

# PDFDevice Performance

Image handling is the one area during PDF generation that can dramatically effect performance. To provide better control over Image quality and performance PDFDevice provides a number of image handling options.

Until PDFDevice version 2.6.5, the biggest known performance killer, besides printing many large high resolution images, is adding large background pictures on every page with the option `kDevPdfImgConvBest` enabled. Until that version, PDFDevice did not maintain information so it knew if the background picture for subsequent pages is the same as that on previous pages. As a consequence it had to do it the hard way by comparing the image data after choosing the best compression. In version 2.6.5 this performance caveat has finally been resolved.

**NOTE:** This document has been updated to account for version 2.6.5 performance improvements. If you have previously read this document, you should pay special attention to the paragraph below and the revised performance table.

## **kDevImgConvBest**

This option causes PDFDevice to choose between run-length encoding or JPEG compression for each individual image. Run-length encoding is very efficient for images with many repeating pixels, such as computer screen images of windows. In fact, run-length encoding can be as much as 900% more efficient without any loss of quality. For more complex images such as photos, JPEG compression will yield better results.

Enabling this option has an immediate impact on performance. In order to determine the best compression for an image, PDFDevice has to compress each image twice and compare the resulting size. Needless to say, this will pretty much double the time it takes to deal with each image.

If performance is your primary concern, turn this option off, but make sure you enable `kDevPdfConvJPEG`.

To produce good quality images balanced with a good compression ratio using JPEG compression, we feel that setting the image quality (`kDevPdfJPEGQuality`) to 75 yields the overall best results.

## **kDevPdfImgStripDup**

This option causes PDFDevice to include data of duplicate images only once, substantially reducing the final size of the PDF file. If your document contains many duplicate images, or background images, we recommend that you turn on this option.

Enabling this option may effect performance a little in the first pass, as PDFDevice has to compare the data with that of previous images that have identical checksums. But this is usually well compensated by having to write much less data to the final PDF file.

We recommend to always turn this option on, unless you know that your documents do not or very rarely contain duplicate images.

## **kDevPdfBkgImgOn**

This option causes PDFDevice to include a background image for each page that is produced.

Turning on this option may affect performance. It will depend on the size of the background image and if the background image changes for different pages. If performance is of utmost importance consider not including a background image.

## Performance comparison table

In order to highlight the impact the different settings can have on performance we have included a performance comparison table. We have used Studio 5.2.2 and PDFDevice 2.6.5 on Macintosh 10.7.5. We have chosen the OWrite group of documents from the Document Manager example, printing most of the chapters totaling 107 pages. IN addition to the Brainy Data background image, this document contains, besides some larger unique images, a number of small images that are duplicated numerous times. The options we have included in the comparison are kDevPdfConvJPEG, kDevPdfBkgImgOn, kDevPdfImgConvBest and kDevPdfImgStripDup.

ConvJPEG	BkgImgOn	ImgConvBest	ImgStripDup	time (sec.)	size (kb)
yes	yes	yes	yes	5.20	913
yes	yes	yes	no	5.23	1100
yes	yes	no	yes	4.10	921
yes	no	yes	yes	4.98	881
yes	no	no	yes	4.01	890
yes	no	no	no	4.06	1200
no*	yes	n/a	yes	5.01	1300
no*	yes	n/a	no	5.03	1500
no*	no	n/a	yes	4.85	989
no*	no	n/a	no	4.85	1200

\*if kDevPdfConvJPEG is disabled, run-length encoding is used

As one can see from the results, there is nothing gained by turning off kDevPdfImgStripDup, other than a bigger file. Whereas turning off kDevPdfImgConvBest increases performance by about 20% in our tests. Turning off kDevPdfBkgImgOn also increases performance but only by a mere 5%. However, smaller documents with fewer pages may yield a greater percentage performance increase when turning off background images. As an additional test we printed another document containing only text on 1 page. This test showed a 12% performance increase when not using a background image.

ConvJPEG	BkgImgOn	ImgConvBest	ImgStripDup	time (sec.)	size (kb)
yes	yes	yes	yes	1.43	98
yes	no	yes	yes	1.26	71