

# PStill

PS/EPS/PDF/JPEG/TIFF to PDF converter

Version 1.65 covered

Copyright Dipl.-Ing. Frank Siegert  
Stauferweg 35, 72555 Metzingen,  
Germany, eMail frank@wizards.de

THIS DOCUMENT IS PART OF THE PSTILL DISTRIBUTION AND MAY NOT BE DISTRIBUTED OR PUBLISHED WITHOUT PERMISSION OF THE AUTHOR.

Usage is restricted by the conditions as stated in the license agreement, see Appendix for the COPYRIGHT NOTICE and the text file 'pstill-license.txt'

## 1. Introduction

This document describes the installation and usage of the PStill/Win product. Its intended audience are users and system administrators. Parts of it, describing the usage of the command line version also apply to the other command line versions of PStill (available for Sun Solaris, SGI IRIX, HP UX, IBM AIX, FreeBSD and Linux at the time of this writing). Users only interested in the command line version may skip the parts about the Windows installation and continue at section 4, however the description of the options in 2.5 may be helpful to them as these correspond to the command line options.

### 1.1 What is PStill

PStill is a EPS/PostScript to PDF file converter. It works by reading and interpreting a PostScript or EPS file and write PDF as output. As alternative input also PDF, TIFF and JPEG files are allowed. PStill can also combine several input files – as mix of all possible input formats – to PDF.

### 2.1 Installation of PStill for Windows (PStill/Win)

The PStill version for MS Windows is distributed as installer application. You need at least Windows 95 to run it, also 98, 98SE, ME, NT, 2000 and XP are supported, Windows 2000 or XP are recommended. Also you need at least 100 MByte free space on C: (for temp files) and 64 MByte of memory. PStill works best the more memory it can get, 256 MByte is recommended, 512 MByte or more is optimum. It needs at least a “Pentium” type processor, it does not yet make use of SSE, SSE2 or MMX but really likes large caches and fast floating point.

You can start the installation process by double clicking on the installer icon. The installation process will create a directory on your hard disk containing all the program helper files and the program executable itself. You can specify the location of this directory during install.

As soon as you start the installer by double clicking its icon it will show you the programs licenses agreement, please read it. By continuing the installation you accept your consent with the conditions and rules of the agreement.

**Note:** If you install a new release of PStill/Win over an old installation, the Installer will overwrite the files 'annots.sl' and 'fontsub.table' you might have edited. Please backup these files before installation if you have modified them!

Deinstallation is just as easy, just open up the 'System'->'Software' preferences of Windows and select "PStill PostScript to PDF converter (remove only)". PStill also create the small text file "pstill.ini" in the Windows system directory where it stores its setting. It is not erased during deinstallation, so you may want to remove it manually in case you like to remove any trace of PStill from your system. Under normal circumstances this small file does not pose any problem and it is perfectly safe to keep it where it is.

## 2.2 Using PStill the first time

When installation has finished PStill should be available on your desktop and the Windows start menu. Please locate the program icon, it should look like this:



PStill is started by double clicking on the program icon.

When you start the first time PStill will display a dialog asking about how fonts should be initialized. You have two choices:

1. [**Preferred**] Let PStill create a default base font set from the system fonts.

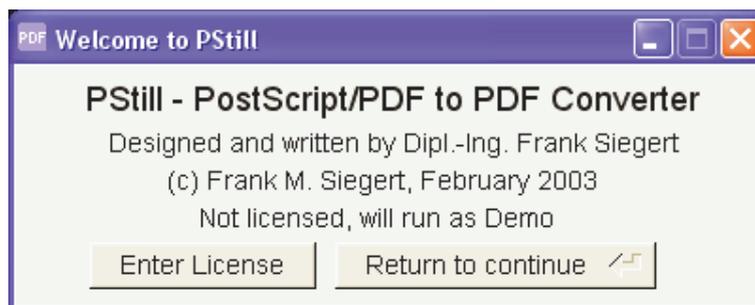
This will usually produce good results but the widths, glyphs and metrics of the fonts may not be accurate enough for highly sensitive artwork. For office applications and normal artwork work however this base font set is usually fine. Use this method if you do not know about PostScript fonts.

2. You may choose to provide PStill with your own fonts.

Go this path if your application demands best precision and you have access to the base font set: Courier, Helvetica, Times-Roman and Symbol. See 'Font installation' later in this document on how to provide the fonts to PStill.

If you choose Method 1. font metric calculations are now taking place. This may take a few seconds up to a few minutes depending on the system you run PStill on. Please do not interrupt this process, otherwise you might find a not fully installed font set which may cause problems later - and this state could only be resolved by a reinstallation of PStill.

As soon as PStill displays its 'Welcome' window it is ready for use and setup:



## 2.3 Licensing PStill

PStill will ask you now for a license code. It is „Shareware“ - that means you can tryout the program but if you want to use it you need to purchase a license code to activate it.

If you have already a license code please choose 'Enter License' and enter your license code in the licensing dialog. The code should look similar to this sample:

***abcdefghijkl=01a02b03c04d05e!ABCDEFGHIJ***

where the first string is a unique identifier of your installation. Please note that the key is case sensitive on input, so be sure to enter it exactly as provided in the registration information you have received from Dipl.-Ing. Frank Siegert or one of his code distributors.

To get a license code or program update please visit the PStill web page:

<http://www.pstill.com>

or open the PDF document 'registration.pdf' which is located in the PStill folder (in the subfolder *docs*). It will explain the registration process by letter. Online ordering by credit card is possible by following the link on the PStill web page.

With your license fee you are directly funding future development of PStill.

## 2.4 Sample session - Conversion of a PostScript file to PDF

To continue with your evaluation click on 'Return to continue', type your 'Return' key. A licensed version of PStill does show the Welcome window only a very short time and you will see the main window almost instantly.

PStills main dialog will now appear. This is the window you will interact with to convert files to PDF, change settings, view the log, etc.



If you have installed PStill using method 1. - that means you choose to have PStill create the base font set, widths and metrics from the system fonts - you are now ready to convert.

Locate a PostScript, EPS, TIFF, JPEG or PDF file using the Windows Browser or Windows Explorer and drag&drop it on this window. Use the supplied *testfile.ps* for your first tests, you can find it in the PStill folder - that is where you just had told the installer to place the program file, by default <c:\Program Files\GPStill>.

Your PStill window will now look like this:



**Hint:** You need to drag&drop onto the window area, but not on the text fields! Otherwise only the text field content is affected. Also dragging multiple jobs will work only if you drag them on the window. Use e.g. the status field 'PStill is ready - ...' as a drop target.

Now click on 'Start Conversion' to let PStill begin its works with the current settings. PStill should start to interpret the PostScript input file and then collect all objects it encounters in a PDF file.

Look into the folder where you dragged testfile.ps from - it should now contain a file *testfile.pdf*.

Double click on it should open your PDF viewer of choice (if installed, otherwise see the appendix for suitable PDF viewers and where to get them). Alternatively you can also click on the button "Open PDF" and show PStill the path to your PDF Viewer program, see the section explaining the *Reset PDF Viewer Button* below to get some insight how PStill starts a remote viewer. E.g. for Acrobat Reader just show GPStill where it is installed and locate the 'Acro32.exe' program file.

Do you see the 'Thumbs up' document in your PDF viewer? Welcome to PStill!

In case you see only 'black dots' instead of the text you need to install your font set before starting a conversion run. At least 'Courier' should be provided to PStill so it has some fallback in case a font is not found. Remember PStill needs all fonts used in the PostScript job installed, if the input file does not contain them (has them embedded).

If a font is not found you will see Courier instead - a message is printed in the log - and since Courier is a monospace'd font most likely your text flow in the PDF will look distorted! This will go away as soon as you provide PStill with the fonts you used to create the PostScript input file.

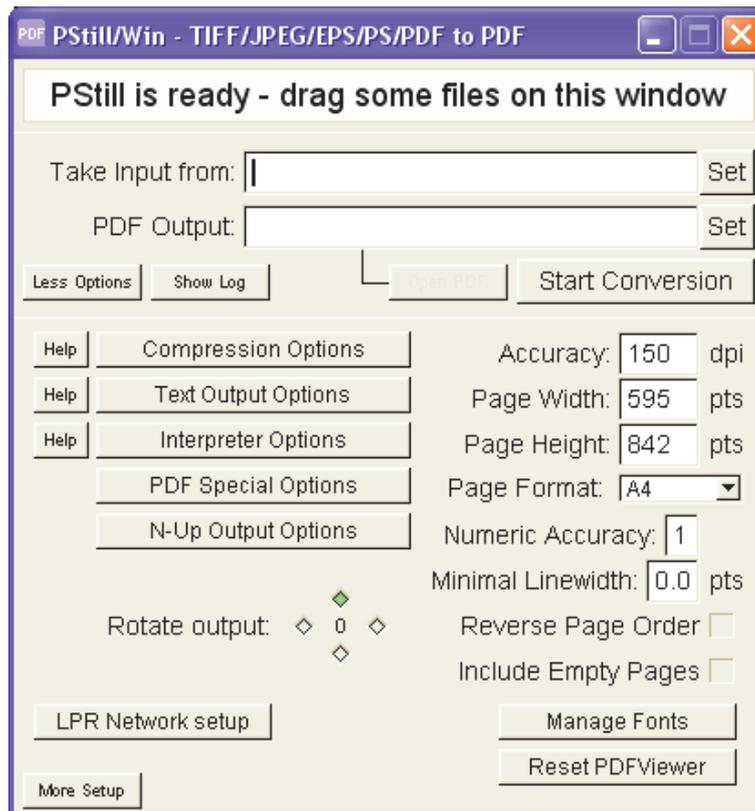
**Important Notice:** If you want to convert your own files right away be sure to setup the system printer driver first according to the needs of PStill. See Section 3.2 (page 16) of this document on how to setup.

## 2.5 Conversion Options

Now on to the Options. By default PStill will use some commonly suitable setting however you might want to change a few later. To open the option section click on the 'More Options' button, the PStill main window will expand to show the option page below the main controls.

**Hint:** Don't panic because there are so much options! There are ready made 'sets' of option for Office, Web and Prepress work. Just open 'More setup' and select the option set. This will set all options at once so you don't need to fiddle with the option dialogs. Using the 'sets' you can also store your own setup under a name and restore it later.

These are the options PStill will show you when installed the first time:



## Options and their effects

*„You can't tune-a-fish but you can tune a PStill“*

By modifying the options you can control how PStill will convert its input - how graphical elements are taken over to the result, how text is converted and what kind of special processing PStill does.

Users of GPStill do not need to worry about the command line options, these are only given for power users of the batchable command line version, they should also read section 4.

**Hint:** *If you want to find out which options are needed for the command line version of PStill you can also use GPStill as its conversion log shows the 'options' as needed for the command line program too. This can be helpful if you want to find out what options to use for a certain type of conversion.*

## Set Compression Options

### **Write flate compressed PDF [default: on, Level 7]**

*Command line options: -c, -c -c, -c -c -c and -c -c -c -c (the more -c the better the compression)*

Will cause PStill to write compressed and binary PDFs. The used lossless compression method is compatible to the PDF specification 1.1 so you need at least Acrobat(R) Reader(TM) Version 3.0 to view these files. You will notice compression makes the PDF files much smaller so it is recommended you leave this option on.

PStill will use this compression for page contents (text and artwork), fonts and raster images. You will notice no visible difference to uncompressed PDFs as the compression method is lossless. Think of it as a kind of “ZIP”ed file (in fact the used method is similar to ZIP).

You can specify a compression level here, from 6 to 9. 7 is the default level and quite fast while 9 is the slowest level but gain a few bytes in output size.

### **Write JPEG-compressed images [default: off]**

*Command line option: -J #, where # is the JPEG quality*

Will cause PStill to write suitable images DCT (another term for 'JPEG') compressed in the output. Suitable images are gray and color raster images. This option has no effect on monochrome bit-level images or mask. The used compression is “lossy”, so the image may degrade in quality.

The quality setting allows to setup the compression level, where 0 means very small output but bad quality and 100 large output but best quality. Good values are 60 to 90.

***Hint:** For some kind of graphics, e.g. screenshots of dialog windows in technical documentations JPEG is entirely unsuited and will cause large and ugly results (this is not a problem of PStill, but a general issue with using JPEG for artificial images, JPEG is best suited for real-world photographs, not for plain surfaces) you will get better results using only the 'flate' compression.*

### **Reduce resolution for raster images [default: off]**

*Command line option: -d placement\_dpi/gray\_dpi/mono\_dpi*

While not strictly a form of „compression“ it is placed in this dialog as it can be used to lower the output file size. By default PStill takes over raster images in their native (input) resolution, however using this option PStill can resample images to a lower DPI setting. This will cause smaller output size (E.g. for web publishing) but is not recommended for PDF generated for printing purposes.

You can specify two values here. The first defines the output dpi for raster images (gray/color), the second for bit-level monochrome images and mask. If PStill encounters an image in the input that is higher in print density than the given values it will resample it in the output. If the image dpi is lower or equal however it will not change the image.

Currently linear rescaling (fixed point accuracy of 32 bit) is used for the downsampling method.

***Hint:** For pure web publishing you should lower the dpi to 72 for color/gray images and 150 dpi for monochrome images. Lower resolutions are also possible, depending on the kind of job.*

The next section explains the text and fonts options:

### Set Text Output Options

#### **Partial (subset) Fonts [default: off]**

*Command line option: -p*

Will cause PStill to include font outlines only for the used glyphs in the resulting PDF file. This will make the result smaller and may be needed depending on your font license agreements (of the fonts you provide to PStill). On the downside some non-standard fonts may cause problems with this options, so if your results may missing text turn this option off. PStill has a build in font checker that will determine in most cases if a font could be subsetted. If the font does not conform it will include it in full regardless of this option. You need to check the log to see if PStill has chosen to include in full or subsetted.

#### **Include Fonts [default: on]**

*Command line option: -i*

Will cause PStill to include fonts that could not be handled by the Multiple Master mechanism in the PDF documents. It is recommended to leave this option on. You should turn it off only to check problematic cases.

#### **Include all Fonts [default: off or unselectable]**

*Command line option: -s*

Will cause PStill to include all and every font in the PDF result. Usually this is not a good idea and may cause larger files without giving much advantage, however some PDF „high end“ workflows needs this option to be turned on. This option may be unselectable if PStill has used the system fonts to create its base font set, only users who did provide their own fonts to PStill at installation can enable it.

#### **Regenerate not-extractable fonts [default: on]**

*Command line option: by default enabled, disable with -W*

Will cause PStill to regenerate problem fonts (that cannot be embedded in the PDF due to their nature. These are in most cases dynamic created fonts by certain application and driver versions. Such fonts may cause problems later so PStill flag them for non inclusion). It is recommended to leave this option on except if you encounter problems. Turning it off causes PStill to emit outlines for such fonts which works all the time, but causes larger output – also screen display is not so good as with fonts. (Hint: Turn on 'smooth vector graphics' in Acrobat Reader for better screen display)

#### **Always convert text to graphics [default: off]**

*Command line option: -P*

Will cause PStill to convert all text to graphics artwork. Depending on the contents of the input it may cause larger output files, however it will solve all problems with embedded fonts - there are no text or fonts in the PDF result anymore, only graphics. Do not enable it for office use, apart from the larger size of the PDFs they become unsearchable/unindexable also. When you create a PDF for

placing in a publication (newspaper ad or similar) this option may help you to create PDF results that do look exactly as you want them, but distributed without giving out any fonts!

**Prefer installed fonts to job fonts [default: on]**

*Command line option: -Q*

Normally PStill will extract embedded fonts from the input and use these in the PDF result even if a font with the same name is available in its own font pool. If you enable this option PStill will however prefer fonts from the font pool to embedded fonts and use and include these in the PDF result.

**Write fonts complete encoded [default: on]**

*Command line option: -K*

This is an important option if you want to postprocess („edit“) the PDF results. By default PStill defines a font encoding (a mapping which glyph can be found under which character value) in relation to an encoding known to PDF, normally the so-called „MacRomanEncoding“. This is perfectly allowed by the PDF specification. However it seems that some postprocessing programs cannot handle this way of specifying an encoding for operations like 'insert pages' and 'edit text' (while displaying them fine). So if you plan to postprocess or „submit“ the PDF turn this option on. It will cause slightly larger PDF results but enhances compatibility of the PDF result.

**Autoconvert dvips bitmap fonts [default: off]**

*Command line option: -D*

This option causes PStill to replace all marked DVIPS (TeX) bitmap fonts inside an input PS job with Type 1 versions. For this purpose the Type 1 version of the fonts (e.g. Bakoma set) must be installed and available for PStill. See the log for more info during a conversion run. Mind that if DVIPS has not written the font names during job creation ('no comments' options or very old version) PStill will not be able to replace the fonts. The log will tell you what happens.

**The next section explains the PostScript and general input processing related options:**

### Set PS/Output Options

#### **PostScript Level 2 Definition [default: on]**

*Command line option: -2*

PStill is a full Level 2 „clone“ PostScript interpreter, supporting 99.9 % of the Level 2 language set and some extensions of Level 3. If you experience problems and PS error you can try to turn this option off. However it is recommended for to leave it on for best compatibility.

#### **Use PS color/gray transfer functions [default: on]**

*Command line option: -t*

Will cause PStill to take over any changes to the color transfer function in the PDF result. Color transfer is used in PostScript to change the appearance of colors or grays, e.g. to display a 'positive' image as 'negative' (inverted). It is advisable to leave this on, except if you experience problems with wrongly colored images and vice versa. This option has no effect on text or artwork, only raster images are affected.

PStill will take over any changes directly by changing the affected color values. It will never include 'transfer curves' or similar in the output result.

#### **PDF output converted to RGB color [default: off]**

*Command line option: -C*

By default PStill will write any color representation as CMYK. This may lead to darker colors in some PDF Viewers however. If your PDF should be printed on a separating workflow (e.g. newspaper ads) and contains CMYK colors just leave this option off as this will produce the best results. If your PDF has too dark colors in a PDF Viewer or is destined for pure office work or web display turn this option on, it will also diminish file size of the PDF result a bit. The output size will also become smaller since three color values takes less space than four.

#### **PDF output converted to Gray [default: off]**

*Command line option: -B*

This option will cause PStill to emit B/W results only. All color values are recalculated to grays. The output size will also become smaller since the data can usually be much better compressed (if you have flate compression turned on).

#### **Convert spot colors to process [default: on]**

*Command line option: turn off with -m Xspot (will cause spot colors to be written as spots)*

This option will cause PStill to convert spot colors to their CMYK (or RGB/Gray depending on your setting of above options) process colors instead and have these written in the output. This is recommended for normal use, only PDF generated for pre-press purposes should use native spot colors.

**Take over overprinting settings [default: off]**

Command line option: `-m Xoverprint`

Overprinting are setup information that only have a meaningful context in a color separation environment. They specify if a graphical element should 'knock-out' (paint in white) on other color plates. This will prevent 'flashing' if the color plates are not positioned exactly for printing, however can lead to color problems if used incorrect. You cannot see overprinting in normal printout or on the screen! So enable this option only if you want to color separate the results and you know that the input job overprinting settings are correct.

**Convert Separation to Composite [default: off]**

Command line option: `-m Xseptocomp`

PStill can auto convert pre-separated PostScript input jobs to composite color results simulating the process of a printing machine. The results will be usable for e.g. Web or proof purposes. This option is only available if you have purchased the separate „S2C“-Option and the „septocomp“-module is installed.

**Image gamma correction [default: 1.0 – no change]**

Command line option: `-G #.#`, where #.# is the gamma value

PStill can apply „gamma“ correction to images in the output, making them darker or lighter in appearance. This option has only effect on raster images (gray/color), monochrome bit-level images and mask are not affected.

**PDF/X Options**

Command line options: `-m XPDFX="INTENT" -m XICCPProfile=path\to\profile -m -m XPDFXVERSION=1A` (optional, otherwise PDF/X-3)

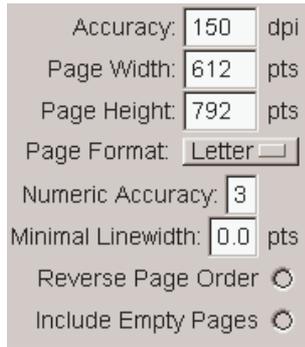
PDF/X is a focused subset of PDF for 'secure' production of PDF.

Please see the PDF/X info in GPStill and check out <http://www.pdfx.info>

This option is not needed for normal desktop/office work. If you need to submit PDFs for publishing purposes however it can become quite handy. Your publisher will tell you the values for the intent name / version and the color profile in this case.

***Note:** If you supply a color profile here, PStill will only include it in the PDF result, but not use it otherwise (as per PDF/X spec.). The PDF result can get larger, so if you get suddenly very large PDF results be sure to uncheck this option – maybe you have left it on for some reason.*

**In the main option windows you can find the page setup and output size options:**



The image shows a screenshot of a page setup dialog box. It contains several input fields and checkboxes. The fields are: Accuracy (150 dpi), Page Width (612 pts), Page Height (792 pts), Page Format (Letter), Numeric Accuracy (3), and Minimal Linewidth (0.0 pts). There are two checkboxes: Reverse Page Order and Include Empty Pages, both of which are currently unchecked.

**Accuracy [default: 150 dpi]**

*Command line option: -d #, where # is between 72 and 1000*

*See also 'Compression setup' and 'reducing image density' for extended options*

This is the internal accuracy PStill calculates any placement of elements with. It defines the device raster of the interpreter which is used to 'flatten' paths. The 'true' resolution is much higher, at least double this value.

150 is usually fine for normal office work, however if your PDF result shows

- slightly misplaced characters in text (usually horizontally by a tiny amount)
- contains high resolution graphics (e.g. CAD or fine artwork drawings)
- demands the best precision

change it to higher values, 300 dpi or 600 dpi are usually good choices. Please note PStill will need more memory for larger values and conversion time may increase a bit. 600 actually means an effective resolution of 1200 dpi for element placement, so don't get carried away by using very large values – they are often unnecessary and increase only processing time!

**Page Width [default: 612 pts (US letter)]**

*Command line option: -w #, where # is the width in dpi (1/72 inch)*

and

**Page Height [default: 792 pts (US letter)]**

*Command line option: -h #, where # is the height in dpi (1/72 inch)*

Here you can tell PStill how large you like to have the page size of the resulting PDF document. Any graphics outside this area will be clipped.

**Page Format [default: letter]**

*Command line option: -F <format>, where <format> is a2-a5, letter or legal*

Here you can tell PStill how large you like to have your page size by selecting a format directly, so you don't need to enter the width/height values yourself. If you choose 'job' here PStill will try to determine the page size from the contents of the input PostScript or EPS file. Please note that PStill can only make the width or height smaller in this case, so you need to provide an appropriate large enough size in this case, otherwise partial clipping will occur. Use Page Height /Width directly to select other formats!

### **Numeric accuracy [default: 1]**

*Command line option: -a #, where # is between 1 and 6*

This setting determines the number of digits floating point values are written in the PDF result. '1' means '2+1' that is '3' digits behind the comma. You may want to change this setting to higher values if you experience slightly misplaced graphics under very high resolutions or for CAD work. Higher values will also cause slightly larger output files (because of the additional digits).

*Note: This option will also define how accurate text elements are 'placed'. Low settings causes larger visual differences but smaller output size. For a setting of „1“ raster images are also rewritten for smaller output size.*

### **Minimal Linewidth [default: 1.0]**

*Command line option: -l #.#, where #.# is between 0.0 and 100.0*

Often the width of lines in PostScript is given wrongly as zero. These PostScript files display great on screen but when printed on a high resolution device the lines vanish completely. PDF has inherited this 'hairline' problem from PostScript. PStill will solve this by promoting too small lines to visible values. If you want PStill not change the line size set this to zero. Otherwise depending on your PDF needs values between 0.1 and 1.0 are fine. Higher values than 1.0 may cause strange effects.

### **Reverse Page Order [default: off]**

*Command line option: -r*

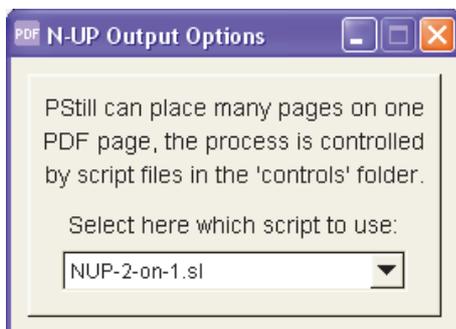
This will cause PStill to reorder the pages from back to front in the PDF result. Use this option if your input file is already reverse ordered.

### **PStill can apply N-UP (many pages on one and rescaling/poster making) to the result**

Set N-Up output options

*Command line option: -M pagecontrol=path\to\script.sl*

This button opens a dialog that displays the available 'scripts' in the folder „controls“ inside the GPStill directory. Such a script defines the placement and scaling of pages on the result. Several scripts come with PStill, e.g. 2-on-1.sl:



This option affects all input types, allowing to reorder an existing PDF or a set of images, output as Poster or other tricks. Please check the supplied collection of scripts for examples.

## Page rotation and saving of current setups and their restoration:



### Rotate Page (at the bottom, using the selectors for 0, 90, 180 and 270 default: 0]

Command line option: `-R #`, where # is either 0, 90, 180 or 270

(0 will cause PStill to honor the PostScript `%%Orientation: DSC` comment)

Will cause PStill to rotate the output. Use it to force landscape or portrait mode.

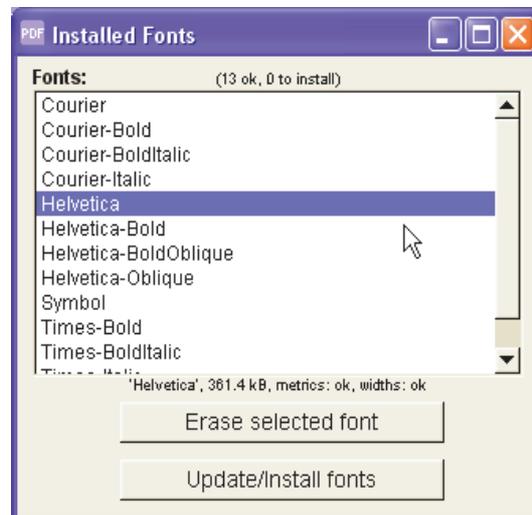
### Setup [Button]



The Setup button will open up the 'save' setup area. You can save all your defined options by name and load them later. This way you can simply 'return' to a complete setup without going through all dialogs.

### ManageFonts [Button]

Command line option: `-I`, install only - usually used together with option `-v` as `'-v -I'`



Will cause PStill to display the current font pool and allows to check all installed fonts and regenerate the fonts metrics and widths table. A newly installed font (See next section) is only visible to PStill after you have chosen this option and you have recalculated the metrics and widths tables.

Prepare to wait a few seconds, each font is run through the complete PS/PDF machinery and is subjected to a set of tests and rules to see if it checks out fine for PDF work. After work see the PStill log ('Show Log' button) to see a list of installed and ready to use fonts.

**For infos about the 'LPR Network Setup' see page 24 of this document.**

## Reset PDF Viewer [Button]

*Command line option: N/A*

PStill can open freshly converted PDF in a PDF viewer directly by clicking the 'Open PDF' button. However to do so it needs the location of the PDF viewer application. So the first time you choose this function it will display a file selector dialog where you should show him this program. It is - assuming you are using Acrobat(R) Reader(TM) - usually named 'AcroRd32.exe' but may be any program that accepts a single PDF file as parameter on the command line to cause this file to open or process (e.g. an automatic emailer, archiving system, printing tool or file repository).

However if you change the path of your reader program later or you like to tell PStill to use a different program you should use this option to force PStill to forget its current setting and open the selector dialog again.

All options correspond to the appropriate command option of the command lines version of PStill. See its section 4 to see how the options are used. The command line version has several additional features not accessible by the option dialog, however for most work the option list of GPStill will be sufficient.

***Hint:** GPStill will display the used 'command line' option list in the log. You can save the log to file by double-clicking in the log area.*

## 2.6 Reading the log

*Enable the log out for the command line version using option -v (verbose)*

PStill presents a detailed log for every conversion run. You will find a lot information about fonts, the PostScript conversion step, the PDF collection step and font embedding step and at the bottom if the conversion process produced any errors.

The sections that may be of particular interest are prefixed with

```
PS-Messages> . . . . .
```

as these are the output of the PS interpreter step and any problems during the conversion and error codes will show here or a few lines above this section.

Also of interest may be the font inclusion step later as here a user can check if a font was included in full or partial (subsetted) and if any problems may have occurred concerning the font sets inside the PostScript file.

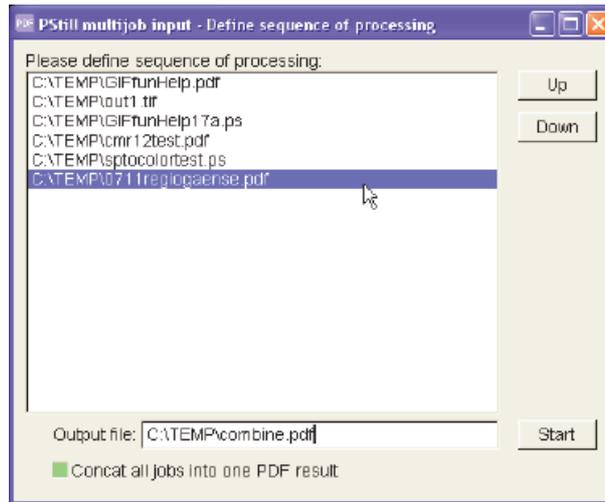
Even if the conversion run produced no errors the result may be not suitable for your purpose. This depends on the setup of PStill and the limits of the conversion itself. Also due to the multitude of different PostScript code, application creating PostScript and driver configurations and versions no guarantee can be given PStill will convert it 'correctly' all the time - PostScript is bound to 'interpretation' and this is meant literally in this case. You are advised to control the results of the conversion yourself.

Don't worry, usually PStill produces good results if setup'd correctly and fed with appropriate data.

### 3.0 Converting single and multiple files

You can drag&drop single or multiple files on PStills window – all supported input formats can be mixed. In case of multiple files PStill will display a dialog that let you set the order of the input files for processing and allows to combine the input files to one PDF result or alternatively process each input file separately to a PDF.

In the later case the results are placed always in the folder where the input originates otherwise for single files you are free to change the destination by typing the complete path in the PDF output field.



For all conversions the options are given at the start of the conversion run apply. It is not possible to change the options during a conversion run as PStills converter engine only reads them once at the start then run with the settings for the rest of the conversion.

*Hint: You can drag&drop more files on this window while it is open.*

### 3.1 Installation of fonts

*Command line users should copy the files to the PSFonts folder and run 'pstill -v -I'*

To install new fonts just select the font files - usually files with the postfix '.pfb' or '.pfa' - and drag them to the PStill main window. PStill will automatically identify them as fonts and place them in the PStill font pool, a folder called 'PSFonts' which resides inside the PStill installation folder. But still these fonts must be checked and their metric calculated. So remember to select 'Install New Fonts' in the options to make them known to PStill.

PStill accepts Type 1 (PFB or PFA format) and Type 3 (PFA format) PostScript fonts. If you have your fonts as TrueType (TTF) it will convert them to PostScript Type 1 first. Power users who can work the command line may use the supplied tool 'tff2pt1.exe' which is placed by the installation process in the PStill program folder. See also the file 'makefonts.bat' on hints how to use this tool. Remember however 'tff2pt1.exe' is only provided for convenience, you are using it and its results entirely at your own risk.

*Hint: PStill can extract the fonts as found in your PS Job if they are written as TrueType and available on the local system. You do not need to install them with PStill in this case. See the next section on how to find out how to setup the printer driver for this to work.*

### 3.2 Creating suitable input files (PostScript and EPS) for PStill

PostScript isn't PostScript - each printer driver is free to write special commands, elements and other settings into a PostScript file, usually this process is controlled by the setup of the printer driver or the application setting in the case of EPS files. It is of great importance to create suitable input files for PStill as it cannot know about the specialties of every printer driver in existence.

To create PostScript for PStill please install a new printer in Windows:

1. **Open the dialog: Workplace -> Printers -> New Printer**
2. **Select 'continue'**
3. **Make it a local printer**
4. **Select an appropriate type, for best results use either the settings for**

**„HP Color Laserjet PS“**

or

**„Canon PS-IPU LaserCopier“**

Other printer types may work too, including several „HP“ and „XEROX“ printer drivers. As you can change the printer type later again you can experiment with the setup here.

Please consult the PStill Network Print whitepaper (See page 24 of this file) for detailed setup on using the networking option.

#### **If you selected „HP Color Laserjet PS“:**

Recommended for Windows XP or 2000 or Windows 98/ME using the Adobe Printer Driver Version 4.x (in the later case please consult its license agreement first if you are allowed to use it for PS generation for Non-Adobe output devices). This additional setup should be done for the “HP Color Laserjet PS” printer type for best results :

- Choose **Start -> Settings -> Printers** and select the printer, then:
- Chose **File -> Printing preferences** and **Advanced** in the printing preferences dialog
- Set True Type Font to **Download as Softfont**
- Under *PostScript* options select

- **TrueType** for the True Type Font Download option

You can optional also leave this on 'Automatic' of set to 'Outline', but using TrueType fonts here will give you the smallest results, while 'Outline' will give your faster processing time and less memory consumption. You may want to play with these options.

- Be sure to leave the PS Output option to **'optimize for speed'**

## If you selected „ Canon PS-IPU LaserCopier“:

Recommended for Windows 98/ME and 95, but will work too on NT, XP or 2000. This printer produce good output but you may notice that non-standard TrueType fonts are written as outlines with can cause unnecessary large output.

This additional setup should be done for the “Canon PS-IPU LaserCopier” printer type for best results :

- Choose **Start -> Settings -> Printers** and select the printer, then:
- Chose **File -> Printing preferences** and **Advanced** in the printing preferences dialog
- Set True Type Font options to **Use printer fonts (replace by device fonts)**
- Under *PostScript* options select
  - **Outline** for the True Type Font Download option

Other printer types are working too, e.g. the type “*XEROX DocuColor40 w/Cyclone*” but above are the best tested variants. PStill can handle a lot different printer types. If you want to check if your choice was good convert a two page document using different fonts and graphics can see if the log of PStill gives the line:

```
* Job's printer driver setup looks good for Type 1 font extraction
```

This line indicates that PStill is happy and will use Type 1 (embedded fonts) as fonts, not converting them to outlines. If your driver emits 'TrueTypes' instead of 'Outlines' this is not of importance, as these are handled differently, however if your download options is set to 'automatic' or 'outlines' check this line.

Fortunately you need to do this only once, as soon as the setup is completed it will 'just work' for all print jobs from applications using the Windows GDI PostScript driver.

### General advise for PostScript generation:

The 'Language Level' gives an indication what PostScript commands and options a interpreter can handle. For this purpose an interpreter (like PStill) does advertise its language level to the PostScript job, so it can check if the needed command set is available. Level 1 is the most basic, Level 2 adds to it and Level 3 adds even more options and commands.

For EPS or manual PostScript generation it is usually a good idea to select "Export as PS Level 1". Otherwise the driver may write fonts as TrueTypes in the PostScript job. PStill can use embedded TrueTypes but will auto convert them to Type 1 PostScript fonts in the PDF result, using native PostScript fonts may look a tad better and process faster.

Mind that PStill can process Level 2 commands so it should work for most Level 2 PS or EPS files without problems, just in case you get such data from a remote source.

Avoid Level 3 PostScript, PStill can handle several Level 3 commands but does not implement some of the special colorspace or transparency options of PS Level 3 at the moment.

## 4.0 Command line usage

GPStill has a companion, a command line version for batch or embedded (calling from e.g. Visual Basic) work. It is called '*pstill.exe*' and resides in the PStill folder.

**Attention: Do not use 'gpstill.exe' for this purpose, it does not understand the options and will not work in a scripted environment. It is meant for interactive use only!**

To run pstill.exe now open a command shell and change to the PStill directory:

```
cd c:\programme\gpstill (may be different on your system!)
```

This step is not strictly needed, if you like to run pstill.exe from another directory you need to call it however with the full path, e.g.

```
c:\programme\gpstill\pstill.exe ...options to follow...
```

To get some help on the usage now type:

```
pstill -H | more
```

The "| more" is only needed to help you letting the text view as there is more than one page of information. To write the usage info to file type:

```
pstill -H > usage.txt
```

Now you can read the usage description and the list of possible options using your text editor of choice. As you can see options to pstill.exe must be given in the \*NIX notation, that means a '-' starts an option, not a '/' like in the DOS world.

### 4.1 Example conversion using the Command line converter

To convert a PS or EPS file to PDF using the command line version use this call

```
pstill ...options... -o output.pdf input.ps
```

e.g.

```
pstill -c -c -c -g -i -p -t -J 70 -o c:\pdf\report.pdf c:\psfiles\report.ps
```

To see what PStill is doing during conversion also add option -v, it causes the program to write the conversion log to stdout. You can easily place this log in a file of your choice using the 'pstill ...option.... > log.txt' redirection.

Return codes are based on the success of the program. 0 means no error occurred, otherwise see the appendix for a list of error codes.

PStill/Win version 1.55.4 and later allow to place the command line in a file and give this file as input to pstill.exe

```
pstill @c:\tmp\options_in_a_file.txt
```

This resolves the problem of the short command line in Windows.

For defining the output name PStill needs the option -o 'path' be given before the PS input file is defined. If you do not specify the output name PStill is using 'out.pdf' instead.

## 4.2 Output name specification using macros

To make processing easier the following macros may be used in the output file name:

**@TITLE@** this will be the contents of the %%Title: ... comment in the PostScript input job, usually the name of the document as created by an application. If this comment is not found PStill defaults to 'tmp-xxxxx', where xxxxx is a number (the process id pstill runs at)

(Version 1.55.4 or later also allow:)

**@SNAME@** the file name of the input, stripped of the path and postfix, e.g. if the input is 'c:\tmp\input.ps' @SNAME@ will result in 'input'.  
*Does not work with STDIN input!*

**@LNAME@** the file name of the input, stripped of the postfix only, e.g. if the input is 'c:\tmp\input.ps' @LNAME@ will result in 'c:\tmp\input'.  
*Does not work with STDIN input!*

You can use these macros quite handily in a bat script:

```
c:\...fullpath...\pstill.exe ... -o c:\out\@SNAME@.pdf c:\tmp\input.ps
```

On \*NIX you can use the 'basename' shell extension instead.

## 4.3 Allow input from Standard In (stdin)

To allow the use of PStill as a 'filter', e.g. in connection with RedMon (see

<http://www.wizards.de/~frank/vprinting.html>

for more info) the command line version allows the use of the special input name 'STDIN'. Usage is like

```
c:\programme\gpstill\pstill.exe ... -o c:\output\pdf\output.pdf STDIN
```

PStill then collect the data and waits for the EOF then starts processing.

## 4.4 Combine several files in one result

If you specify more than one input file on the command line, PStill will concatenate them together in one PDF output file, e.g.

```
pstill ...options... -o output.pdf input1.ps input2.ps input3.ps
```

will result in one PDF file called output.pdf containing the page from input1 until input3 in that order. You can also mix different inputs types.

## 4.5 Convert several files

PStill does not support the '\*' wildcard for file selection. This stems from its \*NIX heritage where the shell does all wildcard processing. However there is an easy way to convert a batch of files on Windows, just use a BAT script alike:

(name this *convert.bat* or similar)

```
@echo off
rem setup where to find pstill:
set pst=c:\programme\gpstill
set in=d:\tmp\pstill_in
set out=d:\tmp\pstill_out
echo Starting PDF conversion loop for %in%, results in %out%
pause
:loop
  echo Checking source directory...
  for %%f in (%in%\*.ps) do call single.bat %%f %out%
  echo Waiting 30 seconds...
  rem use sleep.exe from e.g. http://unxutils.sourceforge.net/
  sleep 30
goto loop
```

and to convert a single file use this BAT-Script as template:

(name this *single.bat* or similar)

```
@echo off
set pstillpath=c:\program files\gpstill
set pstilloptions=-v -c -c -g -i -p -t -J 60
if .%1==. goto FAIL
if .%2==. goto FAIL
%pstillpath%\pstill.exe %pstilloptions% -o %2\@TITLE@.pdf %1
rem you might want to uncomment the following line to erase the input
rem del %1
goto END
:FAIL
echo Script needs two options: inputfile and output directory
:END
```

These scripts are only templates, pstill can also be run from other environments like 'Perl' or 'Visual Basic Scripting Host'. Please adapt these scripts to your environment. Using such scripts you can easily build a 'watched folder' mechanism to convert a lot PS files to PDF.

## 4.6 Flattening mode (PostScript to PostScript processing)

PStill allows not only to write to PDF but to a normalized PostScript format that will only use a handful well defined PostScript command and has a standard coordinate system as base. This mode can be quite helpful in several cases:

- You have a PS file but you need a placeable EPS file
- You need to post-process PostScript files, e.g. to enter field values, pages or watermarks
- You want to sanitize a PostScript file (remove unneeded elements and make sure it works later in a PS interpreter)
- You need to provide PS files to others but only with subsetted fonts
- You need to import text, vector, graphics and raster images for PS or EPS files into another program or convert them to another format - just parse the results of PStill

Mode -m X is your friend in this case. Use:

-m p	to produce a normalized PS file for parsing or processing
-m e	to produce a normalized EPS file
-m P	same as -p but postprocess the results for smaller size and include needed fonts as subsetted form (if possible)
-m E	same as -e but also postprocess and include subsetted fonts

Additional 'high quality' modes for prepress purposes are available but needs to be licensed separately:

-m Prewrite	to produce a normalized PS file suitable for prepress work
-m Erewrite	to produce a normalized EPS file suitable for prepress work

## 5. Helper files and user configurable elements

Some file in the PStill distribution can be configured to improve conversion quality and to implement special behaviour.

***Attention:** We are now moving on high ground, this section is not meant for the mere user of PStill, some understanding of script programming or fonts should be in your backpack. Make backups of the files in question before changing them and test your changes before putting them into production.*

### 5.1 Configuration file 'fontsub.table', located in the PStill folder

This file controls what font is used if a font is called by the interpreter. You can think of it as font replacement table. Its internal syntax is PostScript. By default it is empty and only contains comment lines, that are lines starting with '% '.

To define a mapping enter a line like

```
[ /Arial /Helvetica ]
```

Above line tells PStill to use the font with the PostScript name 'Helvetica' everytime the font 'Arial' is called. Remember we don't use the PFB font file name here as it is artificial shorted for PFB formatted font files. To find out the real PostScript name of a font just import it into PStill and look in the folder 'PSFonts' after metric calculation has taken place. PStill will automatically rename the font to its 'true' name.

You can define up to 65535 mappings, just put every single one on a new line.

There are some caveats on using a mapping however:

1. A mapping does not change the text widths as defined in the PostScript job, that means is is now possible text flow is distorted since the new font has a different glyph width than the old. Be sure to map only fonts that have similar or identical widths.
2. A 'mapped' font is always included in the PDF output if it is not part of the base font set. PStill determines this by looking up the name of the font after the mapping has taken place. So if you use a replacement font instead of e.g. Helvetica or Times-Roman it will most likely be included in the PDF result and the PDF grows in size.
3. The base font "Courier" cannot be mapped using above method. It must be available under its name all the time. This does not apply to its variants like "Courier-Bold" etc.
4. Every single font variant must be covered. That means for e.g. Helvetica you need to define a mapping for 'Helvetica', 'Helvetica-Bold', 'Helvetica-Oblique' and 'Helvetica-BoldOblique' to cover the complete font family.
5. The font names are case sensitive and must start with a '/' character
6. PStill will tell you in its *PS-Message*> ... log section what fonts are not found so you can use this as a reference to setup this table.

## 5.2 Configuration file 'annots.sl' located in the PStill folder and N-Up control scripts

These files is a S-Lang (see <http://www.s-lang.org> for the language description) script file that controls the conversion of annotations (also called 'pdfmarks') from their PostScript form to the PDF form. The syntax of the language is a lot like 'C' with pieces of 'Forth' thrown in for good measure.

Everytime the PDF backend sees an annotation it will process it as for each single key/value pair and call up the S-Lang function 'action\_pdfmark' with two parameters:  
The string value of the current key/value pair and a number as a counter.

e.g. The following mark in PostScript

```
[ /Rect [ 145 210 332 228 ] /Color [1 0 0]  
/Action << /S /URI /URI (http://www.this.net/~frank) >>  
/Subtype /Link /ANN pdfmark
```

Will cause action\_pdfmark to be called with

```
/ANN  
/Subtype /Link  
/Action << /S /URI /URI (http://www.this.net/~frank) >>  
/Color [1 0 0]  
/Rect [ 145 210 332 228 ]  
<empty string>
```

The script can now work on the single annotations key/values and is responsible to collect all pairs into a single string to be returned to the PDF backend for embedding into the PDF result.

For this purpose three utility functions exists:

**objToAnnotation(int pagenumber, string annotation);**

Will place the annotation on the defined page (absolute page numbering required)

**objToOutline(annotstring);**

Will define an new outline entry

**objToName(annotstring);**

Will define a named destination

As the script has full control over the contents and the script language allows also file i/o it could be used for quite some nice effects:

- Automatically write a list of annotations in the PDF in a file during conversion
- Reencode the text (e.g. change 'öäü' characters to make them visible to a viewer)
- programmatically create annotations without the need to place them in the PostScript source

Before changing this script please build up some knowledge about the use and syntax of S-Lang. A single error in the script would cause the conversion engine to abort with an S-Lang error.

More info on S-Lang is available on its web page: <http://www.s-lang.org>

All N-Up scripts are also written in the S-Lang language. You can find them in the directory 'controls' in the GPStill folder. They define the function:

**pdf\_page\_setup(pagenum)**

which is called by the PDF generator to setup the placement of pages. The parameters supplied is the current page number. The script can then change the page setup by calling

**setupPage ( float translate\_x, float translate\_y, float rotate,  
float scale\_x, float scale\_y, int keepOnSamePage )**

All translate values are given in pts, rotation in degrees. Be advised to consult the supplied examples for some deeper insight. You can also experiment, any error in the scripts will be reported to the log.

For more info about the S-Lang scripting language see <http://www.s-lang.org>

### **5.3 Optional installation of additional helper programs**

PStill can access outside helper programs to let it perform additional tasks, e.g. decompression of LZW-compressed images (e.g. As found in TIFF and PDF). By default PStill does not decompress LZW-compressed data in these formats but it can optionally call out to an external and separate program 'gzip.exe' to handle decompression.

You may want to put 'gzip.exe' in the gpstill folder to support these data types. GZIP can be fetched from its home page: <http://www.gzip.org>.

Background information: PStill will never compress data using the LZW method, it uses the much more effective 'flate' compression instead.

GZIP is a standard compression program that comes with all LINUX distributions and several other UNIX variants and is used to compress data with the 'flate' method. It also has the ability to decompress most other data types. It is considered 'patent free' by its authors (see its web page) and is published under the GPL. It is expressly not part of PStill, not part of the PStill distribution and its use falls entirely under its own license.

Do not use this method if LZW *decompression* is patented for software in your country. (It is my believe that only LZW *compression* is patented in the USA but IANAL)

# How to setup the Windows printer driver to print directly to PStill using the Network Printing Facility.

**Note:** You also need to turn on the network support for PStill to make this work, see the options of PStill and 'LPR Network'. The results will be placed in the 'target directory' you can specify there. PStill will not overwrite files in there but add their name with versioning informations.

## 1. Install the Windows components to support LPR (TCP/IP) printing:

PStill supports the LPR printing service which is included in Windows XP, 2000 and NT but is a separate third-party add-on for 98 and ME. This section explains how to install the component in Windows as this particular print service is not installed by default. The component is named „TCP/IP printing“ or „UNIX printing services“ in Windows.

*Attention: Have your Windows System CD-ROM handy - when asked for your Windows CD-ROM, please insert the CD-ROM and follow the instructions as given in the dialog.*

On **Windows XP** follow this documentation as published by Microsoft:

<http://www.microsoft.com/windowsxp/pro/using/itpro/networking/printservicesunix.asp>

On **Windows 2000** follow this path (which is very similar to the 'XP' way):

- From **Start -> Settings -> Control Panel**, open **Network and Dial-up Connections**.
- From the **Advanced** file menu choose **Optional Networking Components...**
- Select **Other Network File and Print Services**, and choose **Details...**
- Check the box for **Print Services for Unix** and choose **OK** and **Next**.

On **Windows NT** click with the right mouse button on **Network Neighborhood**:

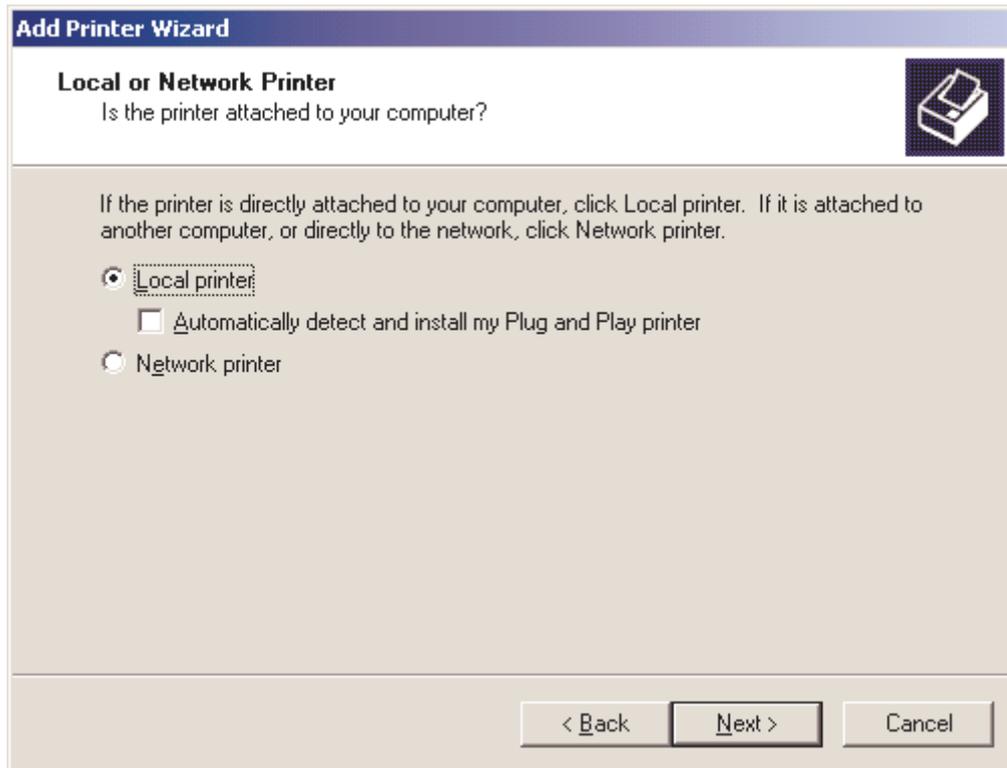
- From the menu that appears, select **Properties**.
- In the **Network** window that appears, click on the **Services** tab.
- Browse through the list of *Network Services*
- If **Microsoft TCP/IP Printing** is not listed, click on the **Add...** button.
- In the **Select Network Service** window, select **Microsoft TCP/IP Printing** from the list of Network Services.
- Click **OK**.

## 2. Install a printer port to allow to print to PStill

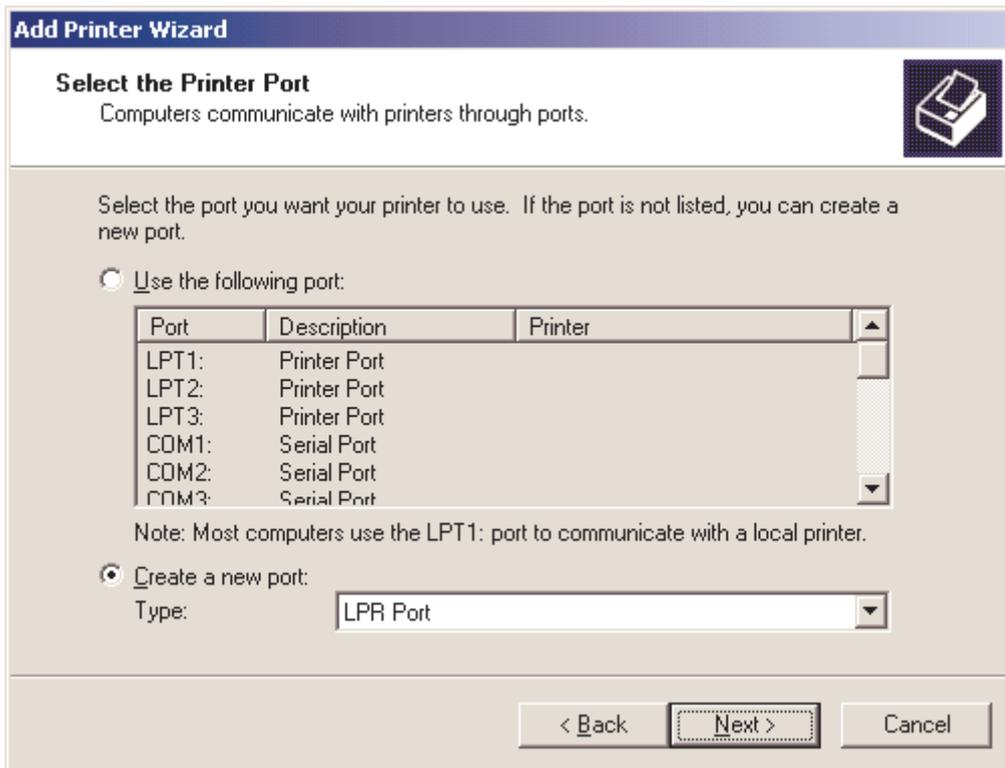
The setup on Windows 2000 is given as example, it is very similar to Windows XP only details of the dialogs differ slightly.

From **Start -> Settings -> Printers**, select **Add Printer**.

- From the „Add Printer Wizard“ window, choose **Next>**
- Choose **Local printer**.



- Select **Create a new port:** and **LPR Port**
- **Note:** If you have already installed a LPR port for PStill you may want to select this from the list of available ports instead! (See 'Use the following port:' list)



- Now enter 127.0.0.1 for the hostname/address and 'pstill' for the printer queue



- Now select the appropriate **Manufacturer** and **Printer model**, e.g. „**HP**“ and „**HP Color Laserjet PS**“
- When prompted to name your printer, enter a descriptive name and decide whether this should be the default printer. E.g. „PStill“ or „PDF Conversion“ makes sense here.
- In the **Printer Sharing** window, select: **Do not share this printer**
- This concludes the network setup. PStill should now be able to receive data as soon as you turn on the network option in the 'LPR Network Setup' dialog (see 'More Options'). If PStill does complain about being unable to establish the network services you may need to reboot once. This is due to the fact that initial installation of the „UNIX printing services“ on Windows seems to block TCP/IP port 515. This block is gone after a reboot.
- Before sending jobs to PStill your printer should be configured for best conversion results, see page 16 of this document on „Print Setup“.

## 6. Technical Appendix

### PDF Reader programs

Some links to PDF reader and viewer programs can be found on the PStill web page:

**<http://www.pstill.com>**

These links are given for convenience and in the hope they provide useful but constitute no special support for these programs. The links lead to external sites out of the control of the author of PStill.

### For PStill to convert the input job successfully it must match some criterias:

PStill contains a generic ("clone") PostScript interpreter. It must interpret the input job in order to gain knowledge about the data and visual display within. This interpretation process depends on certain criterias to be successful. You are advised to test your input jobs with PStill before putting the converter into a production chain to check the results. The input job must meet the following criterias:

-It must not contain commands that are undefined in PStill, e.g. it may not contain device specific PostScript that only exists for a given printer or output device) or Level 3 PostScript commands.

*Attention: Use of undefined commands in the input job will cause a PostScript error and incomplete conversion!*

-It must be created for a reasonable output device: e.g. for Microsoft Windows 95 or better: A suitable way to create PostScript files for use with PStill is to define a printer type of "Canon PS-IPU Laser Copier", use the generic Windows PostScript driver, define all fonts to be written as PostScript equivalents and use the printer fonts. Two compatible printer drivers are e.g.:

For B/W:	"Apple Laserwriter NT or NTX"
For Color and B/W:	"Canon PS-IPU Laser Copier"

It is recommended to define some kind of input file policy for users which create PostScript for use with PStill. To check this policy the program "PSCHECK" (available separately) can be used as a PostScript validator - however it will only validate the "correctness" of the PostScript program code not the correctness of the visual display. If your application demands high correctness of the visual output you should build some kind of "user validation" into it, by displaying the output job to the user and querying a "correct yes/no?" question.

*Attention: Use of a unsuited PostScript driver or settings for PostScript input job generation may cause PostScript errors or 'wrong' display and results.*

### Notes about conversion of text elements:

PStill will usually convert text (font usage) in the given input into suitable text elements in the output. However there are some cases where this cannot work.

- Font is missing:

PStill needs all PostScript fonts referenced by the input job in its "PSFonts" directory. If a font is missing both in the input job and in this directory "Courier" is replaced in the resulting output. Please note that PStill can only take over a font from within the input file to the output if they are marked with suitable DSC comments, as the extraction process depends on the comments to 'find' a font. It is therefore recommended to have all fonts available in your PSFonts directory that are needed for processing.

- Font is not embedded as text:

PStill cannot reconstruct text that is written as paths or bitmap, in this case the output contains the vector path or the bitmap of the font

- Font is unsuitable for embedding:

If a font is used that is build dynamically (glyphs are added during PS processing or the font structure is

changed "on-the-fly" during PostScript processing) PStill is unable to take it over to the output. In this case it will write the text elements as paths. Also Type 3 PostScript fonts will be written as paths or bitmaps as these could use any PostScript command to paint themselves. This way PStill will always preserve the visual display of the input best as possible.

**Notes about the processing environment:**

PStill need a large area for creating its temporary files. By default this is the Windows Temp directory. You may change this directory with the option '-T path'. It is recommended to have at least tripple (3x) the size of the input job for space here.

***Attention: If you run multiple instances of PStill each instance needs its own temporary directory! Otherwise the effects, results and output are undefined.***

## PStill Error codes (process return values)

The "pstill.exe" program will return these codes upon completion of a job. It is recommended to check these values in your own application or script. Please note that a value of 0 (noerror) does not indicate that really no error has occurred for all input jobs as PostScript programs are free to intercept error conditions on their own.

noerror	0	ok, no error to report
errdictfull	1	A directory was filled with too much entries
errdictstackoverflow	2	Too many directories open
errdictstackunderflow	3	Too many directories closed
errexecstackoverflow	4	Too long procedures found
errinterrupt	5	User interrupt
errinvalidaccess	6	Write access on a read only area
errinvalidexit	7	exit command without suitable environment
errinvalidfileaccess	8	file I/O error
errinvalidfont	9	font does not compute or is bad
errinvalidrestore	10	unable to restore
errinvalidstop	11	unable to stop (no suitable environment)
errioerror	12	general I/O error
errlimitcheck	13	a limit was reached (e.g. path too long)
errnocurrentpoint	14	no starting point defined
errrangecheck	15	a limit was reached (e.g. string size too short)
errstackoverflow	16	tried to put more than 5000 elements on the stack
errstackunderflow	17	tried to pop too much elements from the stack
errsyntaxerror	18	some command has a wrong syntax
errtimeout	19	job took too long (could not happen with psrip)
errtypecheck	20	a command got a wrong parameter
errundefined	21	an unknown command was found
errundefinedfilename	22	tried to access a nonexisting file
errundefinedresult	23	tried to compute a nonexisting result (div by zero)
errunmatchedmark	24	no mark was found but requested
errunregistered	25	no registration was found but requested
errVMerror	26	out of memory
errmemoryallocation	27	error during memory access or allocation
errkill	28	interpreter was killed due to signal
errmax	29	too many errors occurred (error handler was flooded)
syserr	>29	a system error has occurred during processing

If an error occurs the log may tell you where in the input file and some details if available.

## Limits of the current interpreter version (valid for PStill 1.65)

The interpreter used in PStill/PSRIP is a 'clone' interpreter for the PostScript[tm] page description language. PStill understands all Level 1, most of the Level 2 commands and some DisplayPostScript operators as defined in the Adobe PostScript Reference Manual [Second Edition]. The interpreter has some inherent limits a user job must not exceed:

max. number of fonts per job	7200
max. path length	512000 device points
max. userdict storage	32767 elements
max. errordict storage	40 elements
max. local vm segments	254
max. local vm segment size	1 MByte (for a total of 254 * 1024 kByte memory)
max. global vm area	16 MByte
max. operand stack size	32767 elements
max. exec stack size	64000 elements
max. dict stack size	20 dictionaries deep
max. no of opened files	unlimited (by memory and system only)
max. string length	65535 chars

Please note that the total job size may exceed this memory size by far as the job is processed and not stored during conversion. So the actual memory consumption of the PostScript job is usually not a function of its file size and can only be determined by e.g. running the PSCHECK product in log mode.

**The known implementation limitations are as follows:**

- Dictionaries are not expanded unlimited automatically for PStill (dictfull error possible), only to a size 160 elements larger than the base set
- Some PostScript operators are emulated as functions and therefore reside in systemdict or userdict as functions not operators (e.g. show and other ops)
- No Level 3 transparency (image type 4) support and some other L3 additions
- setdistillerparams are not evaluated and cause no effect

**PSTILL COPYRIGHT NOTICE**

PStill is a copyrighted work of Frank Siegert, June 2003 All rights reserved.  
PStill (c) Copyright Frank Siegert 1989-1992, 1996-2003, Alle Rechte vorbehalten

Contact information:

*Dipl.-Ing. Frank Siegert*  
*Stauferweg 35*  
*72555 Metzingen*  
*eMail frank@this.net or frank@wizards.de*  
*Office (Germany) (0)7123 963929*  
*Office Fax (Germany) (0)7123 15463*

PStill is licensed not sold. Usage and distribution is subject to the following conditions. If you do not accept these conditions you may not use PStill.

*Usage is subject to the following conditions*

***IF YOU DO NOT ACCEPT THESE CONDITIONS YOU MAY NOT USE PSTILL***

- \* You acknowledge the 'shareware' status of this software. Without valid key it will create a watermark in the produced PDF files and writes a UNREGISTERED message in the PDF info section. A unique key is needed for each system you like to run PStill on (except for special site keys or multiple keys which allow usage on a number of systems for each key)
- \* You will not reverse engineer or disassemble the software to gain knowledge of the inner workings. You may not change the software to try to disable the license code.
- \* You do not gain ownership of the PStill software. PStill is licensed not sold.
- \* You must keep the PStill license key or the licensed version secret. The licensee must take proper precautions not to reveal his key to others.
- \* PStill must not be bundled with any other product without written consent of Frank Siegert. Please send mail to frank@this.net for more information and licensing information.
- \* An unlicensed PStill must not be used in any production environment or for any commercial purpose nor may it be bundled with any commercial software package or distributed for money on media or the internet. Only the distribution of the unchanged PStill package free of charge is permitted provided no 'consulting fee' or 'download costs' are charged.
- \* You are alone responsible for any copyright violations the inclusion of fonts in generated pdf files might inflict. Please read your font license agreement before distributing such pdfs files or use such fonts.  
PStill can use the local TrueType system fonts as templates for its base fonts,

these fonts are renamed alike the Type 1 base fonts, their names are needed as identification for the correct working of the PDF creation machinery. The trademark of the font names is still held by their respectful owners. The naming scheme is only used for identification purpose within the converter system.

\* You are using PStill entirely at your own risk:

Frank Siegert disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall Frank Siegert be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

To the best knowledge of Frank Siegert the software is free of copyright claims of other third parties or patent claims. However since patents are granted on a nation-based scale not guarantee can be given that the software is free of claims or its use and/or methodes of working constitutes an infringement of third party rights in your respective environment. You are alone responsible for any such infringement and your are advised to check if the use of this software collides with any patent claim in your particular legal environment before using or deploying this product.

All names and symbols within the program or its documentation are used for identification purpose only. All trademarks belong to their respective owners. All trademarks mentioned in this document are used for identification purpose only. PSTILL is a registered trademark of Frank Siegert.

### 3. Acknowledgements

PStills core PS interpreter engine was originally based on the 'post' postscript interpreter originally written by Adrian Aylward. He gave me kind permission to use and extend it, much thanks to him. The t1asm and t1disasm helper programs are copyright by I. Lee Hetherington, thanks to his kind permission to use them in PStill. Zlib is Copyright Jean-Loup Gailly and Mark Adler. FLTK is an LGPL'd Window Toolkit by Bill Spitzak, see <http://www.fltk.org/> for more info.

PStill uses the S-Lang Script language as embedded control language for doing its PDFMark (annotation) support. Thanks to John E. Davis for releasing it under the Artistic License. See <http://www.s-lang.org/> for more info.

Conservative C Garbage Collector, [http://www.hpl.hp.com/personal/Hans\\_Boehm/gc](http://www.hpl.hp.com/personal/Hans_Boehm/gc)  
Copyright 1988, 1989 Hans-J. Boehm, Alan J. Demers  
Copyright (c) 1991-1996 by Xerox Corporation. All rights reserved.  
Copyright (c) 1996-1999 by Silicon Graphics. All rights reserved.  
Copyright (c) 1999 by Hewlett-Packard Company. All rights reserved.

JPEGLIB is copyright (C) 1991-1998, Thomas G. Lane.

TIFFLIB is copyright (C) Sam Leffler and others

XPDF is copyright (C) Glyph & Cog, LLC, licensed for commercial use by Frank Siegert

### 4. Trademarks

ALL NAMES AND SYMBOLS ARE USED FOR IDENTIFICATION PURPOSE ONLY. POSTSCRIPT, ADOBE, ACROBAT, DISTILLER, READER, PDF AND OTHERS NOT MENTIONED HERE ARE TRADEMARKS OR REGISTERED TRADEMARKS OF ADOBE SYSTEMS INCORPORATED. ALL OTHER BRAND OR PRODUCT NAMES ARE THE TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE HOLDERS. PDF AND PDF LIST OF OPERATORS AND DATA STRUCTURES ARE COPYRIGHT ADOBE SYSTEMS INCORPORATED. PSTILL WAS WRITTEN ACCORDING TO THE COPYRIGHT CONDITIONS AS OUTLINED IN THE PORTABLE DOCUMENT FORMAT REFERENCE MANUAL VERSION 1.2, PAGE 22/23.

PSTILL IS A REGISTERED „WORDMARK“ WITH THE GERMANY PATENT AND TRADEMARK OFFICE.

## 5. Third party tools

This distribution comes with a binary of ttf2pt1 (Version 3.22SNAP), this is its copyright:

ttf2pt1 is Copyright (c) 1997-2000 by the AUTHORS:

Andrew Weeks <ccsaw@bath.ac.uk>

Frank M. Siegert <fms@this.net>

Mark Heath <mheath@netspace.net.au>

Thomas Henlich <thenlich@rcs.urz.tu-dresden.de>

Sergey Babkin <babkin@bellatlantic.net>, <sab123@hotmail.com>

Turgut Uyar <uyar@cs.itu.edu.tr>

Rihardas Hepas <rch@WriteMe.Com>

Szalay Tamas <tomek@elender.hu>

Johan Vromans <jvromans@squirrel.nl>

Petr Titera <P.Titera@sh.cvut.cz>

All rights reserved.

Redistribution and use of ttf1pt1 in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:  
This product includes software developed by the TTF2PT1 Project and its contributors.

For the source code please refer to:

<http://www.netspace.net.au/~mheath/ttf2pt1/>

TTF2PT1 IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THIS PDF DOCUMENT WAS GENERATED BY PSTILL, JUNE 2003

PSTILL WAS CREATED IN GERMANY.