘以 m 行。无论页面数是多少, 当页面布局被设为 MODE\_SINGLE 时, ActiveX 窗口一次只显示一行。当页面布局设置成 MODE\_CONTINUOUS 时, ActiveX 窗口将可同时显示相邻的几行。

原型:

Void SetLayoutShowMode (BrowseMode nShowMode, short nFacingCount);

参数:

nShowMode - 值可以设成以下两种:

MODE\_SINGLE =0.

MODE\_CONTINUOUS =1.

nFacingCount - 纵列数。

返回值:

[无]

## **SetFacingCoverLeft**

在浏览器模式里，这个函数会将 PDF 文档的封面页显示在左侧。

原型:

Void SetFacingCoverLeft (BOOL bLeft)

参数:

bLeft - BOOL 值表明是否将 PDF 文档封面页显示在左侧

返回值:

[无]

## **ShowNavigationPanels**

切换导航面板的显示

原型:

BOOL ShowNavigationPanels(BOOL bShow)

参数:

bShow - 如果参数为 TRUE 则显示导航面板，如果参数为 FALSE 则隐藏导航面板。

返回值:

如果成功则返回为 True，反之则 False。

## **EnableToolTip**

开关工具提示的显示。

原型:

BOOL EnableToolTip (BOOL bEnable)

参数:

bEnable - 如果参数为 TRUE 则显示工具提示，如果参数为 FALSE 则隐藏工具提示。

返回值:

如果成功则返回为 True，反之为 False。

## **GetLayoutShowMode**

获得当前布局模式。

原型:

Void GetLayoutShowMode(short\* pnShowMode, short\* pnFacingCount)

参数:

pnShowMode - 连续页显示的开关  
pnFacingCount - 封面页显示的开关  
返回值：  
[无]

### 3) 导航

#### **ExistForwardStack**

检测下个视图的存在。  
原型：  
BOOL ExistForwardStack ();  
参数：  
[无]  
返回值：  
如果下个视图存在则返回为 `true`，反之则返回为 `false`。  
批注：  
经常，当用户在一个 PDF 文件里浏览时想回到之前的阅读点。视图的定义是指某些阅读点或者显示状态。某些用户的行为将会创建新的视图。例如，如果用户到一个新页面，然后放大页面，这两个行为将创建 2 个新的视图。程序会调用这组方法以允许用户方便地在不同的视图里跳转。

#### **GoForwardStack**

跳转到下个视图。  
原型：  
Void GoForwardStack ();  
参数：  
[无]  
返回值：  
[无]

#### **ExistBackwardStack**

检测上一视图的存在。  
原型：  
BOOL ExistBackwardStack ();  
参数：  
[无]  
返回值：  
如果上一视角存在则为 `true`，反之则为 `false`。

#### **GoBackwardStack**

跳转到上一视图。

原型:

```
void GoBackStack();
```

参数:

[无]

返回值:

[无]

## SetViewRect

显示当前 PDF 页面中特定矩形区域的内容。

原型:

```
Void SetViewRect (float Left, float Top, float Width, float Height);
```

参数:

Left	-	左上角的水平坐标
Top	-	左上角的垂直坐标
Width	-	矩形的宽度
Height	-	矩形的长度

返回值:

[无]

标注:

这个函数将显示当前 PDF 页面中特定矩形区域的内容。坐标是 PDF 坐标，而不是设备坐标。单位是 PDF 点。此函数将保持 ActiveX 窗口位置和大小不变，并且调整当前 PDF 页面的位置和缩放系数以此来在 ActiveX 窗口完全展示当前 PDF 页面的指定矩形区域。一种典型的应用程序就是：终端用户用鼠标点击并且拖动矩形区域，然后放开鼠标，那么程序将调用 ConvertClientCoordToPageCoord，转换鼠标坐标到 PDF 坐标，接着调用 SetViewRect 在全视图中显示指定的区域。

## ConvertClientCoordToPageCoord

转换 ActiveX 控件窗口的客户端坐标到 PDF 页面坐标。

原型:

```
BOOL ConvertClientCoordToPageCoord (long nClientX, long nClientY,
                                    long* pnPageIndex, float* pPageX, float*
                                    pPageY);
```

参数:

nClientX - 在 ActiveX 控件窗口的客户端坐标的 X 坐标，单位为像素

nClientY - 在 ActiveX 控件窗口的客户端坐标的 Y 坐标，单位为像素

pnPageIndex - 返回给定点在 PDF 文件中所属的页码

pPageX - 返回给定点 PDF 页面中的 X 坐标点（在 PDF 坐标系统）

pPageY - 返回给定点 PDF 页面中的 y 坐标点（在 PDF 坐标系统）

返回值:

返回值表明转换是否成功。客户端区域包含正在显示的 PDF 页面以及一些灰色的背景。

如果点落在灰色的背景，则转换失败。

### ConvertPageCoordToClientCoord

转换 PDF 页面坐标为 ActiveX 控件窗口的客户端区域的坐标。

原型:

```
BOOL ConvertPageCoordToClientCoord (long nIndex, float dPageX, float dPageY,
                                    long* pnClientX, long* pnClientY);
```

参数:

nIndex	-	页码
dPageX	-	PDF 页面里的 X 坐标 (在 PDF 坐标系统)
dPageY	-	PDF 页面里的 Y 坐标 (在 PDF 坐标系统)
pnClientX	-	返回 ActiveX 控件窗口的客户端区域里的 X 坐标。如果结果是否定的，则表明这个点落在 ActiveX 控件窗口的客户端区域外。
pnClientY	-	返回 ActiveX 控件窗口的客户端区域里的 Y 坐标。如果结果是否定的，则表明这个点落在 ActiveX 控件窗口的客户端区域外。

返回值:

返回值表明转换是否成功。如果文档没有正确的打开或者页码不正确，则返回值为 `false`。反之则为 `true`。

### GotoPageDest

跳转到 PDF 文档的指定区域。

原型:

```
Void GotoPageDest (ILink_Dest * link_dest);
```

参数:

link_dest	-	从 OnHyperLink 事件获得的一个 ILink_Dest 接口。
-----------	---	--------------------------------------

返回值:

[无]

### GoToPagePos

跳转到 PDF 文档的指定位置。

原型:

```
Void GoToPagePos (long nIndex, float PageX, float PageY);
```

参数:

nIndex	-	想查看页面的索引。
PageX	-	由 nIndex 指定的检索页面的指定 X 坐标 (在 PDF 坐标系统)
PageY	-	由 nIndex 指定的检索页面的指定 Y 坐标

返回值:

[无]

### GetVisibleLeftTopPage

获得左上角视图的页码。

原型:

```
Long GetVisibleLeftTopPage();
```

参数:

[无]

返回值:

左上角视图的页码

## ScrollView

通过 dx, dy 滚动当前视角, 单位为设备像素。

原型:

```
Void ScrollView (long dx, long dy);
```

参数:

dx - 滚动行为的水平距离。

dy - 滚动行为的垂直距离。

返回值:

[无]

## GetScrollLocation

在当前页面获得当前滚动的位置。

prototype:

```
long GetScrollLocation (long *dx, long *dy);
```

参数:

dx - 返回当前滚动位置的 X 坐标

dy - 返回当前滚动位置的 Y 坐标

返回值:

当前页面的索引值。

## GoToNextPage

跳转到当前打开的 PDF 文档的下一页。

原型:

```
Void GoToNextPage();
```

参数:

[无]

返回值:

[无]

注意:

在以后的版本中将删除此函数。您可以调用 GoToPage 函数。

## GoToPrevPage

跳转到当前打开的 PDF 文档的上一页。

原型:

Void GoToPrevPage();

参数:

[无]

返回值:

[无]

注意:

在以后的版本中将删除此函数。您可以调用 GoToPage 函数。

## 4) 查找

### FindFirst

在文档中查找一个字符串。如果此函数找到了这个字符串的第一个匹配项，它将跳到此页，更新 CurPage 属性，高亮该匹配项然后返回 true。否则则返回 false。

原型:

BOOL FindFirst (BSTR search\_string, BOOL bMatchCase, BOOL bMatchWholeWord)

参数:

SearchString - 您想查找的字符串

BMatchCase - 是否区分大小写

BMatchWholeWord - 是否全字匹配

返回值:

如果查找到这个字符串的匹配项，则返回值不为 0，否则则为 0

### FindFirstEx

FindFirst 函数的扩展。提供了一个接口来搜索文档中的字符串。

原型:

```
BOOL FindFirstEx(const VARIANT FAR& search_string, BOOL bMatchCase,
                  BOOL bMatchWholeWord)
```

参数:

SearchString - 您想查找的字符串

BMatchCase - 是否区分大小写

BMatchWholeWord - 是否全字匹配

返回值:

如果查找到这个字符串的匹配项，则返回值不为 0，否则则为 0

### FindNext

在整个文档中搜索已给字符串的下个匹配项。如果 ActiveX 找到了下个匹配项，它会跳转到页面，更新 CurPage 属性并且高亮此匹配项，然后返回 true。反之，则返回为 false。

请注意 `FindNext` 和 `FindFirst` 方法使用相同的搜索条件，包括 `bMatchCase` 和 `bMatchWholeWord`。如果 `bSearchDown` 是 `true`，那么将往下搜索。如果 `bSearchDown` 是 `false`，那么往上搜索。

原型:

```
BOOL FindNext (BOOL bSearchDown);
```

参数:

`bSearchDown` - 向下搜索 (`True`) 或者向上搜索 (`False`)。

返回值:

如果搜索到下一个匹配项，返回值不为 0。否则则为 0

## FindFileFirst

在文档中查找字符串，如果找到字符串，则返回为 `IFindResult` 接口。反之则返回空。此方法允许您在不打开文档的情况下搜索文档。例如，如果您想在一个文件夹的所有 PDF 文件里搜索一个关键字，您需要迭代那个文件夹里的所有 PDF 文件一个一个地寻找关键字。如果 ActiveX 在一个 PDF 文件中找到了一个匹配项，ActiveX 将返回包含了所有匹配项细节的 `IFindResult`。接下来，您就可以使用 `GoToSearchResult` 来打开这个文件，跳转到页面并且高亮该匹配项。

原型:

```
IFindResult* FindFileFirst(BSTR file_path, BSTR search_string,BOOL bMatchCase,  
BOOL bMatchWholeWord)
```

参数:

<code>file_path</code>	- PDF 文件路径
<code>SearchString</code>	- 您想查找的字符串
<code>BMatchCase</code>	- 是否区分大小写
<code>BMatchWholeWord</code>	- 是否全字匹配

返回值:

如果搜索到匹配项则返回 `IFindResult` 接口。否则返回空。

## FindFileNext

寻找在 `FindFileFirst` 函数中指定的字符串的下一个匹配项。

原型:

```
IFindResult * FindFileNext ();
```

参数:

[无]

返回值:

如果搜索到了下一个匹配项，则返回值为 `IFindResult`。否则返回空。

## GoToSearchResult

显示并且高亮搜索结果。

原型:

Void GoToSearchResult (IFindResult\* findresult);

参数:

Findresult - 由 FindFileFirst 或者 FindFileNext 返回的一个 IFindResult 接口。

返回值:

[无]

### **\*SearchAndHighlightAllTextOnPage**

在指定页面高亮特定的关键字的所有实例。

原型:

```
void SearchAndHighlightAllTextOnPage(BSTR searchstring, BOOL bMatchCase,
                                     BOOL bMatchWholeWord, long PageNo);
```

参数:

searchstring - 您想查找的字符串

BMatchCase - 是否区分大小写

BMatchWholeWord - 是否全字匹配

PageNo - 您想搜索的页码。

返回值:

[无]

## 5) 大纲

### **GetOutlineFirstChild**

获取当前大纲的第一子项。

原型:

```
IPDFOutline* GetOutlineFirstChild( IPDFOutline* Outline)
```

参数:

Outline - 返回第一个子项。

如果您想获得大纲树的根节点, 请把参数值设为空。

返回值:

如果指定的项拥有子项, 第一个子项将被返回。否则返回为空。

### **GetOutlineNextSibling**

获得下一个兄弟节点。

原型:

```
IPDFOutline* GetOutlineNextSibling ( IPDFOutline* Outline)
```

参数:

Outline - 大纲项的下一个兄弟节点将被返回。

返回值:

如果下一兄弟节点存在, 那么将成功返回。否则则返回空。

## 6) 保存

### SaveAs

保存当前加载的 PDF 文档到一个文件里。

原型:

Void SaveAs (BSTR FileName)

参数:

FileName - 指定即将被保存的文件的文件名。

返回值:

[无]

### Save

保存当前加载的 PDF 文档。

原型:

Void Save()

参数:

[无]

返回值:

[无]

### SaveToStream

保存当前加载的 PDF 文档到存储器。

原型:

IStream\* SaveToStream()

参数:

[无]

返回值:

IStream 接口支持读写数据到包含 PDF 文件数据的数据流对象中

## 7) \*注释

### \*ExportAnnotsToFDFFile

从当前文档导出批注到表单数据格式 (FDF) 文件

原型:

BOOL ExportAnnotsToFDFFile(BSTR FDFFFileName)

参数:

FDFFFileName - PDF 文件路径。

返回值:

返回值表明操作是否成功。

### **\*ImportAnnotsFromFDFFile**

从一个表单数据格式(FDF)文件导入批注到当前的文档。

原型:

```
BOOL ImportAnnotsFromFDFFile(BSTR FDFFFileName)
```

参数:

FDFFFileName - FDF 文件路径

返回值:

返回值表明操作是否成功。

### **\*SetBDrawAnnot**

设置注释标志。

原型:

```
void SetBDrawAnnot(BOOL bDrawAnnot);
```

参数:

bDrawAnnot - 如果参数是 true, 则可画注释。  
如果参数是 false, 则不可画注释。

返回值:

[无]

### **\*ShowAllPopup**

弹出所有注释。

原型:

```
Void ShowAllPopup (BOOL bShow);
```

参数:

bShow - 如果参数是 true, 则将弹出注释。

返回值:

[无]

## **8) 表单**

### **\*ExportFormToFDFFile**

导出 PDF 表单数据到表单数据格式(FDF)文件。

原型:

```
BOOL ExportFormToFDFFile (BSTR FDFFFileName)
```

参数:

FDFFFileName - FDF 文件路径。

返回值:

返回值表明操作是否成功。

## \*ImportFormFromFDFFile

从表单数据格式 (FDF) 文件导入数据到 PDF 表单。

原型:

```
BOOL ImportFormFromFDFFile(BSTR FDFFFileName)
```

参数:

FDFFFileName - FDF 文件路径。

返回值:

返回值表明操作是否成功。

## \*FindFormFieldsTextFirst

在表单域查找文本。

原型:

```
BOOL FindFormFieldsTextFirst (BSTR searchstring, BOOL bMatchCase);
```

参数:

Searchstring - 您想查找的字符串。

bMatchCase - 是否区分大小写。

返回值:

返回值表明想要搜索的字符串是否被找到。

## \*FindFormFieldsTextNext

在表单域里搜索文本。

原型:

```
BOOL FindFormFieldsTextNext ()
```

参数:

[无]

返回值:

返回值表明想要搜索的字符串是否被找到。

## #GetCurrentForm

获得表单接口的指针。添加新的 PDF 表单元素需要先调用这个函数。

原型:

```
IPDFForm* GetCurrentForm()
```

参数:

[无]

返回值:

如果成功则返回表单接口指针。否则返回空。

## 9) 绘画

### \*AddWaterMark

插入显示为水印的文本到文档。

原型:

```
BOOL AddWaterMark (short page, BSTR string, float left, float bottom,
                    short fontsize, OLE_COLOR fontcolor,
                    short textmode, short alpha, short rotate);
```

参数:

Page	- 即将被添加水印的文档页码。
String	- 当做水印显示的文本。
Left	- 水印的水平 X 位置。
Bottom	- 水印的水平 Y 位置。
Fontsize	- 文本大小
Fontcolor	- 文本颜色
Textmode	<p>- 值必须是 0, 1, 2</p> <p>0 表示填充文本, 表示描边文本</p> <p>2 表示填充文本后描边文本</p>
alpha	- 从 0 到 255 的一个数, 确定 alpha 数值。
rotate	- 水印旋转的角度。

返回值:

返回值表明水印是否被添加。

### \*AddImageObject

插入图片到文档。

原型:

```
BOOL AddImageObject (long nPageIndex, float left, float bottom, float width,
                     float height, BSTR BmpFileName, short alpha, short rotate);
```

参数:

nPageIndex	- 添加图像的文档页码。
left	<p>- 图像的水平 X 位置</p> <p>在最左边的像素从 0 开始</p>
bottom	<p>- 图像的水平 Y 位置</p> <p>在最底部的扫描线从 0 开始</p>
width	- 位图的宽度
height	- 位图的高度
BmpFileName	- 图像的文件路径
alpha	- 从 0 到 255 的一个数字, 确定 alpha 的值
rotate	- 图像将被旋转的角度

返回值:

返回值表明是否成功添加图像。

## 10) \*运行 Javascript

### \*ShowDocJsDialog

弹出 Javascript 对话框。

原型:

```
void ShowDocJsDialog();
```

参数:

[无]

返回值:

[无]

### \*ShowJsConsoleDialog

弹出 Javascript 控制台对话框。

原型:

```
void ShowJsConsoleDialog();
```

参数:

[无]

返回值:

[无]

## 11) \*超级链接

### \*CountHyperLinks

计算出 PDF 页面中的超级链接数目

原型:

```
short CountHyperLinks(short nPageIndex);
```

参数:

nPageIndex - PDF 文件指定页的索引。

返回值:

如果成功则返回链接数。如果没有链接则返回 0。如果失败则返回-1

### \*HighlightHyperLink

高亮 PDF 页面里指定的超级链接。

原型:

```
void HighlightHyperLink(short nPageIndex, short nLinkIndex)
```

参数:

nPageIndex - 指定的页码

nLinkIndex - 超级链接的索引

返回值:

[无]

### **\*GetHyperLinkRect**

获得指定的超级链接的位置。

原型:

```
BOOL GetHyperLinkRect(short nPageIndex, short nIndex, float* top, float* left,
                      float* bottom, float* right)
```

参数:

nPageIndex	- 指定的页码
nLinkIndex	- 超级链接的索引
top	- 返回顶部坐标的指针
left	- 返回左边坐标的指针
bottom	- 返回底部坐标的指针
right	- 返回右边坐标的指针

返回值:

如果成功则返回 True, 否则返回 False。

### **\*GetHyperLinkInfo**

获得一个指定的超级链接的链接信息。

原型:

```
BOOL GetHyperLinkInfo(short nPageIndex, short nIndex, BSTR* linktype,
                      BSTR* linkdata, L PDISPATCH* linkdest)
```

参数:

nPageIndex	- 指定的页码
nLinkIndex	- 超级链接的索引
linktype	- 返回包含超级链接类型的指针
linkdata	- 返回包含超级链接字符串数据的指针
linkdest	- 返回包含该超级链接重定向目的地的指针

返回值:

如果成功则返回 True, 否则返回 False。

### **\*EnableHyperLink**

启用/禁用超级链接。

原型:

```
Void EnableHyperLink(BOOL bEnable)
```

参数:

bEnable - TRUE 为启用超级链接。FALSE 为禁用超级链接。

返回值:

[无]

## 12) 其他

### **GetSelectedText**

获得当前选定的文本。

原型:

BSTR GetSelectedText();

参数:

[无]

返回值:

当前选定的文本。

### **OpenFileForPrinter**

在不显示 PDF 文件的情况下打印 PDF 文件。

原型:

IPDFPrinter\* OpenFileForPrinter(BSTR file\_path)

参数:

file\_path - PDF 文件路径 (包括扩展名)

返回值:

返回用于控制打印机的 IPDFPrinter 接口

### **\*Highlight**

在该文档的指定页面高亮一个指定的矩形区域。

原型:

void Highlight(long nIndex, float left, float top, float right, float bottom)

参数:

nPageIndex - 将要被高亮的指定矩形区域的页码

left - 矩形区域左上角的 X 坐标

top - 矩形区域左上角的 Y 坐标

right - 矩形区域右下角的 X 坐标

bottom - 矩形区域右下角的 Y 坐标

返回值:

[无]

### **\*RemoveAllHighlight**

删除当前文档的所有高亮。

原型:

void RemoveAllHighlight()

参数:

[无]

返回值:

[无]

## GetPageText

在当前加载的 PDF 文件中的一个 PDF 页面提取文本内容。

原型:

BSTR GetPageText (long nPageIndex)

参数:

nPageIndex - 您想提取文档的页码。

返回值:

返回被提取的文本。

## \*GetPageTextW

从当前加载的 PDF 文件中的一个页面提取文本内容。

原型:

long GetPageTextW(long nPageIndex, long FAR\* pBuffer, long FAR\* nBuflen)

参数:

nPageIndex - 您要提取文本的页码

pBuffer - 放置页面内容的缓存

nBuflen - 缓存的长度

返回值:

如果成功则返回-1, 否则则返回 0

## CountTools

在当前 ActiveX 版本中获得可用的工具数目。

原型:

short CountTools()

参数:

[无]

返回值:

工具数目。

## GetToolByIndex

获得工具名称。

原型:

BSTR GetToolByIndex (short nIndex)

参数:

nIndex - nIndex 的范围是: 0 <= nIndex < CountTools().

返回值:

返回工具名。

## GetDocPermissions

获得文档文件的权限标志。

原型:

```
long GetDocPermissions ()
```

参数:

[无]

返回值:

32-bit 整数表明权限标志。详细描述您可以参照 PDF 参考 (PDF Reference)。如果该文档不受保护则返回为 0xffffffff。

## ShowDocumentInfoDialog

弹出文档属性对话框

原型:

```
void ShowDocumentInfoDialog()
```

参数:

[无]

返回值:

[无]

## SetModulePath

设置 fpdfcjk.bin 文件的路径。

原型:

```
void SetModulePath(LPCTSTR lpFolderName)
```

参数:

lpFolderName - fpdfcjk.bin 文件的路径。

返回值:

[无]

## SetCurrentLanguage

ActiveX 用户界面可以动态地转换成 30 多种语言中的一种，但是这需要额外的语言文件 (xml 格式) 来配合 ActiveX。如果您想要指定的语言文件，请联系 [sales@foxitsoftware.com](mailto:sales@foxitsoftware.com)。

原型:

```
void SetCurrentLanguage(short LanguageID);
```

参数:

LanguageID - 语言标识符

从 0 到 30 的值代表不同的语言

返回值:

[无]

## **SetCurrentLanguageByString**

ActiveX 用户界面可以动态地转换成 30 多种语言中的一种，但是这需要额外的语言文件（xml 格式）来配合 ActiveX。如果您想要指定的语言文件，请联系 [sales@foxitsoftware.com](mailto:sales@foxitsoftware.com)。

原型:

void SetCurrentLanguageByString(BSTR FileName);

参数:

FileName - 语言文件名。比如“lang\_en\_us.xml”。

返回值:

[无]

## **SetCurrentWnd**

当 ActiveX 在多实例中运行时用户调用这个函数来设置当前的实例。

原型:

void SetCurrentWnd(long hWnd);

参数:

hWnd - 一个 OCX 实例的 HWND。

返回值:

[无]

## **SetLogFile**

在您的应用程序中您可以调用这个函数来设置日志文件。您调用的每一个函数都将记录到该日志文件。

原型:

BOOL SetLogFile(BSTR filepath);

参数:

filepath - 该日志文件的路径。

返回值:

布尔值说明该日志文件是否被设置。

## **\*IsDualPage**

检查页面类型。

原型:

BOOL IsDualPage(short pageIndex);

参数:

pageIndex - 文档页码

返回值:

布尔值指定页面是否有两个图层。  
两个图层即一个页面拥有隐藏文本的图像。

## \*ExportPagesToPDF

导出当前文档的几个页面到一个 PDF 文件中。

原型:

BOOL ExportPagesToPDF (BSTR lpszPDFFFileName, BSTR lpszPageRangeString);

参数:

lpszPDFFFileName	-	插入页面的文件路径。
lpszPageRangeString	-	页面范围。比如"0, 2, 3-5", 请注意, "5-2"是无效的。 换句话说, 破折号前面的值不能大于破折号后面的值。

。

返回值:

布尔值确定页面是否被导出。

## \*GetBitmap

以位图的方式来获得 PDF 页面指定区域的显示内容。

原型

```
long GetBitmap(short nIndex, long pixelWidth, long pixelHeight, float rectLeft,
               float rectTop, float rectRight, float rectBottom, long
               PixelFormat);
```

参数:

nPageIndex	-	文档的页码。
pixelWidth	-	位图的宽度。
pixelHeight	-	位图的高度。
rectLeft	-	设备坐标显示区域的左边像素位置。
rectTop	-	设备坐标显示区域的上方像素位置。
rectRight	-	设备坐标显示区域的右边像素位置。
rectBottom	-	设备坐标显示区域的下方像素位置。
PixelFormat	-	位图的像素格式

返回值:

位图的句柄。

## GetPageHeight

获得页面高度

原型:

float GetPageHeight(short nIndex);

参数:

nPageIndex	-	文档的页码
------------	---	-------

返回值:

页面高度（不包括没有显示出来的区域），以点为单位。

1 点等于 1/72 英寸（大约 0.3528 毫米）

## GetPageWidth

获取页面宽度

原型:

`float GetPageWidth(short nIndex);`

参数:

`nPageIndex` - 文档的页码

返回值:

页面宽度（不包括没有显示出来的区域），以点为单位。

1 点等于 1/72 英寸（大约 0.3528 毫米）

## AboutBox

弹出 about 窗口

原型:

`void AboutBox();`

参数:

[无]

返回值:

[无]

## PrintWithDialog

显示发送打印输出时的 Windows 对话框。

原型:

`void PrintWithDialog();`

参数:

[无]

返回值:

[无]

## UnLockActiveX

利用从福昕公司获得的授权密钥来解锁 ActiveX。

原型:

`void UnLockActiveX(BSTR lisence_id, BSTR unlock_code)`

参数:

`license_id` - 用福昕授予的字符串辨别 SDK 授权。

`unlock_code` - 用福昕授予的字符串解锁 ActiveX

返回值:

[无]

批注:

对于试用试用版本 ActiveX 的用户, 您无需调用该函数, 但是所有页面都将保留水印。

对于付费用户, 在调用 ActiveX 其他函数时您必须调用该函数。

## **UnLockActiveXEx**

利用福昕公司授予的授权密钥来解锁 ActiveX。

原型:

```
void UnLockActiveXEx(BSTR strLicense)
```

参数:

strLicense - 用福昕授予的字符串识别 SDK 授权。

返回值:

[无]

批注:

这个函数的功能与 UnlockActiveX 相同。

## **\*SetUserPassword**

给当前 PDF 文件设置用户密码。

原型:

```
BOOL SetUserPassword(LPCTSTR lpsznewValue)
```

参数:

lpsznewValue - 密码字符串

返回值:

如果成功则返回 True, 否则返回 False。

## **\*SetUserPermission**

给当前 PDF 文件设置用户权限。

原型:

```
BOOL SetUserPermission(long dwPermission)
```

参数:

dwPermission - 返回的用户权限标志。

返回值:

如果成功则返回 True, 否则则为 False。

## **\*SetOwnerPassword**

给当前 PDF 文件设置所有者密码。

原型:

```
BOOL SetOwnerPassword(LPCTSTR lpsznewValue)
```

参数:

IpszNewValue - 密码字符串

返回值:

如果成功则返回 `True`, 否则则为 `False`。

## **SetContextMenuString**

给右击上下文相关菜单设置菜单项字符串

原型:

`Void SetContextMenuString(LPCTSTR string)`

参数:

`String` - 包含右击上下文菜单数据的字符串, 比如 “a,b,c,d,e”。与 `OnContextMenuIndex` 事件一起调用该函数。“a, b, c, d, e”代表不同的菜单项。

返回值:

[无]

## **SetPDFMeasureUnit**

设置 PDF 文档度量单位。

原型:

`BOOL SetPDFMeasureUnit(short nType);`

参数:

`nType` - 计量单位: 0 = 点; 1 = 英寸; 2 = 厘米; 3: 像素

返回值:

如果成功则返回 `True`, 否则则为 `False`。

## **GetCurrentWnd**

获得当前窗口的指针。

原型:

`Long GetCurrentWnd();`

参数:

[无]

返回值:

成功返回窗口点。否则返回空。

## **GetCtrlInstance**

获得控件实例句柄。

原型:

`Long GetCtrlInstance();`

参数:

[无]

返回值:

成功返回控件实例句柄。否则则为无效。

## 事件

### BeforeDraw

在描绘视图内容之前触发。

原型:

Void BeforeDraw (long dc)

参数:

dc - 设备上下文句柄。

### AfterDraw

在描绘视图内容完成后触发。

原型:

Void AfterDraw (long dc)

参数:

dc - 设备上下文句柄。

### OnZoomChange

改变 Zoomlevel 属性时触发。

原型:

Void OnZoomChange ()

参数:

[无]

### OnPageChange

改变页面时触发（从一个页面跳转到另一个页面）

原型:

Void OnPageChange ()

参数:

[无]

### OnOpenPassword

当您打开有密码保护的 PDF 文档时触发事件。

原型:

Void OnOpenPassword (BSTR\* password, BOOL\* cancel)

参数:

Password - PDF 文档密码

Cancel - 当这个参数设置 False 时会一直触发此事件，直到密码正确。

## \*OnHyperLink

当点击超文本时触发。

原型:

Void OnHyperLink(BSTR linktype, BSTR linkdata, Link\_Dest\* dest, BOOL\* cancel)

参数:

Linktype - 包含超级链接类型信息的字符串。

linktype sting are:

Goto, 在当前文档中跳转到不同的页面, linkdata 是空的字符串。dest 包含即将进行浏览的位置信息。

GoToR 跳转到本地磁盘一个不同的 PDF 文件中, 如果查看新的文档要求新窗口, 那么 linkdata 信息将包含文件名并且文件名字符串后 紧跟着数字 1, 否则, 文件名字符串后则紧跟着数字 0 。dest 包含即将进行浏览的位置信息。

Launch 启动一个外部应用程序, 如果查看一个新的文本需要新窗口, 那么 linkdata 信息将包含文件名并且文件名字符串后紧跟着数字 1, 否则, 文件名字符串后则紧跟着数字 0 。

URI 打开一个 uri, linkdata 包含 uri 字符串。

Cancel - 如果次参数设置为 true, 控件将不跟随超级链接。

linkData - 包含额外信息被字符隔开的字符串。

## OnSearchProgress

搜索文档时触发。

原型:

Void OnSearchProgress (long pageNumber, long pageCount)

参数:

pageNumber - 正在进行搜索的页面

pageCount - 总页数

## OnOpenFile

当文件打开失败时触发事件

原型:

void OnOpenFile(short Error);

参数:

Error - 返回错误代码

## OnFilePathInvalidate

文件操作验证失败时触发事件

原型:

void OnFilePathInvalidate(BSTR WarnString);

参数:

WarnString - 返回错误消息

## OnShowSavePrompt

关闭修改过的文档时触发事件。

原型:

void OnShowSavePrompt(BOOL\* bShow, short \* nResult);

参数:

bShow - 这个值指定是否在 ActiveX 中显示默认的消息框。  
nResult - 这个值指定是否保存这个修改的文档。

## OnOpenDocument

打开一个 PDF 文档时触发。

原型:

Void OnOpenDocument (BSTR filepath)

参数:

Filepath - PDF 文件路径。

## OnCloseDocument

关闭一个文档时触发。

原型:

Void OnCloseDocument (BSTR filepath)

参数:

Filepath - PDF 文件路径

## OnDocumentChange

PDF 文档内容改变时触发。

原型:

Void OnDocumentChange ()

参数:

[无]

## CustomFileGetSize

当使用 OpenCustomFile 方法打开 PDF 文档时触发。

原型:

Void CustomFileGetSize (long\* size)

参数:

size - [out] 用于接受 PDF 长度的数字指针。

将其设置成 PDF 文件长度。

## CustomFileGetBlock

当使用 `OpenCustomFile` 方法打开 PDF 文档时触发。

原型:

`Void CustomFileGetBlock (long pos, long pBuf, long size)`

参数:

`pos` - [in] 文件开头的偏移字节。  
`pBuf` - [out] 用于接受 PDF 数据的缓存指针。  
`Size` - [in] 缓存大小

批注:

从特定的位置获取数据块。位置是指从文件开始的偏移字节。位置和大小不会超过文件长度的范围。

## OnContextMenuIndex

当点击右键菜单中的一个项目时触发事件。使用此事件时, 请与 `SetContextMenuString` 接口一起使用。

原型:

`void OnContextMenuIndex(short nIndex);`

参数:

`nIndex` - 所选菜单项的索引。

## OnClick

在点击时触发。

原型:

`Void OnClick (long hWnd, long ClientX, long ClientY);`

参数:

`hWnd` - 这个窗口的句柄。  
`ClientX` - ActiveX 控件窗口的客户端区域的 X 坐标。  
`ClientY` - ActiveX 控件窗口的客户端区域的 Y 坐标。

## OnDbClick

当双击时触发。

原型:

`Void OnDbClick (long hWnd, long ClientX, long ClientY);`

参数:

`hWnd` - 这个窗口的句柄。  
`ClientX` - ActiveX 控件窗口的客户端区域的 X 坐标。  
`ClientY` - ActiveX 控件窗口的客户端区域的 Y 坐标。

## OnRButtonClick

当右击时触发事件

原型:

```
void OnRButtonClick(long hWnd, long ClientX, long Client);
```

参数:

hWnd	-	这个窗口的句柄。
ClientX	-	ActiveX 控件窗口的客户端区域的 X 坐标。
ClientY	-	ActiveX 控件窗口的客户端区域的 Y 坐标。

## OnDownLoadFinish

从 Internet 成功下载文档的时候触发事件。

原型:

```
void OnDownLoadFinish();
```

参数:

[无]

## #FormFieldError

配置 PDF 表单域出现错误时触发事件。

原型:

```
Void FormFieldError(long nErrorCode);
```

参数:

nErrorCode - 配置 PDF 表单域时返回错误代码

## #PDFForm

### 方法

#### AddField

添加一个新的表单区域

原型原型:

```
LPDISPATCH AddField(LPCTSTR bstrFieldName, LPCTSTR bstrFieldType,
                     long pageIndex, float left, float top, float right, float bottom);
```

参数:

bstrFieldName	-	区域名称。
bstrFieldType	-	新建区域的字段原型。

有效原型包括:

- text
- buttoncombobox
- listbox

checkboxradio button pageIndex - 页码 (0 为起始数)。

left - 矩形区域的左坐标。

top - 矩形区域的上坐标。

right - 矩形区域的右坐标。

bottom - 矩形区域的底坐标。

返回值:

新建的区域对象。

标注:

矩形区域坐标是在可旋转的页面空间中测量, 即不管是否为旋转页面, [0,0]总是位于左下角。

## RemoveField

删除一特定表单区域

原型:

Void RemoveField (LPCTSTR bstrFieldName);

参数:

bstrFieldName - 删除的区域名称。如果字段含有多重子标注, 则所有子标注将被删除; 如果多区域拥有相同的名称, 则所有该名称的区域将被删除。

返回值:

[无]

## ExportToPDF

将特定表单元素导出至 FDF 文件。

原型:

Void ExportToPDF (LPCTSTR bstrFullPath, BOOL bEmptyFields,  
const VARIANT FAR& arrFields)

参数:

bstrFullPath - 导出的 FDF 文件的完整路径

bEmptyFields - True 表示导出在 VARIANT& arrField 中指定的 ONLY 元素;  
False 表示导出所有元素, 除了在 VARIANT& arrField 指定的元素。

ArrFields - 导出表单元素数组

返回值:

[无]

## ImportFromPDF

从 FDF 文件中导入

原型:

void ImportFormFromPDF(LPCTSTR bstrFullPath);

参数:

bstrFullPath - FDF 文件全路径

返回值:

[无]

### **GetFieldByIndex**

获取指定表单区域的指针

原型:

LPDISPATCH GetFieldByIndex(long index)

参数:

index - 表单区域的索引

返回值:

若成功, 返回指针至表单区域, 否则返回 Null。

### **GetFieldByName**

获取指定表单区域的指针

原型:

LPDISPATCHs GetFieldByName(LPCTSTR bstrFieldName)

参数:

bstrFieldName - 表单区域的全名

返回值:

若成功, 返回表单区域的指针, 否则返回 Null。

### **GetFieldsCount**

获取所有表单区域的数量

原型:

long GetFieldsCount()

参数:

[无]

返回值:

若成功, 则返回表单区域数量, 否则返回-1。

## PDFFormField

### 属性

#### Alignment

类型:

String

描述:

文本区域内的文本对齐（左对齐、居中、右对齐）

标注:

仅对文本区域

#### BorderStyle

类型:

String

描述:

表单区域边框原型

标注:

对所有表单区域

边框原型设置包括: 1) 实线 2) 虚线 3) 斜线 4) 插入线 5) 下划线

#### BorderWidth

类型:

short

描述:

表单区域的宽度

标注:

对所有表单区域

#### ButtonLayout

类型:

short

描述:

按钮的布局外观, 有效值包括:

- 0 – 只有文本; 按钮只有标题没有图标。
- 1 – 只有图标; 按钮只有图标没有标题。
- 2 – 图标覆盖文本; 图标应显示在标题的上一层。
- 3 – 文本覆盖图标; 文本应显示在图标的上一层。
- 4 – 图标连着文本; 图标应显示在标题的左侧。
- 5 – 文本连着图标; 图标应显示在标题的右侧。

6 – 文本覆盖图标；文本应被覆盖在图标的上一层。

标注：

对所有按钮布局

## **CalcOrderIndex**

类型：

short

描述：

当前表单区域的索引

标注：

对所有表单区域

## **CharLimit**

类型：

short

描述：

文本区域内字符数量的限制

标注：

仅对文本区域

## **DefaultValue**

类型：

String

描述：

表单区域的默认值

标注：

对所有表单区域

## **IsEditable**

类型：

Boolean

描述：

标示组合框是否可编辑

标注：

只对组合框

## **Behavior**

类型：

String

描述:

None, Invert, Outline, Push

N (None) 无高亮

I (Invert) 反转注释矩形框内容

O (Outline) 反转注释边框

P (Push) 如它被推到内嵌于页面表面那样显示注释

## **IsHidden**

类型:

Boolean

描述:

标识表单区域是否被隐藏

标注:

对所有表单区域

## **IsMultiline**

类型:

Boolean

描述:

标识文本区域是多行还是单行

标注:

只对文本区域

## **IsPassword**

类型:

Boolean

描述:

标识是否当作密码输入那样掩饰输入

标注:

只对文本区域

## **IsReadOnly**

类型:

Boolean

描述:

是否设置表单区域为只读

标注:

对所有表单区域

## **IsRequired**

类型:

Boolean

描述:

判断区域是否必须为非空

标注:

对组合框、单选按钮、文本区域

## **Name**

类型:

String

描述:

当前表单区域名称

标注:

只读, 对所有表单区域

## **NoViewFlag**

类型:

Boolean

描述:

是否显示表单元素, 1=显示; 0=隐藏

标注:

对所有表单

## **PrintFlag**

类型:

Boolean

描述:

打印品中是否包含表单元素, 1=显示; 0=隐藏

标注:

对所有表单

## **Style**

类型:

CString

描述:

设置复选框与单选按钮的形状:

1) 方格, 2) 交叉, 3) 菱形, 4) 圆形, 5) 星形, 6) 四方形

标注:

对复选框与单选按钮

## TextFont

类型:

String

描述:

字体 (Reference 1.7 416)

可设置为:

Courier  
Courier-Bold  
Courier-Oblique  
Courier-BoldOblique  
Helvetica  
Helvetica-Bold  
Helvetica-Oblique  
Helvetica-BoldOblique  
Symbol  
Times-Roman  
Times-Bold  
Times-Italic  
Times-BoldItalic  
ZapfDingbats

描述:

对除了复选框、单选按钮外的所有表单区域

## TextSize

类型:

Short

描述:

表单区域中文本的大小

标注:

对除了复选框、单选按钮外的所有表单区域

## Type

类型:

String

描述:

表单类型

标注:

对所有表单

可设置为: 文本、按钮、组合框、列表框、复选框、单选按钮

## Value

类型:

String

描述:

当前值

标注:

表单元素有以下属性:

1) 文本区域 2) 组合框 3) 单选按钮 4) 复选框 <Yes&Off> 5) 列表框

### **Tooltip**

类型:

String

描述:

显示工具提示条

标注:

对所有表单

### **Orientation**

类型:

Short

描述:

表单的文本旋转

标注:

对所有表单

## 方法

### **PopulateListOrComboBox**

把值列入列表框和组合框的条目中

原型:

```
Void      PopulateListOrComboBox ( const VARIANT& arrItems,  
                                  const VARIANT& arrExportVal);
```

参数:

arrItem - 一字符串数组, 每个元素代表一个项目名称

arrExportVal - 一字符串数组, 与第一个参数有相同大小, 每个元素代表一个输出值

返回值:

[无]

### **SetBackgroundColor**

设置表单区域的背景色

原型:

```
Void      SetBackgroundColor (LPCTSTR bstrColorSpace, float redC, float greenM,  
                           float blueY, float AlphaK);
```

参数:

bstrColorSpace - 可为以下任一个： 透明、 灰色、 RGB 或 CMYK 颜色空间， 使用 T、 G、 RGB 和 CMYK 来代表他们。  
 对于 T 和 G, redC 是必须的；对于 RGB, redC, greenM 和 blueY 是必须的；对于 CMYK, redC,greenM,blueY 和 AlphaK 是必须的。  
 redC, greenM 和 blueY 的值从 0 到 1。

返回值：

[无]

### **SetBorderColor**

设置边框颜色

原型：

```
void SetBorderColor (LPCTSTR bstrColorSpace, float redC, float greenM, float
                     blueY, float AlphaK);
```

参数：

bstrColorSpace - 可为以下任一个： 透明、 灰色、 RGB 或 CMYK 颜色空间， 使用 T、 G、 RGB 和 CMYK 来表示他们。  
 对于 T 和 G, redC 是必须的；对于 RGB, redC, greenM 和 blueY 是必须的；对于 CMYK, redC,greenM,blueY 和 AlphaK 是必须的。  
 redC, greenM 和 blueY 的值从 0 到 1。

返回值：

[无]

### **SetForegroundColor**

设置前景色

原型：

```
Void SetForegroundColor (LPCTSTR bstrColorSpace, float redC, float greenM,
                        float blueY, float AlphaK);
```

参数：：

bstrColorSpace - 可为以下任一个： 透明、 灰色、 RGB 或 CMYK 颜色空间， 使用 T、 G、 RGB 和 CMYK 来表示他们。  
 对于 T 和 G, redC 是必须的；对于 RGB, redC, greenM 和 blueY 是必须的；对于 CMYK, redC,greenM,blueY 和 AlphaK 是必须的。  
 redC, greenM 和 blueY 的值从 0 到 1。

返回值：

[无]

### **SetButtonCaption**

设置按钮上的标题

原型：

```
void SetButtonCaption (LPCTSTR bstrFace, LPCTSTR bstrCaption);
```

参数：

bstrFace - 一字符串确定了按钮的标题显示状况  
 有效字符串包括：  
 N — 正常显示

D — 向下显示

R — 翻转显示

bstrCaption - 按钮的标题。

返回值:

[无]

### **SetButtonIcon**

设置按钮的图标

原型:

Void SetButtonIcon (LPCTSTR bstrFace, LPCTSTR bstrFilePath);

参数:

bstrFace - 一字符串确定了按钮的标题显示状况

有效字符串包括:

N — 正常显示

D — 向下显示

R — 翻转显示

bstrFilePath - 图形文件的完整路径。

返回值:

[无]

### **SetExportValues**

当输出单选按钮和复选框后, 为不同的选择设置输出值 (选择、未选择、选中、未选中, 等)

原型:

void SetExportValues (const VARIANT& arrExportVal);

参数:

arrExportVal - 输出值数组

返回值:

[无]

### **SetJavaScriptAction**

设置 JavaScript 的动作

原型:

Void SetJavaScriptAction (LPCTSTR bstrTrigger, LPCTSTR bstrJavaScript);

参数:

bstrTrigger - 为动作指定触发器的字符串

有效字符串包括:

up

down

enter

exit

calculate

validate

format

**keystroke**

bstrJavaScript - 脚本本身。

返回值：

[无]

**SetResetFormAction**

设置表单区域的重置功能动作

原型：

```
void SetResetFormAction (LPCTSTR bstrTrigger, long bFlags, const VARIANT&
                        arrFields);
```

参数：

bstrTrigger - 一字符串指定某触发器为功能所使用  
有效字符串包括：

up — Mouse up  
down — Mouse down  
enter — Mouse enter  
exit — Mouse exit

bFlags - 定义各种事件特性的标志集合。

arrFields - 导出表单元素数组

返回值：

[无]

**SetSubmitFormAction**

设置提交表单的动作

原型：

```
Void SetSubmitFormAction (LPCTSTR bstrTrigger, LPCTSTR bstrURL, long
                        bFlags, const VARIANT& arrFields);
```

参数：

bstrTrigger - 一字符串指定某触发器为动作所使用  
有效字符串包括：

up - Mouse up  
down - Mouse down  
enter - Mouse enter  
exit - Mouse exit

bstrURL - 包含有 URL 的字符串

bFlags - 定义各种事件特性的标志集合。

arrFields - 提交表单元素数组

返回值：

[无]

## PDFPrinter

你可以使用 `IPDFPrinter` 接口来控制打印机并发送打印输出。

### 属性

#### **PrinterName**

类型:

`BSTR`

描述:

设置打印机名称，该名称将会用在打印输出中。

#### **printerRangeMode**

类型:

`PrinterRangeMode`

描述:

设置打印机名称，该名称将会用在打印输出中。

设置打印范围为:

```
PRINT_RANGE_ALL      = 0,  
PRINT_RANGE_CURRENT_VIEW = 1,  
PRINT_RANGE_CURRENT_PAGE = 2,  
PRINT_RANGE_SELECTED = 3,
```

#### **printerRangeFrom**

类型:

`short`

描述:

指定第一个要打印的页码。

你必须先将 `PrinterRangeMode` 设置 为 `PRINT_RANGE_SELECTED`

#### **printerRangeTo**

类型:

`short`

描述:

制定最后一个要打印的页码。

你必须先将 `PrinterRangeMode` 设置 为 `PRINT_RANGE_SELECTED`

#### **numOfCopies**

类型:

`short`

描述:

确定打印份数。

## **Scaling**

类型:

short

描述:

设置打印对话框中的缩放比例。

## **AutoRotate**

类型:

boolean

描述:

设置打印对话框中的自动旋转的参数。

## **AutoCenter**

类型:

boolean

描述:

设置打印时是否自动居中, 1 = 是; 0 = 否

## **Collate**

类型:

boolean

描述:

设置在打印对话框中是否选择自动分页选项, 1 = 是; 0 = 否.

## **Rotation**

类型:

short

描述:

设置打印时是否进行文档旋转。

## **RangeSubset**

类型:

short

描述:

设置打印时是否包含子集。

## ReversePage

类型:

boolean

描述:

设置是否以反转顺序打印文档。

## PageBorder

类型:

boolean

描述:

设置是否打印页面边框。

## 方法

### PrintWithDialog

发送打印输出时显示 Windows 对话框。

原型:

void PrintWithDialog();

参数:

[无]

返回值:

[无]

### PrintQuiet

发送打印输出至指定打印机时, 不显示打印对话框。

原型:

void PrintQuiet();

参数:

[无]

返回值:

[无]

### SetPaperSize

对选定打印机设置页面大小, 关于有效的页面大小值, 请阅读 Windows SDK 程序说明书。

原型:

void SetPaperSize (long paperSize);

参数:

paperSize - 可使用的纸张大小。

返回值:

[无]

## PDFOutline

获取 PDF 大纲对象信息。

### 方法

#### **NavigateOutline**

导航至大纲对象所指定的目标。

原型:

void NavigateOutline();

参数:

[无]

返回值:

[无]

#### **GetOutlineTitle**

获取大纲对象的标题。

原型:

BSTR GetOutlineTitle()

参数:

[无]

返回值:

返回大纲对象的标题。

#### **GetOutLineTitle2**

以 VARIANT 类型的形式获取大纲对象的标题。

原型:

VARIANT GetOutLineTitle2()

参数:

[无]

返回值:

返回大纲对象的标题。

## PDFDocumentInfo

文档属性相关的信息

### 属性

#### Author

类型:

BSTR

描述:

PDF 文档的作者。

#### Subject

类型:

BSTR

描述:

PDF 文档的主题。

#### CreatedDate

类型:

BSTR

描述:

PDF 文档的创建日期。

#### ModifiedDate

类型:

BSTR

描述:

PDF 文档的修改日期。

#### Keywords

类型:

BSTR

描述:

PDF 文档的关键字。

#### Creator

类型:

BSTR

描述:

PDF 文档的创建者。

## **Producer**

类型:

BSTR

描述:

PDF 文档的创建器。

## **Title**

类型:

BSTR

描述:

PDF 文档的标题。

## FindResult

如果执行搜索并且找到匹配项，那么 `FindResult` 类就代表了一种搜索结果。

### 方法

#### GetFindPageNum

获取搜索结果的页面索引。

原型:

`long GetFindPageNum();`

参数:

[无]

返回值:

页面索引。

#### GetFindFileName

获取搜索结果的查询文件名称。

原型:

`BSTR GetFindFileName();`

参数:

[无]

返回值:

文件名称。

#### GetFindString

获取搜索结果的上下文。

原型:

`BSTR GetFindString()`

参数:

[无]

返回值:

返回搜索结果的上下文。