



*PDF Image Stream
PDF/Flash
Documentation*

FyTek, Inc.

Web site: <http://www.fytek.com>

FyTek's PDF Image Stream

Trademarks

FyTek, FyTek PDF Image Stream and the FyTek logo are registered trademarks or trademarks of FyTek Incorporated in the United States and/or other countries. Acrobat, Adobe, Adobe PDF and Adobe Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product names, logos, designs, titles, words or phrases mentioned within this publication may be trademarks, servicemarks, or tradenames of FyTek, Inc. or other entities and may be registered in certain jurisdictions including internationally.

FyTek Disclaimer

FYTEK, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE. Copyright © 2000–2007 FyTek, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of FyTek, Inc.

This guide may contain links to third-party websites that are not under the control of FyTek, and FyTek is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. FyTek provides these links only as a convenience, and the inclusion of the link does not imply that FyTek endorses or accepts any responsibility for the content on those third-party sites.

Acknowledgments

Software Development: Mike Bernardo
Writing: Mike Bernardo

FyTek, Inc.
2335 Pontiac Lake Road
Waterford, MI 48328

Introduction

PDF Image Stream is a program to convert an image or group of images to PDF and, optionally, flash (.swf) format. The images may be bitmap, jpeg, gif, png or tif (single or multi-page) type files. Some features of this software product are:

- Combine multiple images into one PDF or flash file
- Create a PDF for each image
- Scan a directory and, optionally, its subdirectories for images
- Create a PDF from an image or images on the Windows clipboard
- Specify a list of images to convert
- Size the image to the page or page to the image
- Include bookmarks, captions and page numbers

This software is available as freeware with limited functionality. The documentation covers all the commands available with the commercial version. The freeware version creates a PDF or PDFs from jpeg images only. You can convert other image types such as bitmap, gif, png and tif images but the words "FyTek, Inc. Image Stream Demo" will be printed across the page. The commercial version does not print this. In addition, the commercial version will allow you to convert from one image type to another and create thumbnail pages from a set of images.

Each image occupies its own page in the PDF by default. The image is centered on the page by default and not scaled unless it is larger than the page size. You may chose to size the page to image size or vice-versa. The -layout option or setLayout method can be used to place multiple images on each page. The following sections detail the options available using the exe or DLL.

Multi-page tifs will be broken out such that each image in the tif will be treated as if it were its own file. Multi-page tifs will take longer to process due to the complexity of the file structure.

You can create a slide show presentation using Acrobat or Acrobat Reader. Here's how to do it using Acrobat Reader 5. First, open the preferences menu under Edit|Preferences. Select "Full Screen" from the list and set your preferences, such as automatic page advance. Finally, select View|Full Screen from the menu.

Using the Executable

The program `pdfimage.exe` is the Windows executable program. It takes as input either a directory name followed by the types of images to include or a file containing a list of images. The name of the output file comes next. For example, run the following to convert all jpeg and gif images to a single PDF in the directory "my documents":

```
pdfimage.exe "d:\my documents" *.jpg,*.gif mypics.pdf
```

This example reads the list of images from a file call `mylist.dat`. Note the file input mask is left out in this case the input file name is preceded by the `@` symbol:

```
pdfimage.exe @mylist.dat mypics.pdf
```

You may also create a PDF for each image rather than combine all images into one PDF. Specify an output directory for the PDFs in this case. Each PDF will be named after the image and have an extension of `pdf`.

```
pdfimage.exe "d:\my documents" *.jpg d:\images -single
```

The output file will be named based on the first image to process if you leave the `-single` option off and do not specify an output file. In this case, the image base file name is used and a `.pdf` extension added.

The input file may also be a Web page or image from the Web. For HTML pages, the page is searched for `IMG` tags and those images are stored locally in order to create the PDF. These temporary images are removed once the PDF is built.

```
pdfimage.exe http://www.fytek.com/pdfimg.html *.jpg,*.gif mypics.pdf
```

Specify the URL of an image if you want to pull just a single image.

```
pdfimage.exe http://www.fytek.com/logo.jpg * mypics.pdf
```

Using a web address will print "Demo" across each page in the free version.

You can create a PDF based on an image or list of image files currently on the Windows clipboard. Use Explorer to mark a set of images and copy them to the clipboard or copy an image from any Windows application. Note the input directory and file input mask is left out in this case. You may optionally supply an output file name (for a single PDF) or directory (for multiple PDFs if `-single` is used). The temp directory will contain the PDF(s) when no output is specified. Use any of the other standard options on the command line as well:

```
pdfimage.exe -clip mypics.pdf -open -zoom 100
```

The following lists the various options you may include on the command line.

PDF Image Stream

Executable Options

Parameter	Description
<i>The following options are used when creating PDFs</i>	
-s	Search in subdirectories as well when specifying an input directory and file mask.
-pboff	Suppress the dialog box that normally appears when the application is running. Note the progress dialog box only works in the executable version and is not available with the DLL.
-pbt <i>title</i>	Sets the title of the dialog box.
-open	Automatically opens the newly created PDF. Only works when creating a PDF and not using the -single option.
-clip	The source is the current contents of the Windows clipboard. The PDF(s) will reside in the temp directory if no output file name is given.
-sort <i>sortby</i>	Used to specify the sort order when creating the images in the PDF. Pass in one of the following for sortby: f = file name b = file size in bytes d = last modify date n = numerically by name You may also use an r along with any of the above to sort in descending order. For example, rf or fr sorts by name in descending order. The numerical option is for when your files are named 1.tif, 2.tif, ..., 30.tif and so on.
-single	Create a single PDF for each image. The files are created in the source directory if the output file path is left off.
-page <i>width,height</i>	Sets the width and height of the page in inches. Default is 8.5,11. Be careful not to leave a space before or after the comma.
-size <i>width,height</i>	Sets the width and height for all images in pixels.
-layout <i>across,down,margin</i>	Sets the number of images across and down on each page. The margin is in inches and is used to set the spacing around each image. The default is one image per page. Be careful not to leave a space before or after the commas. Using this option will print "Demo" across each page in the free version.

PDF Image Stream

Executable Options

Parameter	Description
-notes <i>directory</i>	A directory containing .txt files with the same basename as the images. For example, if an image is named sunset.jpg the program will look for sunset.txt in the directory specified with this option. A pop-up note is added to the page the image is on if the text file is found. The contents of the note are taken from the text file. Using this option will print "Demo" across each page in the free version.
-search	Use along with the -notes option. This option places the notes text as invisible text on the page behind the image. This allows you to use the search feature to lookup an image in the PDF based on this text.
-searchonly	Use along with the -notes option. Works the same as the -search option except the pop-up notes are not placed in the PDF. The notes are only used as searchable text in this case.
-bkg <i>file</i>	An image file to use as the background for each page. Using this option will print "Demo" across each page in the free version.
-int	Applies an image interpolation algorithm during rendering. This helps to smooth out the transition between pixels when the resolution of the image is lower than that of the output device. At higher magnification levels images on the screen typically appear jagged. This option removes most of the jagged appearance.
-nodpi	Images over 72 dpi are automatically scaled down to retain their original height and width. No resolution is lost when this is done - the image is simply scaled to match its height and width based on 72 dpi. This option turns off the automatic scaling.
-dpi <i>number</i>	Sets the dpi for images in some situations where the program might not be able to determine the dpi of the image. This is only needed when the image should be scaled and the automatic scaling is not happening.
-margin <i>width</i>	Sets the width of the margin in inches. Default is .75.
-pgrotate <i>angle</i>	Sets the rotation angle (clockwise) for each page in the PDF. Must be either 0 (the default), 90, 180 or 270.

PDF Image Stream

Executable Options

Parameter	Description
-imgrotate <i>angle</i>	Sets the rotation angle (clockwise) for each image in the PDF. May be any value from 0 to 360.
-header <i>text</i>	Text for the page heading. Use for line breaks.
-headerpoint <i>number</i>	The font point size for the heading text. Default is 12.
-headeroffy <i>number</i>	The offset in points (1/72 of an inch) for the header. Positive values move the heading down, negative move it up.
-footer <i>text</i>	Text for the page footer. Use for line breaks.
-footerpoint <i>number</i>	The font point size for the footer text. Default is 12.
-footeroffy <i>number</i>	The offset in points (1/72 of an inch) for the footer. Positive values move the footer down, negative move it up.
-cap	Displays the caption at the bottom of the image. The caption is the file name unless specified using an input file or the -ctext option. A comma separates the caption from the file name in the input file. Use a tag for a line break. Note the text may flow into the image area depending on the image size and amount of text. See the Input File section for details on using an input file.
-ccolor <i>color</i>	The color for the caption text as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
-ccolorbg <i>color</i>	The color for the background for the caption text as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
-capoffy <i>number</i>	An offset in points (1/72 of an inch) to adjust the caption location by. A positive value moves the caption down while a negative value moves it up.

PDF Image Stream

Executable Options

Parameter	Description
-transparency <i>x[,text]</i>	<p>Used to specify the transparency value and mode. Requires Acrobat or Reader 5.0 or higher to view the transparency. This is the transparency for the caption background when you have multiple lines that overlay the image.</p> <p>The default mode, Normal, uses the transparency value to determine how transparent the text background will be. The opacity value is from 1 to 100. For example, in Normal mode, the higher the number, the less of the background shows through. Optionally, specify the mode of transparency. The valid values are:</p> <ul style="list-style-type: none">Normal (Default)MultiplyScreenOverlayDarkenLightenColorDodgeColorBurnHardLightSoftLightDifferenceExclusionHueSaturationColorLuminosity <p>Example:</p> <ul style="list-style-type: none">-transparency 50or-transparency 75,Screen
-bm	<p>Add bookmarks. The description for the bookmark is the same as the caption.</p>

PDF Image Stream

Executable Options

Parameter	Description
-ctext <i>text</i>	<p>The caption for each image. The following may be used as variables in the string for the caption:</p> <ul style="list-style-type: none">%f - file name%d - directory%b - file size%w - width%h - height <p>For example, "Image %f - %b, %wX%h" will come out looking something like "Image abc.jpg - 25148, 200X350". Use two %'s in a row if you're running the program from within a batch file (i.e. "Image %%f - %%b"). Be sure to also use the -cap option to show the caption in the PDF.</p>
-pg	Adds page numbers centered at the bottom of each page.
-point <i>size</i>	Sets the point size used for captions and page numbers. Default is 12.
-fit <i>image/page</i>	Set to "image" to size each page to the size of the image + the margin. Set to "page" to size each image to the size of the page - the margin. Sizing to "page" will make the image large enough to fit either width or length wise without losing the aspect ratio. That is, the image will not be stretched more in one direction than the other. The default is to leave the image as is centered on the page and only reduce in size if it doesn't fit on the page.
-zoom <i>percentage</i>	Sets the zoom factor to use when the document is opened. Setting to 100 means to open the document at 100%.
-land	Sets page to landscape mode. This is 11 by 8.5 unless the -page option was used.
-border <i>width</i>	Places a border around each image at the specified width. The image is modified to include the border. Use the -lineborder option to instead draw the border in the PDF without changes to the image. This results in faster execution when you don't need a raised or sunken border.

PDF Image Stream

Executable Options

Parameter	Description
-lineborder	Draws the border using line commands in the PDF. Use this option for faster execution and generally smaller PDF size when you don't need to create raised or sunken image borders. Use along with the -border option.
-bordercolor <i>color</i>	The color of the border as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
-borderraised <i>width</i>	The width for a raised border. You do not have to use the -border option along with this one. The edges of the image are lightened/darkened to provide a raised effect.
-bordersunken <i>width</i>	The width for a sunken border. You do not have to use the -border option along with this one. The edges of the image are lightened/darkened to provide a sunken effect.
-trans <i>text</i>	One of the following transition types: Split Blinds Box Wipe Dissolve Glitter Random The default page transition is to just replace the current page with the new page. The effects listed above can be used for presentation purposes. The random option shown above will pick one of the transition types at random for each page.
-transsecs <i>number</i>	The length of time in seconds to spend on the page transition. Default is 1 if a transition type is used.
-transdir <i>H/V</i>	The direction of the transition, horizontal or vertical (H or V). Valid only for Split and Blinds transition types.
-transmdir <i>I/O</i>	The direction of motion for the transition effect, inward or outward (I or O) from the center of the page. Valid only for Split and Box transition types.
-force	Turns off the prompt to overwrite the output file if it already exists.

PDF Image Stream

Executable Options

Parameter	Description
-pdftitle <i>text</i>	Sets the title of the PDF.
-pdfauthor <i>text</i>	Sets the author of the PDF.
-pdfsubject <i>text</i>	Sets the subject of the PDF.
-pdfcreator <i>text</i>	Sets the creator of the PDF.
-pdfproducer <i>text</i>	Sets the producer of the PDF.
-pdfkeywords <i>text</i>	Sets the keywords for the PDF.
<i>The following options are used when creating images (these are features of the commercial version only)</i>	
-make <i>type</i>	Sets the type of images to create. One of the following: bmp, jpg, gif, tif or png. The output file name should be set to the directory you want the images placed in or the current directory will be used. Images created will be of the same base name as the original but not overwrite any existing images. For example, if you converting jpegs to gif and you have a file called mypic.jpg you'll get a file called mypic.gif. If you convert again, you'll get an image called mypic_1.gif and, if that name exists mypic_2.gif and so on.
-quality <i>high/low</i>	Sets the quality of jpeg images created to high (larger file size but better image quality) or low (smaller file size and less quality).
-xscale <i>number</i>	Amount to scale each image in the x direction entered as a percentage. For example, 50% will scale all images by 50 percent in width while 50 will set each image to 50 pixels wide.
-yscale <i>number</i>	Amount to scale each image in the y direction entered as a percentage. For example, 50% will scale all images by 50 percent in height while 50 will set each image to 50 pixels high.
-imoff	Turns off the standard processing of images that converts them to PDF. Instead, images are taken directly to PDF with little or no modification. This allows for faster conversion and sometimes smaller PDF size but does not work on as many image types. The images must be either a jpeg using 256 color grayscale or 24-bit color or a 256-color gif non-interlaced, version 87a. Some types of tiffs are supported as well - must be black & white, packbits or CCITT encoded with a single image strip.

PDF Image Stream

Executable Options

Parameter	Description
<i>-thumbnail size</i>	Use this option to create one thumbnail image containing all the individual images. Set to the size of each image along with spacing in pixels formatted as #x##+## where the # sign is a number of pixels. For example, 50x75+5+10 means create images 50 pixels wide by 75 pixels high and leave 5 pixels in the x direction between images and 10 pixels in the y direction.
<i>-flash filename</i>	Creates a flash file containing the pages of the PDF. You access the pages by clicking on the flash movie to activate then use the page-up and page-down keys to move from page to page. You may also click on the image to advance to the next one. Do not enter the .swf extension when using -flashmult. Note that not all images may translate to flash output so you'll want to test on your target platform with the types of images you plan on using. As a general rule, jpeg images should work the best when converting to flash.
<i>-flashmult</i>	Creates a flash file for each page of the PDF. The default is to build a single flash file containing all the pages. Do not enter the .swf extension for the -flash option when using this option. The output files will be named using the file name passed along with the page number. For example, if you use -flash myfla and the PDF has 3 pages, you will get three files - myfla_1.swf, myfla_2.swf and myfla_3.swf.
<i>-flashbg color</i>	Optional HEX formatted color for the flash file background. The default is FFFFFFFF or white.
<i>-flashscale number</i>	A scaling factor for the flash output. The default is 100 for 100%.
<i>-flashprescale number</i>	Windows platform only. A scaling factor for the images prior to being converted to flash output. The default is 100 for 100%. This can be useful to reduce the flash file size.
<i>The following options are available only when using the -thumbnail option</i>	
<i>-tnacross number</i>	The number of thumbnail images to place in a single row. For example, setting to 5 means put five images across from one another before dropping down to the next row to start another five images.

PDF Image Stream

Executable Options

Parameter	Description
-tnbgcolor <i>color</i>	The background color in hex form - red, green, blue pairs. For example, #000000 is black and #FFFFFF is white.
-tncap <i>caption</i>	The caption for each image in the thumbnail. The following may be used as variables in the string for the caption: %f - file name %d - directory %e - extension %b - file size %w - width %h - height For example, "Image %f - %b, %wX%h" will come out looking something like "Image abc.jpg - 25K, 200X350".
-tnpoint <i>size</i>	The pointsize for the captions. Default is 10.
-tnfont <i>file</i>	The truetype font file to use for the caption.
-tnshadow	Set this to place a shadow behind each thumbnail image.
-tnframe	Set this to place a frame around each thumbnail image.

PDF Image Stream

Examples

Create a single file with page numbers and bookmarks sorting files by name

```
pdfimage "d:\my documents" *.jpg,*.gif mypics.pdf -pg -bm -sort f
```

Create a single file with page numbers and size the page to the image

```
pdfimage . *.tif,*.jpg,*.gif mypics.pdf -pg -fit image -margin 0
```

Get the list of images to convert from a file called mylist.dat

```
pdfimage @mylist.dat mypics.pdf -pg -fit page -zoom 100 -cap -ctext "File %f"
```

Scan the current directory and all subdirectories for jpegs and create a single PDF for each image in the directory d:\pdfpics.

```
pdfimage . *.jpg d:\pdfpics -s -single -pg -fit page -zoom 100 -cap -ctext "File %d%f"
```

Pull the jpeg and gif images from a Web page and place them in a PDF.

```
pdfimage http://www.fytek.com/pdfimg.html *.jpg,*.gif mypics.pdf -bm -cap
```

The following examples only apply to the commercial version of the software.

Scan the current directory and all subdirectories for jpegs and convert to gif images to be placed in d:\gifpics.

```
pdfimage . *.jpg d:\pdfpics -s -make gif
```

Scan the directory c:\images for gifs and jpegs and create a thumbnail image. Size the thumbnails to 75 by 100 with 5 pixels between images. Allow up to 8 images across and place a caption below each image.

```
pdfimage c:\images *.gif,*.jpg thumb.gif -thumbnail 75x100+5+5 -tnacross 8 -tncap "%f"
```

Using the DLL (Dynamic Link Library)

The file pdfimage.dll is the dynamic link library. This file should reside in your Windows or Winnt directory under the system32 sub-directory. You first must register the DLL on your system (note this step happens automatically when you run the setup program). Do this by running

```
regsvr32 pdfimage.dll
```

You should see a message box that reads:

DllRegisterServer in pdfimage.dll succeeded.

Click OK to continue. You are now ready to use the DLL.

The .NET version contains the same methods as the standard DLL. The .NET DLL is named pdfimagedn.dll.

The properties of Image.Stream are:

<u>Property</u>	<u>Description</u>
pdfPages	Returns the total number of pages created in the PDF. The property may be checked after the method buildImage has been called.

PDF Image Stream

DLL Properties and Methods

The methods of Image.Stream are:

<u>Method</u>	<u>Description</u>
buildImage	<p>Call this method create the PDF, images or thumbnail. Set the parameters described in the following methods first then call this method. This function returns the name of the PDF file when using setOutput. Otherwise it will return the PDF stream which is useful if you want to send it to a web browser.</p> <p>The following are error codes which may be returned so you'll need to check for these: -1 = Cannot open output file -2 = No images to convert</p>
setCLA(text)	<p>Can be used as a shortcut to setting the methods described below. Use setCLA (set command line attributes) to set a string of attributes that would normally be sent to the executable version. For example, you can run setCLA("d:\my pics\images" "*.jpg,* .gif d:\temp\output.pdf -s") and set everything at once or use setDirectory, setMask, setOutput and setSubDir and set individually. You can use all of the command line options with setCLA.</p>
setDirectory(path or @file)	<p>Provide either the path to use for scanning for images or a file name preceded by an @ symbol. For example, setDirectory("d:\my documents") or setDirectory("@d:\temp\mylist.dat"). You can pass image data from memory as well using the setFileDataB or setFileDataH methods. This may also be a Web page or image. For example, setDirectory("http://www.fytek.com/pdfimg.html") or setDirectory("http://www.fytek.com/logo.jpg"). For Web pages, the page text is scanned for IMG tags. Those images are placed in the PDF. Using a web address will print "Demo" across each page in the free version.</p>
setMask(comma separated list)	<p>Provide a comma separated list of file names to include. This option is only necessary when a path is used for the setDirectory method. You do not need to set this when providing a file to setDirectory.</p>

PDF Image Stream

DLL Properties and Methods

Method	Description
setFileDataB id, byte()	Used to pass data (images) in from a byte array in memory. Pass the byte array containing the data. The "id" value is what you will be using to refer to the file which can be the actual file name or some other unique ID. The "byte()" value is the raw file data contained in the byte array. You can set the id value to an image name (like myimg.jpg) or some unique value (like myimage1). You then use the id with the setDirectory or setMask methods as you would any other file name.
setFileDataH id, hexstring	Used to pass data (images) in from a hex string in memory. Pass an ASCII hex string containing the data. The "id" value is what you will be using to refer to the file which can be the actual file name or some other unique ID. The "hexstring" value is the raw file data in hex format (0-9 and a-f/A-F characters only - any other characters will be removed before processing). You can set the id value to an image name (like myimg.jpg) or some unique value (like myimage1). You then use the id with the setDirectory or setMask methods as you would any other file name.
setOutput(path-file)	Provide the path and name of the output file. This should be a file with a .pdf extension. This should be set to a directory only when using the setSingle method described below. setSingle will create a separate PDF file for each image. The directory passed to setOutput will be used to store the PDF files in. You may also leave this blank if you want the PDF contents returned as output from the buildImage function.
setSubDir	Search in subdirectories as well when specifying and input directory and file mask. This is not used when a file name is passed to setDirectory.
setOpen	Automatically opens the newly created PDF. Only works when creating a PDF and not using the setSingle method.

PDF Image Stream

DLL Properties and Methods

Method	Description
setClipboard	The source is the current contents of the Windows clipboard. The PDF(s) will reside in the temp directory if no output file name is given.
setSort(sortby)	Used to specify the sort order when creating the images in the PDF. Pass in one of the following for sortby: f = file name b = file size in bytes d = last modify date n = numerically by name You may also use an r along with any of the above to sort in decending order. For example, rf or fr sorts by name in decending order. The numerical option is for when your files are named 1.tif, 2.tif, ..., 30.tif and so on.
setSingle	Create a single PDF for each image. The files are created in the source directory if the output file path is left off. You set the output path with the setOutput method when using this option.
setPageSize width, height	Sets the width and height of the page in inches. Default is 8.5, 11.
setImageSize width, height	Sets the width and height for all images in pixels.
setLayout across, down, margin	Sets the number of images across and down on each page. The margin is in inches and is used to set the spacing around each image. The default is one image per page. Using this method will print "Demo" across each page in the free version.
setNotes(directory)	A directory containing .txt files with the same basename as the images. For example, if an image is named sunset.jpg the program will look for sunset.txt in the directory specified with this option. A pop-up note is added to the page the image is on if the text file is found. The contents of the note are taken from the text file. Using this option will print "Demo" across each page in the free version.

PDF Image Stream

DLL Properties and Methods

Method	Description
setSearch	Use along with the setNotes method. This option places the notes text as invisible text on the page behind the image. This allows you to use the search feature to lookup an image in the PDF based on this text.
setSearchOnly	Use along with the setNotes method. Works the same as the setSearch method except the pop-up notes are not placed in the PDF. The notes are only used as searchable text in this case.
setBackground(file)	An image file to use as the background for each page. Using this method will print "Demo" across each page in the free version.
setInterpolate	Applies an image interpolation algorithm during rendering. This helps to smooth out the transition between pixels when the resolution of the image is lower than that of the output device. At higher magnification levels images on the screen typically appear jagged. This option removes most of the jagged appearance.
setNoDPI	Images over 72 dpi are automatically scaled down to retain their original height and width. No resolution is lost when this is done - the image is simply scaled to match its height and width based on 72 dpi. This method turns off the automatic scaling.
setDPI(number)	Sets the dpi for images in some situations where the program might not be able to determine the dpi of the image. This is only needed when the image should be scaled and the automatic scaling is not happening.
setMargin(width)	Sets the width of the margin in inches. Default is .75.
setPageRotate(angle)	Sets the rotation angle (clockwise) for each page in the PDF. Must be either 0 (the default), 90, 180 or 270.
setImageRotate(angle)	Sets the rotation angle (clockwise) for each image in the PDF. May be any value from 0 to 360.
setHeader (text)	Text for the page heading. Use for line breaks.

PDF Image Stream

DLL Properties and Methods

<u>Method</u>	<u>Description</u>
setHeaderPoint (number)	The font point size for the heading text. Default is 12.
setHeaderOffsetY (number)	The offset in points (1/72 of an inch) for the header. Positive values move the heading down, negative move it up.
setFooter (text)	Text for the page footer. Use for line breaks.
setFooterPoint (number)	The font point size for the footer text. Default is 12.
setFooterOffsetY (number)	The offset in points (1/72 of an inch) for the footer. Positive values move the footer down, negative move it up.
setCaption	Displays the caption at the bottom of the image. The caption is the file name unless specified using an input file or the setCaptionText method. A comma separates the caption from the file name in the input file. Use a tag for a line break. Note the text may flow into the image area depending on the image size and amount of text. See the Input File section for details on using an input file.
setCaptionColor(color)	The color for the caption text as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
setCaptionColorBG(color)	The color for the background for the caption text as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
setCaptionOffsetY(number)	An offset in points (1/72 of an inch) to adjust the caption location by. A positive value moves the caption down while a negative value moves it up.

PDF Image Stream

DLL Properties and Methods

<u>Method</u>	<u>Description</u>
setTransparency x [,text]	<p>Used to specify the transparency value and mode. Requires Acrobat or Reader 5.0 or higher to view the transparency. This is the transparency for the caption background when you have multiple lines that overlay the image.</p> <p>The default mode, Normal, uses the transparency value to determine how transparent the text background will be. The opacity value is from 1 to 100. For example, in Normal mode, the higher the number, the less of the background shows through. Optionally, specify the mode of transparency. The valid values are:</p> <ul style="list-style-type: none">Normal (Default)MultiplyScreenOverlayDarkenLightenColorDodgeColorBurnHardLightSoftLightDifferenceExclusionHueSaturationColorLuminosity <p>Example: -transparency 50 or -transparency 75,Screen</p>
setBookmarks	<p>Add bookmarks. The description for the bookmark is the same as the caption.</p>

PDF Image Stream

DLL Properties and Methods

Method	Description
setCaptionText(text)	<p>The caption for each image. The following may be used as variables in the string for the caption:</p> <ul style="list-style-type: none">%f - file name%d - directory%b - file size%w - width%h - height <p>For example, "Image %f - %b, %wX%h" will come out looking something like "Image abc.jpg - 25148, 200X350". Be sure to also use the setCaption method to show the caption in the PDF.</p>
setPageNum	Adds page numbers centered at the bottom of each page.
setPoint(size)	Sets the point size used for captions and page numbers. Default is 12.
setFit("image" or "page")	Set to "image" to size each page to the size of the image + the margin. Set to "page" to size each image to the size of the page - the margin. Sizing to "page" will make the image large enough to fit either width or length wise without losing the aspect ratio. That is, the image will not be stretched more in one direction than the other. The default is to leave the image as is centered on the page and only reduce in size if it doesn't fit on the page.
setZoom(percent)	Sets the zoom factor to use when the document is opened. Setting to 100 means to open the document at 100%.
setLandscape	Sets page to landscape mode. This is 11 by 8.5 unless the setPageSize method was used.
setBorder(width)	Places a border around each image at the specified width. The image is modified to include the border. Use the setLineBorder method to instead draw the border in the PDF without changes to the image. This results in faster execution when you don't need a raised or sunken border.

PDF Image Stream

DLL Properties and Methods

<u>Method</u>	<u>Description</u>
setLineBorder	Draws the border using line commands in the PDF. Use this method for faster execution and generally smaller PDF size when you don't need to create raised or sunken image borders. Use along with the setBorder method.
setBorderColor(color)	The color of the border as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
setBorderRaised(width)	The width for a raised border. You do not have to use the setBorder method along with this one. The edges of the image are lightened/darkened to provide a raised effect.
setBorderSunken(width)	The width for a sunken border. You do not have to use the setBorder method along with this one. The edges of the image are lightened/darkened to provide a sunken effect.
setTrans(text)	One of the following transition types: Split Blinds Box Wipe Dissolve Glitter Random The default page transition is to just replace the current page with the new page. The effects listed above can be used for presentation purposes. The random option shown above will pick one of the transition types at random for each page.
setTransSecs(number)	The length of time in seconds to spend on the page transition. Default is 1 if a transition type is used.
setTransDir(H or V)	The direction of the transition, horizontal or vertical (H or V). Valid only for Split and Blinds transition types.
setTransMDir(I or O)	The direction of motion for the transition effect, inward or outward (I or O) from the center of the page. Valid only for Split and Box transition types.

PDF Image Stream

DLL Properties and Methods

Method	Description
setForce	Turns off the prompt to overwrite the output file if it already exists.
setPDFTitle(text)	Sets the title of the PDF.
setPDFAuthor(text)	Sets the author of the PDF.
setPDFSubject(text)	Sets the subject of the PDF.
setPDFCreator(text)	Sets the creator of the PDF.
setPDFProducer(text)	Sets the producer of the PDF.
setPDFKeywords(text)	Sets the keywords for the PDF.
<i>The following options are used when creating images (these are features of the commercial version only)</i>	
setMake(type)	Sets the type of images to create. One of the following: bmp, jpg, gif, tif or png. The output file name should be set to the directory you want the images placed in or the current directory will be used. Images created will be of the same base name as the original but not overwrite any existing images. For example, if you converting jorges to gis and you have a file called mypic.jpg you'll get a file called mypic.gif. If you convert again, you'll get an image called mypic_1.gif and, if that name exists mypic_2.gif and so on.
setQuality("high" or "low")	Sets the quality of jpeg images created to high (larger file size but better image quality) or low (smaller file size and less quality).
setXScale(number)	Amount to scale each image in the x direction entered as a percentage. For example, 50% will scale all images by 50 percent in width while 50 will set each image to 50 pixels wide.
setYScale(number)	Amount to scale each image in the y direction entered as a percentage. For example, 50% will scale all images by 50 percent in height while 50 will set each image to 50 pixels high.

PDF Image Stream

DLL Properties and Methods

Method	Description
setIMOff	Turns off the standard processing of images that converts them to PDF. Instead, images are taken directly to PDF with little or no modification. This allows for faster conversion and sometimes smaller PDF size but does not work on as many image types. The images must be either a jpeg using 256 color grayscale or 24-bit color or a 256-color gif non-interlaced, version 87a. Some types of tiffs are supported as well - must be black & white, packbits or CCITT encoded with a single image strip.
setThumbnail(size)	Use this option to create one thumbnail image containing all the individual images. Set to the size of each image along with spacing in pixels formatted as #x#+## where the # sign is a number of pixels. For example, 50x75+5+10 means create images 50 pixels wide by 75 pixels high and leave 5 pixels in the x direction between images and 10 pixels in the y direction.
setFlashFile(path-file)	Creates a flash file containing the pages of the PDF. You access the pages by clicking on the flash movie to activate then use the page-up and page-down keys to move from page to page. You may also click on the image to advance to the next one. Do not enter the .swf extension when using the method setFlashMult. Note that not all images may translate to flash output so you'll want to test on your target platform with the types of images you plan on using. As a general rule, jpeg images should work the best when converting to flash.
setFlashMult	Creates a flash file for each page of the PDF. The default is to build a single flash file containing all the pages. Do not enter the .swf extension for the setFlashFile method when using this option. The output files will be named using the file name passed along with the page number. For example, if you use setFlashFile("myfla") and the PDF has 3 pages, you will get three files - myfla_1.swf, myfla_2.swf and myfla_3.swf.

PDF Image Stream

DLL Properties and Methods

Method	Description
setFlashBG(color)	Optional HEX formatted color for the flash file background. The default is FFFFFFFF or white.
setFlashScale(number)	A scaling factor for the flash output. The default is 100 for 100%.
setFlashPreScale(number)	A scaling factor for the images prior to being converted to flash output. The default is 100 for 100%. This can be useful to reduce the flash file size.
<i>The following options are available only when using the -thumbnail option</i>	
setTNAcross(number)	The number of thumbnail images to place in a single row. For example, setting to 5 means put five images across from one another before dropping down to the next row to start another five images.
setTNBGcolor(color)	The background color in hex form - red, green, blue pairs. For example, #000000 is black and #FFFFFF is white.
setTNCap(text)	The caption for each image in the thumbnail. The following may be used as variables in the string for the caption: %f - file name %d - directory %e - extension %b - file size %w - width %h - height For example, "Image %f - %b, %wX%h" will come out looking something like "Image abc.jpg - 25K, 200X350".
setTNPoint(size)	The pointsize for the captions. Default is 10.
setTNFont(path-file)	The truetype font file to use for the caption.
setTNSHadow	Set this to place a shadow behind each thumbnail image.
setTNFrame	Set this to place a frame around each thumbnail image.

PDF Image Stream

Examples

Create a single file with page numbers and bookmarks

```
Set PDF = CreateObject("Image.Stream")
PDF.setDirectory ("d:\my documents")
PDF.setMask ("*.jpg,*.gif")
PDF.setOutput ("d:\pdf\mypics.pdf")
PDF.setPageSize 6, 9
PDF.setBookmarks
PDF.setSubDir
PDF.setCaption
PDF.setCaptionText("%f")
PDF.setPageNum
PDF.setZoom(100)
PDF.buildImage
Set PDF = Nothing
```

Create a single file with page numbers and bookmarks and size to image

```
Set PDF = CreateObject("Image.Stream")
PDF.setDirectory ("d:\my documents")
PDF.setMask ("*.jpg,*.gif")
PDF.setOutput ("d:\pdf\mypics.pdf")
PDF.setBookmarks
PDF.setSubDir
PDF.setFit("image")
PDF.setZoom(100)
PDF.buildImage
Set PDF = Nothing
```

Get the list of images to convert from a file called mylist.dat

```
Set PDF = CreateObject("Image.Stream")
PDF.setDirectory ("@d:\my documents\mylist.dat")
PDF.setOutput ("d:\pdf\mypics.pdf")
PDF.setBookmarks
PDF.setSubDir
PDF.setFit("image")
PDF.setZoom(100)
PDF.buildImage
Set PDF = Nothing
```

PDF Image Stream

Examples

Scan the directory "d:\my documents" and all subdirectories for jpegs and create a single PDF for each image in the directory d:\pdfpics.

```
Set PDF = CreateObject("Image.Stream")
PDF.setDirectory ("d:\my documents")
PDF.setMask ("*.jpg")
PDF.setOutput ("d:\pdfpics")
PDF.setBookmarks
PDF.setSubDir
PDF.setSingle
PDF.buildImage
Set PDF = Nothing
```

The following two examples only apply to the commercial version of the software.

Scan the directory "d:\my documents" and all subdirectories for jpegs and convert to gif images to be placed in d:\gifpics.

```
Set PDF = CreateObject("Image.Stream")
PDF.setDirectory ("d:\my documents")
PDF.setMask ("*.jpg")
PDF.setOutput ("d:\pdfpics")
PDF.setSubDir
PDF.setMake ("gif")
PDF.buildImage
Set PDF = Nothing
```

Scan the directory c:\images for gifs and jpegs and create a thumbnail image. Size the thumbnails to 75 by 100 with 5 pixels between images. Allow up to 8 images across and place a caption below each image.

```
Set PDF = CreateObject("Image.Stream")
PDF.setDirectory ("c:\images")
PDF.setMask ("*.jpg,*.gif")
PDF.setOutput ("d:\pdfpics\mythumb.gif")
PDF.setThumbnail ("75x100+5+5")
PDF.setTNAcross (8)
PDF.setTNCap ("File %f")
PDF.buildImage
Set PDF = Nothing
```

PDF Image Stream

Examples

An example calling the DLL from PowerBuilder

```
OLEObject PDF
PDF = CREATE OLEObject
li_rc = PDF.ConnectToNewObject("Image.Stream")
PDF.setDirectory ("d:\my documents")
PDF.setMask ("*.jpg,*.gif")
PDF.setOutput ("d:\pdf\mypics.pdf")
PDF.setPageSize 6, 9
PDF.setBookmarks
PDF.setSubDir
PDF.setCaption
PDF.setCaptionText("%d%f Size %w x %h")
PDF.setPageNum
PDF.setZoom(100)
PDF.buildImage
```

Here is an ASP example sending the output directly to the browser. Note the use of the fytek.unicode object. The VBScript will treat the returned PDF stream as Unicode (2-byte characters) which will not work with Response.binaryWrite. The method StrToByte will convert the Unicode string into a single byte string which can be streamed to the browser. This method is in the file fytek.dll included with the installation.

```
<%
Dim PDF
Dim binaryData
Set obj = Server.CreateObject("fytek.unicode")
Set PDF = Server.CreateObject("Image.Stream")
PDF.setDirectory ("d:\my documents")
PDF.setMask ("*.jpg,*.gif")
PDF.setPageSize 6, 9
PDF.setBookmarks
PDF.setSubDir
PDF.setCaption
PDF.setCaptionText("%f")
PDF.setPageNum
PDF.setZoom(100)
pdfOut = PDF.buildImage
binaryData = obj.StrToByte(pdfOut)
Response.ContentType = "application/pdf"
Response.binaryWrite binaryData
set PDF = nothing
set pdfOut = nothing
set obj = nothing
set binaryData = nothing
%>
```

PDF Image Stream

Examples

Here's a VB script that reads an image from a file and passes it to PDF Image Stream. This is to show how to pass data from a byte array in memory rather than a file on disk.

```
Dim b() As Byte
Open "mypic.jpg" For Binary As #1
ReDim b(1 To LOF(1))
Get 1, , b
Close #1
. . .
PDF.setFileDataB "image1.jpg", b
```

Here's a C# script that reads an image from a file and passes it to PDF Image Stream. This is to show how to pass data from a byte array in memory rather than a file on disk. Note that marshalling the byte array can be slow - if you find it too slow, consider using setFileDataH that takes a hex string instead of a byte array.

```
string FileName = "mypic.jpg";
FileInfo oFileInfo = new FileInfo(FileName);
long FileSize = oFileInfo.Length;
BinaryReader FS = new BinaryReader(File.Open(FileName, FileMode.Open));
byte[] Buffer = new byte[FileSize];
Buffer = FS.ReadBytes((int) FileSize);
FS.Close();
. . .
PDF.setFileDataB ("image1.jpg", Buffer);
-- OR --
string Hexs = BitConverter.ToString( Buffer );
PDF.setFileDataH("image1.jpg", Hexs);
```

Of course, you can use any other method to create the byte array. Most likely you'll want to use this method in situations where you're creating the file dynamically from some other software and want to pass the output into PDF Image Stream without creating a disk file.

PDF Image Stream

Examples

Here is an example using C.

```
#include <iostream.h>

// The import directive reads the typelib information from the DLL
// and creates pdfimage.tlh and pdfimage.tli, which are included.
// These define wrappers for each of the pdfimage object methods.

#import <pdfimage.dll>

// Using VC++ 5.0 Smart Pointers makes this much easier.
// The parameter string for a method is converted to Unicode, allocated
// and passed as a variant. The wrappers call IDispatch::Invoke
// This is all compatible with MFC (use AfxOleInit instead of CoInitialize, etc.).
int main(int argc, char* argv[])
{
    HRESULT      hr;

using namespace PDFImageStream_TypeLib;

    hr = CoInitialize (NULL);    // Initialize COM
    if (SUCCEEDED(hr))
    {
        try    // Each of the following lines can throw exceptions
        {
            // Create the instance and get a pointer to the interface
            IPDFImageStreamPtr pPDF(__uuidof(PDF_Image_Stream));
            pPDF->setDirectory (_bstr_t(L"d:\\my documents"));
            pPDF->setMask (_bstr_t(L"*.*.jpg,*.gif"));
            pPDF->setOutput (_bstr_t(L"d:\\pdf\\mypics.pdf"));
            pPDF->setPageSize (_bstr_t(L"6"), _bstr_t(L"9"));
            pPDF->setBookmarks ();
            pPDF->setSubDir ();
            pPDF->setCaption ();
            pPDF->setCaptionText (_bstr_t(L"File %f"));
            pPDF->setPageNum ();
            pPDF->setZoom (_bstr_t(L"100"));
            _variant_t outval = pPDF->buildImage (); // Build the PDF file
        }
        catch (_com_error e)
        {
            cout << e.ErrorMessage() << endl;
        }
    }
    else
        cout << "CoInitialize Failed" << endl;

    CoUninitialize();    // Uninitialize COM

    return 0;
}
```

PDF Image Stream

Input Files

An input file can be used to specify the list of images and captions to use. For example:

```
d:\my images\pic1.jpg,At the cottage
d:\my images\pic2.gif,Ski trip<BR>photo taken November 20, 20xx
d:\my images\pic3.gif
d:\my images\pic4.jpg,View of the lake from porch
```

Each image must be on a separate line when using an input file. The caption must be on the same line as the image and not span lines in the file.

An input file may also be tagged based. This allows the added flexibility of changing the page layout and storing other options in the file itself. For example:

```
<IMAGESTREAM CAPTION>
<PAGE LAYOUT="1,2,.5">
<IMG SRC="d:\my images\pic1.jpg">
At the cottage
</IMG>

<IMG SRC="d:\my images\pic2.gif" BORDER=3>
Ski trip<BR>photo taken November 20, 20xx
</IMG>

<IMG SRC="d:\my images\pic3.gif" ROTATE=90>
</IMG>

<IMG SRC="d:\my images\pic4.jpg">
View of the lake from porch
</IMG>
```

Each tag goes between the < and > symbols. Each tag goes on a line by itself. The file must start with the tag IMAGESTREAM.

For images (the IMG tag), the description is placed between the opening IMG tag and the corresponding closing /IMG tag. Each IMG tag must have a closing /IMG tag.

The VALUE for the LAYOUT tag uses the same options as the -layout option or setLayout method. It is used to specify the number of images across on a page, the number down and, optionally, the spacing in inches around each image.

Be sure to use the -cap option or setCaption method to show the caption in the PDF. You can also set this with the CAPTION option on the IMAGESTREAM tag.

For Executable:

Note the file input mask is left out in this case and the input file name is preceded by the @ symbol:

```
pdfimage.exe @mylist.dat mypics.pdf
```

For DLL:

Use a '@' in front of the file name and pass to setDirectory. For example, PDF.setDirectory("@d:\my documents\mylist.dat"). Leave the setMask method out.

PDF Image Stream

Input Files

```
<IMAGESTREAM
FIT="text"
CAPTION
CAPTIONCOLOR=color
CAPTIONCOLORBG=color
CAPTIONOFFSETY=number
TRANSPARENCY="text"
BOOKMARKS
PAGENUM
POINT=number
INT
IMOFF>
```

Place this tag at the top of the file so it is recognized as a tagged based input file. All of the settings are optional.

<u>Parameter</u>	<u>Description</u>
FIT="text"	Set to "image" to size each page to the size of the image + the margin. Set to "page" to size each image to the size of the page - the margin. Sizing to "page" will make the image large enough to fit either width or length wise without losing the aspect ratio. That is, the image will not be stretched more in one direction than the other. The default is to leave the image as is centered on the page and only reduce in size if it doesn't fit on the page.
CAPTION	Show the image captions. May also be set with the -cap option or setCaption method.
CAPTIONCOLOR=color	The color for the caption text as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
CAPTIONCOLORBG=color	The color for the background for the caption text as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
CAPTIONOFFSETY=number	An offset in points (1/72 of an inch) to adjust the caption location by. A positive value moves the caption down while a negative value moves it up.

PDF Image Stream

Input Files

<u>Parameter</u>	<u>Description</u>
TRANSPARENCY="text"	<p>Used to specify the transparency value and mode as "number" or "number,type". Requires Acrobat or Reader 5.0 or higher to view the transparency. This is the transparency for the caption background when you have multiple lines that overlay the image.</p> <p>The default mode, Normal, uses the transparency value to determine how transparent the text background will be. The opacity value is from 1 to 100. For example, in Normal mode, the higher the number, the less of the background shows through. Optionally, specify the mode of transparency. The valid values are:</p> <p>Normal (Default) Multiply Screen Overlay Darken Lighten ColorDodge ColorBurn HardLight SoftLight Difference Exclusion Hue Saturation Color Luminosity</p> <p>Example: "50" or "75,Screen"</p>
BOOKMARKS	Add bookmarks. The description for the bookmark is the same as the caption.
PAGENUM	Adds page numbers centered at the bottom of each page.
POINT=number	Sets the point size used for captions and page numbers. Default is 12.

PDF Image Stream

Input Files

	<u>Parameter</u>	<u>Description</u>
	INT	Applies an image interpolation algorithm during rendering. This helps to smooth out the transition between pixels when the resolution of the image is lower than that of the output device. At higher magnification levels images on the screen typically appear jagged. This option removes most of the jagged appearance.
	IMOFF	Turns off the standard processing of images that converts them to PDF. Instead, images are taken directly to PDF with little or no modification. This allows for faster conversion and sometimes smaller PDF size but does not work on as many image types. The images must be either a jpeg using 256 color grayscale or 24-bit color or a 256-color gif non-interlaced, version 87a. Some types of tiffs are supported as well - must be black & white, packbits or CCITT encoded with a single image strip.

PDF Image Stream

Input Files

```
<FOOTER  
  SIZE=number  
  OFFSETY=number>  
[footer text]  
</HEADER>
```

Used to set the page footer. The footer text is placed between the opening and closing tags.

<u>Parameter</u>	<u>Description</u>
SIZE=number	The font point size for the footer. Default is 12.
OFFSETY=number	The offset in points (1/72 of an inch) for the footer. Positive values move the heading down, negative move it up.

PDF Image Stream

Input Files

```
<HEADER  
  SIZE=number  
  OFFSETY=number>  
[footer text]  
</HEADER>
```

Used to set the page header. The header text is placed between the opening and closing tags.

<u>Parameter</u>	<u>Description</u>
SIZE=number	The font point size for the header. Default is 12.
OFFSETY=number	The offset in points (1/72 of an inch) for the header. Positive values move the heading down, negative move it up.

PDF Image Stream

Input Files

```
<IMG  
  SRC="text"  
  ROTATE=number  
  BORDER=number  
  BORDERCOLOR=color  
  X=number  
  Y=number  
  XSCALE=number  
  YSCALE=number>  
[caption]  
</IMG>
```

Used to insert an image. The caption, if any, is placed before the closing /IMG tag.

<u>Parameter</u>	<u>Description</u>
SRC="text"	Only required option for the tag. The path and file name of the image you wish to include. Place quotes around this value. You may also use a web address that starts with http:// to pull an image from a web site. Note that the image will be stored locally while the PDF is built. Also, downloading large images over a slow modem will take time.
ROTATE=number	Sets the rotation angle (clockwise) for the image. May be any value from 0 to 360.
BORDER=number	Draws a border around the image at the specified thickness using line commands in the PDF.
BORDERCOLOR=color	The color of the border as a hex value of red, green and blue values from 00 to FF. Use a # in front of the value. For example, black is #000000 while a shade of blue would be #3366FF.
X=number	Width of image in points (default is the image width).
Y=number	Height of image in points (default is the image height).
XSCALE=number	Scaling factor percentage (default value is 100).
YSCALE=number	Height of image in points (default value is 100).

PDF Image Stream

Input Files

```
<PAGE  
  HEIGHT=number  
  WIDTH=number  
  MARGIN=number  
  UNITS="text"  
  LAYOUT="text">
```

Used to setup the page size or modify the layout. May use this tag multiple times throughout the file. You may set either the page height/width or the layout or both.

This option causes a page break and begins a new page of the specified size and layout. Page breaks will continue when necessary throughout the file so it's not necessary to set this for every image or set of images that occupy a page.

<u>Parameter</u>	<u>Description</u>
WIDTH=number	Sets the page width based on the UNITS setting. If UNITS are not used, the value is assumed to be in points (1/72 of an inch).
HEIGHT=number	Sets the page height based on the UNITS setting. If UNITS are not used, the value is assumed to be in points (1/72 of an inch).
MARGIN=number	Sets the page margin based on the UNITS setting. If UNITS are not used, the value is assumed to be in points (1/72 of an inch).
UNITS="text"	The unit of measure the WIDTH, HEIGHT and MARGIN are specified in. Use one of the following: in - Inches cm - Centimeters mm - Millimeters pt - Points
LAYOUT="text"	Sets the number of images across and down on the page. The margin is in inches and is used to set the spacing around each image. Using this option will print "Demo" across each page in the free version.