

## Open Prepress Interface (OPI)

An extension of the PostScript page-description language that lets you design pages with low-resolution placeholder images and replace the images with high-resolution images when creating separations.

PDF files containing OPI-images are commonly referred to as 'thin PDF files'. PDF files containing only high-resolution images are called 'fat PDF files'.

### What is OPI?

The OPI industry-standard convention defines how to embed instructions in a PostScript output file to tell the output device where and how to merge the various text and graphics components of a page. OPI enables users to work with low-res preview images in their page-makeup programs, and keep the high-res graphic images close to the image setter. This maximizes workstation productivity and minimizes network traffic.

In general, OPI works with TIFF files. OPI supports all the cropping and sizing commands issued in the page makeup program. When the page makeup program creates a PostScript output file of the job for the printer, it appends these commands, along with the pathname and filename, as PostScript comments in the job stream. When the OPI-compliant output device reads these comments, it acts upon them by retrieving and merging the high-res image.

### Why remove OPI comments from the final PDF?

The only advantage is actually file size. Thin PDF files can be small and quick to process. But there is a long list of disadvantages of using OPI within PDF-files:

1. One of the advantages of PDF is that you can visually check the files using Acrobat Reader or Exchange. But if you have an OPI-workflow using 'omit images', the PDF file will only contain OPI-references and you will not be able to view the file properly.
2. PDF files can be quite small due to the excellent compression options it offers. So using OPI to limit the file size of PDF-files is often not necessary.
3. Preflighting software or PDF-editing tools are often useless with thin PDFs because the files don't contain all the final data.
4. Using Distiller is a great way of eliminating possible PostScript errors early on in the production chain. This advantage disappears when an OPI-solution can still insert corrupted images or incorrect PostScript code after having processed the clean PDF-file.

All of these arguments prove that a thin PDF is not always a great idea. In a lot of cases, it makes more sense to keep using OPI but to have the OPI-system deliver a fat PostScript to Distiller so that a fat PDF can be processed in the workflow.

### How to remove OPI comments?

You should always deactivate the 'Preserve OPI Comments' option in Distiller if no OPI is used. This way such OPI-comments are deleted while creating the PDF-files and post processing tools are not confused by irrelevant OPI-comments.