



# 用户手册

**Foxit<sup>®</sup> PDF SDK (ActiveX)**

编程指南

适用于Windows

**Microsoft<sup>®</sup> Partner**

Gold Independent Software Vendor (ISV)

©2013福昕软件 保留所有权

© 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

|                                     |           |
|-------------------------------------|-----------|
| 目录 .....                            | 3         |
| 概述 .....                            | 15        |
| 入门教程 .....                          | 17        |
| <b>标准版&amp;专业版 .....</b>            | <b>17</b> |
| 1. 打开一个 PDF 文件 .....                | 17        |
| 2. 跳转到指定页面 .....                    | 17        |
| 3. 缩放页面 .....                       | 17        |
| 4. 旋转页面视图 .....                     | 18        |
| 5. 打印 PDF 文件 .....                  | 18        |
| 6. 隐藏或显示用户界面元素 .....                | 18        |
| 7. 遍历整个大纲目录树 .....                  | 18        |
| 8. 检索 PDF 文件 .....                  | 18        |
| <b>签名模块 .....</b>                   | <b>19</b> |
| <b>批注工具 .....</b>                   | <b>19</b> |
| <b>表单模块的应用 .....</b>                | <b>19</b> |
| <b>IFoxitPDFSDK .....</b>           | <b>20</b> |
| <b>属性 .....</b>                     | <b>20</b> |
| 1) FilePath .....                   | 20        |
| 2) Password .....                   | 20        |
| 3) PageCount .....                  | 20        |
| 4) CurPage .....                    | 20        |
| 5) Rotate .....                     | 21        |
| 6) Zoomlevel .....                  | 21        |
| 7) CurrentTool .....                | 21        |
| 8) Printer .....                    | 23        |
| 9) DocumentInfo .....               | 23        |
| 10) ActiveXVersion .....            | 23        |
| 11) *bHasFormFields .....           | 23        |
| 12) *bHighlightFormFields .....     | 24        |
| 13) *FormFieldsHighlightAlpha ..... | 24        |
| 14) *FormFieldsHighlightColor ..... | 24        |
| <b>方法 .....</b>                     | <b>24</b> |
| 1. 解锁 .....                         | 24        |
| 1) UnLockActiveX .....              | 24        |
| 2) UnLockActiveXEx .....            | 25        |
| 3) *RemoveEvaluationMark .....      | 25        |
| 2. 全局设置 .....                       | 25        |
| 1) SetCurrentLanguage .....         | 25        |
| 2) SetCurrentLanguageByString ..... | 27        |
| 3) SetModulePath .....              | 27        |

|     |                                       |    |
|-----|---------------------------------------|----|
| 4)  | SetCrashLog .....                     | 27 |
| 5)  | SetLogFile .....                      | 28 |
| 3.  | 打开和关闭 PDF 文件 .....                    | 28 |
| 1)  | OpenFile .....                        | 28 |
| 2)  | OpenMemFile .....                     | 28 |
| 3)  | OpenBuffer .....                      | 29 |
| 4)  | OpenStream .....                      | 29 |
| 5)  | OpenCustomFile .....                  | 29 |
| 6)  | OpenFtpFile .....                     | 30 |
| 7)  | CloseFile .....                       | 30 |
| 8)  | SetFileStreamOption .....             | 30 |
| 9)  | OpenFileAsync .....                   | 31 |
| 4.  | 页面导航 .....                            | 31 |
| 1)  | ExistForwardStack .....               | 31 |
| 2)  | GoForwardStack .....                  | 32 |
| 3)  | ExistBackwardStack .....              | 32 |
| 4)  | GoBackwardStack .....                 | 32 |
| 5)  | SetViewRect .....                     | 32 |
| 6)  | ConvertClientCoordToPageCoord .....   | 33 |
| 7)  | ConvertClientCoordToPageCoordEx ..... | 33 |
| 8)  | ConvertPageCoordToClientCoord .....   | 34 |
| 9)  | GotoPageDest .....                    | 34 |
| 10) | GoToPagePos .....                     | 35 |
| 11) | GetVisibleLeftTopPage .....           | 35 |
| 12) | ScrollView .....                      | 35 |
| 13) | GetScrollLocation .....               | 36 |
| 14) | GetScrollLocationEx .....             | 36 |
| 15) | GoToPage .....                        | 36 |
| 16) | GoToNextPage .....                    | 36 |
| 17) | GoToPrevPage .....                    | 37 |
| 5.  | 查看 .....                              | 37 |
| 1)  | AboutBox .....                        | 37 |
| 2)  | ShowDocumentInfoDialog .....          | 37 |
| 3)  | SetPDFMeasureUnit .....               | 38 |
| 4)  | GetPageWidth .....                    | 38 |
| 5)  | GetPageHeight .....                   | 38 |
| 6)  | CountTools .....                      | 38 |
| 7)  | GetToolByIndex .....                  | 39 |
| 8)  | ForceRefresh .....                    | 39 |
| 9)  | *IsDualPage .....                     | 39 |
| 10) | *Highlight .....                      | 40 |
| 11) | *RemoveAllHighlight .....             | 40 |
| 6.  | 超链接 .....                             | 40 |

|     |                                 |    |
|-----|---------------------------------|----|
| 1)  | *CountHyperLinks .....          | 40 |
| 2)  | *HighlightHyperLink .....       | 41 |
| 3)  | *GetHyperLinkRect .....         | 41 |
| 4)  | GetHyperLinkInfo .....          | 41 |
| 5)  | EnableHyperLink .....           | 42 |
| 7.  | 文本.....                         | 42 |
| 1)  | GetPageText.....                | 42 |
| 2)  | *GetPageTextW .....             | 42 |
| 3)  | GetSelectedRectByIndex .....    | 43 |
| 4)  | GetSelectedRectsCount.....      | 43 |
| 5)  | GetSelectedText .....           | 43 |
| 6)  | GetSelectedTextEx .....         | 44 |
| 8.  | 保存.....                         | 44 |
| 1)  | SaveAs.....                     | 44 |
| 2)  | Save .....                      | 44 |
| 3)  | SaveToStream .....              | 45 |
| 4)  | *UploadCurFileToFTP .....       | 45 |
| 9.  | 自定义用户界面.....                    | 45 |
| 1)  | ShowTitleBar.....               | 45 |
| 2)  | ShowToolBar .....               | 46 |
| 3)  | ShowToolBarButton .....         | 46 |
| 4)  | ShowStatusBar .....             | 46 |
| 5)  | EnableToolTip.....              | 46 |
| 6)  | ShowNavPanelByString .....      | 47 |
| 7)  | ShowNavigationPanels .....      | 47 |
| 8)  | GetPanelStatus.....             | 47 |
| 9)  | SetLayoutShowMode .....         | 48 |
| 10) | GetLayoutShowMode .....         | 48 |
| 11) | SetFacingCoverLeft.....         | 48 |
| 12) | *ShowContextMenu .....          | 49 |
| 13) | *ShowFormFieldsMessageBar ..... | 49 |
| 10. | 打印.....                         | 49 |
| 1)  | PrintWithDialog.....            | 49 |
| 2)  | OpenMemFileForPrinter .....     | 49 |
| 3)  | OpenFileForPrinter .....        | 50 |
| 11. | 搜索.....                         | 50 |
| 1)  | FindFirst .....                 | 50 |
| 2)  | FindFirstEx .....               | 50 |
| 3)  | FindNext.....                   | 51 |
| 4)  | FindFileFirst .....             | 51 |
| 5)  | FindFileFirstEx .....           | 52 |
| 6)  | FindFileNext.....               | 52 |
| 7)  | GoToSearchResult.....           | 52 |

|     |                                       |    |
|-----|---------------------------------------|----|
| 8)  | FindPageNext.....                     | 53 |
| 9)  | FindClose.....                        | 53 |
| 10) | FindPageFirst.....                    | 53 |
| 11) | FindMemFileFirst.....                 | 54 |
| 12) | *SearchAndHighlightAllTextOnPage..... | 54 |
| 12. | 书签.....                               | 55 |
| 1)  | GetOutlineFirstChild.....             | 55 |
| 2)  | GetOutlineNextSibling.....            | 55 |
| 13. | 安全.....                               | 55 |
| 1)  | GetDocPermissions.....                | 55 |
| 2)  | * CheckOwnerPassword.....             | 55 |
| 3)  | *SetOwnerPassword.....                | 56 |
| 4)  | *SetUserPermission.....               | 56 |
| 5)  | *SetUserPassword.....                 | 56 |
| 14. | 注释.....                               | 57 |
| 1)  | *ExportAnnotsToFDFFile.....           | 57 |
| 2)  | *ImportAnnotsFromFDFFile.....         | 57 |
| 3)  | *SetBDrawAnnot.....                   | 57 |
| 4)  | *ShowAllPopup.....                    | 57 |
| 5)  | ^GetPageAnnots.....                   | 58 |
| 6)  | ^GetFormatTool.....                   | 58 |
| 15. | 表单.....                               | 58 |
| 1)  | *ExportFormToFDFFile.....             | 58 |
| 2)  | *ImportFormFromFDFFile.....           | 59 |
| 3)  | *FindFormFieldsTextFirst.....         | 59 |
| 4)  | *FindFormFieldsTextNext.....          | 59 |
| 5)  | * SubmitForm.....                     | 59 |
| 6)  | #GetCurrentForm.....                  | 60 |
| 16. | 页面组织.....                             | 60 |
| 1)  | *InsertPage.....                      | 60 |
| 2)  | *DeletePage.....                      | 60 |
| 3)  | *SwapPage.....                        | 61 |
| 4)  | *SetPageIndex.....                    | 61 |
| 5)  | *SetPageCropBox.....                  | 61 |
| 6)  | * SetPageRotate.....                  | 62 |
| 7)  | *FlattenPage.....                     | 62 |
| 8)  | *InsertNewPage.....                   | 62 |
| 9)  | *ExportPagesToPDF.....                | 63 |
| 17. | JavaScript.....                       | 63 |
| 1)  | * RunJScrip.....                      | 63 |
| 2)  | *ShowDocJsDialog.....                 | 64 |
| 3)  | *ShowJsConsoleDialog.....             | 64 |
| 18. | 多实例.....                              | 64 |

|     |                              |           |
|-----|------------------------------|-----------|
| 1)  | SetCurrentWnd .....          | 64        |
| 2)  | GetCurrentWnd.....           | 64        |
| 3)  | GetCtrlInstance .....        | 65        |
| 19. | 签名.....                      | 65        |
| 1)  | &GetPDFSignatureMgr.....     | 65        |
| 2)  | &GetLastSigModuleError.....  | 65        |
| 3)  | &GetLastSigModuleErrMsg..... | 66        |
| 20. | 其他.....                      | 66        |
| 1)  | *AddImageObject.....         | 66        |
| 2)  | *GetBitmap .....             | 66        |
| 3)  | *AddWaterMark.....           | 67        |
| 4)  | * GetBarcodeBitmap.....      | 67        |
| 5)  | *ConvertPDFPageToImage.....  | 68        |
| 6)  | *ImportImageToPdf .....      | 69        |
| 7)  | SetContextMenuString .....   | 69        |
| 事件  | .....                        | <b>70</b> |
| 1)  | BeforeDraw .....             | 70        |
| 2)  | AfterDraw.....               | 70        |
| 3)  | OnZoomChange .....           | 70        |
| 4)  | OnPageChange .....           | 70        |
| 5)  | OnOpenPassword.....          | 70        |
| 6)  | OnSearchProgress .....       | 71        |
| 7)  | OnOpenFile .....             | 71        |
| 8)  | OnOpenDocument .....         | 71        |
| 9)  | OnFilePathInvalidate .....   | 71        |
| 10) | OnShowSavePrompt .....       | 72        |
| 11) | OnCloseDocument .....        | 72        |
| 12) | OnDocumentChange.....        | 72        |
| 13) | CustomFileGetSize .....      | 72        |
| 14) | CustomFileGetBlock.....      | 72        |
| 15) | OnClick .....                | 73        |
| 16) | OnDbClick.....               | 73        |
| 17) | OnRButtonClick .....         | 73        |
| 18) | OnDownloadFinish .....       | 74        |
| 19) | OnErrorOccurred.....         | 74        |
| 20) | OnUploadFinish.....          | 74        |
| 21) | OnTextHyperLink.....         | 74        |
| 22) | OnExcuteMenuItem .....       | 75        |
| 23) | *OnDoGoToRAction .....       | 75        |
| 24) | *OnHyperLink.....            | 75        |
| 25) | *OnContextMenuIndex.....     | 76        |
| 26) | *OnAddMenuItemAction.....    | 76        |
| 27) | #FormFieldError .....        | 76        |

|                                   |           |
|-----------------------------------|-----------|
| <b>ILink_Dest</b> .....           | <b>78</b> |
| 方法.....                           | <b>78</b> |
| 1)    GetPageIndex .....          | 78        |
| 2)    GetZoomMode .....           | 78        |
| 3)    GetZoomParamCount .....     | 78        |
| 4)    GetZoomParam .....          | 79        |
| 5)    GetDestName.....            | 79        |
| <b>IPDFAction</b> .....           | <b>80</b> |
| 方法.....                           | <b>80</b> |
| 1)    GetURIPath.....             | 80        |
| 2)    GetFilePath.....            | 80        |
| 3)    GetType .....               | 80        |
| 4)    GetDest .....               | 81        |
| <b>IPDFOutline</b> .....          | <b>82</b> |
| 方法.....                           | <b>82</b> |
| 1)    GetOutlineDest .....        | 82        |
| 2)    GetOutlineAction .....      | 82        |
| 3)    GetOutLineColor .....       | 82        |
| 4)    NavigateOutline .....       | 82        |
| 5)    GetOutlineTitle.....        | 83        |
| 6)    GetOutLineTitle2.....       | 83        |
| 7)    GetOutlineExpandValue ..... | 83        |
| <b>IFindResult</b> .....          | <b>84</b> |
| 方法.....                           | <b>84</b> |
| 1)    GetFindRectsCount .....     | 84        |
| 2)    GetFindRectByIndex.....     | 84        |
| 3)    GetFindPageNum .....        | 84        |
| 4)    GetFindFileName .....       | 85        |
| 5)    GetFindString .....         | 85        |
| <b>IPDFPrinter</b> .....          | <b>86</b> |
| 属性.....                           | <b>86</b> |
| 1)    PrinterName.....            | 86        |
| 2)    PrinterRangeMode.....       | 86        |
| 3)    PrinterRangeFrom .....      | 86        |
| 4)    PrinterRangeTo.....         | 86        |
| 5)    NumOfCopies .....           | 87        |
| 6)    Scaling .....               | 87        |
| 7)    AutoRotate .....            | 87        |
| 8)    AutoCenter .....            | 87        |
| 9)    Collate .....               | 87        |
| 10)   Rotation.....               | 88        |
| 11)   RangeSubset.....            | 88        |
| 12)   ReversePage .....           | 88        |

|                         |                                  |           |
|-------------------------|----------------------------------|-----------|
| 13)                     | PageBorder.....                  | 88        |
| 方法                      | .....                            | 88        |
| 1)                      | PrintWithDialog.....             | 88        |
| 2)                      | PrintQuiet.....                  | 89        |
| 3)                      | SetPaperSize.....                | 89        |
| 4)                      | GetSystemPrinterCount.....       | 89        |
| 5)                      | GetSystemPrinterNameByIndex..... | 89        |
| 6)                      | SetPaperSizeByPage.....          | 90        |
| <b>IPDFDocumentInfo</b> | .....                            | <b>91</b> |
| 属性                      | .....                            | <b>91</b> |
| 1)                      | Author.....                      | 91        |
| 2)                      | Subject.....                     | 91        |
| 3)                      | CreatedDate.....                 | 91        |
| 4)                      | ModifiedDate.....                | 91        |
| 5)                      | Keywords.....                    | 92        |
| 6)                      | Creator.....                     | 92        |
| 7)                      | Producer.....                    | 92        |
| 8)                      | Title.....                       | 92        |
| <b>#IPDFForm</b>        | .....                            | <b>93</b> |
| 方法                      | .....                            | <b>93</b> |
| 1)                      | #AddField.....                   | 93        |
| 2)                      | #RemoveFieldsByName.....         | 93        |
| 3)                      | #ExportToFDF.....                | 94        |
| 4)                      | #ImportFromFDF.....              | 94        |
| 5)                      | #GetFieldByIndex.....            | 94        |
| 6)                      | #GetFieldsCount.....             | 95        |
| 7)                      | #RemoveField.....                | 95        |
| <b>#IPDFFormField</b>   | .....                            | <b>96</b> |
| 属性                      | .....                            | <b>96</b> |
| 1)                      | #Alignment.....                  | 96        |
| 2)                      | #BorderStyle.....                | 96        |
| 3)                      | #BorderWidth.....                | 96        |
| 4)                      | #ButtonLayout.....               | 97        |
| 5)                      | #CalcOrderIndex.....             | 97        |
| 6)                      | #CharLimit.....                  | 97        |
| 7)                      | #DefaultValue.....               | 97        |
| 8)                      | #IsEditable.....                 | 98        |
| 9)                      | #Behavior.....                   | 98        |
| 10)                     | #IsHidden.....                   | 98        |
| 11)                     | #IsMultiline.....                | 98        |
| 12)                     | #IsPassword.....                 | 99        |
| 13)                     | #IsReadOnly.....                 | 99        |
| 14)                     | #IsRequired.....                 | 99        |

|                             |                               |            |
|-----------------------------|-------------------------------|------------|
| 15)                         | #Name .....                   | 99         |
| 16)                         | #NoViewFlag .....             | 100        |
| 17)                         | #PrintFlag .....              | 100        |
| 18)                         | #Style.....                   | 100        |
| 19)                         | #TextFont.....                | 100        |
| 20)                         | #TextSize.....                | 101        |
| 21)                         | #Type .....                   | 101        |
| 22)                         | #Value.....                   | 102        |
| 23)                         | #Tooltip .....                | 102        |
| 24)                         | #Orientation .....            | 102        |
| 25)                         | #DirtyFlag .....              | 102        |
| <b>方法 .....</b>             |                               | <b>103</b> |
| 1)                          | #PopulateListOrComboBox ..... | 103        |
| 2)                          | #SetBackgroundColor.....      | 103        |
| 3)                          | #SetBorderColor.....          | 103        |
| 4)                          | #SetForegroundColor .....     | 104        |
| 5)                          | #SetButtonCaption .....       | 104        |
| 6)                          | #SetButtonIcon .....          | 105        |
| 7)                          | #SetExportValues .....        | 105        |
| 8)                          | #SetJavaScriptAction .....    | 105        |
| 9)                          | #SetResetFormAction .....     | 106        |
| 10)                         | #SetSubmitFormAction .....    | 106        |
| 11)                         | #GetPageIndex .....           | 107        |
| 12)                         | #GetRectTop.....              | 107        |
| 13)                         | #GetRectLeft .....            | 107        |
| 14)                         | #GetRectRight .....           | 108        |
| 15)                         | #GetRectBottom .....          | 108        |
| <b>^IPDFPageAnnots.....</b> |                               | <b>109</b> |
| <b>方法 .....</b>             |                               | <b>109</b> |
| 1)                          | ^GetAnnot.....                | 109        |
| 2)                          | ^AddAnnot .....               | 109        |
| 3)                          | ^RemoveAnnot .....            | 110        |
| 4)                          | ^GetAnnotIndex.....           | 110        |
| 5)                          | ^GetAnnotsCount .....         | 110        |
| <b>^IPDFAnnot.....</b>      |                               | <b>112</b> |
| <b>属性 .....</b>             |                               | <b>112</b> |
| 1)                          | ^Thickness.....               | 112        |
| 2)                          | ^BorderStyle.....             | 112        |
| 3)                          | ^Color .....                  | 113        |
| 4)                          | ^LineStartingStyle .....      | 113        |
| 5)                          | ^LineEndingStyle .....        | 114        |
| 6)                          | ^FillColor .....              | 114        |
| 7)                          | ^Opacity .....                | 114        |

|                |                            |            |
|----------------|----------------------------|------------|
| 8)             | ^Author .....              | 115        |
| 9)             | ^Subject.....              | 115        |
| 10)            | ^CreationDate .....        | 115        |
| 11)            | ^ModificationDate .....    | 115        |
| 12)            | ^Locked .....              | 116        |
| 13)            | ^Print.....                | 116        |
| 14)            | ^ReadOnly .....            | 116        |
| 15)            | ^Description .....         | 117        |
| <b>方法.....</b> |                            | <b>117</b> |
| 1)             | ^GetType .....             | 117        |
| 2)             | ^GetSubType .....          | 117        |
| 3)             | ^GetContents .....         | 118        |
| 4)             | ^SetContents.....          | 118        |
| 5)             | ^IsPopupOpen.....          | 118        |
| 6)             | ^SetPopupOpen .....        | 118        |
| 7)             | ^HasPopup .....            | 119        |
| 8)             | ^GetRect.....              | 119        |
| 9)             | ^SetRect .....             | 119        |
| 10)            | ^SetLinkGoToAction .....   | 120        |
| 11)            | ^SetLinkURLAction .....    | 120        |
| 12)            | ^DoAction.....             | 121        |
| 13)            | ^HasAction .....           | 121        |
| 14)            | ^GetMarkedState .....      | 121        |
| 15)            | ^SetMarkedState.....       | 121        |
| 16)            | ^GetReviewState .....      | 122        |
| 17)            | ^SetReviewState .....      | 122        |
| 18)            | ^GetMigrationState.....    | 123        |
| 19)            | ^SetMigrationState .....   | 123        |
| 20)            | ^SetStartingPoint .....    | 123        |
| 21)            | ^GetStartingPoint.....     | 124        |
| 22)            | ^SetEndingPoint .....      | 124        |
| 23)            | ^GetEndingPoint .....      | 124        |
| 24)            | ^SetMediaPoster .....      | 125        |
| 25)            | ^SetMultimedia.....        | 125        |
| 26)            | ^SetLinkQuadPoints .....   | 126        |
| 27)            | ^SetPolygonVertices.....   | 126        |
| 28)            | ^SetPencilVertices .....   | 126        |
| 29)            | ^AttachFile .....          | 126        |
| 30)            | ^ GetReplyList .....       | 127        |
| 31)            | ^ UpdateAnnotReplies ..... | 127        |
| <b>事件.....</b> |                            | <b>127</b> |
| 1)             | ^OnAnnotCreated .....      | 127        |
| 2)             | ^OnAnnotDeleted .....      | 128        |

|                                   |                              |            |
|-----------------------------------|------------------------------|------------|
| 3)                                | ^OnAnnotModified .....       | 128        |
| 4)                                | ^OnAnnotReplyCreated .....   | 128        |
| 5)                                | ^OnAnnotReplyDeleted .....   | 128        |
| 6)                                | ^OnAnnotReplyModified .....  | 129        |
| 7)                                | ^OnAnnotRButtonDown .....    | 129        |
| 8)                                | ^OnAnnotRButtonUp .....      | 130        |
| 9)                                | ^OnAnnotLButtonDbClick ..... | 130        |
| 10)                               | ^OnAnnotMoving .....         | 130        |
| 11)                               | ^OnAnnotMouseEnter .....     | 131        |
| 12)                               | ^OnAnnotMouseExit .....      | 131        |
| <b>^ IPDFAnnotReplyList .....</b> |                              | <b>132</b> |
| 方法 .....                          |                              | <b>132</b> |
| 1)                                | ^ GetCount .....             | 132        |
| 2)                                | ^ GetItem .....              | 132        |
| 3)                                | ^ Remove .....               | 132        |
| 4)                                | ^ RemoveAll .....            | 132        |
| 5)                                | ^ Add .....                  | 133        |
| <b>^IPDFAnnotReply .....</b>      |                              | <b>134</b> |
| 方法 .....                          |                              | <b>134</b> |
| 1)                                | ^ SetCreator .....           | 134        |
| 2)                                | ^ GetCreator .....           | 134        |
| 3)                                | ^ SetContent .....           | 134        |
| 4)                                | ^ GetContent .....           | 134        |
| 5)                                | ^ GetChildren .....          | 135        |
| 6)                                | ^ GetParent .....            | 135        |
| 7)                                | ^ GetCreationDate .....      | 135        |
| 8)                                | ^ SetCreationDate .....      | 135        |
| 9)                                | ^ SetReadOnly .....          | 136        |
| 10)                               | ^ GetReplyID .....           | 136        |
| <b>^IPDFFormatTool .....</b>      |                              | <b>137</b> |
| 方法 .....                          |                              | <b>137</b> |
| 1)                                | ^SetFontName .....           | 137        |
| 2)                                | ^GetFontName .....           | 137        |
| 3)                                | ^SetFontSize .....           | 137        |
| 4)                                | ^GetFontSize .....           | 138        |
| 5)                                | ^SetFontColor .....          | 138        |
| 6)                                | ^GetFontColor .....          | 138        |
| 7)                                | ^SetBorderColor .....        | 139        |
| 8)                                | ^GetBorderColor .....        | 139        |
| 9)                                | ^SetFillColor .....          | 139        |
| 10)                               | ^GetFillColor .....          | 139        |
| 11)                               | ^SetFontBold .....           | 140        |
| 12)                               | ^GetFontBold .....           | 140        |

|                                      |                           |            |
|--------------------------------------|---------------------------|------------|
| 13)                                  | ^GetFontBoldEnable .....  | 140        |
| 14)                                  | ^SetFontItalic .....      | 141        |
| 15)                                  | ^GetFontItalic.....       | 141        |
| 16)                                  | ^SetFontItalicEnable..... | 141        |
| 17)                                  | ^SetAlign .....           | 141        |
| 18)                                  | ^GetAlign.....            | 142        |
| 19)                                  | ^SetCharSpace .....       | 142        |
| 20)                                  | ^GetCharSpace.....        | 142        |
| 21)                                  | ^SetCharHorzScale .....   | 143        |
| 22)                                  | ^GetCharHorzScale .....   | 143        |
| <b>&amp;IPDFSignatureMgr .....</b>   |                           | <b>144</b> |
| 方法.....                              |                           | <b>144</b> |
| 1)                                   | &Add .....                | 144        |
| 2)                                   | &SignDocument .....       | 144        |
| 3)                                   | &Verify .....             | 145        |
| 4)                                   | &GetCounts.....           | 146        |
| 5)                                   | &Get .....                | 146        |
| 6)                                   | &Clear .....              | 146        |
| 7)                                   | &Remove.....              | 146        |
| 8)                                   | &VerifyAll .....          | 147        |
| 9)                                   | &InitStraddleValue .....  | 147        |
| <b>&amp;IPDFSignatureField .....</b> |                           | <b>148</b> |
| 属性.....                              |                           | <b>148</b> |
| 1)                                   | &Reason .....             | 148        |
| 2)                                   | &Location .....           | 148        |
| 3)                                   | &Signer.....              | 148        |
| 4)                                   | &Filter .....             | 149        |
| 5)                                   | &SubFilter .....          | 149        |
| 6)                                   | &State.....               | 149        |
| 方法.....                              |                           | <b>150</b> |
| 1)                                   | &SetAPOptions.....        | 150        |
| 2)                                   | &SetAPText.....           | 150        |
| 3)                                   | &SetAPImage .....         | 151        |
| 4)                                   | &IsSigned .....           | 151        |
| 5)                                   | &SetSignerDN.....         | 152        |
| 6)                                   | &SetStatusImage.....      | 152        |
| 7)                                   | &GetPageIndex .....       | 153        |
| 8)                                   | &GetSourceBuffer .....    | 153        |
| 9)                                   | &GetSourceBufferLen ..... | 153        |
| 10)                                  | &GetSignedBuffer .....    | 153        |
| 11)                                  | &GetSignedBufferLen ..... | 154        |
| 12)                                  | &CreateSignedDoc .....    | 154        |
| 13)                                  | &SetVerifyResult .....    | 155        |

---

|                  |                                      |            |
|------------------|--------------------------------------|------------|
| 14)              | &SetCertPath.....                    | 155        |
| 15)              | &SetCertData .....                   | 156        |
| 16)              | &SetCertContext .....                | 156        |
| 17)              | &SetAPIImageData.....                | 156        |
| 18)              | &SetStatusImageData .....            | 157        |
| 19)              | &TurnGray .....                      | 158        |
| 20)              | &TurnBlur .....                      | 158        |
| 21)              | &SetVisible .....                    | 158        |
| 22)              | &GrayPrint.....                      | 159        |
| 23)              | &SetStraddleType .....               | 159        |
| 24)              | &SetStraddlePos.....                 | 160        |
| 25)              | &SetStraddleBitmap.....              | 160        |
| 26)              | &SetStraddlePages .....              | 160        |
| 27)              | &SetStraddleFirstPagePercent .....   | 161        |
| <b>事件.....</b>   |                                      | <b>161</b> |
| 1)               | &OnSetSignatureInfo .....            | 161        |
| 2)               | &OnSigning .....                     | 161        |
| 3)               | &OnVerifying.....                    | 162        |
| 4)               | &OnShowSignaturePropertyDialog ..... | 162        |
| <b>联系我们.....</b> |                                      | <b>163</b> |

## 概述

**Foxit PDF SDK ActiveX** 是一款可视编程组件，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 PDF SDK ActiveX** 使用与福昕阅读器 [Foxit Reader](#) 相同的核心技术，能够快速、准确、高质量的渲染 PDF 文件。

相对于 Foxit PDF SDK DLL 而言，**Foxit PDF SDK ActiveX** 使用更加灵活，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。**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。

**Foxit PDF SDK ActiveX 5.0** 添加了签名模块和其他高级的 PDF 功能，允许开发人员使用 Windows 系统证书中心存储的第三方数字证书签署 PDF 文档，同时，也支持数字签名验证（用于验证含数字签名文档的真实性和完整性）、签名时间戳、签名打印控制等。

此外，**Foxit PDF SDK ActiveX 5.0** 中还添加了许多新功能和事件，允许开发人员更好地控制组件、更好地使用 PDF 文档。

**Foxit PDF SDK ActiveX 5.0** 提供两个不同的版本，即标准版和专业版。这两个版本提供不同的 GUID，您可以在同一电脑上同时注册和使用。与标准版比较，专业版包含更多的特性：创建/编辑批注、导入/导出表单数据、执行 JS 脚本、将 PDF 导出为文本格式、数字签名等。您可以根据您的应用程序来选择 **ActiveX** 的版本。您可以在线购买专业版或标准版 **ActiveX**，如果您想要版本 5.0 中的表单模块、批注模块和电子签章模块，请联系我们的销售团队（发送邮件至 [sales@foxitsoftware.cn](mailto:sales@foxitsoftware.cn)）了解更多授权信息。在本开发手册中，所有带\*号的属性和功能只属于专业版本，所有标记(#)的属性和方法只属于表单模块，所有标记(^)的属性和方法只属于批注模块，所有标记(&)的属性和方法只适用于电子签章模块。

**Foxit PDF SDK ActiveX** 可运行在 Windows 95/NT 或之后的版本上，是一个独立的控件，无需再安装任何的 PDF 应用程序。需要注意的是，在 Windows 系统下，只有 Admin 权限的用户才能成功的注册 **ActiveX** 控件。

福昕已经为开发人员编写 VC++ 示例代码，通过示例代码即可快速的掌握如何使用 **ActiveX** 提供的属性和方法。请访问 [www.foxitsoftware.cn](http://www.foxitsoftware.cn) 下载相关的示例代码。

**Foxit PDF SDK ActiveX 5.0** 专业版增加了签名模块，需要另行授权许可。如需使用签名模

块，请通过邮件（[sales@foxitsoftware.cn](mailto:sales@foxitsoftware.cn)）与我们联系。

**解锁码：**如果您已经购买了 Foxit PDF SDK ActiveX 并且已经拿到正式的 ActiveX 控件和解锁码，在调用任何一个函数前，请确保在您的程序中至少调用过一次 [UnLockActiveX](#) or [UnLockActiveXEx](#) 函数。该函数将在后续的章节中有详细描述。如果您只是想评估 ActiveX，则无需调用该函数。

标准版 GUID：0F6C092B-6E4C-4976-B386-27A9FD9E96A1

专业版 GUID：F53B7748-643C-4A78-8DBC-01A4855D1A10

# 入门教程

福昕 ActiveX 控件提供的是一个独立的 OCX 文件。通过执行命令"regsvr32 OCX name"即可成功注册。需要注意的是，必须指定 OCX 的完整的路径。

ActiveX 控件不仅提供人机的交互界面，同时提供一系列属性和方法以便应用程序能够方便的操作 ActiveX 控件。

在以下列出的示例代码中，假定我们已经创建了一个 ActiveX 控件的实例，该实例的名称是 FoxitReaderSDK。

## 标准版&专业版

### 1. 打开一个 PDF 文件

我们将打开一个名为“testdoc.pdf” 的 PDF 文档。

// 提示: 如果您是在试用 ActiveX，则无需进行解锁。

// 试用的水印将在所有页面上显示。

// 对于已经付费的用户，则首先需要进行解锁。

```
FoxitReaderSDK.UnlockActiveX("license_id","unlock_code");
```

```
FoxitReaderSDK.OpenFile("testdoc.pdf","");
```

### 2. 跳转到指定页面

我们将跳转到当前 PDF 文档的第三页。

```
FoxitReaderSDK.GoToPage(2). //页码从 0 开始计算
```

### 3. 缩放页面

如果您想显示 PDF 页面的原始尺寸，可在 VC 中添加如下代码:

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

或者

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

如果您想设置缩放比率为 200%，可使用如下代码:

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

查看函数说明章节了解更多的细节。

## 4. 旋转页面视图

一个 PDF 页面可以在四个不同方向上显示: 垂直正向、旋转 90 度、旋转 180 度或旋转 270 度。您可以调用 `SetRotate` 函数或者设置 `Rotate` 属性来改变页面显示的方向。

如果您想将页面顺时针旋转 90 度, 可在 VC 中使用如下代码:

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

## 5. 打印 PDF 文件

只需调用 `PrintWithDialog` 方法, 即会弹出一个打印设置窗口, 完成设置后即可将 PDF 文件打印出来。如果想实现静默打印, 即不弹出打印设置窗口, 则需要通过 [PDFPrinter](#) 提供的接口来完成打印操作。

## 6. 隐藏或显示用户界面元素

调用 [ShowToolBar](#) 函数即可显示或隐藏工具栏。类似的, 调用 `ShowStatusBar` 可以显示或隐藏状态栏。如果您想创建自己的工具栏, 只需隐藏 `ActiveX` 控件内置工具栏, 然后在控件外部创建属于自己的工具栏。

## 7. 遍历整个大纲目录树

调用 [GetOutlineFirstChild](#) 和 [GetOutlineNextSibling](#) 即可遍历整个大纲目录树, 通过 [PDFOutline](#) 提供的接口即可查看各节点的信息。

## 8. 检索 PDF 文件

调用 [FindFirst](#) 函数即可定位到给定文本在整个 PDF 文档中的第一个位置。如果没有找到与给定文本相匹配的内容, 则函数返回值为 0。如果找到与给定文本相匹配的内容, 则函数返回非零值, 同时高亮对应的文本内容。重复调用 [FindNext](#) 函数, 可定位到下一个与给定文本相匹配的内容。

如果想要在不打开和不显示的情况对其他 PDF 文件进行检索, 只需调用 [FindFileFirst](#) 函数和 [FindFileNext](#) 函数。

## 签名模块

开发人员可以使用 Windows 系统证书中心存储的第三方数字证书签署 PDF 文档。签名模块支持数字签名验证（用于验证含数字签名文档的真实性和完整性）、签名时间戳、签名打印控制等。

## 批注工具

通过使用不同的批注工具，最终用户可以在 PDF 文件中绘制直线、圆形以及其他各种图形。要实现这样的功能，只需在应用程序中改变 [CurrentTool](#) 属性的值。例如：将 [CurrentTool](#) 属性的值设置为“Line Tool”。

从版本 4.0 开始，Foxit PDF SDK ActiveX 提供了操作 PDF 批注对象的接口。通过这些接口，可以创建属于自己的批注。

## 表单模块的应用

从版本 3.0 开始，Foxit PDF SDK ActiveX 提供了 PDF 表单的操作接口。通过调用 [GetCurrentForm](#) 函数，可获取当前 PDF 文件的 IForm 接口指针，然后使用 [IPDFForm](#) 和 [IPDFFormField](#) 提供的属性和方法来执行针对 PDF 表单的操作。如下列出的代码展示的是如何在一个 PDF 页面中添加一个按钮：

```
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; NULL; Outline;  
button1.SetTooltip("reset all form");  
button1.SetTextFont("Courier");  
button1.SetTextSize(15);  
button1.SetJavaScriptAction("down","app.alert(\"Mouse Down!\")");
```

# IFoxitPDFSDK

本章节主要讲解的是 ActiveX 提供的所有的属性和方法。需要注意的是，所有的描述均基于 C 语法。如果您使用的编程语言不同于 C/C++，请遵守该种编程语言的语法。

备注： 标记(\*)的函数属于专业版，标记 (#)的函数属于表单模块，标记 (^)的函数属于批注模块中使用，标记(&)的函数属于电子签章模块。

## 属性

### 1) FilePath

类型:

BSTR, 只读

描述:

当前打开的 PDF 文件的完整路径。如果当前没有打开的 PDF 文件，或者 PDF 文件是通过缓存或者流中打开的，则该属性的值为空。

### 2) Password

类型:

BSTR,只读

描述:

PDF 文件的密码。

### 3) PageCount

类型:

long, 只读

描述:

当前打开的 PDF 文件的总页数。

### 4) CurPage

类型:

long, 只读。

描述:

当前显示的 PDF 页面的索引值，索引值从 0 开始计算。

## 5) Rotate

类型:

short, 读/写

描述:

当前旋转的方向，其值如下所列：

- 0: 垂直正向;
- 1: 顺时针旋转 90 度;
- 2: 旋转 180 度;
- 3: 逆时针旋转 90 度。

## 6) Zoomlevel

类型:

long, 读/写

描述:

一般情况下，缩放的比率在 10~1600 之间。也可使用如下特定的值:

- 0: 显示页面的原始尺寸，相当于 100%。
- 1: 以适当的缩放比率显示页面，确保客户端显示窗口能够显示整个 PDF 页面。
- 2: 以适当的缩放比率显示页面，确保在水平方向上页面充满整个客户端显示窗口。

## 7) CurrentTool

类型:

BSTR, 读/写

描述:

读取或设置当前使用的工具，可使用如下所列的值:

- “Hand Tool”
- “ZoomOut Tool”
- “ZoomIn Tool”
- “Select Text Tool”
- “Find Text Tool”

“Snapshot Tool”  
\*“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”  
\*“Typewriter”  
\*“CallOut”  
\*“Textbox”  
\*“Image Tool”

\*"Sound Tool"  
\*"Movie Tool"  
\*"FileAttachment Tool"  
\*"Attach a file"

备注:

通过调用 `CountTools` 函数可以获取当前 `ActiveX` 版本所支持的工具的数量, 然后调用 `GetToolByIndex` 函数来获取每一个工具的名称。

## 8) Printer

类型:

[IPDFPrinter](#), 只读

描述:

该打印机属性返回的是一个 [IPDFPrinter](#) 接口, 通过接口可实现打印机的管理和打印操作。

## 9) DocumentInfo

类型:

`IPDFDocumentInfo*`, 只读

描述:

该属性返回的一个 [IPDFDocumentInfo](#) 接口, 通过接口可获取文档的其他信息, 如作者、创建者、创建时间、关键词、修改时间、创建工具、主题和标题。

## 10) ActiveXVersion

类型:

`BSTR`, 只读

描述:

获取 `ActiveX` 控件的版本信息。

## 11) \*bHasFormFields

类型:

`BOOL`, 只读

描述:

如果当前文档中含有 `PDF` 表单域, 则该属性的值为 `True`, 否则为 `False`。

## 12) \*bHighlightFormFields

类型:

BOOL, 读/写

描述:

设置该属性值为 True 将会高亮所有的 PDF 表单域, 突出 PDF 表单域, 给出提示的作用。

## 13) \*FormFieldsHighlightAlpha

类型:

short, 读/写

描述:

调整 PDF 表单域在被高亮时的透明度。0 代表完全透明, 255 代表不透明。

## 14) \*FormFieldsHighlightColor

类型:

OLE\_COLOR, 读/写

描述:

高亮 PDF 表单域时使用的颜色。

# 方法

## 1. 解锁

### 1) UnLockActiveX

使用从福昕公司获取的正式授权文件来解锁 ActiveX。

函数原型:

```
Void      UnLockActiveX(BSTR lisence_id, BSTR unlock_code)
```

参数:

license\_id - 授权标识字符串。  
unlock\_code - 解锁码字符串。

返回值:

无

备注:

如果只是试用 **ActiveX**, 则不需要调用该函数。试用水印将出现在所有页面上。  
如果是正式授权的 **ActiveX**, 必须首先调用该函数以去除试用水印, 再进行其他函数的调用。

## 2) **UnLockActiveXEx**

使用从福昕公司获取的正式授权文件来解锁 **ActiveX**。

函数原型:

```
Void      UnLockActiveXEx(BSTR strLicense)
```

参数:

strLicense - 福昕公司提供的 SDK 授权码

返回值:

无

备注:

该函数的功能与 **UnlockActiveX** 相同。

## 3) **\*RemoveEvaluationMark**

在拥有正式授权的情况下, 移除试用水印。

函数原型:

```
BOOL RemoveEvaluationMark()
```

参数:

无

返回值:

成功则返回 **True**, 反之则返回 **False**。

# 2. 全局设置

## 1) **SetCurrentLanguage**

在存在对应的 **XML** 文件的情况下, 根据语言识别号切换相应的语言。

函数原型:

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

参数:

LanguageID - 语言识别号。0-30 每个数字代表一种语言。

- 1-阿拉伯语
- 2-保加利亚语
- 3-匈牙利语
- 4-加泰罗尼亚语
- 5-捷克语
- 6-简体中文
- 7-繁体中文
- 8-丹麦语
- 9-荷兰语
- 10-英语
- 11-爱沙尼亚语
- 12-芬兰语
- 13-法语
- 14-加利西亚语
- 15-德语
- 16-希腊语
- 17-意大利语
- 18-韩语
- 19-拉托维亚语
- 20-立陶宛语
- 21-挪威语
- 22-波兰语
- 23-葡萄牙语
- 24-巴西葡萄牙语
- 25-罗马语
- 26-俄罗斯语
- 27-斯洛维尼亚语
- 28-西班牙语
- 29-瑞典语
- 30-土耳其语
- 31-希伯来语
- 32-日语
- 33-泰语
- 34-瓦伦西亚语

返回值:

无

备注:

ActiveX 支持对用户界面进行动态的语言切换，支持的语言多达 30 多种。只需加载额外的语言包(xml 文件)。如需指定的语言包，请发邮件至 [sales@foxitsoftware.cn](mailto:sales@foxitsoftware.cn) 与我们联系。

## 2) SetCurrentLanguageByString

通过指定语言包（XML 文件）来切换程序界面语言。

函数原型:

```
void SetCurrentLanguageByString(BSTR FileName);
```

参数:

FileName - XML 语言包的路径

返回值:

无

备注:

ActiveX 支持对用户界面进行动态的语言切换，支持的语言多达 30 多种。只需加载额外的语言包(xml 文件)。如需指定的语言包，请发邮件至 [sales@foxitsoftware.cn](mailto:sales@foxitsoftware.cn) 与我们联系。

## 3) SetModulePath

指定 fpdfcjk.bin 组件的路径,用以显示 PDF 文件中的中文、日文和韩文文本。

函数原型:

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

参数:

lpFolderName - fpdfcjk.bin 组件的完整路径

返回值:

无

备注:

fpdfcjk.bin 组件用以显示 PDF 文档中的中文、日文和韩文文本。

## 4) SetCrashLog

在应用程序中调用该函数可设置一个日志文件，该日志文件将记录程序运行中每次崩溃的情况。

函数原型:

```
BOOL SetCrashLog (BSTR filepath);
```

参数:

filepath - 指定日志文件的路径，对应的文件必须已经存在。

返回值:

返回值指明是否成功设置了日志文件。

## 5) SetLogFile

在应用程序中调用该函数可设置一个日志文件, 该日志文件将记录程序中调用的每一个接口。

函数原型:

```
BOOL SetLogFile(BSTR filepath);
```

参数:

filepath - 指定日志文件的路径, 对应的文件必须已经存在。

返回值:

返回值指明是否成功设置了日志文件。

# 3. 打开和关闭 PDF 文件

## 1) OpenFile

从本地磁盘中或从 http 网络服务器上打开一个 PDF 文件。

函数原型:

```
BOOL OpenFile (BSTR FilePath, BSTR Password)
```

参数:

FilePath - 本地 PDF 文件路径或者 HTTP 服务器上的 URL 地址。

Password - PDF 文件的打开密码。

返回值:

如果 PDF 文件打开成功, 则函数返回非零值, 反之则返回零。

备注:

通过该接口打开的 PDF 文件, 不会被独占, 其他应用程序还可同时打开同一个 PDF 文件。

## 2) OpenMemFile

打开内存中 PDF 文件。

函数原型:

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

参数:

pBuffer - 内存中指向 PDF 数据的指针

- Size - PDF 数据的大小
- Password - PDF 文件的打开密码

返回值:

如果 PDF 文件打开成功则返回一个非零值，反之返回值为零。

### 3) OpenBuffer

从缓存中打开一个 PDF 文件。

函数原型:

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

参数:

- Buffer - 包含 PDF 数据的字节数组
- Size - PDF 数据的大小
- Password - PDF 文档的打开密码

返回值:

如果 PDF 文件打开成功则返回一个非零值，反之返回值为零。

### 4) OpenStream

通过从 IStream 类提供的接口中获取 PDF 数据的方式打开一个 PDF 文件。

函数原型:

```
BOOL OpenStream (IStream* Stream, BSTR Password)
```

参数:

- Stream - 一个 IStream 的指针
- Password - PDF 文件的打开密码。如果文档没有设置密码，则此值为空即可。

返回值:

如果 PDF 文件打开成功则返回一个非零值，反之返回值为零。

### 5) OpenCustomFile

通过用户自定义方式打开一个 PDF 文件。

函数原型:

```
BOOL OpenCustomFile(BSTR Password)
```

参数:

- Password - PDF 文件的打开密码。如果没有密码，则设置为空字符串。

返回值:

如果 PDF 文件打开成功则返回一个非零值，反之返回值为零。

备注:

当应用程序调用该方法时, ActiveX 控件将触发 [CustomFileGetSize](#) 事件和 [CustomFileGetBlock](#) 事件。在这两个事件中, 应用程序将按照用户自定义的格式打开一个 PDF 文件, 并返回文件的大小和数据。详情请参照 [CustomFileGetSize](#) 和 [CustomFileGetBlock](#) 的描述。

## 6) OpenFtpFile

打开一个 FTP 服务器上的 PDF 文件。

函数原型:

```
BOOL    OpenFtpFile(BSTR ftpName,BSTR username, BSTR userPassword,
                    long port,BSTR filePath, BSTR filePassword, 30sernam Passive);
```

参数:

|              |   |               |
|--------------|---|---------------|
| ftpName      | - | FTP 服务器的名称    |
| Username     | - | FTP 服务器的登录用户名 |
| userPassword | - | FTP 服务器的登录密码  |
| port         | - | FTP 服务器端口     |
| filePath     | - | FTP 文件的路径     |
| filePassword | - | PDF 文件的密码     |
| Passive      | - | 是否主动连接服务器     |

返回值:

如果 PDF 文件打开成功则返回一个非零值, 反之返回值为零。

## 7) CloseFile

关闭当前打开的 PDF 文件。

函数原型:

```
Void    CloseFile()
```

参数:

无

返回值:

无

## 8) SetFileStreamOption

当打开一个 PDF 文件时, 设置文件流标记。

函数原型:

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

参数:

**bFileStream** - 布尔型, 文件流标记

返回值:

无

备注:

如果一个 PDF 文件被频繁的访问, 可以通过该方法将内容流加载到内存中以提高执行效率。当然这样将会消耗更多的内存。

## 9) OpenFileAsync

主要用于打开线性化 PDF 文档。可在 B/S 构架中快速下载并渲染 PDF 数据。

函数原型:

```
BOOL OpenFileAsync(LPCTSTR strURL, LPCTSTR strPDFPassword,  
LPCTSTR strUserName, LPCTSTR strUserPassword)
```

参数:

**strURL** - PDF 文档的 URL 下载地址, 可以 FTP 和 HTTP 类型;

**strPDFPassword** - PDF 文档的打开密码, 如果无密码, 则值为空;

**strUserName** - FTP 服务器的登陆用户名, 如果不需要, 则值为空;

**strUserPassword** - FTP 服务器的登陆密码, 如果不需要密码, 则值为空。

返回值:

调用成功则返回 TRUE, 反之返回 FALSE。

## 4. 页面导航

### 1) ExistForwardStack

查询是否存在下一个视图。

函数原型:

```
BOOL ExistForwardStack ();
```

参数:

无

返回值:

如果存在下一个视图, 则返回值为 true, 反之则返回值为 false。

备注:

当用户在浏览 PDF 过程中, 经常需要返回到上一个阅读的位置。我们将阅读到 PDF 文档的某一个位置或者 PDF 文档显示的某一个状态称作视图。用户的每一个操作都会产生一个新的视图。例如, 如果用户跳转到一个新的页

面，然后进行了缩放页面，那么这两个操作将产生两个新的视图。程序通过调用这一系列的方法，可以很方便的在不同的视图间进行切换。

## 2) GoForwardStack

跳转到下一个视图。

函数原型:

```
Void    GoForwardStack ();
```

参数:

无

返回值:

无

## 3) ExistBackwardStack

查询是否存在前一个视图。

函数原型:

```
BOOL    ExistBackwardStack ();
```

参数:

无

返回值:

如果存在前一个视图，则返回值为 true，反之则返回值为 false。

## 4) GoBackwardStack

跳转到前一个视图。

函数原型:

```
Void    GoBackwardStack ();
```

参数:

无

返回值:

无

## 5) SetViewRect

显示当前 PDF 页面指定区域的内容。

函数原型:

```
VoidSetViewRect (float Left, float Top, float Width, float Height);
```

参数:

|        |   |          |
|--------|---|----------|
| Left   | - | 左上角 x 坐标 |
| Top    | - | 左上角 y 坐标 |
| Width  | - | 指定区域的宽度  |
| Height | - | 指定区域的高度  |

返回值:

无

备注:

该函数将显示当前 PDF 页面指定区域的内容。这里提到的坐标指的是 PDF 的坐标，而非设备坐标。同样的，这里提到的长度单位是 PDF 的点。该函数不会改变 ActiveX 控件窗口的位置和尺寸大小，而是通过改变当前 PDF 页面的显示位置和缩放比率，以保证设定的区域内的 PDF 内容能够充满整个 ActiveX 控件窗口。一个典型的应用场景是:最终用户通过鼠标点击和拖拽划定一个区域后，松开鼠标，应用程序首先调用 `ConvertClientCoordToPageCoord` 接口将鼠标坐标转换为 PDF 坐标，然后调用 `SetViewRect` 接口将之前选中的区域内的 PDF 页面内容充满整个显示窗口。

## 6) ConvertClientCoordToPageCoord

将 ActiveX 控件的窗口坐标转换为 PDF 页面坐标。

函数原型:

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

参数:

|             |   |                                         |
|-------------|---|-----------------------------------------|
| nClientX    | - | ActiveX 控件的窗口坐标，x 坐标点，像素                |
| nClientY    | - | ActiveX 控件的窗口坐标，y 坐标点，像素                |
| pnPageIndex | - | 返回值。保存的是指定的坐标点转换成 PDF 页面坐标后，所在页面的页码索引值。 |
| pPageX      | - | 返回值。保存的是 PDF 页面坐标的 x 坐标点。               |
| pPageY      | - | 返回值。保存的是 PDF 页面坐标的 y 坐标点。               |

返回值:

函数返回值表示转换是否成功。给定的区域不仅会包括当前正在显示的 PDF 页面，也可能会包括灰色的页面背景区。

## 7) ConvertClientCoordToPageCoordEx

`ConvertClientCoordToPageCoord` 函数的扩展。

函数原型:

```
BOOL ConvertClientCoordToPageCoordEx (long nClientX, long nClientY,  
VARIANT* pnPageIndex, VARIANT* pPageX, VARIANT* pPageY);
```

参数:

- nClientX - ActiveX 控件的窗口坐标, x 坐标点, 像素
- nClientY - ActiveX 控件的窗口坐标, y 坐标点, 像素
- pnPageIndex - 返回值。保存的是指定的坐标点转换成 PDF 页面坐标后, 所在页面的页码索引值。
- pPageX - 返回值。保存的是 PDF 页面坐标的 x 坐标点。
- pPageY - 返回值。保存的是 PDF 页面坐标的 y 坐标点。

返回值:

函数返回值表示转换是否成功。给定的区域不仅会包括当前正在显示的 PDF 页面, 也可能会包括灰色的页面背景区。如果坐标点落在灰色的页面背景区, 则转换失败。

## 8) ConvertPageCoordToClientCoord

将 PDF 页面坐标转换为 ActiveX 控件的窗口坐标。

函数原型:

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

参数:

- nPageIndex - 页面索引值
- dPageX - PDF 页面坐标的 x 坐标点
- dPageY - PDF 页面坐标的 y 坐标点
- pnClientX - 返回值。保存的是 ActiveX 控件的窗口坐标的 x 坐标点。如果该坐标点超出了控件的窗口区域, 则被视为无效的转换结果。
- pnClientY - 返回值。保存的是 ActiveX 控件的窗口坐标的 y 坐标点。如果该坐标点超出了控件的窗口区域, 则被视为无效的转换结果。

返回值:

函数返回值表示转换是否成功。如果当前没有打开文档或者页码错误, 则返回值为 FALSE。否则, 返回值为 TRUE。

## 9) GotoPageDest

跳转到 PDF 文档中一个指定的位置。

函数原型:

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

参数:

link\_dest - 从 [OnHyperLink](#) 事件中获取的 [ILink\\_Dest](#) 接口的一个实例

返回值:

无

## 10) GoToPagePos

跳转到 PDF 文档中一个指定的位置。

函数原型:

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

参数:

nPageIndex - 跳转到该页码索引值指定的页面

PageX - nPageIndex 指定的页面内的 x 坐标点

PageY - nPageIndex 指定的页面内的 y 坐标点

返回值:

无

## 11) GetVisibleLeftTopPage

获取显示区域中左上角的页面的页码。

函数原型:

```
Long GetVisibleLeftTopPage ();
```

参数:

无

返回值:

返回显示区域中左上角的页面的页码。

## 12) ScrollView

将当前视图在水平方向和垂直方向上进行 dx 和 dy 距离的滚动，采用的是设备坐标。

函数原型:

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

参数:

dx - 水平方向的滚动的距离

dy - 垂直方向的滚动的距离

返回值:

无

### 13) GetScrollLocation

获取显示区域的滚动条当前的位置。

函数原型:

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

参数:

dx        -    返回水平滚动条的位置  
dy        -    返回垂直滚动条的位置

返回值:

无

### 14) GetScrollLocationEx

GetScrollLocation 方法的扩展。

函数原型:

```
void GetScrollLocationEx (VARIANT * HPos, VARIANT * VPos);
```

参数:

HPos      -    返回水平滚动条的位置  
VPos      -    返回垂直滚动条的位置

返回值:

无

### 15) GoToPage

跳转到当前打开的 PDF 文档指定的页面。

函数原型:

```
void GoToPage( long page_index )
```

参数:

page\_index    -    指定页面的页码。

返回值:

无

### 16) GoToNextPage

跳转到 PDF 文档当前页面的下一页。

函数原型:

```
Void    GoToNextPage ();
```

参数:

无

返回值:

无

备注:

该函数在后续的版本中将被删除，而以 `GoToPage` 函数代之。

## 17) GoToPrevPage

跳转到 PDF 文档当前页面的前一页。

函数原型:

```
Void    GoToPrevPage ();
```

参数:

无

返回值:

无

备注:

该函数在后续的版本中将被删除，而以 `GoToPage` 函数代之。

## 5. 查看

### 1) AboutBox

弹出“关于”对话框。

函数原型:

```
Void    AboutBox()
```

参数:

无

返回值:

无

### 2) ShowDocumentInfoDialog

弹出文档的属性对话框。

函数原型:

```
void    ShowDocumentInfoDialog()
```

参数:

无

返回值:

无

### 3) SetPDFMeasureUnit

设置 PDF 文档的测量单位。

函数原型:

```
BOOL SetPDFMeasureUnit(short nType);
```

参数:

nType - 指定测量的单位，它包括: 0 = 点; 1 = 英寸; 2 = 厘米; 3=像素

返回值:

成功则返回 True，反之返回 False。

### 4) GetPageWidth

获取指定的 PDF 页面的宽度。

函数原型:

```
float GetPageWidth(short nPageIndex);
```

参数:

nPageIndex - 指定页面的页码

返回值:

返回可显示区域的页面宽度（以点为单位）。

1 点= 1/72 英寸 (约等于 0.3528 mm)

### 5) GetPageHeight

获取指定 PDF 页面的高度。

函数原型:

```
float GetPageHeight(short nPageIndex);
```

参数:

nPageIndex - 指定页面的页码

返回值:

返回可显示区域的页面高度（以点为单位）。

1 点= 1/72 英寸 (约等于 0.3528 mm)

### 6) CountTools

获取当前 ActiveX 版本中可使用工具的数量。

函数原型:

```
short    CountTools()
```

参数:

无

返回值:

返回可使用工具的数量。

## 7) GetToolByIndex

根据索引值获取工具的名称。

函数原型:

```
BSTR    GetToolByIndex (short nIndex)
```

参数:

nIndex - 索引值，范围是大于等于 0，小于可使用工具的总数量

返回值:

返回工具的名称。

## 8) ForceRefresh

函数原型:

```
Void ForceRefresh()
```

备注:

该函数用于提高高亮显示的效率。在旧版本中，用户每添加一次高亮文本，页面就会刷新一次，若用户多次添加高亮文本，则页面将会多次刷新，这就降低了显示效率。从 5.0 版本开始，我们将高亮功能的添加和刷新动作进行分离，即在用户完成高亮添加后再调用 ForceRefresh 函数刷新页面。

## 9) \*IsDualPage

判断 PDF 文档的页面是否是双层的。

函数原型:

```
BOOL    IsDualPage(short pageIndex);
```

参数:

pageIndex - 指定的页码

返回值:

返回值指明该 PDF 页面是否是双层的。双层的意思是 PDF 页面包含一个图

片和与图片对应的隐藏的文本。

## 10) \*Highlight

在指定的文档页面上，将选定的矩形区域进行高亮。

函数原型:

```
void Highlight(long nPageIndex, float left, float top, float right, float bottom)
```

参数:

|            |   |                 |
|------------|---|-----------------|
| nPageIndex | - | 待高亮区域页面的页码      |
| left       | - | 指定矩形区域的左上角的水平坐标 |
| top        | - | 指定矩形区域的左上角的垂直坐标 |
| right      | - | 指定矩形区域的右下角的水平坐标 |
| bottom     | - | 指定矩形区域的右下角的垂直坐标 |

返回值:

无

## 11) \*RemoveAllHighlight

移除当前打开的 PDF 文档中所有的高亮。

函数原型:

```
void RemoveAllHighlight()
```

参数:

无

返回值:

无

# 6. 超链接

## 1) \*CountHyperLinks

统计指定页面所包含的超链接的数量。

函数原型:

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

参数:

|            |   |       |
|------------|---|-------|
| nPageIndex | - | 指定的页码 |
|------------|---|-------|

返回值:

成功则返回超链接的数量，返回值为 0 表示指定页面不包含任何超链接，返回值为-1 表示函数执行失败。

## 2) \*HighlightHyperLink

高亮显示指定页面中某一个超链接。

函数原型:

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

参数:

|            |   |         |
|------------|---|---------|
| nPageIndex | - | 指定页面的页码 |
| nLinkIndex | - | 超链接索引   |

返回值:

无

## 3) \*GetHyperLinkRect

获取指定超链接的位置。

函数原型:

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

参数:

|            |   |           |
|------------|---|-----------|
| nPageIndex | - | 指定页面的索引   |
| nLinkIndex | - | 超链接的索引    |
| top        | - | 顶部坐标返回的指针 |
| left       | - | 左边坐标返回的指针 |
| bottom     | - | 底部坐标返回的指针 |
| right      | - | 右边坐标返回的指针 |

返回值:

成功则返回 True，反之，则返回 False。

## 4) GetHyperLinkInfo

获取指定超链接的链接信息。

函数原型:

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

参数:

|            |   |       |
|------------|---|-------|
| nPageIndex | - | 指定的页面 |
|------------|---|-------|

- nLinkIndex - 超链接索引
- linktype - 含有超链接的返回的指针
- linkdata - 含有字符串数据的返回的指针
- linkdest - 含有超链接重定向目的地的返回的指针

返回值:

成功则返回 **True**，反之，则返回 **False**。

备注:

目前，仅支持获取 GoTo、GoToR、Lanuch 和 URI 类型链接的信息。

## 5) EnableHyperLink

启用/禁用超链接。

函数原型:

Void EnableHyperLink(BOOL bEnable)

参数:

- bEnable - 值为 **TRUE** 则启用超链接，值为 **FALSE** 则禁用超链接。

返回值:

无

## 7. 文本

### 1) GetPageText

从当前加载的 PDF 文件中提取指定页的文本内容。

函数原型:

BSTR GetPageText (long nPageIndex)

参数:

- nPageIndex - 指定页面的页码

返回值:

返回提取的文本。

### 2) \*GetPageTextW

从当前加载的 PDF 文件中提取指定页的文本内容。

函数原型:

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

参数:

- nPageIndex - 指定页面的页码
- pBuffer - 接收页面内容的缓存
- nBuflen - 实际提取到的页面内容的长度

返回值:

成功返回值为-1，反之为 0。

### 3) GetSelectedRectByIndex

获取指定索引的选择区域

函数原型:

```
long GetSelectedRectByIndex(long nIndex, long* nPageIndex, float* left,
float* bottom, float* right, float* top);
```

参数:

- nIndex - 当前选定区域的索引，索引值从 0 开始
- nPageIndex - 选定区域所在页面的页码
- Left - 选定区域左边的坐标
- Bottom - 选定区域底端的坐标
- Right - 选定区域右边的坐标
- Top - 选定区域顶端的坐标

返回值:

成功则返回 True，失败则返回 False。

### 4) GetSelectedRectsCount

获取文本选中区域的数量。

函数原型:

```
long GetSelectedRectsCount()
```

返回值:

返回文本选中区域的总数量。

### 5) GetSelectedText

获取当前选中的文本。

函数原型:

```
BSTR GetSelectedText ();
```

参数:

无

返回值:

函数返回当前选中的文本内容。

## 6) GetSelectedTextEx

GetSelectedText 函数的扩展。

函数原型:

```
long    GetSelectedTextEx(long* pBuffer, long nBuflen);
```

参数:

pBuffer - 接收选定文本内容的缓存

nBuflen - 选定文本内容的长度

返回值:

返回选定文本内容的长度。

# 8. 保存

## 1) SaveAs

将当前打开的文档另存为一个新的文件。

函数原型:

```
Void          SaveAs (BSTR FileName)
```

参数:

FileName - 指定当前 PDF 文件要保存的路径（仅支持本地路径）。

返回值:

无

## 2) Save

保存当前打开的文档。

函数原型:

```
Void          Save ()
```

参数:

无

返回值:

无

备注:

该函数只能保存本地磁盘中的 PDF 文件，无法保存从 URL 中打开的 PDF 文件。

### 3) SaveToStream

将当前打开的文档保存到内存流中。

函数原型:

```
IStream*      SaveToStream()
```

参数:

无

返回值:

函数返回值是一个 IStream 指针。IStream 提供的接口支持将 PDF 文件数据从 IStream 对象中读取，也支持将 PDF 数据写入 IStream 对象中。

### 4) \*UploadCurFileToFTP

将当前的 PDF 文件上传到一个 FTP 服务器。

函数原型:

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

参数:

|              |   |              |
|--------------|---|--------------|
| ftpName      | - | FTP 服务器名称    |
| userName     | - | FTP 服务器登录用户名 |
| userPassword | - | FTP 服务器登录密码  |
| port         | - | FTP 服务器端口    |
| filePath     | - | FTP 文件路径     |

返回值:

如果 PDF 文件上传成功则返回一个非零值，反之返回值为零。

## 9. 自定义用户界面

### 1) ShowTitleBar

显示或隐藏标题栏。

函数原型:

```
Void      ShowTitleBar(BOOL bShow);
```

参数:

bShow - 值为 FALSE, 则隐藏标题栏; 值为 TRUE, 则显示标题栏。

返回值:

无

## 2) ShowToolBar

显示或隐藏工具栏。

函数原型:

Void ShowToolBar(BOOL bShow);

参数:

bShow - 值为 FALSE, 则隐藏工具栏。值为 TRUE, 则显示工具栏。

返回值:

无

## 3) ShowToolBarButton

显示或隐藏工具栏上的按钮。

函数原型:

Void ShowToolBarButton (short nIndex, BOOL bShow)

参数:

nIndex - 按钮的索引值

bShow - 值为 FALSE, 则隐藏该按钮; 值为 TRUE, 则显示该按钮;

返回值:

无

## 4) ShowStatusBar

显示或隐藏状态栏。

函数原型:

Void ShowStatusBar(BOOL bShow);

参数:

bShow - 值为 FALSE, 则隐藏状态栏; 值为 TRUE, 则显示状态栏。

返回值:

无

## 5) EnableToolTip

显示或隐藏工具的提示信息。

函数原型:

**BOOL EnableToolTip (BOOL bEnable)**

参数:

**bEnable** - 值为 **TRUE** 表示显示提示信息, 值为 **FALSE** 隐藏提示信息。

返回值:

返回值为 **True** 表示设置成功, 反之则设置失败。

## 6) ShowNavPanelByString

显示指定名称的导航栏面板。

函数原型:

**BOOL ShowNavPanelByString(LPCTSTR lpszPanelName)**

参数:

**lpszPanelName** - 导航栏面板的名称, 包括"Bookmarks", "Pages", "Layer", "Attachments "

返回值:

成功则返回值为 **TRUE**, 反之则返回值为 **FALSE**。

## 7) ShowNavigationPanels

控制导航栏的显示或隐藏。

函数原型:

**BOOL ShowNavigationPanels(BOOL bShow)**

参数:

**bShow** - 值为 **TRUE**, 则显示整个导航栏。反之, 则为 **FALSE**。

返回值:

成功则返回值为 **True**, 反之则为 **False**。

## 8) GetPanelStatus

获取整个导航栏的显示状态。

函数原型:

**BOOL GetPanelStatus()**

参数:

无

返回值:

返回值为 **TRUE** 表示导航栏为显示状态, 反之表示导航栏为隐藏状态。

## 9) SetLayoutShowMode

设置页面显示布局的模式。

函数原型:

```
Void SetLayoutShowMode (BrowseMode nShowMode, short  
nFacingCount);
```

参数:

|              |   |                  |
|--------------|---|------------------|
| nShowMode    | - | 可以设置如下的值:        |
| 0            | = | MODE_SINGLE.     |
| 1            | = | MODE_CONTINUOUS. |
| nFacingCount | - | 每行显示的列数          |

返回值:

无

备注:

在显示区域, 可按照 **N\*M** 的方式显示 PDF 文档。当显示布局的模式设定成 **MODE\_SINGLE** 时, 在 **ActiveX** 控件的显示区域每行将只显示一页, **nFacingCount** 的值将不起作用。当显示布局的模式设定成 **MODE\_CONTINUOUS** 时, 在 **ActiveX** 控件的显示区域每行将可以显示多页。

## 10) GetLayoutShowMode

获取页面显示布局的模式。

函数原型:

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

参数:

|               |   |                  |
|---------------|---|------------------|
| pnShowMode    | - | 返回当前的模式          |
| pnFacingCount | - | 返回当前水平方向上显示页面的数量 |

返回值:

无

## 11) SetFacingCoverLeft

在双页显示模式下, 调用该函数将决定 PDF 文档的封面是否显示在左侧。

函数原型:

```
Void SetFacingCoverLeft (BOOL bLeft)
```

参数:

|       |   |                    |
|-------|---|--------------------|
| bLeft | - | 布尔值, 设定是否将封面显示在左侧。 |
|-------|---|--------------------|

返回值:

无

## 12) \*ShowContextMenu

显示或隐藏右键菜单

函数原型:

```
void ShowContextMenu(BOOL bShow)
```

参数:

bShow        -        是否显示右键菜单

说明:

bShow 的值为 TRUE，则显示右键菜单，反之，则隐藏右键菜单。

## 13) \*ShowFormFieldsMessageBar

显示或隐藏表单域信息提示栏。

函数原型:

```
Void            ShowFormFieldsMessageBar(BOOL bShow)
```

参数:

bShow        -        如果值为 FALSE，则信息提示栏不显示；如果值为 TRUE，  
                         则信息提示栏显示。

返回值:

无

# 10. 打印

## 1) PrintWithDialog

打印 PDF 文档时弹出打印对话框。

函数原型:

```
Void            PrintWithDialog();
```

参数:

无

返回值:

无

## 2) OpenMemFileForPrinter

直接对内存中的 PDF 文件进行打印，但不显示该 PDF 文件。

函数原型:

```
IPDFPrinter* OpenMemFileForPrinter(long buffer, long size)
```

参数:

Buffer - 内存中 PDF 文件的数据。

Size - 缓存中数据的长度。

返回值:

函数返回一个 [IPDFPrinter](#) 类的实例，以便对打印机进行控制。

### 3) OpenFileForPrinter

直接打开一个 PDF 文件并进行打印，但不显示该 PDF 文件。

函数原型:

```
IPDFPrinter* OpenFileForPrinter(BSTR file_path)
```

参数:

file\_path - PDF 文件的路径（包括文件的扩展名）

返回值:

函数返回一个 [IPDFPrinter](#) 类的实例，以便对打印机进行控制。

## 11. 搜索

### 1) FindFirst

在 PDF 文档中搜索一个字符串。如果 PDF 页面包含搜索的字符串，则跳转到那个页面，并自动更新 CurPage 属性值，高亮页面上的字符串，返回值为 TRUE，反之返回值为 False。

函数原型:

```
BOOL FindFirst (BSTR search_string, BOOL bMatchCase, BOOL  
bMatchWholeWord)
```

参数:

SearchString - 要搜索的字符串

BMatchCase - 是否区分大小写

BMatchWholeWord - 是否匹配整个字符串

返回值:

如果文档中存在搜索的字符串，则返回非零值，反之，返回 0。

### 2) FindFirstEx

**FindFirst** 方法的扩展。提供在 PDF 文档中进行搜索的接口。

函数原型:

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

参数:

|                        |             |
|------------------------|-------------|
| <b>Search_string</b>   | - 要搜索的字符串   |
| <b>bMatchCase</b>      | - 是否区分大小写   |
| <b>bMatchWholeWord</b> | - 是否匹配整个字符串 |

返回值:

如果文档中存在搜索的字符串，则返回非零值，反之，返回 0。

### 3) FindNext

在整个 PDF 文档中搜索下一个与给定字符串相匹配的文本内容。如果找到下一个，则直接跳转到那个页面，自动更新 **CurPage** 属性值，并高亮页面中匹配的内容，函数返回值为 **True**。反之，函数返回值为 **False**。需要注意的是，**FindNext** 的搜索条件与 **FindFirst** 方法相同，也包括是否区分大小写，是否匹配整个字符串。如果函数参数值为 **true**，则向文档尾部方向继续搜索，如果函数参数值为 **false**，则向文档开始方向进行搜索。

函数原型:

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

参数:

|                    |                                                            |
|--------------------|------------------------------------------------------------|
| <b>bSearchDown</b> | - 搜索方向。设置为 <b>True</b> ，向文档尾部方向；设置为 <b>False</b> ，向文档开始方向。 |
|--------------------|------------------------------------------------------------|

返回值:

如果找到下一个匹配的内容，则返回值为非零，反之返回值等于零。

### 4) FindFileFirst

在一个 PDF 文件中搜索指定的字符串，如果文件中包含指定的字符串则返回一个 **IFindResult** 接口的实例，否则反之值为 **NULL**。该方法允许在不打开文件的情况下直接进行搜索。例如，想要在一个目录中对所有的 PDF 文件进行关键字搜索，只需循环遍历目录中的每一个 PDF 文件，对每一个文件的内容进行关键字搜索即可。如果在一个 PDF 文件中找到与之相匹配的内容，函数的返回值是一个包含相关细节的 **IFindResult** 接口类的示例。然后，调用 **GoToSearchResult** 打开文件，自动跳转到相应的页面并高亮页面中匹配的内容。

函数原型:

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

参数:

|                 |   |             |
|-----------------|---|-------------|
| file_path       | - | PDF 文件的完整路径 |
| search_string   | - | 想要搜索的字符串    |
| bmatchCase      | - | 是否区分大小写     |
| BMatchWholeWord | - | 是否全字匹配      |

返回值:

如果找到匹配的内容, 则返回一个 [IFindResult](#) 接口类实例, 否则返回 NULL。

## 5) FindFileFirstEx

对 FindFileFirst 方法的扩展。

函数原型:

```
IFindResult * FindFileFirstEx(BSTR file_path, BSTR password, VARIANT  
search_string, boolean bMatchCase, boolean bMatchWholeWord);
```

参数:

|                 |   |             |
|-----------------|---|-------------|
| file_path       | - | PDF 文件的完整路径 |
| Password        | - | PDF 文件的密码   |
| search_string   | - | 想要搜索的字符串    |
| bMatchCase      | - | 是否区分大小写     |
| BMatchWholeWord | - | 是否全字匹配.     |

返回值:

如果找到匹配的内容, 则返回一个 IFindResult 接口类实例, 否则返回 NULL。

## 6) FindFileNext

按照 FindFileFirst 指定的搜索条件, 继续搜索并定位到下一个符合条件的结果。

函数原型:

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

参数:

无

返回值:

如果找到下一个符合条件的结果, 则返回一个 IFindResult 接口类实例, 否则返回 NULL。

## 7) GoToSearchResult

显示并高亮搜索结果。

函数原型:

```
Void GoToSearchResult (IFindResult* findresult);
```

参数:

Findresult - 一个从 [FindFileFirst](#) 或 [FindFileNext](#) 返回的 [IFindResult](#) 接口类的实例

返回值:

无

## 8) FindPageNext

搜索下一个实例。

函数原型:

```
IFindResult* FindPageNext()
```

参数:

无

返回值:

成功则返回 [IFindResult](#) 接口类实例，反之则返回 NULL。

备注:

首次搜索时调用 [FindPageFirst](#) 函数，然后调用 [FindPageNext\(\)](#)接收下一个搜索结果。

## 9) FindClose

释放 FindMemFileFirst 当中分配的内存。

函数原型:

```
Void FindClose()
```

## 10) FindPageFirst

在页面中搜索第一个实例。

函数原型:

```
IFindResult* FindPageFirst(long nPageIndex, BSTR search_string, BOOL  
bMatchCase, BOOL bMatchWholeWord)
```

参数:

- nPageIndex - PDF 页面索引，起始页为 0。
- Search\_string - 所要查找的字符串
- bMatchCase - 搜索字符串是否区分大小写
- bMatchWholeWord - 搜索字符串是否全词匹配

返回值:

成功则返回 IFindResult 接口类实例，反之则返回 NULL。

## 11) FindMemFileFirst

在内存当中查找文件

函数原型:

```
IFindResult* FindMemFileFirst(VARIANT buffer, long fileSize, BSTR password,  
BSTR search_string, BOOL bMatchCase, BOOL bMatchWholeWord)
```

参数:

- Buffer - PDF 的文件数据
- fileSize - PDF 的文件大小
- Password - PDF 的文件打开密码
- Search\_string - 所要查找的字符串
- bMatchCase - 匹配大小写
- bMatchWholeWord - 匹配完整关键字

## 12) \*SearchAndHighlightAllTextOnPage

在指定的页面上搜索关键字，并高亮所有符合条件的结果。

函数原型:

```
Void SearchAndHighlightAllTextOnPage(BSTR searchstring, BOOL  
bMatchCase, BOOL bMatchWholeWord, long pageNo);
```

参数:

- PageNo - 要搜索的页面

返回值:

无

## 12. 书签

### 1) GetOutlineFirstChild

获取大纲目录树中某个节点的第一个子节点。

函数原型:

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

参数:

Outline - 大纲目录树中的某一个节点。如果想获取大纲目录树的根节点，则该参数为 NULL。

返回值:

如果该节点存在子节点，则返回第一个子节点，反之则返回值为空。

### 2) GetOutlineNextSibling

获取大纲目录树中某个节点的下一个兄弟节点。

函数原型:

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

参数:

Outline - 大纲目录树中的某一个节点。

返回值:

如果该节点存在下一个兄弟节点，则返回该兄弟节点，反之返回值为空。

## 13. 安全

### 1) GetDocPermissions

获取 PDF 文件的权限控制标识。

函数原型:

```
long GetDocPermissions ()
```

参数:

无

返回值:

返回值是一个 32 位的整型数，详情请参照 PDF 规范。如果文档不存在权限控制，则返回值是 0xffffffff。

### 2) \* CheckOwnerPassword

判断文档的所有者密码（文档修改密码）是否正确。

函数原型:

**BOOL CheckOwnerPassword(BSTR lpszPermPsw)**

参数:

**lpszPermPsw** - 输入文档的所有者密码（文档修改密码）用来进行校验

返回值:

成功则返回 **True**，反之返回 **False**。

### 3) \*SetOwnerPassword

为当前的 PDF 文件设置所有者密码（文档修改密码）。

函数原型:

**BOOL SetOwnerPassword(LPCTSTR lpszNewValue)**

参数:

**lpszNewValue** - 设置的新的所有者密码（文档修改密码）字符串

返回值:

成功则返回 **True**，反之返回 **False**。

### 4) \*SetUserPermission

设置当前 PDF 文件的权限控制。

函数原型:

**BOOL SetUserPermission(long dwPermission)**

参数:

**dwPermission** - 权限控制标识

返回值:

成功则返回 **True**，反之返回 **False**。

### 5) \*SetUserPassword

为当前的 PDF 文件设置打开密码。

函数原型:

**BOOL SetUserPassword(LPCTSTR lpszNewValue)**

参数:

**lpszNewValue** - 设置的新的打开密码字符串

返回值:

成功则返回 **True**，反之返回 **False**。

## 14. 注释

### 1) \*ExportAnnotsToFDFFile

将批注从当前打开的文档中导出，批注数据保存为一个独立的 FDF 文件。

函数原型:

```
BOOL ExportAnnotsToFDFFile(BSTR FDFFileName)
```

参数:

FDFFileName - 导出的批注数据 FDF 文件的路径

返回值:

成功则返回 TRUE，反之则返回 FALSE。

### 2) \*ImportAnnotsFromFDFFile

将批注数据从 FDF 文件导入到当前打开的文档中。

函数原型:

```
BOOL ImportAnnotsFromFDFFile(BSTR FDFFileName)
```

参数:

FDFFileName - 进行批注数据导入的源 FDF 文件路径

返回值:

成功则返回 TRUE，反之则返回 FALSE。

### 3) \*SetBDrawAnnot

设定在显示 PDF 页面时，批注是否显示

函数原型:

```
Void SetBDrawAnnot(BOOL bDrawAnnot);
```

参数:

bDrawAnnot - 参数值为 true 表示显示批注。  
参数值为 false 表示隐藏批注。

返回值:

无

### 4) \*ShowAllPopup

显示所有批注的弹出框。

函数原型:

Void ShowAllPopup (BOOL bShow);

参数:

bShow - 参数值为 true 时, 显示所有批注的弹出框。

返回值:

无

## 5) ^GetPageAnnots

获取指定页面中的注释信息。

函数原型:

IPDFPageAnnots GetPageAnnots(long pageIndex)

参数:

pageIndex - 指定 PDF 页面的页码以获取其注释信息

返回值:

成功则返回 [IPDFPageAnnots](#) 对象, 该对象中包含页面所有的注释信息; 反之则返回 NULL。

## 6) ^GetFormatTool

获取 FormatTool 的句柄。

函数原型:

IPDFFormatTool GetFormatTool ()

参数:

无

返回值:

如果成功, 返回值为 [FormatTool](#) 的句柄。失败则返回值为 NULL。

# 15. 表单

## 1) \*ExportFormToFDFFile

将 PDF 表单数据从当前打开的文档中导出, 并保存为一个独立的 FDF 文件。

函数原型:

BOOL ExportFormToFDFFile (BSTR FDFFileName)

参数:

FDFFileName - 导出的 PDF 表单数据形成的 FDF 文件的路径。

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 2) \*ImportFormFromFDFFile

将 PDF 表单数据从 FDF 文件导入到当前打开的文档中。

函数原型:

```
BOOL ImportFormFromFDFFile(BSTR FDFFileName)
```

参数:

FDFFileName - 要导入到当前 PDF 文档的源表单数据 FDF 文件路径。

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 3) \*FindFormFieldsTextFirst

在表单域中进行文本搜索

函数原型:

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

参数:

Searchstring - 想要搜索的字符串

bMatchCase - 是否区分大小写

返回值:

函数返回值指明是否存在相同的字符串。

## 4) \*FindFormFieldsTextNext

在表单域中继续进行文本搜索

函数原型:

```
BOOL FindFormFieldsTextNext ()
```

参数:

无

返回值:

函数返回值指明是否依然存在相同的字符串。

## 5) \* SubmitForm

将 PDF 表单提交到指定的位置。

函数原型:

```
BOOL SubmitForm(BSTR csDestination)
```

参数:

csDestination - 指定的 URL 地址

返回值:

执行成功则返回 TRUE，反之返回 FALSE。

## 6) #GetCurrentForm

获取一个操作表单接口的句柄。只有通过表单接口才能实现添加 PDF 表单元素的功能。

函数原型:

IPDFForm\* GetCurrentForm()

参数:

无

返回值:

执行成功则返回表单接口的句柄，反之返回值等于 NULL。

# 16. 页面组织

## 1) \*InsertPage

在当前 PDF 文件中插入指定页面。

函数原型:

boolean InsertPage(long nInsertAt, BSTR lpszPDFFileName, BSTR  
lpszPssword, BSTR lpszPageRangeString);

参数:

nInsertAt - 插入新 PDF 页面的位置-在指定的当前 PDF 页面的前面

lpszPDFFileName - 待插入页面所在的 PDF 文件的路径

lpszPssword - 源 PDF 文件密码

lpszPageRangeString- 源 PDF 文件中指定插入页面的页码范围

返回值:

成功则返回 True，反之则返回 False。

## 2) \*DeletePage

删除当前 PDF 文件中的页面。

函数原型:

```
boolean DeletePage(long pageIndex, long count);
```

参数:

|           |   |             |
|-----------|---|-------------|
| pageIndex | - | 待删除页面的起始页页码 |
| count     | - | 待删除页面的总页数   |

返回值:

成功则返回 `True`，反之则返回 `False`。

### 3) \*SwapPage

交换当前 PDF 文件的两个页面的位置。

函数原型:

```
boolean SwapPage(long pageIndex1, long pageIndex2);
```

参数:

|            |   |       |
|------------|---|-------|
| pageIndex1 | - | 交换的页码 |
| pageIndex2 | - | 交换的页码 |

返回值:

成功则返回 `True`，反之则返回 `False`。

### 4) \*SetPageIndex

为当前 PDF 文档中指定的页面设置一个新的页码，即改变其位置。

函数原型:

```
boolean SetPageIndex (long pageOldIndex, long pageNewIndex);
```

参数:

|              |   |      |
|--------------|---|------|
| pageOldIndex | - | 原页面号 |
| pageNewIndex | - | 新页面号 |

返回值:

成功则返回 `True`，反之则返回 `False`。

### 5) \*SetPageCropBox

通过设置剪切区域，裁剪当前 PDF 文件中指定的页面。

函数原型:

```
boolean SetPageCropBox(long pageIndex, float left, float top, float right, float bottom);
```

参数:

|           |   |                |
|-----------|---|----------------|
| pageIndex | - | 指定剪切的 PDF 页的页码 |
| left      | - | 页面左上角的水平坐标     |
| top       | - | 页面左上角的垂直坐标     |
| right     | - | 页面右下角的水平坐标     |
| bottom    | - | 页面右下角的垂直坐标     |

返回值:

成功则返回 True，反之则返回 False。

## 6) \*SetPageRotate

旋转当前 PDF 文件中的指定页面。

函数原型:

```
boolean SetPageRotate(long pageIndex, long rotate);
```

参数:

|           |   |                |
|-----------|---|----------------|
| pageIndex | - | 指定旋转的 PDF 页的页码 |
| rotate    | - | 设置顺时针旋转的角度     |

返回值:

成功则返回 True，反之则返回 False。

## 7) \*FlattenPage

页面扁平化

函数原型:

```
boolean FlattenPage(long pageIndex);
```

参数:

|           |   |    |
|-----------|---|----|
| pageIndex | - | 页码 |
|-----------|---|----|

返回值:

成功则返回 True，反之则返回 False。

## 8) \*InsertNewPage

在当前 PDF 文件的指定位置插入空白页。

函数原型:

```
Boolean InsertNewPage(long nPageIndex, long nPageWidth, long nPageHeight);
```

参数:

- nPageIndex - 待插入的新空白 PDF 页的页码
- nPageWidth - 待插入的新空白 PDF 页的页面宽度
- nPageHeight - 待插入的新空白 PDF 页的页面高度

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 9) \*ExportPagesToPDF

将当前 PDF 文档中的指定页面导出为一个新的 PDF 文件。

函数原型:

```
BOOL ExportPagesToPDF (BSTR lpszPDFFileName, BSTR lpszPageRangeString);
```

参数:

- lpszPDFFileName - 新的 PDF 文件的路径
- lpszPageRangeString - 指定页码范围，格式包括 "0, 2, 3-5"，不能指定为逆序，例如"5-2"。也就是说，破折号前面的值不能大于破折号后面的值。

返回值:

返回中指明指定的页面是否导出成功。

## 17. JavaScript

### 1) \* RunJScript

运行 JavaScript 脚本。

函数原型:

```
void RunJScript(LPCTSTR csJS)
```

参数:

- csJS - 指定待运行的 JavaScript 脚本

## 2) \*ShowDocJsDialog

弹出文档 Javascript 对话框

函数原型:

Void ShowDocJsDialog();

参数:

无

返回值:

无

## 3) \*ShowJsConsoleDialog

弹出 Javascript 控制台对话框。

函数原型:

Void ShowJsConsoleDialog();

参数:

无

返回值:

无

# 18. 多实例

## 1) SetCurrentWnd

当 ActiveX 在多实例情况下，通过该函数可设置当前的实例。

函数原型:

Void SetCurrentWnd(long hWnd);

参数:

hWnd - OCX 实例的窗口句柄

返回值:

无

## 2) GetCurrentWnd

获取当前窗口的句柄。

函数原型:

Long GetCurrentWnd();

参数:

无

返回值:

成功则返回当前窗口句柄，反之返回值等于 **NULL**。

### 3) GetCtrlInstance

获取控件实例句柄。

函数原型:

```
Long GetCtrlInstance ();
```

参数:

无

返回值:

成功则返回控件实例句柄，反之返回 **NULL**。

## 19. 签名

### 1) &GetPDFSignatureMgr

函数原型:

```
IPDFSignatureMgr* GetPDFSignatureMgr();
```

返回值:

返回 [PDFSignatureMgr](#) 类型。

### 2) &GetLastSigModuleError

获取签章接口最后返回的错误索引

函数原型:

```
long GetLastSigModuleError()
```

返回值:

签章接口最后返回的错误索引

|   |   |      |
|---|---|------|
| 0 | - | 成功   |
| 1 | - | 参数错误 |
| 2 | - | 状态错误 |
| 3 | - | 执行错误 |

4 - 证书不存在

### 3) &GetLastSigModuleErrMsg

获取签章接口最后返回的错误提示

函数原型:

```
BSTR GetLastSigModuleErrMsg()
```

返回值:

签章接口最后返回的错误提示。

## 20. 其他

### 1) \*AddImageObject

将图片插入到文档中。

函数原型:

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

参数:

|             |   |                             |
|-------------|---|-----------------------------|
| nPageIndex  | - | 要插入图片的页码                    |
| left        | - | 插入图片的水平坐标，坐标原点是左下角          |
| bottom      | - | 插入图片的垂直坐标，坐标原点是左下角          |
| width       | - | 图片在 PDF 文档中的显示宽度            |
| height      | - | 图片在 PDF 文档中的显示高度            |
| BmpFileName | - | 待插入图片的路径                    |
| alpha       | - | 一个 0 到 255 之间的数字，表示图片显示的透明度 |
| rotate      | - | 图片的旋转度数                     |

返回值:

函数返回值指明图片插入是否成功。

### 2) \*GetBitmap

将 PDF 的页面渲染为一张图片。

函数原型:

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

参数:

|             |   |                   |
|-------------|---|-------------------|
| nPageIndex  | - | 指定进行渲染的页面页码       |
| pixelWidth  | - | 形成的图片的宽度          |
| pixelHeight | - | 形成的图片的高度          |
| rectLeft    | - | 设备坐标系中显示区域左边的像素位置 |
| rectTop     | - | 设备坐标系中显示区域上方的像素位置 |
| rectRight   | - | 设备坐标系中显示区域右边的像素位置 |
| rectBottom  | - | 设备坐标系中显示区域下方的像素位置 |
| PixelFormat | - | 形成的图片的像素格式        |

返回值:

返回值为图片的句柄。

### 3) \*AddWaterMark

将一个文本型水印插入到文档中。

函数原型:

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

参数:

|           |   |                                                            |
|-----------|---|------------------------------------------------------------|
| Page      | - | 要插入水印的页码                                                   |
| String    | - | 水印的内容                                                      |
| Left      | - | 插入水印的水平坐标                                                  |
| Bottom    | - | 插入水印的垂直坐标                                                  |
| Fontsize  | - | 水印文本的字体大小                                                  |
| Fontcolor | - | 水印文本的字体颜色                                                  |
| Textmode  | - | 值只能是 0、1、2<br>0: 表示填充文本<br>1: 表示绘制文本边框<br>2: 表示同时绘制边框和填充文本 |
| alpha     | - | 一个 0 到 255 之间的数字, 表示透明度                                    |
| rotate    | - | 水印旋转的角度                                                    |

返回值:

函数返回值指明添加水印是否成功。

### 4) \*GetBarcodeBitmap

获得二维码图片

函数原型:

```
long GetBarcodeBitmap(BSTR contents, short format, long moduleHSize,
long moduleVSize, short ecLevel);
```

参数:

|              |   |                 |
|--------------|---|-----------------|
| Contents     | - | 需要编码的内容         |
| Format       | - | 编码的格式可以为以下几种:   |
| UNSPECIFY    | = | -1,             |
| CODABAR      | = | 0               |
| CODE_39      | = | 1               |
| CODE_93      | = | 2               |
| CODE_128     | = | 3               |
| EAN_8,       | = | 4               |
| UPC_A,       | = | 5               |
| EAN_13,      | = | 6               |
| ITF,         | = | 7               |
| PDF_417,     | = | 8               |
| RSS_14,      | = | 9               |
| RSS_EXPANDED | = | 10              |
| UPC_E,       | = | 11              |
| DATAMATRIX   | = | 12              |
| QR_CODE      | = | 13              |
| moduleHSize  | - | 模型的单位宽度         |
| moduleVSize  | - | 模型的单位高度         |
| ecLevel      | - | 只对 QR_Code 类型有效 |
| ECLEVEL_L    | = | 0               |
| ECLEVEL_M    | = | 1               |
| ECLEVEL_Q    | = | 2               |
| ECLEVEL_H    | = | 3               |

返回值:

成功则返回二维码的图像，反之则返回 NULL。

## 5) \*ConvertPDFPageToImage

将指定 PDF 页面转换成 JPG 图片。

函数原型:

```
BOOL ConvertPDFPageToImage(LPCTSTR pdfFile, long lnPageIndex, LPCTSTR
imageFilePath, long lnImageWidth, long lnImageHeight)
```

参数:

|               |            |
|---------------|------------|
| pdfFile       | -PDF 文档路径  |
| lnPageIndex   | -指定页面索引    |
| imageFilePath | -图像文件的保存路径 |
| lnImageWidth  | -转换的图片宽度   |
| lnImageHeight | -转换的图片高度   |

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 6) \*ImportImageToPdf

在指定的 PDF 文件中导入图片。

函数原型:

```
BOOL ImportImageToPdf(LPCTSTR pdfFilePath, long index, LPCTSTR  
imageFilePath)
```

参数:

|               |                     |
|---------------|---------------------|
| pdfFilePath   | - PDF 文件的路径         |
| index         | - PDF 文件中要导入图片文件的页码 |
| imageFilePath | - 图片文件的路径           |

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 7) SetContextMenuString

设置右键菜单项的字符名称

函数原型:

```
void SetContextMenuString(BSTR string);
```

参数:

|        |              |
|--------|--------------|
| string | - 右键菜单项的字符名称 |
|--------|--------------|

返回值:

无

## 事件

### 1) BeforeDraw

在开始绘制显示内容之前触发的事件。

函数原型:

Void BeforeDraw (long dc)

参数:

dc - 设备上下文相关的句柄

### 2) AfterDraw

在完成显示内容绘制之后触发的事件。

函数原型:

Void AfterDraw (long dc)

参数:

dc - 设备上下文相关的句柄

### 3) OnZoomChange

当 Zoomlevel 属性发生变化的时候触发该事件。

函数原型:

Void OnZoomChange ()

参数:

无

### 4) OnPageChange

当显示的页面发生变化时触发该事件。比如翻页、页面跳转等操作都会触发该事件。

函数原型:

Void OnPageChange ()

参数:

无

### 5) OnOpenPassword

当试图打开一个包含密码的 PDF 文件时触发该事件。

函数原型:

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

参数:

Password - PDF 文件的密码  
Cancel - 当该参数设置为 False 时, 该事件将一直被触发, 直到密码输入正确为止

## 6) OnSearchProgress

当搜索文档时触发该事件。

函数原型:

```
Void OnSearchProgress (long pageNumber, long pageCount)
```

参数:

pageNumber - 当前正在搜索的页码  
pageCount - 总页码数

## 7) OnOpenFile

当文件打开失败时触发该事件。

函数原型:

```
void OnOpenFile(short Error);
```

参数:

Error - 返回错误代码

## 8) OnOpenDocument

当打开一个 PDF 文件时触发该事件。

函数原型:

```
Void      OnOpenDocument (BSTR filepath)
```

参数:

Filepath - 被打开的 PDF 文件的路径

## 9) OnFilePathInvalidate

当打开文件路径错误时触发该事件。

函数原型:

```
void OnFilePathInvalidate(BSTR WarnString);
```

参数:

WarnString - 返回错误提示信息

## 10) OnShowSavePrompt

当关闭一个已经被修改过的 PDF 文件时触发该事件。

函数原型:

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

参数:

- bShow - 该参数指明是否在 ActiveX 控件中显示默认的提示框。
- nResult - 该参数指明是要保存被修改过的 PDF 文件，默认不保存。

## 11) OnCloseDocument

当关闭一个 PDF 文件时触发该事件。

函数原型:

```
Void OnCloseDocument (BSTR filepath)
```

参数:

- Filepath - 要关闭的 PDF 文件的路径。

## 12) OnDocumentChange

当 PDF 文档内容发生改变时触发该事件。

函数原型:

```
Void OnDocumentChange ()
```

参数:

无

## 13) CustomFileGetSize

当以 [OpenCustomFile](#) 方式打开 PDF 文档时触发该事件。

函数原型:

```
Void CustomFileGetSize (long* size)
```

参数:

- size - [out] 该参数指向一个数值，用于设置 PDF 文件的长度。

## 14) CustomFileGetBlock

当以 [OpenCustomFile](#) 方式打开 PDF 文档时触发该事件。

函数原型:

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

参数:

pos - [in ] 相对于文件起始位置的偏移位置, 单位为字节  
pBuf - [out ]指向一个用于接收 PDF 数据的缓冲  
Size - [in] 缓冲区的大小

备注:

从指定的位置开始获取数据。以字节为单位, 从文件起始位置开始进行偏移。  
偏移量和缓冲区的大小不能超出 PDF 文件的长度。

## 15) OnClick

在鼠标左键点击时触发该事件。

函数原型:

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

参数:

hWnd - 鼠标左键点击时的窗口句柄  
ClientX - 鼠标左键在 ActiveX 控件窗口客户区点击时水平坐标  
ClientY - 鼠标左键在 ActiveX 控件窗口客户区点击时垂直坐标

## 16) OnDbClick

在鼠标左键双击时触发该事件。

函数原型:

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

参数:

hWnd - 鼠标左键双击时的窗口句柄  
ClientX - 鼠标左键在 ActiveX 控件窗口客户区双击时水平坐标。  
ClientY - 鼠标左键在 ActiveX 控件窗口客户区双击时垂直坐标。

## 17) OnRButtonClick

在鼠标右键双击时触发该事件。

函数原型:

void OnRButtonClick(long hWnd, long ClientX, long ClientY);

参数:

hWnd - 鼠标右键双击时的窗口句柄  
ClientX - 鼠标右键在 ActiveX 控件窗口客户区双击时水平坐标。

ClientY - 鼠标右键在 ActiveX 控件窗口客户区双击时垂直坐标。

## 18) OnDownloadFinish

从网络中下载 PDF 文件结束时触发该事件。

函数原型:

```
void OnDownloadFinish();
```

参数:

无

## 19) OnErrorOccurred

在调用 SDK 提供的接口出现异常时触发该事件。目前仅支持 [GetToolByIndex](#) 和 [ShowToolbarButton](#) 两个接口。

函数原型:

```
void OnErrorOccurred(BSTR lpszErrorMsg)
```

参数:

lpszErrorMsg - 异常提示字符串

返回值:

无

## 20) OnUploadFinish

在调用 [UploadCurFileToFTP](#) 时触发该事件。

函数原型:

```
void OnUploadFinish(short nRetCode)
```

参数:

nRetCode - 返回操作结果

返回值:

无

## 21) OnTextHyperLink

当在 ActiveX 控件点击操作文本链接时触发该事件。该事件传递的参数包括该链接和是否对该链接进行响应。

函数原型:

```
void OnTextHyperLink(BSTR csUrl, boolean* cancel)
```

参数:

csUrl - 文本链接的 URL 地址

cancel - 是否响应链接

返回值:

无

## 22) OnExcuteMenuItem

通过 [OnAddmenuItemAction](#) 事件添加的用户自定义菜单项被执行时触发该事件。

函数原型:

```
void OnExcuteMenuItem(BSTR sMenuItem, boolean* bResult)
```

参数:

sMenuItem - 用户自定义的菜单项

bResult - 操作产生的结果 (TRUE 或者 FALSE)

返回值:

无

## 23) \*OnDoGoToRAction

当执行 GoToR (超链接类型) 操作时触发该事件。

函数原型:

```
Void OnDoGoToRAction(BSTR sFilePath, Link_Dest* dest)
```

参数:

sFilePath - 目标文件的路径

dest - 文件中的目标对象

返回值:

无

## 24) \*OnHyperLink

点击超文本时触发该事件。

函数原型:

```
Void OnHyperLink(BSTR linktype, BSTR linkdata, Link_Dest* dest,  
BOOL* cancel)
```

参数:

**Linktype** - 包含超链接类型信息的字符串。

超链接类型字符串包括：

**GoTo** : 跳转至当前文档的相应页面，**linkdata** 为 **None**，**dest** 包含控件导览的位置信息。

**GoToR**: 跳转到本地磁盘中保存的相应 **PDF** 文件。若要在新的窗口中显示文档，则 **linkdata** 中在文件名后加 **1**，反之 **linkdata** 中在文件名后加 **0**。

**Dest** 包含控件导览的位置信息。

**Launch**: 执行外部程序，若要在新的窗口中打开文档，则 **linkdata** 中在文件名后加 **1**，反之则在文件名后加 **0**。

**URI**: 打开 **URI**，**linkdata** 包含 **URI** 字符串。

**Cancel** - **cancel** 变量的值为 **true**，则控件不打开超链接。

**linkData** - 附加信息字符串

## 25) \*OnContextMenuIndex

在鼠标右键弹出的菜单中点击某一项时触发该事件。与

[SetContextMenuString](#) 函数配合使用。

函数原型:

```
void OnContextMenuIndex(short nIndex);
```

参数:

**nIndex** - 菜单的索引值

## 26) \*OnAddMenuItemAction

当执行“添加一个菜单项”操作时触发该事件。

函数原型:

```
void OnAddMenuItemAction(BSTR* pMenuItem)
```

参数:

**pMenuItem** - 菜单项的内容

返回值:

无

## 27) #FormFieldError

在设置 **PDF** 表单域时出现异常会触发该事件。

函数原型:

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

参数:

---

`nErrorCode` - 返回异常错误码

## ILink\_Dest

通过 ILink\_Dest 所提供的接口可以获取链接所关联的 PDF 页面的相关信息。

### 方法

#### 1) GetPageIndex

获取 PDF 页面索引值。

函数原型:

```
long GetPageIndex();
```

返回值:

页面索引值

#### 2) GetZoomMode

获取页面显示模式。

函数原型:

```
long GetZoomMode();
```

返回值:

返回值定义如下:

- 2    -   适合页面
- 3    -   适合宽度
- 4    -   适合宽度

#### 3) GetZoomParamCount

获取缩放参数。

函数原型:

```
long GetZoomParamCount()
```

返回值:

返回缩放参数值。

#### 4) GetZoomParam

获取第 nIndex 个参数的内容

函数原型:

```
double GetZoomParam(long nIndex)
```

返回值:

返回第 nIndex 参数的内容，该内容描述的是缩放的信息。

#### 5) GetDestName

获取目标对象名称。

函数原型:

```
BSTR GetDestName();
```

返回值:

返回目标对象名称。

# IPDFAction

通过 IPDFAction 所提供的接口可以获取 Action 的信息。

## 方法

### 1) GetURIPath

获取 Action 所关联的 URI 路径。

函数原型:

```
BSTR GetURIPath();
```

返回值:

返回 URI 路径。

### 2) GetFilePath

获取 Action 所关联的外部文件的路径。

函数原型:

```
BSTR GetFilePath();
```

返回值:

返回外部文件路径。

### 3) GetType

获取 Action 的类别。

函数原型:

```
int GetType();
```

返回值:

返回值以下几个值:

- 1 - Go to Action;
- 2 - Remote Go-To Actions;
- 4 - Launch Actions;
- 6 - URI Actions;

#### 4) GetDest

获取 Action 所关联的 dest

函数原型:

```
llink_Dest* GetDest();
```

返回值:

返回跳转定义。

# IPDFOutline

通过 IPDFOutline 所提供的接口可以获取 PDF 文件的书签的相关信息。

## 方法

### 1) GetOutlineDest

获取书签的链接目标。

函数原型:

```
ILink_Dest* GetOutlineDest ();
```

返回值:

返回书签的跳转目标。

### 2) GetOutlineAction

获取书签所关联的动作。

函数原型:

```
IPDF_Action* GetOutlineAction ();
```

返回值:

返回书签的跳转动作。

### 3) GetOutlineColor

获取书签文本的颜色（RGB）。

函数原型:

```
DWORD GetOutlineColor ();
```

返回值:

返回书签文本的颜色。

### 4) NavigateOutline

遍历指定的大纲结点所包含的子节点的数据。

函数原型:

```
void NavigateOutline ();
```

参数:

无  
返回值:  
无

## 5) GetOutlineTitle

获取大纲结点的标题。

函数原型:

**BSTR**      **GetOutlineTitle()**

参数:

无

返回值:

返回当前大纲结点的标题。

## 6) GetOutLineTitle2

获取大纲节点的标题，作为一个变量。

函数原型:

**VARIANT**      **GetOutLineTitle2 ()**

参数:

无

返回值:

返回大纲节点的标题，作为一个变量。

## 7) GetOutlineExpandValue

获取大纲子节点的数量。

函数原型:

**Long**      **GetOutlineExpandValue()**

返回值:

返回子节点的数量。

# IFindResult

通过 IFindResult 所提供的接口可以执行搜索操作。

## 方法

### 1) GetFindRectsCount

获取查找的矩形框个数

函数原型:

```
long    GetFindRectsCount();
```

返回值:

成功则返回非负整数，反之则返回 0。

### 2) GetFindRectByIndex

获取查找矩形框

函数原型:

```
Long GetFindRectByIndex
```

```
(long nIndex, float* left, float* bottom, float* right, float* top);
```

参数:

|        |   |                       |
|--------|---|-----------------------|
| nIndex | - | 查找到的区域的索引             |
| left   | - | 查找到的区域的左边坐标（PDF 坐标系统） |
| bottom | - | 查找到的区域的底部坐标（PDF 坐标系统） |
| right  | - | 查找到的区域的右边坐标（PDF 坐标系统） |
| top    | - | 查找到的区域的顶端坐标（PDF 坐标系统） |

返回值:

成功则返回 True，反之则返回 False。

### 3) GetFindPageNum

获取搜索结果的页面索引信息。

函数原型:

```
long    GetFindPageNum();
```

参数:

无

返回值:

返回页面索引值。

#### 4) GetFindFileName

获取执行搜索操作的文件名。

函数原型:

```
BSTR GetFindFileName();
```

参数列表:

空

返回值:

函数返回的是当前 `FindResult` 类包含的搜索结果所对应的文件名。

#### 5) GetFindString

获取搜索结果内容。

函数原型:

```
BSTR GetFindString()
```

参数:

无

返回值:

返回搜索结果内容。

# IPDFPrinter

通过 IPDFPrinter 所提供的接口可以控制打印机和执行打印操作。

## 属性

### 1) PrinterName

类型:

BSTR

描述:

设置将要被执行打印操作的打印机名称。

### 2) PrinterRangeMode

描述:

设置打印范围，可设置如下选项：

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

### 3) PrinterRangeFrom

类型:

short

描述:

设定开始打印的页码。

该属性仅在 [PrinterRangeMode](#) = PRINT\_RANGE\_SELECTED 时有效。

### 4) PrinterRangeTo

类型:

short

描述:

设定打印结束的页码。

该属性仅在 [PrinterRangeMode](#) = PRINT\_RANGE\_SELECTED 时有效。

## 5) NumOfCopies

类型:

short

描述:

设定文档打印的份数。

## 6) Scaling

类型:

short

描述:

在打印对话框中设置打印缩放比例。

## 7) AutoRotate

类型:

boolean

描述:

设置打印对话框中自动旋转的参数。

## 8) AutoCenter

类型:

boolean

描述:

设置打印时是否自动居中。

1 = 自动居中

0 = 不要自动居中。

## 9) Collate

类型:

boolean

描述:

设置是否检查“自动分页”选项。

1 = 检查“自动分页”选项

2 = 不检查“自动分页”选项

## 10) Rotation

类型:

short

描述:

设置打印时是否旋转文档。

## 11) RangeSubset

类型:

short

描述:

设置打印时是否包含子集。

## 12) ReversePage

类型:

boolean

描述:

设置是否以逆序方式打印文档。

## 13) PageBorder

类型:

boolean

描述:

设置是否打印页面边框。

# 方法

## 1) PrintWithDialog

执行打印操作过程中，弹出与打印相关的对话框。

函数原型:

```
void PrintWithDialog();
```

参数:

无

返回值:

无

## 2) PrintQuiet

执行静默打印操作。即在执行打印操作时，不弹出任何与打印相关的对话框。

函数原型:

```
void PrintQuiet();
```

参数:

无

返回值:

无

## 3) SetPaperSize

设置打印机的页面尺寸，具体的参数值请参照 Windows SDK 说明文档。

函数原型:

```
void SetPaperSize (long paperSize);
```

参数:

paperSize - Windows SDK 说明文档规定的值。

返回值:

无

## 4) GetSystemPrinterCount

获取系统打印机的数量。

函数原型:

```
Long GetSystemPrinterCount();
```

返回值:

返回系统中打印机的数量。

## 5) GetSystemPrinterNameByIndex

通过索引获取系统中打印机的名称。

函数原型:

```
BSTR GetSystemPrinterNameByIndex(long index);
```

返回值:

返回通过索引获得的系统打印的名称。

## 6) SetPaperSizeByPage

在静默打印和正常打印过程中勾选“根据 PDF 页面大小选择打印纸张”选项。

函数原型:

```
void SetPaperSizeByPage(BOOL bPage);
```

参数:

**bPage:** 值为 TRUE 则勾选“根据 PDF 页面大小选择打印纸张”选项,值为 FALSE 则不勾选此选项。

# IPDFDocumentInfo

本章节介绍了 PDF 文档属性的相关信息。

## 属性

### 1) Author

类型:

BSTR

描述:

指明文档的作者。

### 2) Subject

类型:

BSTR

描述:

指明文档的主题。

### 3) CreatedDate

类型:

BSTR

描述:

指明文档创建的时间。

### 4) ModifiedDate

类型:

BSTR

描述:

指明文档最后一次被修改的时间。

## 5) Keywords

类型:

BSTR

描述:

指明用于描述文档的的关键词。

## 6) Creator

类型:

BSTR

描述:

指明文档的创建者。

## 7) Producer

类型:

BSTR

描述:

指明创建文档的工具或软件。

## 8) Title

类型:

BSTR

描述:

指明文档的标题。

# #IPDFForm

## 方法

### 1) #AddField

添加一个表单域

函数原型:

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

参数:

|               |   |          |
|---------------|---|----------|
| bstrFieldName | - | 表单全名     |
| bstrFieldType | - | 新建表单域的类型 |

表单域类型包括:

文本域  
按钮  
组合框  
列表框  
复选框  
单选按钮

|           |   |                      |
|-----------|---|----------------------|
| pageIndex | - | 添加表单域的页码（页码从 0 开始计算） |
| Left      | - | 表单域左边的坐标             |
| Top       | - | 表单域顶部的坐标             |
| right     | - | 表单域右边的坐标             |
| Bottom    | - | 表单域底部的坐标             |

返回值:

返回新创建的表单。

备注:

表单域的坐标是在旋转后的页面中测量的，也就是说，无论页面怎么旋转，表单左下角的坐标都是[0,0]。

### 2) #RemoveFieldsByName

删除一个表单域

函数原型:

```
Void RemoveFieldsByName (LPCTSTR bstrFieldName);
```

参数:

**bstrFieldName** - 待删除表单的全名，若表单中含有多个子注释，则所有子注释都将会被删除；若系统中含有多个同名的表单，则所有同名的表单都将会被删除。

返回值:

无

### 3) #ExportToFDF

导出特定表单元素并保存为 FDF 文件。

函数原型:

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

参数:

**bstrFullPath** - 导出后生成的 FDF 文件的路径  
**bEmptyFields** - 值为 True 则仅导出 VARIANT & arrField 函数指定的表单元素。值为 False 则导出除 VARIANT& arrFields 函数指定的元素外的所有表单元素。  
**ArrFields** - 待导出的表单元素数组。

返回值:

无

### 4) #ImportFromFDF

从 FDF 文件导入表单内容。

函数原型:

```
Void      ImportFormFromFDF(LPCTSTR bstrFullPath);
```

参数:

**bstrFullPath** - 待导入的 FDF 文件路径。

返回值:

无

### 5) #GetFieldByIndex

获取特定表单域的指针。

函数原型:

```
LPDISPATCH GetFieldByIndex(long index)
```

参数:

index - 表单域索引

返回值:

成功则返回表单域指针，反之则返回 `NULL`。

## 6) #GetFieldsCount

获取表单域的总数量。

函数原型:

long GetFieldsCount()

参数:

无

返回值:

调用成功，返回表单域的总数量值，否则返回值“-1”。

## 7) #RemoveField

删掉指定表单域。

函数原型:

Void RemoveField(IPDFFormField pFormField)

参数:

pFormField - 输入待删除的表单域

返回值:

无

# #IPDFFormField

## 属性

### 1) #Alignment

类型:

String

描述:

对齐文本框中的文本（左对齐、居中对齐、右对齐）

提示:

仅用于文本域。

### 2) #BorderStyle

类型:

String

描述:

表单域边框类型

备注:

可用于所有表单类型。

边框类型包括:

- 实线
- 虚线
- 斜线
- 插入线
- 下划线

### 3) #BorderWidth

类型:

short

描述:

表单域的宽度

备注:

可用于所有表单类型

#### 4) #ButtonLayout

类型:

short

描述:

按钮的外观。有效值包括:

- 0 – 仅为文本; 按钮有文字说明, 但没有图标。
- 1 – 仅为图标; 按钮有图标, 但没有文字说明。
- 2 – 在文本上显示图标; 按钮图标显示在文字说明的顶部。
- 3 – 在图标上显示文本; 按钮的文本显示在图标的顶部。
- 4 – 先显示图标, 然后显示文本; 按钮的图标显示在文字说明的左边。
- 5 – 先显示文本, 然后显示图标; 按钮的图标显示在文字说明的右边。
- 6 – 在图标下显示文本; 按钮的文本显示在图标顶部的下方。

备注:

仅用于按钮类型

#### 5) #CalcOrderIndex

类型:

short

描述:

优化列阵中当前表单域的索引

备注:

可用于所有的表单类型

#### 6) #CharLimit

类型:

short

描述:

文本域的字符数限制

备注:

仅用于文本域类型

#### 7) #DefaultValue

类型:

String

描述:

表单域的默认值

备注:

可用于所有的表单类型

## 8) #IsEditable

类型:

Boolean

描述:

Combo Box 是否可编辑

备注:

仅用于 Combo Box

## 9) #Behavior

类型:

String

备注:

None, Invert, Outline, Push

N (None) 无高亮显示

I (Invert) 转化注释内容

O (Outline) 转化注释边框

P (Push) 在页面文字的下方显示注释

## 10) #IsHidden

类型:

Boolean

描述:

表单域是否隐藏;

备注:

可用于所有的表单类型

## 11) #IsMultiline

类型:

Boolean

描述:

文本域是单行浏览或者多行浏览。

备注:

仅用于文本域

## 12) #IsPassword

类型:

Boolean

描述:

密码以明文或者密文方式输入。

备注:

仅用于文本域。

## 13) #IsReadOnly

类型:

Boolean

描述:

表单域是否只读。

备注:

可用于所有的表单类型。

## 14) #IsRequired

类型:

Boolean

描述:

是否将表单域设置为非空

备注:

可用于组合框、单选按钮和文本框

## 15) #Name

类型:

String

描述:

当前选中的表单域的名称

备注:

只读，可用于所有表单类型

## 16) #NoViewFlag

类型:

Boolean

描述:

是否显示表单元素。“1”表示隐藏表单元素；“0”表示显示表单元素。

备注:

可用于所有表单类型

## 17) #PrintFlag

类型:

Boolean

描述:

打印内容时是否包含表单元素。“1”表示包含表单元素；“0”表示不包含表单元素。

备注:

可用于所有表单类型

## 18) #Style

类型:

CString

描述:

设置复选框和单选按钮的形状:

- 1) 格子形
- 2) 十字形
- 3) 菱形
- 4) 圆形
- 5) 星形
- 6) 方形

备注:

可用于复选框和单选按钮

## 19) #TextFont

类型:

String

描述:

字体（Reference 1.7 416）

字体可设置为:

Courier

Courier-Bold

Courier-Oblique

Courier-BoldOblique

Helvetica

Helvetica-Bold

Helvetica-Oblique

Helvetica-BoldOblique

Symbol

Times-Roman

Times-Bold

Times-Italic

Times-BoldItalic

ZapfDingbats

备注:

可用于除复选框和单选按钮以外的所有表单

## 20) #TextSize

类型:

Short

描述:

表单域中文本的大小

备注:

可用于除复选框和单选按钮外的所有表单

## 21) #Type

类型:

String

描述:

表单类型

备注:

可用于所有表单类型

表单类型可设置为: 文本域、按钮、组合框、列表框、复选框、单选按钮

## 22) #Value

类型:

String

描述:

当前值

备注:

以下表单形式包含这种属性:

- 1) 文本
- 2) 组合框
- 3) 单选按钮
- 4) 复选框 <是&否>
- 5) 列表框

## 23) #Tooltip

类型:

String

描述:

显示提示信息

备注:

可用于所有表单类型

## 24) #Orientation

类型:

Short

描述:

表单文本的旋转方向

备注:

可用于所有表单类型

## 25) #DirtyFlag

描述:

表单域被改动后设置为 `true`，没改动过为 `false`。

## 方法

### 1) #PopulateListOrComboBox

给列表框或组合框中的条目分配数值

函数原型:

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

参数:

- `arrItem`            -     一个字符串数组，每个元素代表一个项目名称。
- `arrExportVal`    -     一个字符串数组，大小与第一个参数相同，每个元素代表一个导出值。

返回值:

无

### 2) #SetBackgroundColor

设置表单域的背景颜色

函数原型:

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

参数:

`bstrColorSpace`    -     可设置的顏色包括:

透明、灰色、RGB 或 CMYK 色域

使用 T、G、RGB 和 CMYK 分别代表以上四种颜色。

设置为 T 和 G 时，需要设置 `redC`;

设置为 RGB 时，需要设置 `redC`、`greenM` 和 `blueY`;

设置为 CMYK 时，需要设置 `redC`、`greenM`、`blueY` 和 `AlphaK`。`redC`、`greenM` 和 `blueY` 值的范围为 0-1。

返回值:

无

### 3) #SetBorderColor

设置边框颜色。

函数原型:

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

参数:

bstrColorSpace - 可设置的颜色包括:

透明、灰色、RGB 或 CMYK 色域

使用 T、G、RGB 和 CMYK 分别代表以上四种颜色。

设置为 T 和 G 时, 需要设置 redC;

设置为 RGB 时, 需要设置 redC、greenM 和 blueY;

设置为 CMYK 时, 需要设置 redC、greenM、blueY 和 AlphaK。redC、greenM 和 blueY 值的范围为 0-1。

返回值:

无

#### 4) #SetForegroundColor

设置前背景色

函数原型:

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

参数:

bstrColorSpace - 可设置的颜色包括:

透明、灰色、RGB 或 CMYK 色域

使用 T、G、RGB 和 CMYK 分别代表以上四种颜色。

设置为 T 和 G 时, 需要设置 redC;

设置为 RGB 时, 需要设置 redC、greenM 和 blueY;

设置为 CMYK 时, 需要设置 redC、greenM、blueY 和 AlphaK。redC、greenM 和 blueY 值的范围为 0-1。

返回值:

无

#### 5) #SetButtonCaption

对按钮上显示的文本进行设置

函数原型:

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

参数:

bstrFace - 一个字符串, 指定说明文字的朝向

有效的字符串包括:

N — 普通

D — 向下

R — 反转

bstrCaption - 按钮文字说明

返回值:

无

## 6) #SetButtonIcon

设置按钮图标

函数原型:

```
Void SetButtonIcon (LPCTSTR bstrFace, LPCTSTR bstrFilePath);
```

参数:

bstrFace 一个字符串, 指定说明文字的朝向

有效的字符串包括:

N = 普通

D = 向下

R = 反转

BstrFilePath: 用作图标的图片文件的路径。

返回值:

无

## 7) #SetExportValues

导出单选按钮和复选框时, 根据不同的选择 (如: 已选、未选、已勾选、未勾选等) 设置导出值。

函数原型:

```
void SetExportValues (const VARIANT& arrExportVal);
```

参数:

arrExportVal - 导出值数组

返回值:

无

## 8) #SetJavaScriptAction

设置 Javascript 脚本动作

函数原型:

```
Void SetJavaScriptAction (LPCTSTR bstrTrigger, LPCTSTR bstrJavaScript);
```

参数:

bstrTrigger - 一个字符串, 指定动作触发的条件

有效的字符串包括:

up

down

enter

exit

calculate

validate

format

keystroke

bstrJavaScript - 脚本动作

返回值:

无

## 9) #SetResetFormAction

设置重置表单域动作

函数原型:

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

参数:

bstrTrigger - 一个字符串, 指定动作触发的条件

有效的字符串包括:

Up = 鼠标向上移动

down = 鼠标向下移动

enter = 鼠标输入

exit = 鼠标退出

bFlags - 一组定义动作特性的标记集

arrFields - 待导出的表单元素组

返回值:

无

## 10) #SetSubmitFormAction

设置提交表单域的动作

函数原型:

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

参数:

bstrTrigger - 一个字符串, 指定动作触发的条件

有效的字符串包括:

up = 鼠标向上移动

down = 鼠标向下移动

enter = 鼠标输入

exit = 鼠标退出

bstrURL - 含有 URL 的字符串

bFlags - 一组定义动作特性的标记集

arrFields - 待提交的表单元素组

返回值:

无

## 11) #GetPageIndex

获取指定表单域的页码

函数原型:

```
long GetPageIndex()
```

参数:

无

返回值:

返回表单域的页码

## 12) #GetRectTop

获取指定表单域的顶部的坐标

函数原型:

```
float GetRectTop(I)
```

参数:

无

返回值:

返回表单域顶边的坐标

## 13) #GetRectLeft

获取表单域左边的坐标

函数原型:

`float GetRectLeft()`

参数:

无

返回值:

返回表单域左边的坐标

#### **14) #GetRectRight**

获取指定表单域的右边的坐标

函数原型:

`float GetRectRight()`

参数:

无

返回值:

返回表单域右边的坐标

#### **15) #GetRectBottom**

获取指定表单域底边的坐标

函数原型:

`float GetRectBottom()`

参数:

无

返回值:

返回表单域底边的坐标

# ^IPDFPageAnnots

## 方法

### 1) ^GetAnnot

获取特定 PDF 注释的指针

函数原型:

```
CPDFAnnot GetAnnot(long AnnotIndex)
```

参数:

AnnotIndex : 该 PDF 注释的索引。

返回值:

成功则获取特定 PDF 注释的指针，反之则返回 NULL。

### 2) ^AddAnnot

添加一个新的 PDF 注释。

函数原型:

```
CPDFAnnot AddAnnot (LPDISPATCH AnnotToAddAfter, LPCTSTR SubType, float left, float top, float right, float bottom)
```

参数:

AnnotToAddAfter : 保留。设为 NULL。

SubType : 注释的子类型。

子类型包括:

"Cloudy", "Arrow", "Line", "Square", "Rectangle", "Circle", "Ellipse", "Polygon",  
"PolyLine", "Pencil", "Underline", "Highlight", "Squiggly", "StrikeOut",  
"Replace", "Caret", "Note", "Typewriter", "Callout", "Textbox",  
"FileAttachment", "Image", "Movie", "Sound", "Rectangle Link", "Quadrilateral Link".

Left : 该注释矩形框的左边坐标

Top : 该注释矩形框的顶边坐标

Right : 该注释矩形框的右边坐标

**Bottom:** 该注释矩形框的底边坐标

所有坐标均基于 PDF 坐标空间。

返回值:

成功则返回新的注释对象，反之则返回 **NULL**。

### 3) ^RemoveAnnot

删除特定 PDF 注释

函数原型:

**LongRemoveAnnot(LPDISPATCH AnnotToRemove)**

参数:

**AnnotToRemove :** 待删除 PDF 注释的信息

返回值:

成功则返回 0，反之则返回-1。

### 4) ^GetAnnotIndex

获取特定 PDF 注释的索引

函数原型:

**LongGetAnnotIndex(LPDISPATCH Annot)**

参数:

**Annot :** 特定 PDF 注释的信息

返回值:

成功则返回特定 PDF 注释的索引，反之则返回-1。

### 5) ^GetAnnotsCount

获取当前 PDF 页面上 PDF 注释的总数量

函数原型:

**LongGetAnnotsCount()**

参数:

无

返回值:

---

返回 PDF 注释的数量

# ^IPDFAnnot

## 属性

### 1) ^Thickness

类型:

Short 读/写

描述:

PDF 注释边框的宽度

备注:

用于图形标注、测量工具以及其他具有边框的注释（即云形、箭头、线条、方形、矩形、圆形、椭圆形、多边形、折线、铅笔工具、注释框、文本框、链接等）。

### 2) ^BorderStyle

类型:

Short 读/写

描述:

注释边框的类型。

备注:

边框类型包括七种:

- |   |        |   |                         |
|---|--------|---|-------------------------|
| 1 | 实线     | - | 适用于各种类型的注释              |
| 2 | 虚线类型 1 | - | 仅适用于图形标注                |
| 3 | 虚线类型 2 | - | 仅适用于图形标注                |
| 4 | 虚线类型 3 | - | 仅适用于图形标注                |
| 5 | 虚线类型 4 | - | 仅适用于图形标注                |
| 6 | 云形类型 1 | - | 仅适用于云形、多边形、圆形、方形及矩形等注释。 |
| 7 | 云形类型 2 | - | 仅适用于云形、多边形、圆形、方形及矩形等注释。 |

不支持的注释有：

铅笔注释、下划线、高亮、波浪形、删除线、替换标记、备注、打字机注释、文件附件、图片、视频和音频。

### 3) ^Color

类型：

OLE\_COLOR 读/写

描述：

PDF 注释的背景色。

备注：

可用于所有注释

### 4) ^LineStartingStyle

类型：

Short 读/写

描述：

线条的起画格式

备注：

用于线条、箭头、折线及注释框等注释。

包括以下 10 种格式：

- 0 无
- 1 方形
- 2 圆形
- 3 菱形
- 4 张开
- 5 闭合
- 6 对接
- 7 反向张开
- 8 反向闭合
- 9 斜线

## 5) ^LineEndingStyle

类型:

Short 读/写

描述:

线条的结束格式

备注:

用于线条、箭头和折线等注释

包括 10 种类型:

- 0 无
- 1 方形
- 2 圆形
- 3 菱形
- 4 张开
- 5 闭合
- 6 对接
- 7 反向张开
- 8 反向闭合
- 9 斜线

## 6) ^FillColor

类型:

OLE\_COLOR 读/写

描述:

PDF 注释的填充色。

备注:

用于云形、箭头、线条、方形、矩形、圆圈、椭圆、多边形和折线等注释。

## 7) ^Opacity

类型:

Short 读/写

描述:

PDF 注释的透明度值

备注:

不支持链接、视频和音频等注释

## 8) ^Author

类型:

BSTR 读/写

描述:

PDF 注释的作者

备注:

不支持图片、视频、音频和链接等注释

## 9) ^Subject

类型:

BSTR 读/写

描述:

PDF 注释的主题

备注:

不支持图片、视频、音频和链接等注释

## 10) ^CreationDate

类型:

DATE 只读

备注:

PDF 注释的创建时间

## 11) ^ModificationDate

类型:

DATE 只读

描述:

PDF 注释的最后修改日期

## 12) ^Locked

类型:

Boolean 读/写

描述:

标记注释是否被锁定

备注:

可用于所有注释

## 13) ^Print

类型:

Boolean 读/写

描述:

标记是否可以打印注释

备注:

适用于所有注释

## 14) ^ReadOnly

类型:

Boolean 读/写

描述:

标记注释是否为只读的

备注:

适用于所有注释

## 15) ^Description

类型:

BSTR 读/写

描述:

PDF 注释的描述

备注:

仅适用于文件附件注释

## 方法

### 1) ^GetType

获取 PDF 注释的类型

函数原型:

BSTR GetType()

参数:

无

返回值:

返回 PDF 注释的类型

### 2) ^GetSubType

获取 PDF 注释的子类型

函数原型:

BSTR GetSubType()

参数:

无

返回值:

返回 PDF 注释的子类型

### 3) ^GetContents

获取 PDF 注释的内容

函数原型:

`BSTR GetContents()`

参数:

无

返回值:

返回 PDF 注释的内容

### 4) ^SetContents

设置 PDF 注释的内容

函数原型:

`long SetContents(LPCTSTR Contents)`

参数:

`Contents:` 需要设置的内容

返回值:

成功则返回 0，反之则返回-1。

### 5) ^IsPopupOpen

设定是否打开弹出框

函数原型:

`BOOL IsPopupOpened()`

参数:

无

返回值:

打开弹出框打开则返回 `TRUE`，反之则返回 `FALSE`。

### 6) ^SetPopupOpen

设置打开弹出框

函数原型:

Long SetPopupOpened(BOOL Open)

参数:

Open: 所输入的值用来设定是否打开弹出框

返回值:

成功则返回 0，反之则返回-1。

## 7) ^HasPopup

指出注释是否含有弹出框

函数原型:

BOOL HasPopup()

参数:

无

返回值:

注释含有弹出框则返回 TRUE，反之则返回 FALSE。

## 8) ^GetRect

获取注释的矩形区域

函数原型:

Long GetRect(float\* pLeft, float\* pTop, float\* pRight, float\* pBottom)

参数:

pLeft: 输出值，用以接收注释矩形区域左边的坐标

pTop: 输出值，用以接收注释矩形区域顶边的坐标

pRight: 输出值，用以接收注释矩形区域右边的坐标

pBottom: 输出值，用以接收注释矩形区域底边的坐标

返回值:

成功则返回 0，反之则返回-1。

## 9) ^SetRect

设置注释的矩形区域

函数原型:

```
long SetRect(float Left, float Top, float Right, float Bottom)
```

参数:

Left: 输入值, 用以设置注释矩形区域左边的坐标

Top: 输入值, 用以设置注释矩形区域顶边的坐标

Right: 输入值, 用以设置注释矩形区域右边的坐标

Bottom: 输入值, 用以设置注释矩形区域底边的坐标

返回值:

成功则返回 0, 反之则返回-1。

## 10) ^SetLinkGoToAction

设置链接注释的跳转动作

函数原型:

```
Void SetLinkGoToAction(long nPageIndex, float left, float top, float zoom)
```

参数:

nPageIndex: 页码的输入值

left,top : Windows 左上角的位置

zoom: 页面的放大系数

返回值:

无

## 11) ^SetLinkURLAction

设置链接注释的 URL 动作

函数原型:

```
Void SetLinkURLAction(LPCTSTR sURL)
```

参数:

sURL: 动作执行时跳转的位置

返回值:

无

## 12) ^DoAction

执行链接注释的动作（若有）

函数原型：

```
long DoAction()
```

参数：

无

返回值：

成功则返回 0，反之则返回-1。

## 13) ^HasAction

指出链接注释是否含有动作

函数原型：

```
BOOL HasAction()
```

参数：

无

返回值：

如果链接注释含有动作则返回 TRUE，反之则返回 FALSE。

## 14) ^GetMarkedState

获取注释的标记状态

函数原型：

```
Long GetMarkedState()
```

参数：

无

返回值：

未标记时返回值为 0，标记时返回值为 1，出现错误时返回值为-1。

## 15) ^SetMarkedState

设置注释的标记状态（不支持图片、视频、音频和链接等注释）

函数原型:

```
long SetMarkedState(long state)
```

参数:

**state:** 输入值, 用以设置注释的标记状态。

可能的值为 0 (未标记) 或 1 (标记)。

返回值:

成功则返回 0, 反之则返回-1。

## 16) ^GetReviewState

获取注释的审阅状态

函数原型:

```
long GetReviewState()
```

参数:

无

返回值:

返回值可以为 0、1、2、3 和 4, 分别表示 NULL、接受、拒绝、取消和完成。

出现错误则返回值为-1。

## 17) ^SetReviewState

设置注释的审阅状态 (不支持图片、视频、音频和链接等注释)

函数原型:

```
long SetReviewState(long state)
```

参数:

**State:** 输入值, 用以设置注释的审阅状态。该值可以为 0、1、2 和 4,

每个值指定下列相应的审阅状态:

0 = NULL

1 = 接受

2 = 拒绝

3 = 取消

4 = 完成

返回值:

成功则返回 0，反之则返回-1。

## 18) ^GetMigrationState

获取注释的迁移状态

函数原型：

```
long GetMigrationState()
```

参数：

无

返回值：

下列不同的返回值表示注释不同的迁移状态：

0= NULL

1= 未确认

2= 已确认

-1= 其他

## 19) ^SetMigrationState

设置注释的迁移状态（不支持图片、视频、音频和链接等注释）

函数原型：

```
long SetMigrationState(long state)
```

参数：

state: 输入值，用以指定注释的迁移状态。该值可以为 0、1 或 2，分别指定下列状态：

0 = NULL

1 = 未确认

2 = 已确认

返回值：

成功则返回 0，反之则返回-1。

## 20) ^SetStartingPoint

对于线条和箭头注释，设置它们的起点。

函数原型:

```
long SetStartingPoint(float PointX, float PointY)
```

参数:

PointX: 输入值, 用以设置起点的 X 坐标

PointY: 输入值, 用以设置起点的 Y 坐标

返回值:

成功则返回 0, 反之则返回-1。

## 21) ^GetStartingPoint

对于线条和箭头注释, 获取这些注释的起点。

函数原型:

```
long GetStartingPoint(float* PointX, float* PointY)
```

参数:

PointX: 输出值, 用以接收起点的 X 坐标

PointY: 输出值, 用以接收起点的 Y 坐标

返回值:

成功则返回 0, 反之则返回-1。

## 22) ^SetEndingPoint

设置注释的终点。仅适用于线条和箭头注释。

函数原型:

```
long SetEndingPoint(float PointX, float PointY)
```

参数:

PointX: 输入值, 用以设置终点的 X 坐标

PointY: 输入值, 用以设置终点的 Y 坐标

返回值:

成功则返回 0, 反之则返回-1。

## 23) ^GetEndingPoint

获得注释的终点。仅适用于线条和箭头注释。

函数原型:

```
Long GetEndingPoint(float* PointX, float* PointY)
```

参数:

PointX: 输出值, 用以设置终点的 X 坐标

PointY: 输出值, 用以设置终点的 Y 坐标

返回值:

成功则返回 0, 反之则返回-1。

## 24) ^SetMediaPoster

为含海报的注释（如影像注释和音像注释）设置海报（图片）。

函数原型:

```
long SetMediaPoster(LPCTSTR ImageFilePath)
```

参数:

ImageFilePath: 输入图片文件的路径

返回值:

成功则返回 0, 反之则返回-1。

## 25) ^SetMultimedia

设置支持多媒体注释的多媒体内容

函数原型:

```
Long SetMultimedia (LPCTSTR FilePath, LPCTSTR ContentType, BOOL Embed,  
BOOL bShowCtrlBar)
```

参数:

FilePath: 媒体文件路径

ContentType: 媒体数据的 MIME 类型

Embed : 表明是否在 PDF 文件中嵌入媒体

bShowCtrlBar: 表明是否显示控制条

返回值:

成功则返回 0, 反之则返回-1。

## 26) ^SetLinkQuadPoints

设置链接注释的外观

函数原型:

```
long SetLinkQuadPoints(long* PointsArray, long PointsCount)
```

参数:

PointsArray: 点阵

PointsCount: 点阵的大小应为 4

返回值:

成功则返回 0, 反之则返回-1。

## 27) ^SetPolygonVertices

设置多边形注释的外观

函数原型:

```
long SetPolygonVertices(long* PointsArray, long PointsCount)
```

参数:

PointsArray 点阵

PointsCount 点阵的大小

返回值:

成功则返回 0, 反之则返回-1。

## 28) ^SetPencilVertices

设置铅笔注释的外观

函数原型:

```
long SetPencilVertices(long* PointsArray, long PointsCount)
```

参数:

PointsArray 一组线条集, 每条线代表一个点阵

PointCount 点阵的大小

返回值:

成功则返回 0, 反之则返回-1。

## 29) ^AttachFile

为附件注释指定添加的文件

函数原型:

`long AttachFile(LPCTSTR FileName)`

参数:

`FileName`      输入文件的路径

返回值:

成功则返回 0，反之则返回-1。

### 30) ^ GetReplyList

获取注释回复

函数原型:

`IPDFAnnotReplyList* GetReplyList()`

返回值:

成功则返回注释回复，若无注释回复，则返回 NULL。

### 31) ^ UpdateAnnotReplies

更新注释回复

函数原型:

`Void UpdateAnnotReplies(IPDFAnnotReplyList* pReplies)`

参数:

`pReplies`    -    待更新的注释回复

备注:

更新注释的所有回复

## 事件

### 1) ^OnAnnotCreated

创建注释后触发该事件，该事件由第三方完成。

函数原型:

`void OnAnnotCreated(long pageIndex, long annotIndex)`

参数:

`pageIndex`    -    页面索引

annotIndex - 注释索引

## 2) ^OnAnnotDeleted

删除注释后触发该事件，该事件由第三方完成。

函数原型:

```
void OnAnnotDeleted(long pageIndex, long annotIndex)
```

参数:

pageIndex - 页面索引

annotIndex - 注释索引

## 3) ^OnAnnotModified

编辑注释后触发该事件，该事件由第三方完成。

函数原型:

```
void OnAnnotModified(long pageIndex, long annotIndex)
```

参数:

pageIndex - 页面索引

annotIndex - 注释索引

## 4) ^OnAnnotReplyCreated

创建注释回复后触发该事件，该事件由第三方完成

函数原型:

```
void OnAnnotReplyCreated(long pageIndex, long annotindex,  
                          lpctstr replyNM)
```

参数:

pageIndex - 页面索引

annotIndex - 注释索引

pReply - 创建的回复

## 5) ^OnAnnotReplyDeleted

删除注释回复后触发该事件，该事件由第三方完成。

函数原型:

```
void OnAnnotReplyDeleted(long pageIndex,  
long annotindex, CPDFAnnotReply* pReply)
```

参数:

|            |   |          |
|------------|---|----------|
| pageIndex  | - | 页面索引     |
| annotIndex | - | 注释索引     |
| pReply     | - | 待删除的注释回复 |

## 6) ^OnAnnotReplyModified

修改注释回复后触发该事件，该事件由第三方完成。

函数原型:

```
Void OnAnnotReplyModified(long pageIndex,  
long annotindex, CPDFAnnotReply* pReply)
```

参数:

|            |   |          |
|------------|---|----------|
| pageIndex  | - | 页面索引     |
| annotIndex | - | 注释索引     |
| pReply     | - | 所修改的注释回复 |

## 7) ^OnAnnotRButtonDown

右击注释时触发该事件

函数原型:

```
void OnAnnotRButtonDown(IPDFAnnot* Annot, float x, float y,  
BOOL* bDefault);
```

参数:

|          |   |                                  |
|----------|---|----------------------------------|
| Annot    | - | 右击选中的注释                          |
| x        | - | 注释的水平坐标（PDF 坐标系）                 |
| y        | - | 被移动注释的垂直坐标系（PDF 坐标系）             |
| bDefault | - | 值为 True 则启用按钮的默认动作，反之则禁用按钮的默认动作。 |

返回值:

[None]

## 8) ^OnAnnotRButtonUp

释放右键按钮时触发该事件

函数原型:

```
void OnAnnotRButtonUp(IPDFAnnot* Annot, float x, float y, BOOL* bDefault);
```

参数:

|          |   |                                  |
|----------|---|----------------------------------|
| Annot    | - | 右击选中的注释                          |
| x        | - | 被移动注释的水平坐标（PDF 坐标系）              |
| y        | - | 被移动注释的垂直坐标（PDF 坐标系）              |
| bDefault | - | 值为 True 则启用按钮的默认动作，反之则禁用按钮的默认动作。 |

返回值:

[None]

## 9) ^OnAnnotLButtonDbClick

双击注释时触发该事件

函数原型:

```
void OnAnnotLButtonDbClick(IPDFAnnot* Annot, float x, float y,  
BOOL* bDefault);
```

参数:

|          |   |                                  |
|----------|---|----------------------------------|
| Annot    | - | 双击选中的注释                          |
| x        | - | 注释的水平坐标（PDF 坐标系）                 |
| y        | - | 被移动注释的垂直坐标（PD 坐标系）               |
| bDefault | - | 值为 True 则启用按钮的默认动作，反之则禁用按钮的默认动作。 |

返回值:

[None]

## 10) ^OnAnnotMoving

移动注释时触发该事件

函数原型:

```
void OnAnnotMoving(IPDFAnnot* Annot, float x, float y, BOOL* bDefault);
```

参数:

- |          |   |                                  |
|----------|---|----------------------------------|
| Annot    | - | 待移动的注释                           |
| x        | - | 注释的水平坐标（PDF 坐标系）                 |
| y        | - | 被移动注释的垂直坐标（PDF 坐标系）              |
| bDefault | - | 值为 True 则启用按钮的默认动作，反之则禁用按钮的默认动作。 |

返回值:

[None]

### 11) ^OnAnnotMouseEnter

鼠标移至注释时触发该事件

函数原型:

```
void OnAnnotMouseEnter(IPDFAnnot* Annot);
```

参数:

- |       |   |       |
|-------|---|-------|
| Annot | - | 输入的注释 |
|-------|---|-------|

返回值:

[None]

### 12) ^OnAnnotMouseExit

鼠标移出注释时触发该事件

函数原型:

```
Void OnAnnotMouseExit(IPDFAnnot* Annot);
```

参数原型:

- |       |   |         |
|-------|---|---------|
| Annot | - | 鼠标移出的注释 |
|-------|---|---------|

返回值:

[None]

# ^ IPDFAnnotReplyList

## 方法

### 1) ^ GetCount

获取回复的数量

函数原型:

```
long GetCount()
```

返回值:

返回回复的数量

### 2) ^ GetItem

获取指定注释的回复

函数原型:

```
CPDFAnnotReply GetItem(long nIndex)
```

参数:

nIndex            -    回复的索引

返回值:

成功则返回指定的回复，反之则返回 NULL。

### 3) ^ Remove

删除指定的回复

函数原型:

```
void Remove(long nIndex)
```

参数:

nIndex            -    待删除的回复的索引

### 4) ^ RemoveAll

删除回复列表中所有的回复

函数原型:

```
void RemoveAll()
```

## 5) ^ Add

添加新的回复

函数原型:

```
Void Add(LPCTSTR Creator, LPCTSTR Content, DATE CreationDate, long nIndex)
```

参数:

- |              |   |                                    |
|--------------|---|------------------------------------|
| Creator      | - | 回复人                                |
| Content      | - | 回复内容                               |
| CreationDate | - | 回复时间                               |
| nIndex       | - | 添加回复的位置（从 0 开始），若为-1，则在回复列表末尾添加回复。 |

备注:

假设原回复列表中有 **N** 条回复，则 **nIndex** 的范围为[0，N-1]。

# ^IPDFAnnotReply

## 方法

### 1) ^SetCreator

设置注释回复的创建者

函数原型:

```
void SetCreator(LPCTSTR Creator)
```

参数:

Creator - 注释回复的创建者

### 2) ^GetCreator

获取注释回复的创建者

函数原型:

```
BSTR GetCreator()
```

返回值:

返回注释回复的创建者

### 3) ^SetContent

设置注释回复内容

函数原型:

```
void SetContent(LPCTSTR Content)
```

参数:

Content 注释回复内容

### 4) ^GetContent

获取注释回复内容

函数原型:

```
BSTR GetContent()
```

返回值:

返回注释回复的内容

## 5) ^ GetChildren

获取子注释回复

函数原型:

```
IPDFAnnotReplyList* GetChildren()
```

返回值:

成功则返回子注释回复，若无子回复，则返回 **NULL**。

## 6) ^ GetParent

获取父类注释回复

函数原型:

```
CPDFAnnotReply GetParent()
```

返回:

成功则返回父类注释回复，若无子回复，则返回 **NULL**。

## 7) ^ GetCreationDate

获取注释回复的创建时间

函数原型:

```
DATE GetCreationDate()
```

返回值:

返回注释回复的创建时间

## 8) ^ SetCreationDate

设置注释回复的创建时间

函数原型:

```
void SetCreationDate(DATE CreationDate)
```

参数:

CreationDate                    -    注释回复的创建时间

## 9) ^ SetReadOnly

将注释回复设置为只读型

函数原型:

```
void SetReadOnly(BOOL bNewValue)
```

参数

**bNewValue**            值为 TRUE 则设置注释回复为只读型，值为 FALSE 则为读/写型。

## 10) ^ GetReplyID

获取注释回复的唯一 ID

函数原型:

```
long GetReplyID()
```

返回值:

返回注释回复的唯一 ID。

# ^IPDFormatTool

IPDFormatTool 类提供了相应的接口，允许用户在 PDF 应用程序中设置 free text 类型 (FreeTextAnnot) 批注的格式。该接口使用的格式化工具与福昕阅读器注释工具栏中的格式化工具相同。在 ActiveX 控件中 free text 类型的批注分为三种，分别是 Typewriter、Callout 和 TextBox。每一种都有与之对应的格式数据。需要注意的是，在你修改当前 free text 批注的时候，相对应的 CPDFFormatTool 中的的数据也会被修改。IPDFFormatTool 类仅对 free text 类型的批注有效。

## 方法

### 1) ^SetFontName

设置当前字体的名称

函数原型:

```
Void SetFontName(BSTR FontName)
```

参数:

FontName      字体名称

返回值:

无

### 2) ^GetFontName

获取当前使用的字体的名称。

函数原型:

```
BSTR GetFontName()
```

参数:

无

返回值:

返回当前正在使用的字体的名称。

### 3) ^SetFontSize

设置字体的大小。

函数原型:

```
Void SetFontSize(float FontSize)
```

参数:

FontSize            字体大小

返回值:

无

#### 4) ^GetFontSize

获取当前正在使用的字体的字号。

函数原型:

Float GetFontSize.

参数:

无

返回值:

返回当前正在使用的字体的字号。

#### 5) ^SetFontColor

设置当前字体的颜色。

函数原型:

Void SetFontColor(OLE\_COLOR FontColor)

参数:

FontColor        字体颜色

返回值:

无

#### 6) ^GetFontColor

获取字体的颜色。

函数原型:

OLE\_COLOR GetFontColor()

参数:

无

返回值:

返回当前字体的颜色。

## 7) ^SetBorderColor

设置字体边框的颜色。

函数原型:

```
Void SetBorderColor(OLE_COLOR color)
```

参数:

Color            字体边框的颜色。

返回值:

无

## 8) ^GetBorderColor

获取当前字体的边框颜色。

函数原型:

```
OLE_COLOR GetBorderColor()
```

参数:

无

返回值:

返回当前字体的边框颜色。

## 9) ^SetFillColor

设置当前字体的填充颜色。

函数原型:

```
Void SetFillColor(OLE_COLOR FillColor)
```

参数:

FillColor            字体的填充颜色。

返回值:

无

## 10) ^GetFillColor

获取字体的填充颜色。

函数原型:

```
OLE_COLOR GetFillColor()
```

参数:

无

返回值:

返回字体的填充颜色。

### 11) ^SetFontBold

设置文本是否显示为粗体。

函数原型:

```
Void SetFontBold(BOOL FontBold)
```

参数:

FontBold                      表示文本是否显示为粗体。

返回值:

无

备注:

只有在当前字体是 Courier, Helvetica, Times Roman 时, bold 属性和 italic 属性才允许被设置。

### 12) ^GetFontBold

获取文本显示的状态是否为粗体。

函数原型:

```
BOOL GetFontBold()
```

参数:

无

返回值:

如果文本显示状态为粗体, 则返回值等于 TRUE, 反之则等返回 FALSE。

### 13) ^GetFontBoldEnable

指明格式化工具中是否启用了加粗功能

函数原型:

```
BOOL GetFontBoldEnalbe()
```

参数:

无

返回值:

若已启用加粗功能, 则返回 TRUE, 反之则返回 FALSE。

#### 14) ^SetFontItalic

设置文本是否显示为斜体。

函数原型:

```
Void SetFontItalic(BOOL FontItalic)
```

参数:

FontItalic 表示文本是否显示为斜体。TRUE 表示显示为斜体，FALSE 表示不显示斜体。

返回值:

无

#### 15) ^GetFontItalic

获取文本显示的状态是否斜体。

函数原型:

```
BOOL GetFontItalic()
```

参数:

无

返回值:

文本显示为斜体，则返回 TRUE，反之则返回 FALSE。

#### 16) ^GetFontItalicEnable

指明是否可以将字体设置为斜体

函数原型:

```
BOOL GetFontItalicEnable()
```

参数:

无

返回值:

如果文本显示状态为斜体，则返回值等于 TRUE，反之则等返回 FALSE。

#### 17) ^SetAlign

设置文本对齐的方式。

函数原型:

```
Void SetAlign(AlignStyle Style)
```

参数:

|          |               |
|----------|---------------|
| Style    | 文本对齐方式。包括如下选项 |
| ASLEFT   | =0, 左对齐       |
| ASMIDDLE | =1, 居中对齐      |
| ASRIGHT  | =2, 右对齐       |

返回值:

无

## 18) ^GetAlign

获取文本对齐的方式。

函数原型:

AlignStyle GetAlign()

参数:

无

返回值:

返回文本对齐的方式。具体返回值可以是:

|          |          |
|----------|----------|
| ASLEFT   | =0, 左对齐  |
| ASMIDDLE | =1, 居中对齐 |
| ASRIGHT  | =2, 右对齐  |

## 19) ^SetCharSpace

设置字符间距。

函数原型:

Void SetCharSpace(float CharSpace)

参数:

|           |      |
|-----------|------|
| CharSpace | 字符间距 |
|-----------|------|

返回值:

无

## 20) ^GetCharSpace

获取字符间距。

函数原型:

float GetCharSpace()

参数:

无

返回值:

返回字符间距。

## 21) ^SetCharHorzScale

设置字符的水平缩放比率。

函数原型:

```
Void SetCharHorzScale(float CharHorzScale)
```

参数:

CharHorzScale      水平拉伸比率

返回值:

无

## 22) ^GetCharHorzScale

获取字符水平缩放比率。

函数原型:

```
float GetCharHorzScale()
```

参数:

无

返回值:

返回字符水平缩放比率。

# &IPDFSignatureMgr

## 方法

### 1) &Add

添加未签名的签名域

函数原型:

```
IPDFSignatureField* Add(long pageIndex, float left, float top, float right, float bottom);
```

参数:

|           |   |                  |
|-----------|---|------------------|
| pageIndex | - | PDF 中待添加签名域的页面索引 |
| left      | - | 签名矩形区域起始位置的水平坐标  |
| top       | - | 签名矩形区域起始位置的垂直坐标  |
| right     | - | 签名矩形区域结束位置的水平坐标  |
| bottom    | - | 签名矩形区域结束位置的垂直坐标  |

返回值:

新创建的 IPDFSignatureField 对象。

备注:

左下方是 PDF 文件的原始点。

### 2) &SignDocument

对未签名的签名域进行签名，默认签名成功时会关闭源文档，并打开签名后的文档

函数原型:

```
boolean SignDocument(LPDISPATCH pSigField, BSTR signedFilePath, boolean bDefault);
```

参数:

|                |   |                            |
|----------------|---|----------------------------|
| pSigField      | - | 待签名的 IPDFSignatureField 对象 |
| signedFilePath | - | 签名后 PDF 文件的保存路径            |
| bDefault       | - | 是否使用默认签名                   |

返回值:

返回签名是否成功。

备注:

若您使用默认的算法添加签名，则调用该接口来完成签名；若不是使用默认的算法添加签名，则调用以下接口运行第三方签名工具来签署 PDF 文件：

1. 首先，通过 [GetSourceBuffer](#) 和 [GetSourceBufferLen](#) 接口来获取待签署文件的内容流；然后，使用第三方工具添加签名；最后，调用 [CreateSignedDoc](#) 来完成签名。
2. 运用标准算法成功添加签名后，PDF 文件将会重新打开，同时 [IPDFSignatureField](#) 对象也会失效。如果您需要继续操作 [IPDFSignatureField](#) 对象，则需要再次获取该对象。

### 3) &Verify

验证签名域

函数原型:

```
boolean Verify(LPDISPATCH pSigField, boolean bShowStateDialog);
```

参数:

|                   |                                       |
|-------------------|---------------------------------------|
| pSigField:        | 指定待验证的签名域                             |
| bDefaultVerified: | 指明是否采用默认的算法来验证签名域。值为 TRUE 则表示采用默认的算法。 |

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

若您采用默认的算法来添加签名，则调用该接口来完成 PDF 文件验证。

若您采用第三方签名算法来添加签名，则在调用该接口后，调用以下接口来验证 PDF 文件：

首先，通过 [GetSourceBuffer](#) 和 [GetSignedBuffer](#) 接口验证所签署的文件；然后，调用 [SetVerifyResult](#) 接口将验证结果传至 ActiveX；最后，通过 [IPDFSignatureField](#) 的 [GetState](#) 接口获取验证结果。

#### 4) &GetCounts

计算签名个数。

函数原型:

```
long GetCounts();
```

返回值:

签名域的总个数（包括已签名域与未签名域）

#### 5) &Get

通过索引获取签名域

函数原型:

```
LPDISPATCH Get(long index);
```

参数:

index - 通过指定索引获取签名域对象

返回值:

返回 [IPDFSignatureField](#) 对象的索引。

#### 6) &Clear

清空签名域，即把签名的相关信息（包括外观）进行清除，在页面上留下一个未签名的签名域对象

函数原型:

```
boolean Clear(LPDISPATCH pSigField);
```

参数:

pSigField - 待清空的 [IPDFSignatureField](#) 对象

返回值:

返回值指明签名域是否成功清空

#### 7) &Remove

移除未签名的签名域，即删除一个为签名的签名域对象。

函数原型:

```
boolean Remove(LPDISPATCH pSigField);
```

参数:

pSigField - 需要移除的 PDFSignatField 对象

返回值:

返回值指明删除操作是否成功。

## 8) &VerifyAll

验证所有已签名的签名域

函数原型:

```
boolean VerifyAll();
```

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

一般在刚开发文档时要求验证所有签名域对象的情况下使用该接口，这时，在默认情况下，无需弹出状态对话框。

## 9) &InitStraddleValue

使用骑缝签章

函数原型:

```
BOOL InitStraddleValue(IPDFSignatureField* pSigField)
```

参数:

pSigField - 待签名的 IPDFSignatureField 对象

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

调用该接口后，调用正常的签章流程（包括标准签名和第三方签名）来添加骑缝章。

# &IPDFSignatureField

## 属性

### 1) &Reason

类型:

String

描述:

签名原因

操作:

可读可写

### 2) &Location

类型:

String

描述:

签名物理位置

操作:

可读可写

### 3) &Signer

类型:

String

描述:

签名者

操作:

可读可写

#### 4) &Filter

类型:

String

描述:

签名域对象的签名算法 Filter 名称，默认的 Foxit 签名算法 Filter 为 Adobe.PPKLite。

操作:

可读可写

#### 5) &SubFilter

类型:

String

描述:

签名域对象的签名算法 subFilter 名称，默认的 Foxit 签名算法的 SubFilter 为 adbe.pkcs7.detached。

#### 6) &State

类型:

Short

描述:

签章当前状态（如果是默认签名时，该状态由 ActiveX 内部定义，否则，该值由应用定义并用 [SetVerifyResult](#) 传给 ActiveX）

0: 未知签名

1: 验证通过

2: 验证未通过

3: 未签名

操作:

只读

## 方法

### 1) &SetAPOptions

设置客户化外观的外观显示选项值

函数原型:

```
boolean SetAPOptions(long opts);
```

参数:

opts - 客户化外观显示选项设置 (0-511)

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

客户化外观提供了下面的显示选项可供用户选择，需在 `SignDocument` 之前调用。

|              |          |
|--------------|----------|
| 全部不显示        | - 0      |
| 显示全部         | - 511    |
| 显示文本         | - 0x100L |
| 显示图片         | - 0x080L |
| 显示签名者        | - 0x040L |
| 显示位置         | - 0x020L |
| 显示 DN        | - 0x010L |
| 显示时间         | - 0x008L |
| 显示原因         | - 0x004L |
| 显示标签         | - 0x002L |
| 显示 FoxitFlag | - 0x001L |

### 2) &SetAPText

设置签名域 AP 显示的文本

函数原型:

```
boolean SetAPText(BSTR text);
```

参数:

`text` - 设置显示的文本

返回值:

成功则返回 `TRUE`，反之则返回 `FALSE`。

备注:

需要在调用 [SignDocument](#) 之前调用

### 3) `&SetAPIImage`

设置签名域中显示的图片

函数原型:

```
boolean SetAPIImage(BSTR imagePath, boolean bSetMask,  
    OLE_COLOR clrMask);
```

参数:

`imageFilePath` - 指定签名域中图像的路径

`bSetMask` - 该参数数值指定是否给图片背景设置遮罩效果（注：  
图片背景必须纯色的）

`clrMask` - 设置 `Mask` 色值

返回值:

成功则返回 `TRUE`，反之则返回 `FALSE`。

备注:

需在 [SignDocument](#) 之前调用

### 4) `&IsSigned`

判断签名域是否已经被签名

函数原型:

```
boolean IsSigned();
```

返回值:

已签署则返回 `TRUE`，反之则返回 `FALSE`。

## 5) &SetSignerDN

设置证书识别名称

函数原型:

```
boolean SetSignerDN(BSTR dn);
```

参数:

dn -证书识别名称

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

需在 [SignDocument](#) 之前调用

## 6) &SetStatusImage

设置签名状态图章样式

函数原型:

```
boolean SetStatusImage(BSTR imagePath, short sState, short sMode,  
boolean bRotate, boolean bSetMask, OLE_COLOR clrMask);
```

参数:

|           |                                               |
|-----------|-----------------------------------------------|
| imagePath | - 状态图片的路径                                     |
| sState    | - 签章的状态，该值由属性 <b>Status</b> 决定。               |
| sMode     | - 图片显示的模式，0 为默认的图片显示位置（即签章左上角）；1 为图片覆盖整个章区域显示 |
| bRotate   | - 图标是否随着签章对象旋转而旋转（默认为不旋转）。                    |
| bSetMask  | - 是否为图片背景色设置遮罩效果（注：图片背景必须是纯色的）                |
| clrMask   | - 设置 Mask 色值。                                 |

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

设置的图片不会保存至 PDF，只有在 **Signature** 状态改变前设置有效。

## 7) &GetPageIndex

获取签名的页面索引

函数原型:

```
long GetPageIndex();
```

返回值:

获取签名的页面索引

## 8) &GetSourceBuffer

获取待签名或待验证时源文档内容流

函数原型:

```
VARIANT GetSourceBuffer();
```

返回值:

源文档内容流

备注:

- A. 调用该接口前先调用 [SignDocument](#) 或 [Verify](#) 并返回 True。
- B. 内容流存在 VARIVANT 的 parray 域中，vc 中使用 parray->pvData 获得内容流的指针；js 使用 `getArray()` 获取内容流数组。

## 9) &GetSourceBufferLen

获取待签名时或待验证时源文档内容流长度

函数原型:

```
long GetSourceBufferLen();
```

返回值:

内容流长度

备注:

调用该接口前先调用 [SignDocument](#) 或 [Verify](#) 并返回 True。

## 10) &GetSignedBuffer

获取签名后的签名内容流

函数原型:

```
VARIANT GetSignedBuffer();
```

返回值:

签名内容流

备注:

调用该接口前先调用 [Verify](#) 并返回 True。

内容流存在 VARIANT 的 parray 域中，vc 中使用 parray->pvData 获得内容流的指针；js 使用 [getArray\(\)](#) 获取内容流数组。

## 11) &GetSignedBufferLen

获取签名后签名内容流的长度

函数原型:

```
long GetSignedBufferLen();
```

返回值:

返回内容流的长度

备注:

调用该接口前先调用 [Verify](#) 并返回 True。

## 12) &CreateSignedDoc

第三方签名生成所需的签名文档（路径设置在 [SignDocument](#) 接口中）

函数原型:

```
boolean CreateSignedDoc(VARIANT signedBuf, long length);
```

参数:

|           |   |           |
|-----------|---|-----------|
| signedBuf | - | 签名后的内容流   |
| length    | - | 签名后内容流的长度 |

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

调用该接口前需先调用 [SignDocument](#) 设置签名文档保存路径和不使用默认签名。signedBuf 在 vc 使用指针存在 VARIANT 的 pbVal 域中，js 则使用数组对象作为参数。

签名成功后，PDF 会重新打开，[PDFSignatureField](#) 对象会失效，需要重新获取。

### 13) &SetVerifyResult

设置第三方验证结果

函数原型:

```
boolean SetVerifyResult(short sResult);
```

参数:

sResult - 用户自定义验证结果

0: 未知签名

1: 验证通过

2: 验证未通过

3: 未签名

返回值:

成功则返回 TRUE，反之则返回 FALSE。

### 14) &SetCertPath

设置证书内容

函数原型:

```
boolean SetCertPath(BSTR certPath, BSTR pfxPsw);
```

参数:

certPath - 证书路径

pfxPsw - 证书密码

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

pCertData 在 vc 使用指针存在 VARIANT 的 pbVal 域中，js 则使用数组对象作为参数。

## 15) &SetCertData

设置证书内容

函数原型:

```
boolean SetCertData(VARIANT pCertData, long length, BSTR pfxPsw);
```

参数:

|           |   |        |
|-----------|---|--------|
| pCertData | - | 证书内容流  |
| length    | - | 内容流字节数 |
| pfxPsw    | - | 证书密码   |

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

pCertData 在 vc 使用指针存在 VARIANT 的 pbVal 域中，js 则使用数组对象作为参数。

## 16) &SetCertContext

设置证书上下文

函数原型:

```
boolean SetCertContext(long pCertContext);
```

参数:

|              |   |                        |
|--------------|---|------------------------|
| pCertContext | - | 证书上下文类型 PCCERT_CONTEXT |
|--------------|---|------------------------|

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

&SetCertPath/&SetCertData/ &SetCertContext 这三个接口只要设置其中一个，证书就可生效。

## 17) &SetAPIImageData

设置签名域外观显示图像的字节流

函数原型:

```
boolean SetAPIImageData(VARIANT imageDataBuffer, BSTR imageType,
```

```
long dataSize, boolean bSetMask, OLE_COLOR clrMask);
```

参数:

|                 |   |       |
|-----------------|---|-------|
| imageDataBuffer | - | 图片字节流 |
| imageType       | - | 图片类型  |
| dataSize        | - | 字节流长度 |
| bSetMask        | - | 是否遮罩  |
| clrMask         | - | 遮罩颜色  |

返回值:

成功则返回 **TRUE**，反之则返回 **FALSE**。

备注:

pCertData 在 vc 使用指针存在 **VARIANT** 的 pbVal 域中，js 则使用数组对象作为参数。

## 18) &SetStatusImageData

设置签名域外观图片的字节流

函数原型:

```
Boolean SetStatusImageData(VARIANT imageDataBuffer,  
BSTR imageType, long dataSize, short sState, short sMode,  
boolean bRotate, boolean bSetMask, OLE_COLOR clrMask);
```

参数:

|                 |   |                                              |
|-----------------|---|----------------------------------------------|
| imageDataBuffer | - | 图片字节流                                        |
| imageType       | - | 图片类型（可支持的类型包括：bmp、jpg、png 和 gif）             |
| dataSize        | - | 字节流长度                                        |
| sState          | - | 签章状态，该值由属性 <b>Status</b> 决定                  |
| sMode           | - | 图片显示的模式，0 为默认的图片显示位置（即为签章左上角）；1 为图片覆盖整个章区域显示 |
| bRotate         | - | 是否图标随着签章对象旋转而旋转（默认不旋转）                       |
| bSetMask        | - | 是否需要对图片背景色做遮罩（图片背景必须是纯色）                     |
| clrMask         | - | 设置 Mask 色值                                   |

返回值:

成功则返回 **TRUE**，反之则返回 **FALSE**。

备注:

imageDataBuffer 在 vc 使用指针存在 VARIANT 的 pbVal 域中, js 则使用数组对象作为参数。

## 19) &TurnGray

签章显示灰化

函数原型:

```
boolean TurnGray(boolean bGray,boolean bCanModify);
```

参数:

bGray            -    是否灰化  
bCanModify    -    是否可以修改

返回值:

成功则返回 TRUE, 反之则返回 FALSE。

## 20) &TurnBlur

签章显示雾化

函数原型:

```
boolean TurnBlur(boolean bBlur);
```

参数:

bBlur        -    是否雾化

返回值:

成功则返回 TRUE, 反之则返回 FALSE。

备注:

该接口只在控件中改变签章的显示, 未改变 PDF 内容流。

## 21) &SetVisible

签章是否显示

函数原型:

```
boolean SetVisible(boolean bVisible);
```

参数:

bVisible - 是否显示

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 22) &GrayPrint

签章灰化打印

函数原型:

```
boolean GrayPrint(boolean bGrayPrint);
```

参数:

bGrayPrint - 是否灰化打印

返回值:

成功则返回 TRUE，反之则返回 FALSE。

备注:

该接口只在控件中改变签章的显示，未改变 PDF 内容流。

## 23) &SetStraddleType

设置骑缝章的类型

函数原型:

```
BOOL SetStraddleType(short nType)
```

参数:

nType - 数值必须为: 0, 1, 2, 3, 4;

0 指的是中骑缝章;

1 指的是左骑缝章;

2 指的是右骑缝章;

3 指的是上骑缝章;

4 指的是下;

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 24) &SetStraddlePos

设置骑缝章的位置

函数原型:

```
BOOL SetStraddlePos(float fPos)
```

参数:

fPos - 若骑缝章位于顶端或底端，则该参数值为签名域的水平中间点；反之，则为签名域的垂直中间点。

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 25) &SetStraddleBitmap

根据签名状态设置骑缝章的外观图片

函数原型:

```
BOOL SetStraddleBitmap(short nState, BSTR sFilePath)
```

参数

nState -数值必须为： 0, 1, 2, 3;

0 表示未知状态；

1 表示未签署；

2 表示有效；

3 表示无效；

sFilePath - 图片文件的路径

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 26) &SetStraddlePages

设置签名显示的页码范围

函数原型:

```
BOOL SetStraddlePages(BSTR sRange)
```

参数:

sRange -该参数值为跨骑缝章显示的页码范围，如：“0-10” 或

“0-2,3-10”;

返回值:

成功则返回 TRUE，反之则返回 FALSE。

## 27) &SetStraddleFirstPagePercent

设置首个含有骑缝章的 PDF 页面中骑缝章外观的百分比

函数原型:

```
BOOL SetStraddleFirstPagePercent(float fPercent)
```

参数:

fPercent - 该参数值的范围为 0-1

返回值:

成功则返回 TRUE，反之则返回 FALSE。

# 事件

## 1) &OnSetSignatureInfo

单击未签名的签名域或右击选择签名时触发该事件。在该事件中设置 PDFSignatureField 签名必要的属性完成签名。

函数原型:

```
void OnSetSignatureInfo(IPDFSignatureField* pSignature);
```

参数:

pSignature - 单击或右击的签名域对象

## 2) &OnSigning

若对象未签名会继续触发该事件，在该事件中进行默认或第三方签名。

函数原型:

```
void OnSigning(IPDFSignatureField* pSignature);
```

参数:

pSignature - 当前单击的签名域对象

### 3) &OnVerifying

单击已签名对象或右键选择验证时触发该事件，在该事件中进行默认或自定义验证。

函数原型:

```
void OnVerifying(IPDFSignatureField* pSignature);
```

参数:

pSignature - 当前单击的签名域对象

### 4) &OnShowSignaturePropertyDialog

右击签名域对象选择查看签名属性时触发该事件

函数原型:

```
Void OnShowSignaturePropertyDialog (IPDFSignatureField* pSignature,  
  
boolean* bShowProperty);
```

参数:

pSignature - 当前右击的签名域对象  
bShowProperty - 是否弹出签名域的属性对话框

## 联系我们

如有问题，请随时与我们联系，我们将竭诚为您服务。

- **福昕软件（北京）开发有限公司**  
地址：北京市海淀区知春路 56 号中海实业大厦 9 层  
邮编：100098  
电话：010-82111557 010-82116788  
传真：010-82125798
- **福昕软件（福州）开发有限公司**  
地址：福州市铜盘路软件大道 89 号 D 区 23 座  
邮编：350003  
电话：0591-87883901  
技术支持电话：0591-87801729  
传真：0591-87883902
- **福昕软件（台湾）开发有限公司**  
地址：新北市淡水區自強路 9 號 1 樓  
邮编：25162  
电话：00886-2-2809-2969  
传真：00886-2-2808-6566
- **电子邮件：**  
  
销售 – [sales@foxitsoftware.cn](mailto:sales@foxitsoftware.cn)  
市场 – [marketing@foxitsoftware.cn](mailto:marketing@foxitsoftware.cn)  
客服 – [support@foxitsoftware.cn](mailto:support@foxitsoftware.cn)
- **网址：** <http://www.foxitsoftware.cn/>