

ps2pdf: PostScript-to-PDF converter

1 Table of contents

- [Overview](#)
- [Usage](#)
- [Setting page orientation](#)
- [Options](#)
- [Creating a PDF/X-3 document](#)
- [Creating a PDF/A document](#)
- [Limitations](#)
- [Ghostscript PDF Printer Description](#)
- [Known problems](#)
- [Comparison of ps2pdf and Acrobat Distiller](#)
- [Acknowledgments](#)

For other information, see the [Ghostscript overview](#).

2 Overview

ps2pdf is a work-alike for nearly all the functionality (but not the user interface) of Adobe's Acrobat™ Distiller™ product: it converts PostScript files to Portable Document Format (PDF) files.

ps2pdf is implemented as a very small command script (batch file) that invokes Ghostscript, selecting a special "output device" called **pdfwrite**. In order to use **ps2pdf**, the **pdfwrite** device must be included in the makefile when Ghostscript was compiled; see the [documentation on building Ghostscript](#) for details: this is currently the case on all platforms, in Ghostscript as we distribute it.

3 Usage

The usage for **ps2pdf** is

```
ps2pdf [options] input.[e]ps output.pdf
```

or, on Unix systems and some versions of Windows NT and OS/2

```
ps2pdf input.[e]ps
```

which is equivalent to

```
ps2pdf input.[e]ps input.pdf
```

There are actually several different **ps2pdf** *scripts: the name **ps2pdf** above refers to any of them.

- **ps2pdf12** produces PDF 1.2 output (Acrobat 3-and-later compatible).
- **ps2pdf13** produces PDF 1.3 output (Acrobat 4-and-later compatible).
- **ps2pdf14** produces PDF 1.4 output (Acrobat 5-and-later compatible).
- **ps2pdf** per se currently produces PDF 1.2 output. However, this may change in the future. If you care about the compatibility level of the output, use **ps2pdf12**, **ps2pdf13**, or **ps2pdf14**, or use the **-dCompatibilityLevel=1.x** switch in the command line.

Note that if you specify a value for **PDFSETTINGS**, this chooses PDF 1.3 or 1.4 output depending on the value of **PDFSETTINGS**: this overrides the output format specified by the script name. You can still specify the output format by using **-dCompatibilityLevel= after -dPDFSETTINGS=**.

All of these scripts actually call a script named **ps2pdfwr** or **ps2pdfxx**. The Unix **ps2pdfwr** script assumes that the Ghostscript executable is named **gs**; it is unlikely that you will need to change this. The DOS and MS Windows **ps2pdfxx.bat** script uses the value of the **gsc** environment variable, if defined, as the name of the executable; otherwise the script assumes the executable is named **gswin32c**. So in these environments, if the executable has a different name, you must set **gsc** to the name of the executable.

4 Setting page orientation

By default Ghostscript determines viewing page orientation based on the dominant text orientation on the page. Sometimes, when the page has text in several orientations or has no text at all, wrong orientation can be selected.

Acrobat Distiller parameter **AutoRotatePages** controls the automatic orientation selection algorithm. On Ghostscript, besides input stream, Distiller parameters can be given as command line arguments. For instance: **-dAutoRotatePages=/None** OR **/All** OR **/PageByPage**.

When there is no text on the page or automatic page rotation is set to `/None` an orientation value from `setpagedevice` is used. Valid values are: `0` (portrait), `3` (landscape), `2` (upside down), and `1` (seascape). The orientation can be set from the command line as `-c "<</Orientation 3>> setpagedevice"` using Ghostscript directly but cannot be set in `ps2pdf`. See [Limitations](#) below.

Ghostscript passes the orientation values from DSC comments to `pdfwrite` driver but they are effectively ignored there. This appears to be consistent with Distiller 5 behavior.

5 Options

The *options* in the command line may include any switches that may be used with Ghostscript's PostScript and PDF interpreter (see [here](#) for a complete list), although almost none of them are useful with `ps2pdf`. The following may be useful:

`-rresolution`

Sets the resolution for pattern fills and for fonts that must be converted to bitmaps.

`-dProcessColorModel=device_color_space`

Sets the color space to be used for device-dependent colors in the output. *device_color_space* may be `/DeviceGray`, `/DeviceRGB`, or `/DeviceCMYK`; the default value is `/DeviceRGB`. Note that this does not affect images: see [Limitations](#) below.

More importantly, *options* may include `-dparameter=value` or `-sparameter=string` switches for setting "distiller parameters", Adobe's documented parameters for controlling the conversion process. The PostScript `setdistillerparams` and `currentdistillerparams` operators are also recognized when running `ps2pdf`, and provide an equivalent way to set these parameters from within the PostScript input file.

`ps2pdf` also recognizes the following options:

`-dCompressFonts=boolean`

Defines whether `ps2pdf` will compress embedded fonts in the output. The default value is `true`; the `false` setting is intended only for debugging.

`-dMaxInlineImageSize=integer`

Specifies the maximum size of an inline image, in bytes. For images larger than this size, `ps2pdf` will create an XObject instead of embedding the image into the context stream. The default value is `4000`. Note that redundant inline images must be embedded each time they occur in the document, while multiple references can be made to a single XObject image. Therefore it may be advantageous to set a small or zero value if the source document is expected to contain multiple identical images, reducing the size of the generated PDF.

`-dPDFSETTINGS=configuration`

Presets the "distiller parameters" to one of four predefined settings:

- `/screen` selects low-resolution output similar to the Acrobat Distiller "Screen Optimized" setting.
- `/ebook` selects medium-resolution output similar to the Acrobat Distiller "eBook" setting.
- `/printer` selects output similar to the Acrobat Distiller "Print Optimized" setting.
- `/prepress` selects output similar to Acrobat Distiller "Prepress Optimized" setting.
- `/default` selects output intended to be useful across a wide variety of uses, possibly at the expense of a larger output file.

The following option controls a conversion into PDF/X-3:

`-dPDFX=boolean`

Specifies the generated document to follow the PDF/X-3 standard. When true, a `DefaultRGB ColorSpace` resource must be defined, and options `NOSUBSTDEVICECOLORS`, `NOCIE` must not be specified. Default value is `false`.

When generating a PDF/X-3 document, Ghostscript performs the following special actions to satisfy the PDF/X-3 standard :

- All fonts are being embedded.
- `DeviceRGB` color space is being substituted with `DefaultRGB` color space, which must be defined in the `ColorSpace` category. The easiest way is to provide it in the `DefaultRGB` file in the resource directory.
- `DeviceRGB` color values are being passed unchanged. If an user needs an untrivial color adjustment, an untrivial `DefaultRGB` color space to be defined.
- Transfer functions and halftone phases are being skipped.
- `/PS pdfmark` interpretes the `DataSource` stream or file.
- `TrimBox` and `BleedBox` entries are generated in page descriptions. Their values can be changed using the `PDFXTrimBoxToMediaBoxOffset`, `PDFXSetBleedBoxToMediaBox`, and `PDFXBleedBoxToTrimBoxOffset` distiller parameters (see below).

The following options control a conversion into PDF 1.2:

`-dPatternImagemask=boolean`

With `CompatibilityLevel < 1.3` it specifies whether the target viewer handles `ImageMask` with a pattern color. Some old viewers, such as Ghostscript 3.30 fail with such constructs. Setting this option to false, one can get more compatibility, but the mask interpolation is lost. With `CompatibilityLevel >= 1.3` this option is being ignored. Default value is `false`.

`-dMaxClipPathSize=integer`

Specifies the maximum number of elements in the clipping path that the target viewer can handle. This option is used only with `CompatibilityLevel < 1.3` and `PatternImagemask=false`, and only when converting a mask into a clipping path. If the clipping path exceeds the

specified size, the masked image and the clipping path is being decomposed into smaller images. The value of the option counts straight path segments (curved segments are not used for representing a mask). Default value is 12000.

`-dMaxShadingBitmapSize=integer`

With `CompatibilityLevel < 1.3` it specifies the maximum number of bytes allowed for representing a shading as a bitmap. If a shading exceeds this value, the resolution of the output bitmap is being reduces to fit into the specified frame. Note that the number of bytes depends on the number of color components in `ProcessColorModel`, assumes 8 bits per sample, and doesn't account an image compression or filtering. Also note that reducing the resolution results unsmooth shading boundaries. With `CompatibilityLevel >= 1.3` this option is being ignored. Default value is 256000. For the best quality one can set the maximal integer value, but the output file size may dramatically increase. Therefore the user should choose a compromise value.

`-dHaveTrueTypes=boolean`

With `CompatibilityLevel < 1.3` it specifies whether the target viewer can handle TrueType fonts. If not, TrueType fonts are being converted into raster fonts with resolution specified in `HWResolution`. With `CompatibilityLevel >= 1.3` this option is being ignored. Default value is `true`.

The following option controls a conversion into PDF 1.3:

`-dHaveTransparency=boolean`

With `CompatibilityLevel >= 1.4` it specifies whether the target viewer can handle PDF 1.4 transparency objects. If not, transparency objects are being converted into plain images. Default value is `true`.

The following switches are used for creating encrypted documents :

`-sOwnerPassword=string`

Defines that the document be encrypted with the specified owner password.

`-sUserPassword=string`

Defines the user password for opening the document. If empty, the document can be opened with no password, but the owner password is required to edit it.

`-dPermissions=number`

Defines the PDF permissions flag field. Negative values are allowed to represent unsigned integers with the highest bit set. See the PDF Reference manual for the meaning of the flag bits.

`-dEncryptionR=number`

Defines the encryption method revision number - either 2 or 3.

`-dKeyLength=number`

Defines the length (in bits) of the encryption key. Must be a multiple of 8 in the interval [40, 128]. If the length isn't 40, `-dEncryptionR` must be 3.

The following switches are used for generating metadata according to Adobe XMP specification :

`-sDocumentUUID=string`

Defines a DocumentID to be included into the document Metadata. If not specified, Ghostscript generates an UUID automatically. Otherwise the specified string is being copied into the document without checking its syntax or consistence.

Note that Adobe XMP specification requires DocumentID must be same for all versions of a document. Since Ghostscript does not provide a maintenance of document versions, users are responsible to provide a correct UUID through this parameter.

Note that Ghostscript has no assess to the host node ID due to a minimization of platform dependent modules. Therefore it uses an MD5 hash of the document contents for generating UUIDs.

`-sInstanceUUID=string`

Defines a instance ID to be included into the document Metadata. If not specified, Ghostscript generates an UUID automatically. Otherwise the specified string is being copied into the document without checking its syntax or consistence.

Note that Adobe XMP specification requires instance ID must be inique for all versions of document. This parameter may be used to disable an unique ID generation for a debug purpose.

When none of `DocumentUUID` and `InstanceUUID` are specified, the generated DocumentID appears same as instance ID.

`-sDocumentTimeSeq=integer`

Defines an integer to be used as a deconflictor for generating UUIDs, when several invocations of Ghostscript create several PDF documents within same clock quantum (tick). Mainly reserved for very fast computers and/or multithreading applications, which may appear in future. If both `DocumentUUID` and `InstanceUUID` are specified, `DocumentTimeSeq` is being ignored.

`ps2pdf` recognizes all of the Acrobat Distiller 5 parameters defined in the DistillerParameters document included in the [Acrobat SDK](#). Cells in the table containing '=' mean that the value of the parameter is the same as in the "default" column.

Parameter name	Notes	default	screen	ebook	printer	prepress
----------------	-------	---------	--------	-------	---------	----------

AlwaysEmbed	(13)	[]	=	=	=	=
AntiAliasColorImages	(0)	false	=	=	=	=
AntiAliasGrayImages	(0)	false	=	=	=	=
AntiAliasMonoImages	(0)	false	=	=	=	=
ASCII85EncodePages		false	=	=	=	=
AutoFilterColorImages	(1)	true	=	=	=	=
AutoFilterGrayImages	(1)	true	=	=	=	=
AutoPositionEPSFiles	(0)	true	=	=	=	=
AutoRotatePages		/PageByPage	/PageByPage	/All	/None	/None
Binding	(0)	/Left	=	=	=	=
CalCMYKProfile	(0)	()	=	=	=	=
CalGrayProfile	(0)	()	=	=	=	=
CalRGBProfile	(0)	()	=	=	=	=
CannotEmbedFontPolicy	(0)	/Warning	/Warning	/Warning	/Warning	/Error
ColorACSImageDict	(13)	(note 7)	(note 10)	(note 10)	(note 8)	(note 9)
ColorConversionStrategy	(0,6)	/LeaveColorUnchanged	/sRGB	/sRGB	/UseDeviceIndependentColor	/LeaveColorUnchang
ColorImageDepth		-1	=	=	=	=
ColorImageDict	(13)	(note 7)	=	=	=	=
ColorImageFilter		/DCTEncode	=	=	=	=
ColorImageDownsampleThreshold		1.5	=	=	=	=
ColorImageDownsampleType	(3)	/Subsample	/Average	/Bicubic	/Bicubic	/Bicubic
ColorImageResolution		72	72	150	300	300
CompatibilityLevel		1.4	1.3	1.4	1.4	1.4
CompressPages		true	=	=	=	=
ConvertCMYKImagesToRGB		false	=	=	=	=
ConvertImagesToIndexed	(0)	false	=	=	=	=
CoreDistVersion		4000	=	=	=	=
CreateJobTicket	(0)	false	false	false	true	true
DefaultRenderingIntent		/Default	=	=	=	=
DetectBlends	(0)	true	=	=	=	=
DoThumbnails	(0)	false	false	false	false	true
DownsampleColorImages		false	true	true	false	false
DownsampleGrayImages		false	true	true	false	false
DownsampleMonoImages		false	true	true	false	false
EmbedAllFonts		true	false	true	true	true
EmitDSCWarnings	(0)	false	=	=	=	=
EncodeColorImages		true	=	=	=	=
EncodeGrayImages		true	=	=	=	=
EncodeMonoImages		true	=	=	=	=
EndPage	(0)	-1	=	=	=	=
GrayACSImageDict	(13)	(note 7)	(note 7)	(note 10)	(note 8)	(note 9)
GrayImageDepth		-1	=	=	=	=
GrayImageDict	(13)	(note 7)	=	=	=	=
GrayImageDownsampleThreshold		1.5	=	=	=	=
GrayImageDownsampleType	(3)	/Subsample	/Average	/Bicubic	/Bicubic	/Bicubic
GrayImageFilter		/DCTEncode	=	=	=	=
GrayImageResolution		72	72	150	300	300
ImageMemory	(0)	524288	=	=	=	=
LockDistillerParams		false	=	=	=	=
LZWEncodePages	(2)	false	=	=	=	=
MaxSubsetPct		100	=	=	=	=
MonoImageDepth		-1	=	=	=	=
MonoImageDict	(13)	<<K -1>>	=	=	=	=
MonoImageDownsampleThreshold		1.5	=	=	=	=
MonoImageDownsampleType		/Subsample	/Average	/Bicubic	/Bicubic	/Bicubic
MonoImageFilter		/CCITTFaxEncode	=	=	=	=
MonoImageResolution		300	300	300	1200	1200
NeverEmbed	(13)	(note 11)(note 12)	(note 11)	(note 11)	[](note 12)	[](note 12)
			(note 12)	(note 12)		
OffOptimizations		0	=	=	=	=
OPM		1	=	=	=	=
Optimize	(0,5)	false	true	true	true	true
ParseDSCComments		true	=	=	=	=
ParseDSCCommentsForDocInfo		true	=	=	=	=
PreserveCopyPage	(0)	true	=	=	=	=

PreserveEPSInfo	(0)	true	=	=	=	=
PreserveHalftoneInfo		false	=	=	=	=
PreserveOPIComments	(0)	false	false	false	true	true
PreserveOverprintSettings		false	false	false	true	true
sRGBProfile	(0)	()	=	=	=	=
StartPage	(0)	1	=	=	=	=
SubsetFonts		true	=	=	=	=
TransferFunctionInfo	(4)	/Preserve	=	=	=	=
UCRandBGInfo		/Remove	/Remove	/Remove	/Preserve	/Preserve
UseFlateCompression	(2)	true	=	=	=	=
UsePrologue	(0)	false	=	=	=	=

(note 0) This parameter can be set and queried, but currently has no effect.

(note 1) Currently, `AutoFilterxxxImages` doesn't examine the image to decide between JPEG and LZW or Flate compression: it always uses Flate compression. If you want to force JPEG compression, use `-dAutoFilterxxxImages=false`.

(note 2) Because of Unisys's threats regarding the Welch patent, `ps2pdf` does not actually use LZW compression: instead, it treats all requests for LZW compression as calling for Flate compression. Concomitantly, `UseFlateCompression` is treated as always on, and the value of this parameter is ignored as with note 0. Now that the patent has expired, we could change this should it become worthwhile.

(note 3) The `xxxDownsampleType` parameters can also have the value `/Bicubic` (a Distiller 4 feature), which is currently treated as equivalent to `/Average`.

(note 4) Currently, the transfer function is always applied. If the corresponding parameter is set to `/Preserve`, the function setting is also copied into the PDF file.

(note 5) Optimization (linearization) is implemented with a separate program, `pdfopt` *input.pdf output.pdf*; the `Optimize` parameter has no effect.

(note 6) Ghostscript specifics : The value `UseDeviceIndependentColor` requires the device parameter `UseCIEColor` to be set to `true`. The value `UseDeviceIndependentColorForImages` works same as `UseDeviceIndependentColor`. The value `CMYK` works with any `CompatibilityLevel` and requires the device parameter `ProcessColorModel` to be set to `DeviceCMYK`. The value `sRGB` requires the device parameter `ProcessColorModel` to be set to `DeviceRGB`, and actually converts to RGB with the default Ghostscript conversion. The new Ghostscript-specific value `Gray` requires the device parameter `ProcessColorModel` to be set to `DeviceGray`, and converts all colors to `DeviceGray`. The old Ghostscript-specific value `UseDeviceDependentColor` is now deprecated. It is automatically replaced with `sRGB`, `CMYK`, or `Gray`.

(note 7) The default image parameter dictionary is

```
<< /QFactor 0.9 /Blend 1 /HSamples [2 1 1 2] /VSamples [2 1 1 2] >>
```

(note 8) The printer ACS image parameter dictionary is

```
<< /QFactor 0.4 /Blend 1 /ColorTransform 1 /HSamples [1 1 1 1] /VSamples [1 1 1 1] >>
```

(note 9) The prepress ACS image parameter dictionary is

```
<< /QFactor 0.15 /Blend 1 /ColorTransform 1 /HSamples [1 1 1 1] /VSamples [1 1 1 1] >>
```

(note 10) The screen and ebook ACS image parameter dictionary is

```
<< /QFactor 0.76 /Blend 1 /ColorTransform 1 /HSamples [2 1 1 2] /VSamples [2 1 1 2] >>
```

(note 11) The default, screen, and ebook settings never embed the 14 standard fonts (Courier, Helvetica, and Times families, Symbol, and ZapfDingbats).

(note 12) `NeverEmbed` can include CID font names. If a CID font is substituted in `lib/cidfmap`, the substitute font name is used when the CID font is embedded, and the original CID font name is used when it is not embedded. `NeverEmbed` should always specify the original CID font name.

(note 13) The arrays `AlwaysEmbed` and `NeverEmbed` and image parameter dictionaries `ColorACSImageDict`, `ColorACSImageDict`, `ColorImageDict`, `GrayACSImageDict`, `GrayImageDict`, `MonoImageDict` cannot be specified on the `ps2pdf` command line. To specify these, you must use PostScript, either by including it in the PostScript source or by passing the `-c` command-line parameter to ghostscript as described in [Limitations](#) below. For example, including the PostScript string in your file `in.ps`:

```
<</AlwaysEmbed [/Helvetica /Times-Roman]>> setdistillerparams
```

is equivalent to invoking:

```
gs -dBATCH -dSAFER -DNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile=out.pdf -c '.setpdfwrite <</AlwaysEmbed [/Helvetica /Times-Roman]>> setdistillerparams' -f in.ps
```

or using `ps2pdf` with the extra parameters in a file:

```
ps2pdf @params.in out.pdf
```

where the file `params.in` contains:

```
-c '<</AlwaysEmbed [/Helvetica /Times-Roman]>> setdistillerparams' -f in.ps
```

6 Ghostscript PDF Printer Description

To assist with creating a PostScript file suitable for conversion to PDF, ghostscript includes [ghostpdf.ppd](#), a PostScript Printer Description (PPD) file. This allows some [distiller parameters](#) to be set when a PostScript file is generated.

6.1 Windows XP or 2000

To install a "Ghostscript PDF" printer on Windows XP, select the Windows Control Panel, Printers and Faxes, Add a Printer, Local Printer, Use port FILE: (Print to File), Have Disk..., select the directory containing [ghostpdf.ppd](#) and [ghostpdf.inf](#), select "Ghostscript PDF", Replace existing driver (if asked), and answer the remaining questions appropriately. After installing, open the "Ghostscript PDF" properties, select the Device Settings tab, set "Minimum Font Size to Download as Outline" to 0 pixels.

To set distiller parameters, select the "Ghostscript PDF" Printing Preferences, then the Advanced button. The PDF settings are under "Printer Features".

7 Creating a PDF/X-3 document

To create a PDF/X-3 document from a Postscript or a PDF file, you should :

- Specify the `pdfwrite` device or use the `ps2pdf` script.
- Specify the `-dPDFX` option. It provides the document conformity and forces `-dCompatibilityLevel=1.3`.
- Specify `-sProcessColorModel=DeviceGray` OR `-sProcessColorModel=DeviceCMYK` (`DeviceRGB` is not allowed).
- Specify the `-dUseCIEColor` option if necessary (see below).
- Specify a PDF/X definition file before running the input document. It provides additional information to be included into the output document. A sample PDF/X definition file may be found in `gs/lib/PDFX_def.ps`.
- If a registered printing condition is applicable, specify its identifier in the PDF/X definition file. Otherwise provide an ICC profile and specify it in the PDF/X definition file as explained below.
- Provide a `DefaultRGB` resource file in the `ColorSpace` resource category. Either define it in the PDF/X definition file, or provide a definition of `gs/Resource/ColorSpace/DefaultRGB`. Rather `gs/Resource/ColorSpace/DefaultRGB` is usually distributed with Ghostscript, its contents is not necessarily satisfy your needs, see below.

As mentioned above, the PDF/X definition file provides a special information, which the PDF/X-3 standard requires. You can find a sample file in `gs/lib/PDFX_def.ps`, and edit it according to your needs. The file follows Postscript syntax and uses the operator `pdfmark` to pass the special information. For your comfort we marked editable lines in the sample file with the comment `% Customize`. They are explained below.

OutputCondition string

Defines an `OutputCondition` value for the output intent dictionary.

OutputConditionIdentifier string

Defines an `OutputConditionIdentifier` value for the output intent dictionary.

ICCProfile string

May be omitted if `OutputConditionIdentifier` specifies a registered identifier of characterized printing condition (see http://www.color.org/IPA_2003-11_PDFX.pdf). Defines a file name of an ICC profile file to be included into the output document. You may specify either an absolute file name, or a relative path from the working directory.

Title string

Defines the document title. Only useful if the source Postscript file doesn't define a title with DSC comments. Otherwise remove entire line from definition file.

Info string

Defines an `Info` value for the output intent dictionary.

Rather the PDF/X-3 standard requires colors to be adjusted at the document generation time, Ghostscript *does not* perform any special color conversion. Either colors to be adjusted in advance, or a proper color conversion to be specified in `/DefaultGray`, `DefaultRGB`, `DefaultCMYK` resources of the `ColorSpace` resource category, and the `-dUseCIEColor` option to be specified to remap device colors into CIE colors. The choice of the ICC profile is also the user's responsibility.

Please note that if a graphic object can't embed into the output format, Ghostscript converts it into low level objects, using a device color space specified in the `ProcessColorModel` option. If you need to adjust those resulting colors, you may substitute them with CIE colors, *running Ghostscript at second time*. Performing both actions in a single pass is a subject of further improvements.

Ghostscript distribution does not contain an ICC profile to be used for creating a PDF/X-3 document. Users should either create an appropriate one themselves, or use one from a public domain, or create one with the PDF/X-3 inspector freeware.

The PDF/X-3 standard requires a `TrimBox` entry to be written for all page descriptions. This is an array of four offsets that specify how the page is to be trimmed after it has been printed. It is set to the same as `MediaBox` by default unless the `PDFXTrimBoxToMediaBoxOffset` distiller parameter is present. It accepts offsets to the `MediaBox` as an array [*left right top bottom*], e.g., the PostScript input code `<< /PDFXTrimBoxToMediaBoxOffset`

[10 20 30 40] >> `setdistillerparams` specifies that 10 points will be trimmed at the left, 20 points at the right, 30 points at the top, and 40 points at the bottom.

Another page entry is the `BleedBox`. It gives the area of the page to which actual output items may extend; cut marks, color bars etc. must be positioned in the area between the `BleedBox` and the `MediaBox`. The `TrimBox` is always contained within the `BleedBox`. By default, the `PDFXSetBleedBoxToMediaBox` distiller parameter is `true`, and the `BleedBox` is set to the same values as the `MediaBox`. If it is set to `false`, the `PDFXBleedBoxToTrimBoxOffset` parameter gives offset to the `TrimBox`. It accepts a four-value array in the same format as the `PDFXTrimBoxToMediaBoxOffset` parameter.

Here is a sample command line to invoke Ghostscript for generating a PDF/X-3 document :

```
gs -dPDFX -dBATC -dNOPAUSE -dNOOUTERSAVE -dUseCIEColor -sDEVICE=pdfwrite -sOutputFile=out-x3.pdf PDFX_def.ps input.ps
```

8 Creating a PDF/A document

To create a PDF/A document, please follow instructions about [creating a PDF/X-3 document](#), with the following exceptions :

- Specify the `-dPDFA` option.
- A sample PDF/A definition file may be found in `gs/lib/PDFA_def.ps`.
- `Info`, `OutputCondition`, `OutputConditionIdentifier` are not required in the PDF/A definition file.

Here is a sample command line to invoke Ghostscript for generating a PDF/A document :

```
gs -dPDFA -dBATC -dNOPAUSE -dNOOUTERSAVE -dUseCIEColor -sDEVICE=pdfwrite -sOutputFile=out-a.pdf PDFA_def.ps input.ps
```

9 Limitations

`ps2pdf` will sometimes convert PostScript constructs to lower-level ones, even if a higher-level construct is available. For example, if the PostScript file uses `charpath` to set a clipping path consisting of text, `ps2pdf` will write the clipping path as a path in the PDF file, rather than as text, even though PDF is able to express clipping with text. This is only a performance issue, and will be improved incrementally over time.

Some applications, such as HIGZ, produce PostScript files that use ridiculously large coordinates. On such files, `ps2pdf` may cause a `limitcheck` error. If this occurs, try reducing the default internal resolution of 720 dpi by using the `-r` switch, e.g., `ps2pdf -r300 somefile.ps`.

`ps2pdf` ignores the PDF 1.3 (Acrobat 4.x) pdfmarks related to document content structure: `StRoleMap`, `StClassMap`, `StPNE`, `StBookmarkRoot`, `StPush`, `StPop`, `StPopAll`, `StBMC`, `StBDC`, `EMC`, `StOBJ`, `StAttr`, `StStore`, `StRetrieve`, `NamespacePush`, `NamespacePop`, and `NI`. While this causes some structural information to be omitted from the output file, the displayed and printed output are normally not affected.

`ps2pdf` currently has only very limited support for PDF 1.4. It writes out the blend mode, constant alpha, and text knockout graphics state parameters, and it handles images with soft masks, but it does not handle transparency groups, or soft masks in the graphics state. (Note that there is no standard way to specify any of these things in PostScript, so these statements only apply when the input file is already a PDF 1.4 file.)

`ps2pdf` provides a simplified interface to the Ghostscript command line. It is not possible to use `-c` option directly or pass multiple source files. For the unrestricted access to the command line parameters, use Ghostscript directly as in:

```
gs -q -dSAFER -dNOPAUSE -dBATC -sOutputFile=file.pdf [more options] \
-sDEVICE=pdfwrite -c .setpdfwrite -f source1.ps [more files]
```

or create a parameter file with the `-c` option and/or the multiple input files:

```
ps2pdf @params.in out.pdf
```

where the file `params.in` contains:

```
-c Postscript commands -f source1.ps [more files]
```

See [Language.htm](#) for details of the `.setpdfwrite` operator.

10 Known problems

Distiller parameters should only be saved by `save` and restored by `restore`, but they are also saved by `gsave` and restored by `grestore`.

Changing the value of the `CompressPages` parameter after any marks have been made on the page may cause a crash.

Ghostscript has been writing incorrect `ToUnicode` `CMap` without `CMapName` into the PDF since version 8.10 (rev. 3611) . This bug is fixed in version 8.54 (rev. 6201). We recommend to re-generate PDF files created by the affected Ghostscript versions. Since version 8.54 (rev. 6590) Ghostscript can read the incorrect PDF files.

11 Comparison of `ps2pdf` and Acrobat Distiller

According to users, the greatest benefit of `ps2pdf` is that it is more robust than Acrobat Distiller: it will process complex and difficult PostScript files that Acrobat Distiller is not able to handle.

For certain documents, `ps2pdf` is much faster than Adobe Distiller, and may be suitable for run-time conversions. George White, a heavy user of `ps2pdf`, remarks:

I haven't seen a head to head comparison, but Distiller seems slower when running on what should be a faster system (for instance, Distiller on a PPC Mac vs a 25 MHz 68040 NeXT running `ps2pdf`), so I think this is fair -- also, one of Mark Doyle's postings indicated that Distiller was not fast enough for use as a run-time server. In contrast, I find that I can use `ps2pdf` as a post-processor during routine document creation.

On the other hand, there are some documents for which `ps2pdf` may be much slower than Acrobat Distiller. Caveat user.

`ps2pdf` usually produces output that is comparable in size to the output of Acrobat Distiller; however, it sometimes produces much larger output, especially if the input file involves pattern fills.

Many users report that the combination of `ps2pdf` with Acrobat Reader is superior to using a generic PostScript viewer (`psview` or `ghostview`), particularly for documents with many pages where the navigational support in PDF files reduces the overhead involved in navigating conventional PostScript documents.

12 Acknowledgments

Thanks to George N. White III <aa056@chebucto.ns.ca> of the Ocean Sciences Division of the Bedford Institute of Oceanography in Dartmouth, Nova Scotia for extensive testing of early versions of `ps2pdf`, and for contributing most of this writeup.

Thanks to Martin Hosken of SIL International <<http://www.sil.org>> for help with testing `ps2pdf` with a wide variety of international fonts.

Copyright © 1996, 2000, 2004 artofcode LLC. All rights reserved.

This software is provided AS-IS with no warranty, either express or implied. This software is distributed under license and may not be copied, modified or distributed except as expressly authorized under the terms of the license contained in the file LICENSE in this distribution.

Ghostscript version 8.51, 31 December 2005