

.NET PDF Viewer for WinForms

1. Introduction

Author: SkySof Software Inc. at <http://www.skysof.com>

Download URL: <http://www.getfilez.com/pdfvwnet.zip>

Updated: 7/16/10

.NET PDF Viewer is a .NET component for C# and Visual Basic .NET developers which enables your WinForm application to display and interact with PDF files. Simply place the control on your WinForm, pass a PDF file path to the OpenPDF method, and you are all set! .NET PDF Viewer is both 32-bit and 64-bit compatible and contains many functions for printing, setting page number/zoom factor, etc. .NET PDF Viewer was written in .NET to ensure maximum compatibility with your .NET projects and does not require any PDF reading software such as Adobe Reader. Note: If you are looking for a .NET for displaying PDF files on WebForms please download our product .NET PDF Viewer for WebForms at <http://www.getfilez.com/pdfvasp.zip>

2. How to add .NET PDF Viewer to your .NET project

- Open Visual Studio .NET and create a new C# or Visual Basic .NET project
- Right-click on the toolbox and select “Add/Remove Items...” or “Choose Items...”
- Select the .NET Framework Components Tab
- Click the Browse button and add file C:\PDF Viewer Component for .NET\bin\pdfview.dll
- Click the OK button to add the component to the toolbox
- The component should appear in the Windows Forms toolbox as “pdfview”
- Double-click on the component to add to your form
- Resize and move the component on the form as desired
- Set the Anchor property of the component to Top, Bottom, Left, Right
- Add a button to the form
- Double-click on the button to access code editor and enter the following code within the Click event:

```
pdfView1.OpenPDF("http://www.getfilez.com/pdfvwnet.pdf")
```

- Run the application and click on the button. The PDF should appear!

Note: The Visual Basic demo source code for PDF Viewer .NET can be found at C:\PDF Viewer Component for .NET\Demo\Demo.sln. The C# demo source code for PDF Viewer .NET can be found at C:\PDF Viewer Component for .NET\C Sharp Demo\C Sharp Demo.sln.

To open a PDF file with C #:

```
bool b;  
string strFilePath = "C:\\MyPath\\pdfvwnet.pdf";  
b = PdfView1.OpenPDF(strFilePath);
```

3. Including .NET PDF Viewer with your applications

When building an installation program for your application the following file must be included:

pdfview.dll
Aspose.Pdf.Kit.dll

4. Purchasing .NET PDF Viewer

.NET PDF Viewer is one of the lowest priced .NET components available for displaying and interacting with PDF files and is guaranteed to work with any PDF file version (including future releases) or you get your money back! **.NET PDF Viewer may be distributed with your application royalty free and all future upgrades are free!**

The pricing for .NET PDF Viewer through <http://www.regnow.com> is as follows:

\$299.95 USD Individual Developer License

\$499.95 USD Site Developer License (Unlimited number of developers per site)

Go to this link to purchase using RegNow:

<https://www.regnow.com/softsell/nph-softsell.cgi?item=4459-63>

Save 30% by buying .NET PDF Viewer through www.paypal.com!!!

\$209.95 USD Individual Developer License

\$349.95 USD Site Developer License (Unlimited number of developers per site)

Go to this link to buy a Single Developer License using PayPal:

https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=.NET+PDF+Viewer+Single+Developer+License&amount=209.95

Go to this link to buy a Site Developer License using PayPal:

https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=.NET+PDF+Viewer+Site+Developer+License&amount=349.95

Please send emails to kusluski@bellsouth.net for additional ordering methods. Please visit <http://www.skysof.com> for other fine products.

5. Methods

CloseForm - Method to close the active form.

CloseStream - Method to close the active file stream.

DownloadFile – Method to download a file.

URL As String	The URL of the file to save
localFile As String	The file to save

Returns: Boolean (True if file was downloaded successfully)

ExportBookmarksToTreeView - Export bookmarks from active PDF to specified TreeView control. Note: The OpenPDF method should be used before using this method. The TreeView's AfterSelect Sub can be coded to move to a specific page when a bookmark is clicked. See the sample code below:

tv As TreeView The TreeView control to export the bookmarks too
expandBookmarks As Boolean Expand all bookmark nodes in the TreeView control?

Returns: Boolean (True if bookmarks were successfully added to TreeView control)

Visual Basic .NET code:

```
Private Sub TreeView1_AfterSelect(ByVal sender As System.Object, ByVal e As  
System.Windows.Forms.TreeViewEventArgs) Handles TreeView1.AfterSelect  
    Me.Cursor = Cursors.WaitCursor  
    Pdfview1.Page = CInt(e.Node.Tag)  
    Me.Cursor = Cursors.Default  
End Sub
```

C# code:

```
private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)  
{  
    this.Cursor = Cursors.WaitCursor;  
    Pdfview1.Page = Convert.ToInt32(e.Node.Tag);  
    this.Cursor = Cursors.Default;  
}
```

ExportBookmarksToXMLFile - Export bookmarks from active PDF to specified XML file.

outXMLFile As String The XML file to create

Returns: Boolean (True if XML file was created successfully)

ExportFieldsToFDF - Export form field values to a FDF file.

outFile As String/Stream The FDF file/stream to create

Returns: Boolean (True if file/stream was created successfully)

ExportFieldsToXFDF - Export form field values to a XFDF file.

outFile As String/Stream The XFDF file/stream to create

Returns: Boolean (True if file/stream was created successfully)

ExportFieldsToXML - Export form field values to an XML file.

outFile As String/Stream The XML file/stream to create

Returns: Boolean (True if file/stream was created successfully)

ExtractPages - Extract pages from the active PDF.

startPage As Integer The start page
endPage As Integer The end page
outputPDF As String The PDF file to create

Returns: Boolean (True if PDF file was created successfully)

FillFormField - Method to fill a PDF form field with a value.

fieldName As String	Field name
fieldValue As String	Field value

Returns: Boolean (True if form field was updated successfully)

FirstPage - Method to move to the first page of the active PDF.

FlattenFormField – Method to flatten a form field.

fieldName As String	Field name
---------------------	------------

Returns: Boolean (True if form field was flattened successfully)

GetTotalPages - Method to get the total number of pages in a PDF file.

strPath As String	The PDF file to open
-------------------	----------------------

Returns: Integer (The total number of pages)

HasPassword - Method to determine if the PDF file is protected with a password.

pdf As String/Stream	The PDF file/stream to open
----------------------	-----------------------------

Returns: Boolean (True if the PDF file is password-protected)

ImportFieldsFromFDF - Method to Import form field values into a PDF from FDF file.

fdfFile As String/Stream	FDF file/stream to use
outFile As String/Stream	PDF file/stream to create

Returns: Boolean (True if PDF file was created successfully)

ImportFieldsFromXFDF - Method to Import form field values into a PDF from XFDF file.

xfdfFile As String/Stream	XFDF file/stream to use
outFile As String/Stream	PDF file/stream to create

Returns: Boolean (True if PDF file was created successfully)

ImportFieldsFromXML - Method to Import form field values into a PDF from an XML file.

xmlFile As String/Stream	XML file/stream to use
outFile As String/Stream	PDF file/stream to create

Returns: Boolean (True if PDF file was created successfully)

LastPage - Method to move to the last page of the active PDF.

NextPage – Method to move to the next page of the active PDF.

OpenPDF - Method to open a PDF file/stream.

pdf As String/Stream	PDF file to open
page As Integer	Page to display
resolution As Integer	Resolution to use (1 through 600)
password As String	Password to use (not available for streams)

Returns: Boolean (True if PDF file opened successfully, otherwise False)

OpenPDFfromURL – Method to open a PDF from a URL.

URL As String	The URL of the PDF to open
pdf As String	PDF file to save
page As Integer	Page to display
resolution As Integer	Resolution to use (1 through 600)
password As String	Password to use

Returns: Boolean (True if PDF file opened successfully, otherwise False)

PreviousPage - Method to go to the previous page of the active PDF file.

Print - Method to print a PDF file.

resolution As Integer	Resolution to use (1 through 600)
showPrintPageDialog As Boolean	Show Printer Page Dialog Screen?
autoResize As Boolean	Auto resize pages?
autoRotate As Boolean	Auto rotate pages?
showPrintDialog As Boolean	Show Printer Dialog Screen?

Returns: Boolean (True if the PDF file printed successfully, otherwise False)

PrintPDFWithAPI – Method to print the active/specified PDF using the system's default PDF printer.

file As String	The PDF file to print
----------------	-----------------------

Returns: Boolean (True if the PDF file printed successfully, otherwise False)

Rotate - Method to rotate the active PDF page by specified degrees.

rotateFlipType As System.Drawing.RotateFlipType	rotation degrees
---	------------------

SaveAsText - Method to create a text file from the active PDF.

fileName As String	The text file to save
textMode As Integer	The text mode to use (0 = pure text, 1 = raw text)

Returns: Boolean (True if the text file was created, otherwise False)

SaveAsTIFF - Method to create a TIFF image file from the active PDF.

fileName As String	The TIFF image file to save
--------------------	-----------------------------

resolution As Integer The image resolution to use (1 through 600)

Returns: Boolean (True if the TIFF image file was created, otherwise False)

SaveAsTIFFClassF - Method to create a TIFF Class F image file from the active PDF.

fileName As String The TIFF image file to save
resolution As Integer The image resolution to use (1 through 600)
resolutionX As Integer The image resolution X to use
resolutionY As Integer The image resolution Y to use

Returns: Boolean (True if the TIFF image file was created, otherwise False)

SaveCopy - Method to save a copy of the PDF file.

outFile As String The PDF file to create

Returns: Boolean (True if the PDF file was copied successfully)

SaveForm - Method to save the form with filled values.

flattenFields As Boolean Flatten all fields?

Returns: Boolean (True if the form was saved successfully)

SavePageAsImage - Method to save a PDF file page as an image file.

fileName As String The image file to save
pageNumber As Integer The page number to use
format As ImageFormat The image format to use
resolution As Integer The image resolution to use (1 through 600)

Returns: Boolean (True if the image file was created successfully)

SavePageAsText - Method to save a PDF file page as a text file.

fileName As String The image file to save
pageNumber As Integer The page number to use
textMode As Integer The text mode to use (0 = pure text, 1 = raw text)

Returns: Boolean (True if the text file was created successfully)

SetForm - Method to set the form to be filled with values.

outFile As String The PDF file to create

Returns: Boolean (True if the PDF file was created successfully)

SetPassword - Method to create a PDF from the active PDF with passwords.

outFile As String/Stream The PDF file/stream to create
ownerPassword As String Owner password

newUserPassword As String	New user password
newOwnerPassword As String	New owner password
use128bitEncryption As Boolean	Use 128 bit encryption?

Returns: Boolean (True if the PDF file/stream was created successfully)

StampWithBackgroundImage - Method to create a PDF with a background image.

backgroundFile As String	The image file to use as background
backgroundFileRotation As Integer	Rotation 0 through 360
backgroundFileOpacity As Single	Opacity 0.0 through 1.0
outFile As String	The PDF file to create
pageArray As Array	An array of page numbers to add background image too

Returns: Boolean (True if the PDF file was created successfully)

StampWithText - Method to create a PDF with stamped text.

text As String	The text to use
fontColor As System.Drawing.Color	Font color
fontName As String	Font name
fontSize As Single	Font size
textRotation As Integer	Rotation 0 through 360
textOpacity As Single	Opacity 0.0 through 1.0
outFile As String	The PDF file to create
originX As Integer	Text X origin
originY As Integer	Text Y origin
pageArray As Array	An array of page numbers to add the text too

Returns: Boolean (True if the PDF file was created successfully)

StampWithWatermark - Method to create a PDF with a watermark.

watermarkFile As String	The file containing the watermark
textRotation As Integer	Rotation 0 through 360
textOpacity As Single	Opacity 0.0 through 1.0
outFile As String	The PDF file to create
originX As Integer	Watermark X origin
originY As Integer	Watermark Y origin
pageArray As Array	An array of page numbers to add the text too

Returns: Boolean (True if the PDF file was created successfully)

UnloadPDF - Method to unload the active PDF file.

ZoomIn - Method to zoom in at a specified percentage.

pct As Integer	The percentage to zoom in
----------------	---------------------------

ZoomOrigin - Method to restore the original zoom factor.

ZoomOut - Method to zoom out at a specified percentage.

pct As Integer The percentage to zoom out

6. Properties

ActivePDF - Get the active PDF as string or file stream.

Author – Get the author name (string) of the active PDF file.

Border – Get/set the border style of the active PDF file.

CreationDate – Get the creation date (string) of the active PDF file.

Creator – Get the creator name (string) of the active PDF file.

GetMetaData – Get/set the meta data flag (boolean) for the active PDF file.

IsLoaded – Is a PDF loaded? (boolean)

IsStream – Is the active PDF a File Stream? (boolean)

Keywords – Get the keywords (string) of the active PDF file.

ModDate – Get the modification date (string) of the active PDF file.

Page – Get/set the page number (integer) of the active PDF file.

PagePosition – Get/set the current page's X and Y coordinates (Drawing.Point) of the active PDF file.

Pages – Get the total number of pages (integer) of the active PDF file.

Password – Get/set the password (string) for the active PDF file.

Producer – Get the producer name (string) of the active PDF file.

Resolution – Get/set the resolution (integer) for the active PDF file. Possible values: 1 through 600.

Subject – Get the subject (string) of the active PDF file.

TempPath – Get/set the file path (string) where temporary files will be created.

Title – Get the title (string) of the active PDF file.

Zoom – Get/set the zoom percentage number (integer) of the active PDF file.

7. Events

pageClick – Runs when a mouse click is performed on the component.

pageCursorChanged - Runs when the mouse cursor over the component changes.

pageDoubleClick - Runs when a mouse double-click is performed on the component.

pageDragDrop - Runs when a drop operation is performed on the component.

pageDragEnter - Runs when a drag operation is performed on the component.

pageDragLeave - Runs when a drag operation leaves the component.

pageDragOver - Runs when a drag operation moves over the component.

pageGotFocus - Runs when the component gets focus.

pageInvalidated - Runs when the component is redrawn.

pageLostFocus - Runs when the component loses focus.

pageMouseClicked - Runs when a mouse click is performed on the component.

pageMouseDoubleClick - Runs when a mouse double-click is performed on the component.

pageMouseDown - Runs when a mouse down click is performed on the component.

pageMouseEnter - Runs when the mouse cursor first passes over the component.

pageMouseHover - Runs whenever the mouse cursor hovers over the component.

pageMouseLeave - Runs when the mouse cursor leaves the component.

pageMouseMove - Runs when the mouse cursor is moved over the component.

pageMouseUp - Runs when a mouse up click is performed on the component.

pageMouseWheel - Runs when the mouse wheel is moved while the cursor is on the component.

pagePaint - Runs when the component is redrawn.

pageResize - Runs when the component is resized.

pageSizeChanged - Runs when the component is resized.

8. PDF related products

To download .NET PDF Viewer for WebForms, a web component for ASP .NET developers please click here:

<http://www.getfilez.com/pdfvasp.zip>

To download PDF Maker Class .NET, a .NET component (requires Adobe Acrobat Standard or Professional) for software developers which greatly simplifies creation of PDF files from multiple file types click here:

<http://www.getfilez.com/pdfmcls.zip>

To download PDF Create .NET, a .NET component (does not require Adobe Acrobat Standard or Professional) for software developers which greatly simplifies creation of PDF files from multiple file types click here:

<http://www.getfilez.com/pdfcnet.zip>