

## What is zeonpadpdf ?

zeonpadpdf is library for converting word,excel,ppt,publisher,visio,outlook ,images to PDF. It works for both Desktop and Server application. It generates the high quality PDF file that are adobe Acrobat compatible. Supports both 32bit and 64 bit windows os. We don't support multithreaded conversion.

## SYSTEM Requirements

### Operating System

- Microsoft Windows XP
- Microsoft Windows Server 2003 and 2003 R2
- Microsoft Windows Vista
- Microsoft Windows Server 2008 and 2008 R2
- Microsoft Windows 7
- Microsoft Windows 8 and 8.1
- Microsoft Windows Server 2012 and 2012 R2
- 

### MS Office

- Microsoft Office 2003, 2007, 2010, or 2013
- Microsoft Visio 2003, 2007, 2010, or 2013
- Microsoft Publisher 2003, 2007, 2010, or 2013

**Note:** Install the free "Save as PDF or XPS" add-in from Microsoft for office 2007.

### JDK

- JDK 1.5 and higher

## JAVA SETUP

zeonpadpdf library is currently based on the Java COM bridge called JACOB, a bridge is required between COM and the Java Virtual Machine. This allows the Java application to utilize MS Office APIS.

JACOB is a free Java COM bridge that is actively developed and maintained by its open-source community. It supports both the Win32 and x64 releases of the JRE/JDK version 1.5 or newer.

you can download the latest version of zeonpadpdf library from

<http://www.zeonpad.com/zeonpadpdf-download.html>

The following Java core files are included in our distribution zip file. **zeonpadpdf\_1.0.zip**

|                            |                       |
|----------------------------|-----------------------|
| zeonpadpdf java library    | zeonpadpdf.jar        |
| JACOB native 32-bit binary | jacob-1.15-M3-x86.dll |
| JACOB native 64-bit binary | jacob-1.15-M3-x64.dll |
| JACOB Java library         | jacob.jar             |

Download and install the Java SE Development Kit 1.5 or above from

<http://www.oracle.com/technetwork/java/index.html>

Add a **JAVA\_HOME** environment variable.

Add **zeonpadpdf.jar** and **jacob.jar** into your classpath.

If you are using 64 bit machine then Copy the **Jacob-1.15-M3-x64.dll** file to java\_Home/bin (Example : C:\Program Files\Java\jdk1.6.0\_26\bin).

If you are using 32 bit machine then copy Copy the **jacob-1.15-M3-x86.dll** file to java\_Home/bin (Example : C:\Program Files\Java\jdk1.6.0\_26\bin).

## GenericConvertor Object Description

This object is used to convert word,excel,ppt,publisher,visio,outlook ,images to PDF.

### Example Usage In Java

```
GenericConvertor genericConv = new GenericConvertor();  
genericConv.convert("C:\\Users\\Desktop\\input.doc","C:\\Users\\Desktop\\output.pdf");
```

## WordToPdf Object Description

This object is used to convert ms-word document to pdf. This object contains many ms-word related features. WordToPdf only works with Word 2007 or higher version. Only Word 2007 requires the free "Save as PDF or XPS" add-in for Office 2007 to be installed. This add-in is available from Microsoft.

### Example Usage In Java

```
WordToPdf wordToPdf = new WordToPdf();
```

```
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## Properties

### PdfExportOptimizeFor Property

Specifies the resolution and quality of the PDF document for Screen or Print.

Default its set to PdfExportOptimizeFor.PDF\_EXPORT\_OPTIMIZE\_FORPRINT for Print.

### Type

PdfExportOptimizeFor

|                                     |   |
|-------------------------------------|---|
| PDF_EXPORT_OPTIMIZE_FOR_ONSCREEN =1 | Export for screen, which is a lower quality and results in a smaller file size. |
| PDF_EXPORT_OPTIMIZE_FORPRINT =0     | Export for print, which is higher quality and results in a larger file size.    |

### Usage

Setting Resolution and Quality for print.

```
WordToPdf wordToPdf = new WordToPdf();
```

```
wordToPdf.setPdfExportOptimizeFor(PdfExportOptimizeFor.PDF_EXPORT_OPTIMIZE_FORPRINT);
```

```
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

### PdfExportRange Property

Specifies how much of the document to convert to pdf. Default set to

PdfExportRange.PDF\_EXPORT\_ALL\_DOCUMENT means converting the full document to pdf.

### Type

PdfExportRange

|                             |   |
|-----------------------------|---|
| PDF_EXPORT_ALL_DOCUMENT = 0 | Convert the entire document to pdf.                   |
| PDF_EXPORT_SELECTION = 1    | Convert the contents of the current selection to pdf. |
| PDF_EXPORT_CURRENT_PAGE = 2 | Convert the current page to pdf.                      |

|                       |   |
|-----------------------|---|
| PDF_EXPORT_FROM_TO =3 | Convert the contents of a range using the starting and ending positions to pdf. |
|-----------------------|---|

### Usage

#### Example 1

```
WordToPdf wordToPdf = new WordToPdf();
wordToPdf.setPdfExportRange(PdfExportRange.PDF_EXPORT_ALL_DOCUMENT);
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

#### Example 2

```
WordToPdf wordToPdf = new WordToPdf();
wordToPdf.setPdfExportRange(PdfExportRange.PDF_EXPORT_FROM_TO);
wordToPdf.setStartPage(2);
wordToPdf.setEndPage(6);
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

### DocumentMarkups Property

Specifies whether to export the pdf with markup. Default value is PDFMarkups.PDF\_EXPORT\_WITHOUT\_MARKUP means document markups are omitted.

### Type

PDFMarkups

|                              |                                 |
|------------------------------|---------------------------------|
| PDF_EXPORT_WITHOUT_MARKUP =0 | Exports the pdf without markup. |
| PDF_EXPORT_WITH_MARKUP=7     | Exports the pdf with markup.    |

### Usage

```
WordToPdf wordToPdf = new WordToPdf();
wordToPdf.setDocumentMarkups(PDFMarkups.PDF_EXPORT_WITH_MARKUP);
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

### IncludeDocProps Property

To include document properties in pdf file. Default is set to false.

Set it to True to include document properties from the source document in the PDF.

### Type

Boolean

### Usage

```
WordToPdf wordToPdf = new WordToPdf();  
wordToPdf.setIncludeDocProps(true);  
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## CreateBookmarks Property

Specifies whether to export bookmarks and the type of bookmarks to export into pdf. Default value is set to PdfCreateBookmark.PDF\_CREATE\_NO\_BOOKMARK means there will be no bookmarks in the PDF

### Type

PdfCreateBookmark

|   |  |
|---|--|
| PdfCreateBookmark.PDF_CREATE_NO_BOOKMARK=0      | Do not create bookmarks in the exported pdf document.  |
| PdfCreateBookmark.PDF_CREATE_HEADINGBOOKMARKS=1 | Create a bookmark in the exported pdf document for each Microsoft Office Word heading, which includes only headings within the main document and text boxes not within headers, footers, endnotes, footnotes, or comments. |
| PdfCreateBookmark.PDF_CREATE_WORDBOOKMARKS=2    | Create a bookmark in the exported pdf document for each Word bookmark, which includes all bookmarks except those contained within headers and footers.   |

### Usage

```
WordToPdf wordToPdf = new WordToPdf();  
wordToPdf.setCreateBookmarks(PdfCreateBookmark.PDF_CREATE_WORDBOOKMARKS);  
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## DocStructureTags Property

This is used to include extra data into generated pdf to help screen readers, such as information about the flow and logical organization of the content ,The default is **true**.

### Type

Boolean

### Usage

```
WordToPdf wordToPdf = new WordToPdf();  
wordToPdf.setDocStructureTags(false);  
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## BitmapMissingFonts Property

This is used to include a bitmap of the text or reference the text font. The default is **true** means non-embeddable fonts will be converted to bitmaps. When this property is False, the font is referenced but not embedded in the PDF.

### Type

Boolean

### Usage

```
WordToPdf wordToPdf = new WordToPdf();  
wordToPdf.setBitmapMissingFonts(false);  
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## ConversionTimeOut Property

Set the file conversion time out property. Default is 5 min.

### Type

Int (minute)

### Usage

```
WordToPdf wordToPdf = new WordToPdf();  
wordToPdf.setConversionTimeOut (10); //10 minute  
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## DocumentPassword Property

Sets a password for password protected Excel document.

### Type

String

### Usage

```
WordToPdf wordToPdf = new WordToPdf();  
wordToPdf.setDocumentPassword ("12345"); //10 minute  
wordToPdf.convert("C:\\Users\\Desktop\\input.docx","C:\\Users\\Desktop\\output.pdf");
```

## ExcelToPdf Object Description

This object is used to convert Microsoft Excel to pdf. This object contains many Microsoft Excel related features. ExcelToPdf only works with Microsoft Excel 2007 or higher version. Only Excel 2007 requires the free "Save as PDF or XPS" add-in for Office 2007 to be installed. This add-in is available from Microsoft.

### Example Usage In Java

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## Properties

### DocumentPassword Property

Sets a password for password protected Excel document.

### Type

String

## Usage

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.setDocumentPassword("12345");  
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## PdfExportQuality Property

Specifies the quality of the exported pdf file. Default its set to PDFXIFixedFormatQuality.PDF\_EXPORT\_QUALITY\_STANDARD.

## Type

PDFXIFixedFormatQuality

|                                |                                |
|--------------------------------|--------------------------------|
| PDF_EXPORT_QUALITY_MINIMUM =1  | Export pdf with lower quality  |
| PDF_EXPORT_QUALITY_STANDARD =0 | Export pdf,with higher quality |

## Usage

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.setPdfExportQuality(PDFXIFixedFormatQuality.PDF_EXPORT_QUALITY_STANDARD);  
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## IncludeDocProps Property

To include document properties in pdf file. Default is set to false.

Set it to True to include document properties from the source document in the PDF.

## Type

Boolean

## Usage

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.setIncludeDocProps(true);  
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## IgnorePrintAreas Property

true to ignore any print areas set when exporting; otherwise, false. Default is false.

### Type

Boolean

### Usage

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.setIgnorePrintAreas(true);  
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## PrintAllSheets Property

Default is true to print all sheets.If set to false then need to set **from** and **to**

### Type

Boolean

### Usage

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.setPrintAllSheets(false);  
excelToPdf.setFrom(1);  
excelToPdf.setFrom(10);  
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## ConversionTimeOut Property

Set the file conversion time out property. Default is 5 min.

### Type

Int (minute)

### Usage

```
ExcelToPdf excelToPdf = new ExcelToPdf ();  
excelToPdf.setConversionTimeOut (10); //10 minute
```

```
excelToPdf.convert("C:\\Users\\Desktop\\input.xls","C:\\Users\\Desktop\\output.pdf");
```

## PptToPdf Object Description

This object is used to convert Microsoft PowerPoint to pdf. PptToPdf only works with Microsoft PowerPoint 2007 or higher version. Only PowerPoint 2007 requires the free "Save as PDF or XPS" add-in for Office 2007 to be installed. This add-in is available from Microsoft.

### Example Usage In Java

```
PptToPdf pptToPdf = new PptToPdf();  
pptToPdf.convert("C:\\Users\\Desktop\\input.pptx","C:\\Users\\Desktop\\output.pdf");
```

### Properties

#### ConversionTimeout Property

Set the file conversion time out property. Default is 5 min.

#### Type

Int (minute)

#### Usage

```
PptToPdf pptToPdf = new PptToPdf();  
pptToPdf.setConversionTimeout(1);  
pptToPdf.convert("C:\\Users\\Desktop\\input.pptx","C:\\Users\\Desktop\\output.pdf");
```

## PublisherToPdf Object Description

This object is used to convert Microsoft Publisher to pdf. PublisherToPdf only works with Microsoft Publisher 2007 or higher version. Only Publisher 2007 requires the free "Save as PDF or XPS" add-in for Office 2007 to be installed. This add-in is available from Microsoft.

### Example Usage In Java

```
PublisherToPdf publisherToPdf = new PublisherToPdf();  
publisherToPdf.convert("C:\\Users\\Desktop\\input.pub", "C:\\Users\\Desktop\\output.pdf");
```

## Properties

### ConversionTimeOut Property

Set the file conversion time out property. Default is 5 min.

#### Type

Int (minute)

#### Usage

```
PublisherToPdf publisherToPdf = new PublisherToPdf();  
publisherToPdf.setConversionTimeOut(1);  
publisherToPdf.convert("C:\\Users\\Desktop\\input.pub", " C:\\Users\\Desktop\\output.pdf");
```

### IncludeDocProps Property

To include document properties in pdf file. Default is set to false.  
Set it to True to include document properties from the source document in the PDF.

#### Type

Boolean

#### Usage

```
PublisherToPdf publisherToPdf = new PublisherToPdf();  
publisherToPdf.setIncludeDocProps(true);  
publisherToPdf.convert("C:\\Users\\Desktop\\input.pub", " C:\\Users\\Desktop\\output.pdf");
```

### DocStructureTags Property

This is used to include extra data into generated pdf to help screen readers, such as information about the flow and logical organization of the content ,The default is **true**.

#### Type

Boolean

#### Usage

```
PublisherToPdf publisherToPdf = new PublisherToPdf();  
publisherToPdf.setDocStructureTags(false);  
publisherToPdf.convert("C:\\Users\\Desktop\\input.pub", " C:\\Users\\Desktop\\output.pdf");
```

## BitmapMissingFonts Property

This is used to include a bitmap of the text or reference the text font. The default is **true** means non-embeddable fonts will be converted to bitmaps. When this property is False, the font is referenced but not embedded in the PDF.

### Type

Boolean

### Usage

```
PublisherToPdf publisherToPdf = new PublisherToPdf();
publisherToPdf.setBitmapMissingFonts(false);
publisherToPdf.convert("C:\\Users\\Desktop\\input.pub", "C:\\Users\\Desktop\\output.pdf");
```

## OutlookToPdf Object Description

This object is used to convert Microsoft Outlook to pdf.

### Example Usage In Java

```
OutlookToPdf outlookToPdf = new OutlookToPdf();
outlookToPdf.convert("C:\\Users\\Desktop\\input.msg", "C:\\Users\\Desktop\\output.pdf");
```

### Properties

#### ConversionTimeOut Property

Set the file conversion time out property. Default is 5 min.

### Type

Int (minute)

### Usage

```
OutlookToPdf outlookToPdf = new OutlookToPdf();
outlookToPdf.setConversionTimeOut(1);
outlookToPdf.convert("C:\\Users\\Desktop\\input.msg", "C:\\Users\\Desktop\\output.pdf");
```

## ImageToPDF Object Description

This object is used to convert images into PDF.

### Example Usage In Java

```
ImageToPDF imageToPdf = new ImageToPDF();  
imageToPdf.convert("C:\\Users\\Desktop\\input.jpg", "C:\\Users\\Desktop\\output.pdf");
```

### Properties

#### ConversionTimeOut Property

Set the file conversion time out property. Default is 5 min.

#### Type

Int (minute)

#### Usage

```
ImageToPDF imageToPdf = new ImageToPDF();  
imageToPdf.setConversionTimeOut(1);  
imageToPdf.convert("C:\\Users\\Desktop\\input.jpg", "C:\\Users\\Desktop\\output.pdf");
```

## VisioToPdf Object Description

This object is used to convert Microsoft Visio to pdf. VisioToPdf only works with Microsoft Visio 2007 or higher version. Only Visio 2007 requires the free "Save as PDF or XPS" add-in for Office 2007 to be installed. This add-in is available from Microsoft.

### Example Usage In Java

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

## Properties

### ConversionTimeOut Property

Set the file conversion time out property. Default is 5 min.

#### Type

Int (minute)

#### Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setConversionTimeOut(1);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

### VisioPrintOutRange Property

This is used to define range of document pages to be exported, Default is set to PDFVisioPrintOutRange.VISIO\_PRINT\_ALL.

#### Type

PDFVisioPrintOutRange

|                              |   |
|------------------------------|---|
| VISIO_PRINT_ALL = 0          | Prints all foreground pages.                                  |
| VISIO_PRINT_FROM_TO = 1      | Prints pages between the FromPage value and the ToPage value. |
| VISIO_PRINT_CURRENT_PAGE = 2 | Prints the active page.                                       |
| VISIO_PRINT_SELECTION = 3    | Prints a selection.   |
| VISIO_PRINT_CURRENT_VIEW = 4 | Prints the current view area.                                 |

#### Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setVisioPrintOutRange(PDFVisioPrintOutRange.VISIO_PRINT_CURRENT_PAGE);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

### ColorAsBlack Property

True to render all colors as black to ensure that all shapes are visible in the exported drawing. False to render colors normally. The default is False .

## Type

Boolean

## Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setColorAsBlack(true);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

## IncludeBackground Property

Whether to include background pages in the exported file. The default is True .

## Type

Boolean

## Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setIncludeBackground(true);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

## IncludeDocumentProperties Property

Whether to include document properties in the exported file. The default is True .

## Type

Boolean

## Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setIncludeDocumentProperties(false);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

## IncludeStructureTags Property

Whether to include document structure tags to improve document accessibility. The default is True .

### Type

Boolean

### Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setIncludeStructureTags (false);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```

## NativeOfficePDF Property

Whether the resulting document is compliant with ISO 19005-1 (PDF/A). The default is False .

### Type

Boolean

### Usage

```
VisioToPdf visioToPdf = new VisioToPdf();  
visioToPdf.setNativeOfficePDF (false);  
visioToPdf.convert("C:\\Users\\Desktop\\input.vsd", "C:\\Users\\Desktop\\output.pdf");
```