Refrying PDFs

the good, the bad and the ugly

Author Leonard Rosenthal,
PDF Standards Architect, Adobe Inc.

Date October 2008

Status Revised January 2020

Ghent Workgroup

www.gwg.org

Table of Contents

1 Introduction

1.1 About the Ghent Workgroup

The Ghent Workgroup (GWG) is a worldwide assembly of graphic arts stakeholders (user associations, vendors, consultants, educational institutions, service providers, and end users) founded in 2001. It was formed in response to increased needs for standardization of the different processes in graphic arts workflows, especially in an increasingly globalized service provider landscape. The rules of the group have been carefully conceived to ensure that the group remains practically oriented, and the priority is focused on the needs of the end users.

The GWG focuses on developing best practice guidelines and specifications for graphic arts workflows. While initially focusing mainly on quality control and preflight for PDF workflows in commercial print, that focus has broadened to also include metadata specifications, workflow test suites and increased support for market segments such as packaging.

All material created by the GWG is disseminated free of charge through the website of the GWG (www.gwg.org) and through the vendors and user associations partaking in the work of the group.

The mission statement of the Ghent Workgroup states that the group will "establish and disseminate process specifications for best practices in graphic arts workflows". In practice this means that the group:

- Develops and maintains process specifications and associated documentation for best practices in graphic arts workflows.
- Develops and maintains *reference implementations* to ensure the specifications it develops are usable in the real world.
- Actively promotes adoption of its work in both the graphic arts user and vendor communities.
- Streamlines and coordinates the decision process between its members.

While the group started its work developing guidelines for PDF quality control, it has expanded its scope. The group is now involved in magazine, office, and packaging specific specifications, the development of job ticket metadata specifications for delivering PDF files for advertisements, preflighting PDF files, and in developing test suites to ensure workflows and applications are configured and used correctly.

Much of the work of the group is done through teleconferences and e-mail discussions. Three times a year, the members come together for a three-day face-to-face meeting. To streamline the work and decision process, subcommittees have been organized around specific topics do the actual work. To learn more about the different subcommittees, or to find out how you can contribute to this effort, visit the Ghent Workgroup website (www.gwg.org).

1.2 Background

When Adobe created PostScript, it was designed to be a printer-specific language for describing the contents of each page that was to be rendered. Today, we call such languages PDLs (Page Definition Languages) and they continue to drive printers, RIPs and the like. PostScript, which hasn't been updated since Level 3 in 1998, defines a common "core" set of language features, however, each printer that uses it has added various extensions for their particular needs and that aren't compatible with other devices.

As PostScript became the standard way to produce and exchange files in the prepress & print production industry, it became necessary to define a device-independent subset of PostScript that could be used to represent a single page (or subset of a page) of content to exchange graphics, advertisements and such. This is where EPS (Encapsulated PostScript) came about. Even today, many users still use EPS files as part of their workflows.

Since PDF originally shared a common imaging model with PostScript, it made sense for Adobe to create a tool that converted PostScript and EPS to PDF – Acrobat Distiller, which has been available since Acrobat 1.0. However, PDF's imaging model diverged from PostScript over ten years ago with PDF 1.4's introduction of transparency and has continued to include additional features such as JPEG2000 compression and Optional Content (Layers) that are seeing greater usage in the print area.

What this means is that while creating PDF from PostScript remains an acceptable method for PDF creation, the conversion of PDF back to PostScript or EPS can be an extremely lossy operation and can cause many problems later in the workflow. This paper will examine some of the various issues that can occur and how you can try to avoid them when necessary.

1.3 Terminology

While the history of the usage of the term has been lost to the sands of time, the term refrying has come to refer to the process of converting a PDF file into PostScript (or EPS) and then back again to PDF.

A user can do this operation explicitly, such as with using Adobe Acrobat to first "Save as PostScript" and then feeding that PostScript to Distiller, or more commonly a user just prints from Acrobat to the Adobe PDF Printer. This practice started years ago as a way to convert "troublesome PDFs" into ones that would print properly when there were no other tools to help solve such problems. Unfortunately, while there exists modern tools to address any problems that might be encountered, many users still insist on this process because "it worked in the past, so it must still be good today."

In addition, a user may be working with an application such as Quark, Adobe PageMaker or Adobe FrameMaker that will convert all placed PDFs into EPS. When the entire document is later printed to the Adobe PDF printer to PostScript is generated and fed to Distiller, a refrying is taking place.

There is, however, another similar operation that is sometimes also called refrying, but is more commonly and correctly called repurposing. To repurpose a PDF is to use it in a way that wasn't (necessarily) originally intended. Placing a PDF of an advertisement into a larger publication is an example of repurposing, as is performing an imposition of multiple pages of a document into a new layout or converting a printed piece for use on the web. For this particular document, we will focus strictly on refrying and leave discussions about repurposing for another document covering just that topic.

2 How refrying can go wrong

As mentioned earlier, the process of refrying involves, at least, a double conversion – first from PDF to PostScript and then from PostScript back to PDF. When working in a PostScript/EPS-based workflow, such as Quark, you may even be performing as many as four or eight transformations on your data. With each transformation, of course, comes the opportunity for errors and loss of your data.

2.1 Font and Text

Fonts in PostScript and PDF are almost identical, provided that you were only going to be printing the PostScript (since that is what it is designed for). PDF, however, also incorporates various features that enable the glyphs of the font to not only render correctly, but also be extracted back out as text and usually as Unicode. Therefore, it is quite common that in the process of refrying the ability for text to be extracted – even via a simple copy & paste – will most likely be lost. This also impacts the ability for search engines, including those built into various Operating Systems, to do their job of indexing the content.

In the print production world, it is quite common for a printer to have to what is referred to as a “late stage edit” – where a typo has been located and it easier for them to simply edit (or touch up) the text of the PDF than to go back to the original source. When fonts information undergoes the refrying transformations, there is ample opportunity for fonts to be changed in a myriad of ways, which can severely impact the ability of a printer to edit in this way.

2.2 Resampling

Since PDF supports image compression technologies such as JPEG2000 and JBIG2 that PostScript does not, the conversion process will require that the image data be decompressed and then (possibly) recompressed with another algorithm such as JPEG or CCITTG4, respectively. The PostScript algorithms are not as efficient as the PDF ones, which leads to quite large print streams.

With JPEG being a lossy algorithm, it is expected that image data will be lost in the process. Even if the PDF itself uses only JPEG, it may be that improper settings in both the export and the distillation can lead to image data loss as well.

In addition, the new compression algorithms, PDF supports 16 bits per component (bpc) images, while PostScript remains at 8. Here is another place where image data will be lost, since half of each pixel of the image will need to be thrown away.

2.3 Smooth Shading

PostScript Level 3 introduced a native set of operators for the rendering of high-quality smooth shadings (also known as gradients) and these were also introduced into PDF 1.3. The upshot of this is that any conversion of a PDF with smooth shadings to PostScript/EPS must be done with Level 3 and not the older Level 2 conformance. If the conversion takes place using Level 2, the single smooth shade will be converted to a series of lines or paths which will bring about a significant increase in file size and reduce the quality of rendering due to banding and nonlinear color. In addition, since it will no longer be a single object, the ability to perform late stage editing is eliminated.

2.4 Color Management

As PostScript was designed to be device dependant, all colors specified in such a file are also device dependent and don't offer the rich set of independent spaces present in PDF, including support for the industry standard ICC profiling technology. Therefore, when working with a PDF that is designed to operate in a color-managed workflow, all such colors need to be converted.

If the conversion is to PostScript, the original ICC profiles cannot be maintained to use in the reverse conversion – however, EPS supports a special construct (ie. DSC comment) that enables the ICC profile to be embedded and reused later. This, however, only works for a single profile for the entire content and doesn't serve when the content is prepared with multiple profiles.

3 What to do?

Use EPS and not PostScript!

Because EPS is device independent, offers support for maintaining ICC profiles and can usually be produced more easily from some applications, it is recommended that anytime refrying is unavoidable that EPS be used instead of PostScript. It is also preferable that users should always do the conversions themselves, where the settings can be controlled, rather than letting an application into which you are placing a PDF do it for you. By assuming control and ensuring that the settings are chosen correctly, it is possible to keep the amount of data loss to a minimum.

The following settings are recommended:

- Export as Level 3, Binary

  This is the most modern and compact form of EPS thus ensuring full support for all features possible in the smallest file size.

- Include all embedded fonts

  By including all of the original font data from the PDF file into the EPS, there is never an opportunity for Acrobat Distiller (or a similar process) to replace the font with a different one.

- Never allow font formats to be converted, such as TrueType to Type 1 or a CID-encoded font to be un-encoded.
- Don't enable color conversion, unless the workflow calls for it.

3.1 Alternatives to refrying

While refrying a PDF to integrate it into a non-PDF workflow is inevitable, there are no reasons to refry a PDF simply to modify it in some way for a PDF workflow. There exist many tools for both the desktop and server that enable the transcoding of a PDF into a new PDF with the characteristics required. Common examples such as preparation for posting on the web, performing color conversion, font correction, and file size reduction can all be performed on native PDFs and will always yield a much better PDF at the end.

4 Conclusion

Sometimes a specific workflow requires the implicit refrying of a PDF due to the use of older (or non-PDF-native) tools, in this case there isn't a choice and the conversion must take place. However, there is no reason to let the conversion process be driven by such software. Instead, it is recommended that manual and explicit conversion to EPS be performed using the recommends above.

While implicit refrying is inevitable sometimes, there is never a reason to explicitly refry a PDF. There is nothing that can't be accomplished via refrying that can't also be accomplished with native PDF transcoding tools. Since the process of refrying will cause data loss, it should not be used.

Ghent
Workgroup