
Transparency Best Practices

Author **Jason Lisi**
Documentation Officer
jlisi@ryerson.ca
Date April 12, 2018
Status Reviewed in January 2020



Ghent
Workgroup

www.gwg.org

Table of Contents

1	Introduction	3
1.1	About the Ghent Workgroup	3
1.2	About This Document	4
1.3	More Information	4
2	Transparency: An Overview	5
2.1	Transparency 101	5
2.2	Feathering	6
2.3	Opacity	6
2.4	Blending Modes	6
3	Designing with Transparency: What Actually Happens	8
3.1	General Considerations	8
3.2	Drop Shadows	8
3.3	Inner and Outer Glows	9
3.4	Transparency and final output	10
4	Best Practices for Using Transparency	11
4.1	Design Smartly	11
4.2	Color Model and Default Blend Space	12
4.3	Object Order Matters	14
4.4	Transparency to Create Drop Shadows	15
5	Conclusion	16
6	Other Resources	17
7	References	18

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 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, packaging, digital print and signage, preflighting PDF files, and in developing test suites to ensure workflows and applications are configured and used correctly.

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 tools to implement best practices and/or find areas of improvement
- 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.

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 About This Document

This document is intended to aid designers in better understanding native transparency, and provides best practices that can be used to help avoid some of the most common print problems associated with transparency. This document is not intended to provide a comprehensive technical overview of transparency, nor does it explain the detailed functions of transparency within PDF. For more information on these topics, please refer to the GWG website at www.gwg.org.

1.3 More Information

The GWG website (www.gwg.org) contains a lot of useful information regarding the GWG, its specifications and deliverables, and caters to vendors, associations and end-users alike.

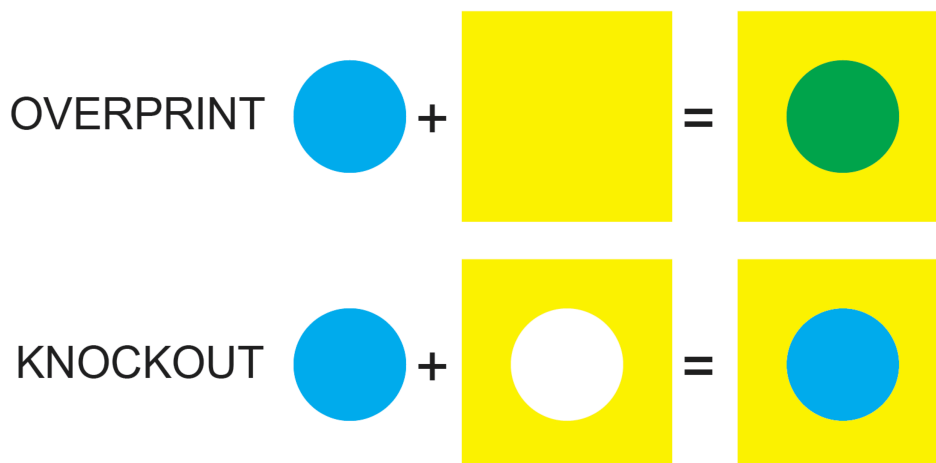
2 Transparency: An Overview

Native transparency, also known as vector-based transparency, has been a useful tool available in many illustration and page layout programs for almost two decades. While the ability to add transparent effects to design objects greatly enhances the creative palette, it can lead to problems on press if not done correctly.

In this document, we will look at the fundamental concept of what transparency is, how it is used, and what can be done in terms of best practices to avoid some of the more common pitfalls associated with using – and printing – transparency.

2.1 Transparency 101

Before we look at ways in which transparency is used, it will be a good idea to have a better understanding of what native transparency is, and how it works. The first important thing to understand is that transparency and overprint are not the same thing. Overprint occurs when two or more colours are printed on top of each other. For example, when a cyan circle is overprinted on top of yellow, the result would be a green circle on a yellow background. If the cyan circle is knocked out of the yellow background, the result would be a cyan circle on a yellow background. Overprints and knockouts are a function of printing, and do not rely on transparency effects in design software to work.



Transparency is different than overprint, and is used primarily to create artistic effects like shadows and feathering. As a graphic artist, you know the ability to create see-through effects using transparency has both esthetic and practical benefits. Transparency effects like soft drop shadows, blending modes, and feathered edges enable us to create eye-catching designs right in illustration and page layout programs, eliminating the need to spend extra time creating these effects in image manipulation programs (such as Adobe Photoshop) and then importing them. This convenience, combined with the ability to export designs as press-ready PDF files, makes native transparency functional, creative, and easy to use.

Let's look at some of the more common uses of transparency in a little more detail.

2.2 Feathering



Feathering softens the edges of an object by fading the object from opaque to transparent over a specified distance. This is done as a gradual, controlled change in opacity.

Feathering the edge of an object causes the object to fade smoothly outward and blend into the page background or any objects behind it. The size (distance), softness, and smoothness of the feather are all controlled by the user.

Most of today's illustration and page layout applications will allow you to apply feathering to vector-based objects, as well as placed bitmap images.

2.3 Opacity

We can vary the degree of transparency of an object or a group by adjusting its opacity. An object with an opacity of 100% is totally opaque (that is, you can't see anything behind it); an object with an opacity of 0% is totally transparent. As you decrease an object's opacity, the page background or any underlying artwork becomes increasingly visible through the object.

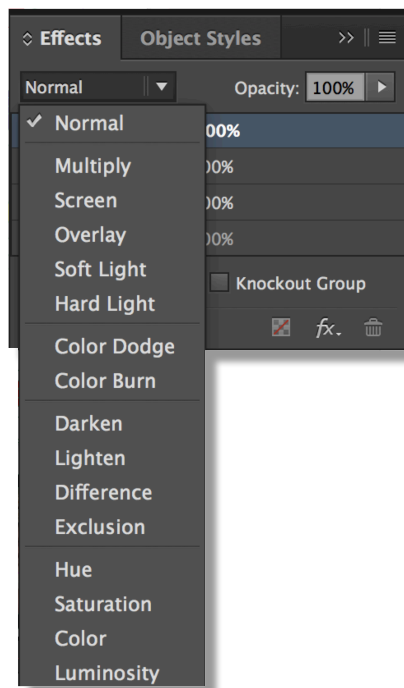
In the example below, the word "OPACITY" has been written three times. The instance on the top is 100% opaque (0% transparent), the middle instance is 60% opaque (40% transparent), and the bottom instance is 20% opaque (80% transparent).



2.4 Blending Modes

Blending modes, also sometimes called transparency effects, are another tool that can be used to create transparency effects. The option to apply any of several blending modes to objects lets you vary the way the colors of objects blend with the colors of underlying objects. Blending can be applied to many different things, including layers, objects, and special effects to name a few. When using blending modes, it can be helpful to think about the three colors that come into play: The base color, the blend color, and the result color. In general terms, these colors can be defined as follows:

- The *base color* is the original color;
- The *blend color* is the color being applied (i.e. the color that is being manipulated); and,
- The *result color* is the color resulting from the blend.



There are several different blend modes that can be applied to a design. Each blend mode will generate a different result. For example, **Multiply** works by assessing color information in each channel and multiplies the base color by the blend color. The result color is usually, but not always, a darker color than the base or blend color. Multiply is used a lot with shadow work.

In contrast, **Screen** examines each channel's color information and multiplies the inverse of the blend and base colors. The result color is always a lighter color than the base or blend color. Screen is often used to create inner and outer glows.

When working with blend modes, the default blending color space becomes very important, especially when working in documents that will be printed. We will discuss blending colour space later in this document.

3 Designing with Transparency: What Actually Happens

When we incorporate native transparency into our design, there are a lot of complex processes happening in the background to make that transparency work. This is exceptionally true considering that it is easy to have several transparency effects applied to a single object. For example, a simple drop shadow uses opacity changes, a blend color space, and feathering!

3.1 General Considerations

Transparency effects can be applied to an object, like a stroke or a fill applied to a piece of vector art. As mentioned previously, you can apply more than one transparency effect to any object. You can also apply transparency effects to a group of objects.

It's also possible to add transparency to a document by importing/placing transparent artwork created in another application. For example, you can place a native Photoshop file with a feathered edge into an Illustrator layout, or you can place a PDF file with a soft drop shadow into InDesign or QuarkXPress.

Not all transparency is equal when it comes to simplicity and complexity. For example, as the number of overlapping transparent objects increases, so does the complexity of the transparency effects. Placing a box with a drop shadow on top of only the page background is simpler than placing shadowed text in front of an imported graphic—in which case the transparency effect (the drop shadow) must be combined with the underlying graphic for display and printing.

When creating transparency effects, care should be taken not to create effects that produce undesirable results. For example, it's possible to feather the edge of small type with fine serifs, but if you do, the resulting text might be difficult or impossible to read or print. Use good judgement when designing with transparency.

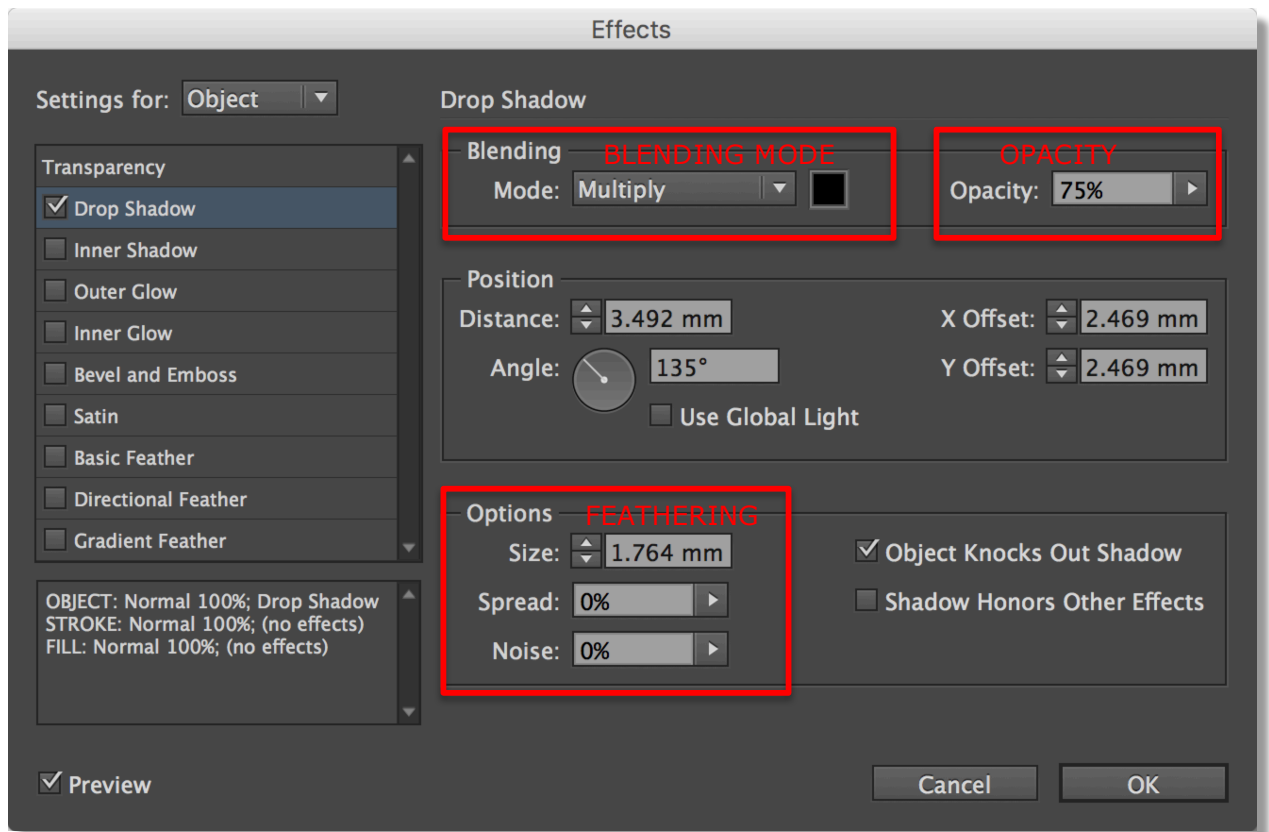
Feathered Serifs

In all the examples that follow, a softmask image is automatically created that is used to control opacity of the transparency effect, such as the drop shadow object or the inner/outer flow edge of objects.

3.2 Drop Shadows

**DROP
SHADOW**

Drop shadows are moderately complicated in the grand scheme of things. There are a few different transparency effects being applied in a single drop shadow, as the following InDesign dialogue box shows:

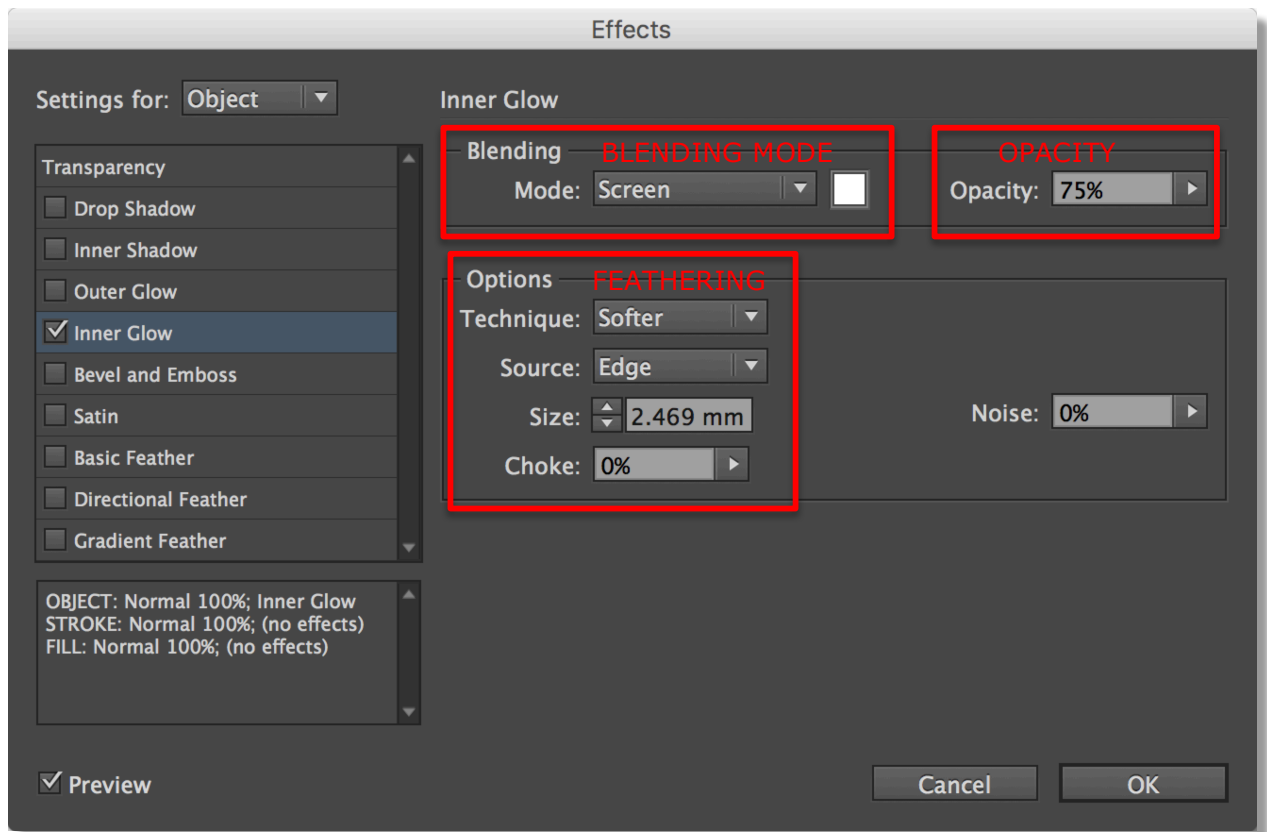


3.3 Inner and Outer Glows



By default, both Inner and outer glows rely on the Screen Blend Mode to create the effect of glowing edges. This can be a useful tool, especially if we want to give something the illusion of being illuminated.

Just like drop shadows, glows employ multiple transparency effects as the following InDesign dialogue box shows:



3.4 Transparency and final output

The Ghent Workgroup endorses the use of the PDF/X-4 standard when creating print-ready PDFs, and as such, the GWG 2015 specification is based on PDF/X-4. One of the advantages of PDF/X-4 is that it can retain native file transparency: That is, the PDF file does not require flattening (resolving) prior to output. When a PDF/X-4 file is output on a RIP with a native PDF interpreter, the transparency blending occurs as part of the render process.

While this model works well the majority of the time, issues can arise when poorly designed transparency effects using various blend modes carry forward to the RIP, resulting in incorrect output or bad processing performance. Also, different RIPs might result in different results when not fully supporting transparency.

The GWG output suite has several patches that will test to final rendering of transparency effects. If you would like to test your settings, you can download the Ghent PDF Output Suite 5.0 for free from the GWG website: <http://www.qwg.org/workflow-tools/test-suites/ghent-output-suite/>.

4 Best Practices for Using Transparency

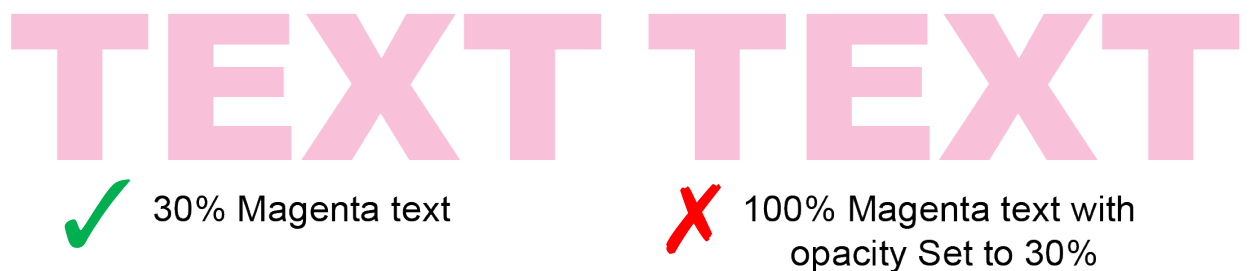
When done correctly, native transparency can add trouble-free creative flair to a design. There are some instances, however, where transparency has caused issues on output, especially for print-based projects. In most cases, the issues that arise when printing documents with transparency can be avoided by following some basic rules and procedures.

In this section, we will discuss some of the things we can do to ensure that our designs print trouble free.

4.1 Design Smartly

4.1.1 Keep It As Simple As Possible

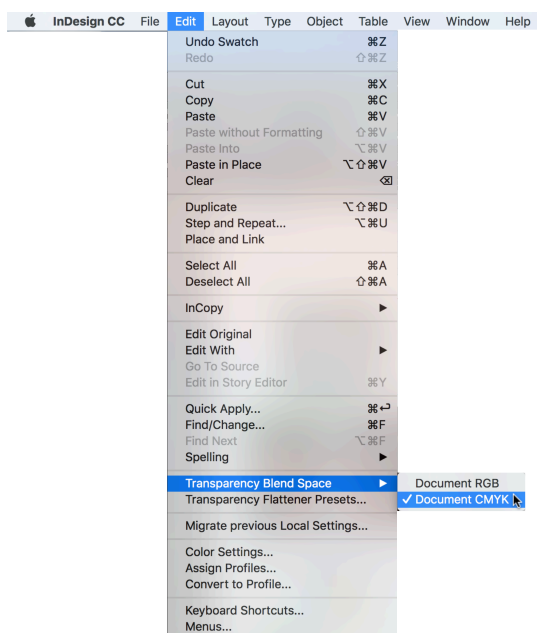
Remember that files can get computationally complex as we add more and more transparency. Keeping the transparency interactions simple can help alleviate a lot of problems. For example, text that is 30% opaque, with a drop shadow, over a box that is set with a transparent gradient, over a box that has an outer glow and be a complex scenario to compute. Try to avoid gratuitous use of transparency. For example, let's say we want to make 100% magenta text on a white background 30% opaque. In this example, it would be better to just color the text as 30% magenta. You will get the same result.



Another design note to keep in mind, especially for work that will be flattened prior to output, is to avoid having vector transparent objects (especially text) only partially overlapping other objects. This can avoid problems where part of a word or sentence is flattened (and therefore rasterized), while the rest remains vector. This issue can result in the flattened text looking "fatter" than the non-rasterized text.



4.2 Color Model and Default Blend Space



One thing that can really mess up transparency is multiple or conflicting blend spaces. The Transparency Blend Space that is selected tells the application you are using which color model to use when computing the colors that result from transparency. The two options here are RGB and CMYK.

Which default you use is really dependant on the workflow that you are using. For example, if you do a lot of cross media design – that is, designing one product that is distributed for both web and print – you might want to consider an RGB Blend Space. In contrast, if you design only for print, a CMYK blend space might be better. It should be noted that the GWG 2015 specification requires PDFs to have a CMYK transparency blend space to comply with the specification.

Where issues arise is when two or more documents are combined, where each has a

different blend space. For example, you may import an Illustrator file into InDesign. The Illustrator file has an RGB Blend Space, while the InDesign file's Blend Space is set to CMYK. Now imagine these two transparent objects, the one in generated in InDesign, and the one imported from Illustrator, with different Blend Spaces, overlap. How will the software manage this? What will be converted? This can lead to all sorts of issues. This can get even more complex when objects within the same file have different Blend Spaces.

The GWG recommends that all imported and embedded files use the same transparency blend space as the destination file. Similarly, the GWG recommends that all objects within the same file have the same transparency blending space, which must be set to be either undefined or DeviceCMYK. This is the requirement for the GWG 2015 specification, and the corresponding preflight profiles will check PDF files against these attributes.

Below we have an example of artifacts that can occur in printing when color spaces are mixed up and have either specified, or improperly specified, page level transparency groups and transparency blending color spaces.

Image 01 shows a direct output from Microsoft PowerPoint 2016 Windows using Acrobat PDFMaker. In the process of creating the PDF, all R=G=B text and vector objects are converted to grayscale. It looks fine on screen because Acrobat/Reader displays everything in RGB. However, when the file is printed, the result can be seen in Image 02.

There are differences/discontinuities in the background grays and blacks in the areas of the shadows and reflections. This is caused by an unnecessary page level transparency group with an ICC Adobe RGB transparency blending space specified.

In order to correct this problem, the PDF must be "hand modified" to eliminate the unnecessary page level transparency group with the ICC Adobe RGB transparency blending space. Printing this yielded the results can be seen in Image 03. The differences/discontinuities in the background grays and blacks are gone, and the file prints as intended.

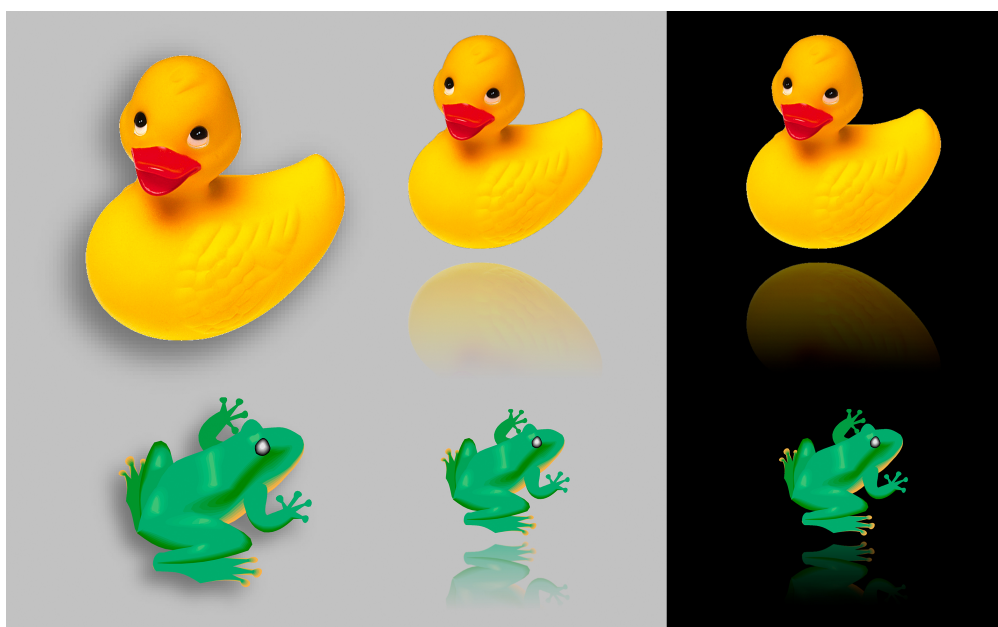


Image 01: PDF file as viewed in Acrobat/Reader

R=G=B Vector Fill Background overlaid with RGB Images & Vector Artwork using Transparency Effects (Drop Shadow & Reflection)

Acrobat.PDFMaker creates DeviceGray for all R=G=B text & vector objects and ICC Adobe RGB for other vector artwork and all images.

Page Transparency Group with ICC Adobe RGB Transparency Blending Color Space. Desired print as DeviceCMYK (0,0,0,K) for gray and black backgrounds with no rich black artifacts

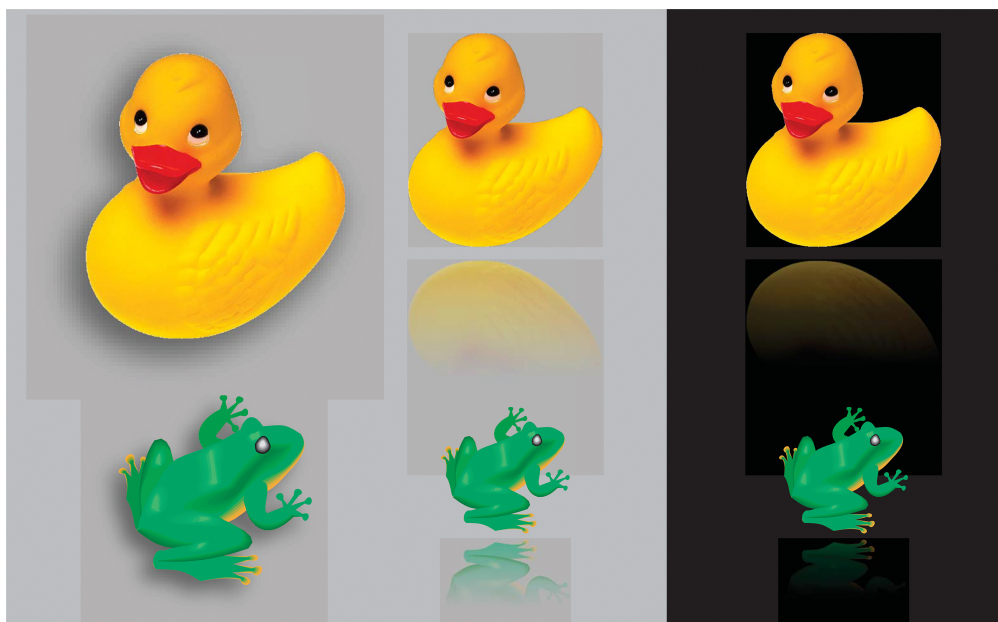


Image 02: Printed result from PDF file

Printed result has clear unwanted artifacts in gray and rich black backgrounds.

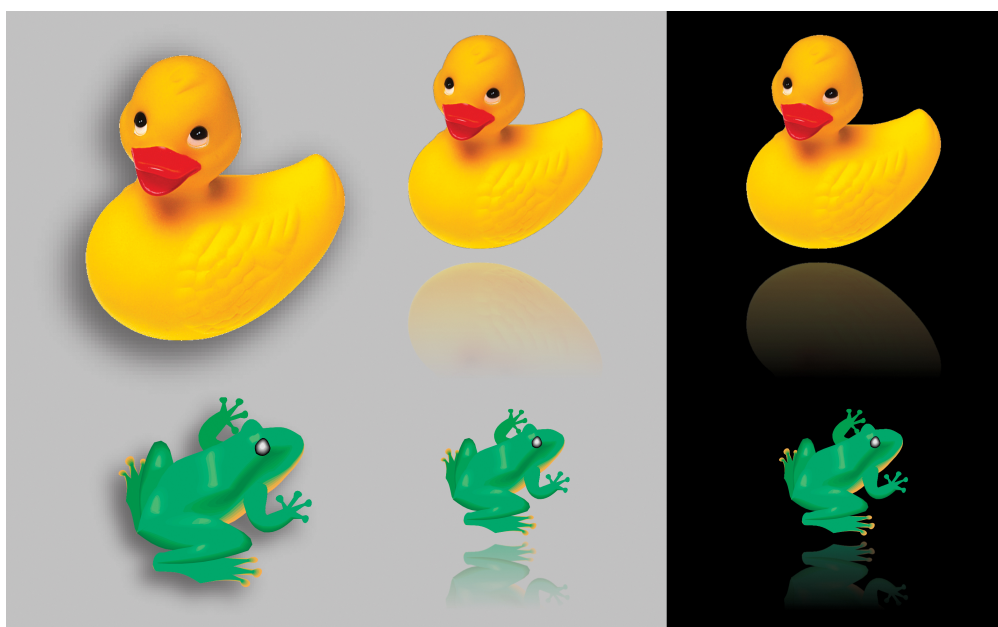


Image 03: Printed result after correcting PDF

Desired print as DeviceCMYK (0,0,0,K) for gray and black backgrounds with no rich black artifacts.

4.3 Object Order Matters



Every time you create a new object, it's placed on top of all existing objects. This creates a stack of objects, one on top of the other (stacking order). Stacking order becomes more complicated when you're working with a group or a layer, each of which has its own stacking order and can have transparency applied differently to an object, a group of objects, or layers of objects. You can adjust the stacking order of an object by moving it forward or backward relative to other objects on the layer or page.

Layers (and the objects in layers) are also in a stacking order—the layers at the top of the list in the Layers palette (Window > Layers) are in front of the layers at the bottom of the list. The stacking order of objects is critical for correct display and printing of transparency.

Transparency effects don't have to involve overlapping objects, but this is often the case. For example, you can apply an opacity value that's less than 100% to lighten any object regardless of whether the object is in front of other objects. However, by applying an opacity value, you not only make an object lighter, you also make it semi-opaque, which causes it to blend with any underlying objects it overlaps.

When you're working with transparent objects that overlap, you should keep in mind that changing the stacking order can change the appearance of overlapping areas. When you create transparency effects that involve overlapping objects, make sure the stacking order is producing the intended results.

Generally, text should be on top of all other objects unless it is supposed to interact with transparency. Placing text frames and spot colors on the top of the stacking order will help keep them from being involved in transparency flattening.

4.4 Transparency to Create Drop Shadows

Drop shadows that have soft edges and allow the colour behind them to show through are soft shadows that contain transparency. Soft shadows are 'constructs' that are not native to PDF, and must be 'created' by the application. That is, an image will be generated from the (e.g.) text, that contains the actual drop shadow, and that image will be placed with some transparency info. As such, the quality of such effects also depend on the bitmap generation of the design application. It is recommended that any drop shadows created in this manner are tested through hard copy output (i.e. a proof) prior to final output.

5 **Conclusion**

Native transparency is a powerful tool that can enhance and streamline the design process. Understanding how to properly employ transparency can avoid costly mistakes down the chain of production.

Best practices for transparency consider the nature of transparency, the fundamentals of its application, and the end use or destination of the file. Remember that not all transparency is created equal, and the more complex the transparency the higher the likelihood of output problems. This is especially of concern when native transparency is resolved in the RIP at output, as the issues may not be evident before this critical stage in the workflow.

In addition to this whitepaper, the GWG has an array of educational materials, tools, and resources to help you increase your success with native transparency. For the latest materials, please refer to gwg.org.

6 Other Resources

The following resources can be downloaded from gwg.org:

- GWG 2015 Specifications
- GWG 2015 Packaging Specification
- 9 Reasons to Preflight

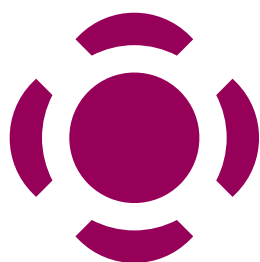
7 References

Lisi, J and Smyth C. (2010). Prepress Workflow Advancements Utilizing PDF/X-4 and the Adobe PDF Print Engine. *Proceedings of the Technical Association of the Graphic Arts TAGA 62nd Annual Technical Conference*, 265-297

Lisi, J and Smyth C. (2011). PDF Workflows for the Real World. *Proceedings of the Technical Association of the Graphic Arts TAGA 63rd Annual Technical Conference*, 403-436

Ghent Workgroup (2016). *GWG2015 PDF/X Workflow*. Available at www.gwg.org

Ghent Workgroup (2013). *PDF Specification in Practice: GWG2012_CMYK*. Available at www.gwg.org



Ghent
Workgroup
