

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
ColorACSIImageDict	(13)	(note 7)	(note 10)	(note 10)	(note 8)	(note 9)
ColorConversionStrategy	(0,6)	LeaveColorUnchanged	RGB	RGB	UseDeviceIndependentColor	LeaveColorUnchanged
ColorImageDepth		-1	=	=	=	=
ColorImageDict	(13)	(note 7)	=	=	=	=
ColorImageFilter		/DCTEncode	=	=	=	=
ColorImageDownsampleThreshold		1.5	=	=	=	=
ColorImageDownsampleType	(3)	/Subsample	/Average	/Average	/Average	/Bicubic
ColorImageResolution		72	72	150	300	300
CompatibilityLevel		1.4	1.3	1.4	1.4	1.4
CompressPages	(14)	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	=	=	=	=
GrayACSIImageDict	(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	/Subsample	/Subsample	/Subsample	/Subsample
MonoImageFilter		/CCITTFaxEncode	=	=	=	=
MonoImageResolution		300	300	300	1200	1200
NeverEmbed	(13)	(note 11)(note 12)	(note 11)(note 12)	(note 11)(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	=	=	=	=
PassThroughJPEGImages	(15)	true	=	=	=	=

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

(note 1) -dAutoFilterxxxImages=false works since Ghostscript version 7.30. Older versions of Ghostscript don't examine the image to decide between JPEG and LZW or Flate compression: they always use Flate compression.

(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), this will use a Mitchell filter. (older versions of pdfwrite simply used Average instead). Note: if a non-integer downsample factor is used the code will clamp to the nearest integer (if the difference is less than 0.1) or will silently switch to the old bicubic filter, NOT the Mitchell filter.

(note 4) The default for transfer functions is to preserve them, this is because transfer functions are a device-dependent feature, a set of transfer functions designed for an RGB device will give incorrect output on a CMYK device for instance. The pdfwrite device does now support /Preserve, /Apply and /Remove (the previous documentation was incorrect, application of transfer functions was not supported). PDF 2.0 deprecates the use of transfer functions, and so when producing PDF 2.0 compatible output if the TransferFunctionInfo is set to /Preserve it will be silently replaced with /Apply. You can instead specifically set TransferFunctionInfo to /Remove when producing PDF 2.0 in order to avoid the transfer function being applied.

(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. With the new color conversion code active it is no longer necessary to set ProcessColorModel when selecting Gray, RGB or CMYK. It is also no longer necessary to set UseCIEColor for UseDeviceIndependentColor to work properly, and the use of UseCIEColor is now strongly discouraged. sRGB is not supported, use RGB instead.

(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). This behavior is intentional but can be overridden by:

```
<< /NeverEmbed [ ] >> setdistillerparams
```

(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
```

(note 14) The default value of CompressPages is false for ps2write and eps2write.

(note 15) When true image data in the source which is encoded using the DCT (JPEG) filter will not be decompressed and then recompressed on output. This prevents the multiplication of JPEG artefacts caused by lossy compression. PassThroughJPEGImages currently only affects simple JPEG images. It has no effect on JPX (JPEG2000) encoded images, or masked images. In addition this parameter will be ignored if the pdfwrite device needs to modify the source data. This can happen if the image is being downsampled, changing colour space or having transfer functions applied. Note that this parameter essentially overrides the 'EncodeColorImages' and 'EncodeGrayImages' parameters if they are false, the image will still be written with a DCTDecode filter. NB this feature currently only works with PostScript or PDF input, it does not work with PCL, PXL or XPS input.

Color Conversion and Management

As of the 9.11 pre-release, the color management in the pdfwrite family has been substantially altered so that it now uses the same Color Management System as rendering (the default is LCMS2). This considerably improves the color handling in both pdfwrite and ps2write, particularly in the areas of Separation and DeviceN color spaces, and Indexed color spaces with images. Despite lengthy careful testing it is possible that there will be bugs in this initial implementation and so the following switch is available.

`-dPDFUseOldCMS=boolean`

The old (non-ICC) colour management code has now been removed and this switch no longer has any effect.

The `ColorConversionStrategy` switch can now be set to `LeaveColorUnchanged`, `Gray`, `RGB`, `CMYK` or `UseDeviceIndependentColor`. Note that, particularly for ps2write, `LeaveColorUnchanged` may still need to convert colors into a different space (ICCbased colors cannot be represented in PostScript for example). `ColorConversionStrategy` can be specified either as a string by using the `-s` switch (`-sColorConversionStrategy=RGB`) or as a name using the `-d` switch (`-dColorConversionStrategy=/RGB`).

ps2write cannot currently convert into device-independent color spaces, and so `UseDeviceIndependentColor` should not be used with ps2write (or eps2write).

All other color spaces are converted appropriately. Separation and DeviceN spaces will be preserved if possible (ps2write cannot preserve DeviceN) and if the alternate space is not appropriate a new alternate space will be created. Eg a `[/Separation (MyColor) /DeviceRGB {...}]` when the `ColorConversionStrategy` is set to `CMYK` would be converted to `[/Separation (MyColor) /DeviceCMYK {...}]`. The new tint transform would be created by sampling the original tint transform, converting the RGB values into CMYK, and then creating a function to linearly interpolate between those values.

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 the pdfwrite driver, and these are compared with the auto-rotate heuristic. If they are different then the DSC value will be used preferentially. If the heuristic is to be preferred over the DSC comments then comment parsing can be disabled by setting `-`