ONYX
THRIVE
FLASH

Max-imize Your Color:

Expert Tips from our Color Scientist Max Derhak

Troubleshooting Transparency Problems

The use of transparency is a relatively new capability added to many creative applications which have opened new opportunities for creating documents by graphic artists. However, this capability also introduces complications in reproducing/printing documents in a predictable fashion. This article discusses how transparency works, common difficulties that can arise, and methods/recommendations relative to its use.

Understanding Transparency

Though the concept of transparency has been around for many years its use became much more prevalent with the introduction of PDF 1.4 and later versions. These versions allow for transparent layer processing to be specified by a PDF file and performed by the RIP.
At a very basic level, transparency between two layers is a fairly simple concept, and can be described as a combination of two raster images (a background image and foreground image) with an opacity value that is used to describe how to combine (blend) the two images' pixels. The following simple mathematical expression can be used to describe how basic transparency blending works for each corresponding pixel in a transparency region:

$$\begin{gathered}\text{blended\_pixel} = \text{background\_pixel} * (1 - \text{foreground\_pixel\_opacity}) + \\ \text{foreground\_pixel} * \text{foreground\_pixel\_opacity}\end{gathered}$$

Equation 1 - Transparency blending equation

In this case, the foreground_pixel_opacity in the above equation is represented as a fractional percentage ranging from 0.0 to 1.0 with 1.0 representing 100% opacity of the foreground pixel and 0.0 representing a totally transparent foreground (resulting in 100% of the background showing). The foreground_pixel_opacity can be a single value for the entire blended image or it can be defined on a pixel by pixel basis (which makes up the contents of an alpha mask).

There are two requirements that are assumed by the above transparency blending equation. The first requirement is that the elements in the both the background and foreground are completely rasterized as pixels (meaning all vector elements need to first be converted to raster (pixel based) images). The second requirement is that the pixel format is identical for both the background pixels and foreground pixels. This may involve color space changes or color conversions of the background pixels, the foreground pixels or both.

The following example can be used to help describe how transparency works and provide a reference for discussion of possible problems that can occur. Suppose that we have a simple document with three element layers. The first layer is a box filled in with the spot color "Pantone Warm Red C". Above this is a circular drop shadow drawn as a black circle which is 75% opaque. On top is a cyan circle drawn at an offset from the drop shadow. Figure 1 shows the final result with the layers labeled.

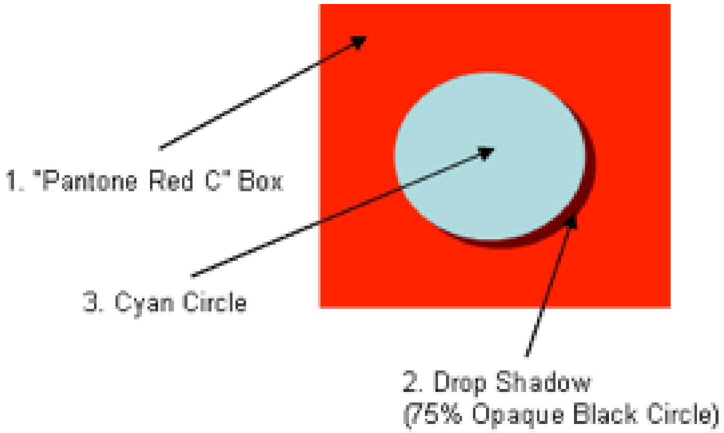


Figure 1 – Simple example with a semi-transparent drop shadow with labeled layers

The transparent portion of this example is made up of only the first two layers, and for the purposes of the transparency blending Layer 1 is the background, and Layer 2 is the foreground. Additionally, there are two important regions (inside versus outside the transparent region) where separate operations are performed to rasterize each region to the final output canvas (See Figure 2).

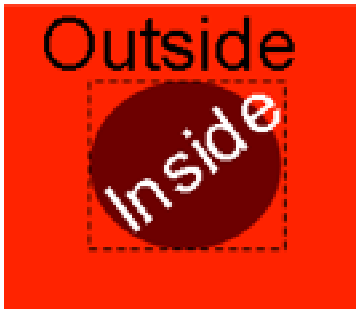
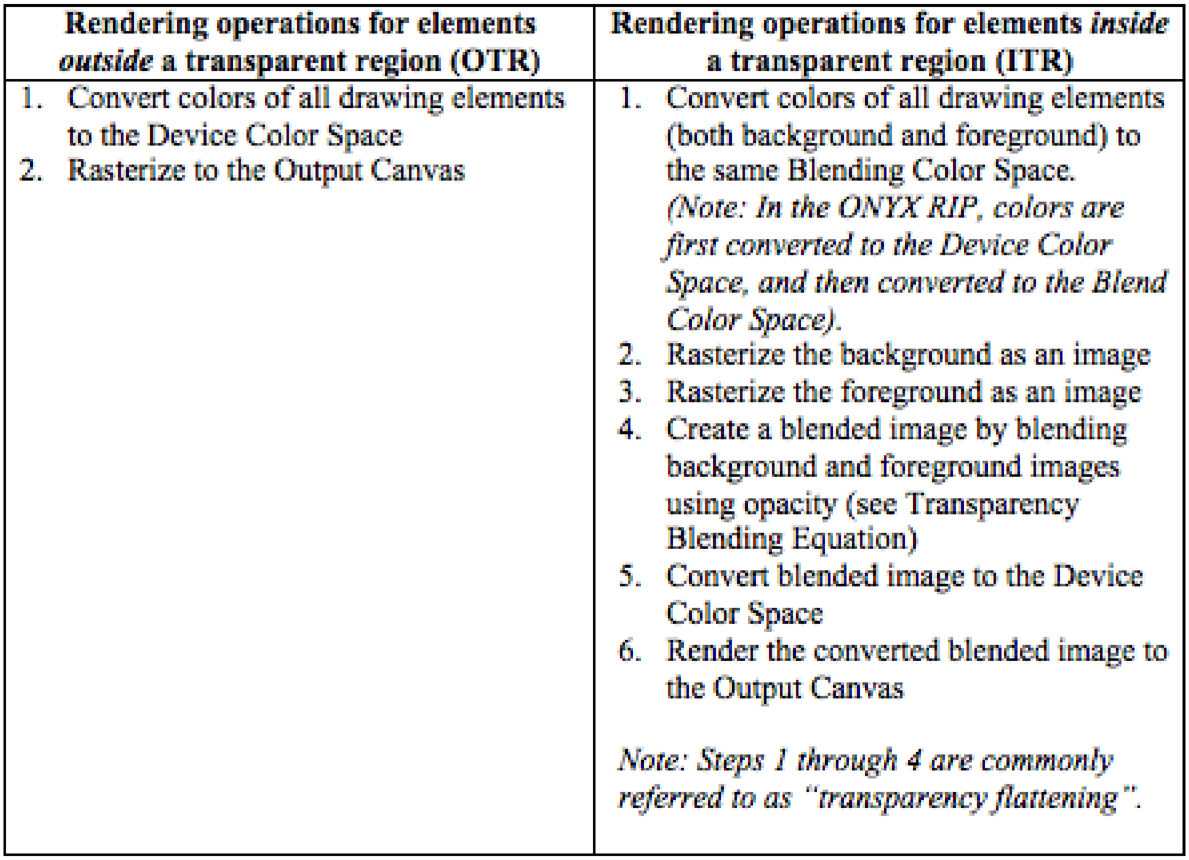


Figure 2 – Transparency regions

The following table (Table 1) presents a comparison of the rendering operations for both inside and outside a transparent region. The rendering operations for elements outside a transparent region (OTR) are fairly simple as shown on the left, while the operations for elements inside a transparent region (ITR) being slightly more involved are shown on the right.



Rendering operations for elements *outside* a transparent region (OTR)	Rendering operations for elements *inside* a transparent region (ITR)
1. Convert colors of all drawing elements to the Device Color Space 2. Rasterize to the Output Canvas	1. Convert colors of all drawing elements (both background and foreground) to the same Blending Color Space. *(Note: In the ONYX RIP, colors are first converted to the Device Color Space, and then converted to the Blend Color Space).* 2. Rasterize the background as an image 3. Rasterize the foreground as an image 4. Create a blended image by blending background and foreground images using opacity (see Transparency Blending Equation) 5. Convert blended image to the Device Color Space 6. Render the converted blended image to the Output Canvas *Note: Steps 1 through 4 are commonly referred to as "transparency flattening".*

Table 1- Comparison of rendering operations outside (left) and inside (right) of a transparency region

Transparency Problems

Problems can occur when there are drawing elements that are both inside and outside a transparency region and the color conversion operations in both regions result in different colors. One common example of this is where the spot color replacement applied by the inside region rendering operations is different that the spot color replacement applied by the outside region rendering operations. An example of this can be seen for our example in Figure 3 where the "Pantone Warm Red C" color has been converted differently by the inside and outside rendering operations.

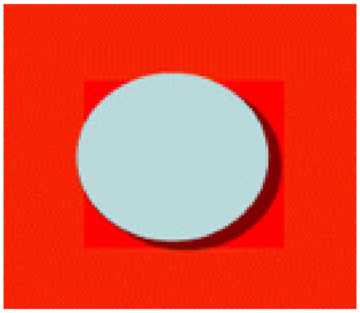


Figure 3 – Example of mismatch between spot color replacement conversions inside and outside the transparency region

In looking at the processing operations for both inside and outside transparency regions (in Table 1) one can see that a color will be the same for both regions if (and only if) the transform from the Original Color to the Blending Color Space and then to the Device Color Space is the same as the transform from the Original Color to the Device Color Space.

There are at least three possible reasons why the color conversions in these two cases can be different:

1. Differences can occur if transparency flattening (Steps 1 through 4 in the ITR case) is performed by the creative application (rather than by the RIP). This can be because the output file format being used doesn't natively support or make use of transparent blend operations. BE AWARE - EPS, PS and PDF (before version 1.4) files do not support transparency blend operations.

   (Note: Some PDF files may be created so that transparency operations are not used even if the PDF file version is specified to be 1.4 or greater. This may be the case of PDF/X).

   In these cases, if a creative application supports transparency, the creative application will need to perform the transparency flattening thus creating raster images for all transparent regions that are then placed in the saved output file. Any colors in these regions will be converted using the creative application's color transforms, and in some cases these transforms can be very different than the color transforms that will be applied by the RIP for the colors outside the transparent regions.

   (Note: This can become further complicated by embedding EPS files created by multiple (different) applications that perform their own transparency flattening with their own transforms for color converting to the blending color space they use).

   Two particular cases of color transforms relative to transparency are important to consider:
   a. Spot Color Replacement. The spot color replacement transforms are very likely to be different between a creative application and the RIP since the RIP is more likely to have access to the Print Mode's color management data to get a closer match to the spot color. However, most likely the default spot color replacement encoded in the file will match between the inside and outside regions. Turning OFF spot color replacement will most likely result in getting the same results between the inside and outside regions since you are turning off a different outside region transform.

   b. Color Management Differences between document element types. Since the inside regions are defined by images and the outside regions are likely to be defined as vector data it is important that these two element types are treated identically. This means that for Image and Vector Data, both the ICC profiles and the Rendering Intents should be the same.

2. Differences can occur if information is lost in the transform from Device Color Space to Blending Color Space to Device Color Space. The PDF specification allows the blending space to be determined by the PDF file. Generally, the blending space will be the same as the working color space in the creative application that creates the PDF file. If the working space is significantly different (encoding fewer channels) than the Device Color Space implied by the Print Media in the RIP then information will be lost when going from Device Color Space to the Blending Color Space. Two examples of this are going from CMYK to RGB or CMYKRGB to CMYK. In both of these cases color separation information is lost. Converting back to the device color space may be close (in appear-

ance), but with a loss of information the conversion could just as well be rather different. Ideally for a PDF workflow with transparency blending being performed in the RIP it is recommended that the working space in the creative application be the same as the device working space defined by the Print Media.

3. Difference can occur due to differences that exist between code paths in the RIP for inside and outside transparency regions. In this case there may be a problem that probably needs to be fixed. Transparency processing is fairly complicated and small changes to fix one problem can potentially introduce other problems. This is not desirable behavior, and it is recommended that problem files be escalated through tech support in order to find a resolution through a change in an updated version of the RIP. Every effort should be first made to ensure that the first two problems are not the cause.

Transparency Recommendations

For better results with transparency the following is recommended:

- Use PDF 1.7 (non-PDF/X), resulting in all transparency processing being deferred to the RIP. This allows spot color replacement to better match inside and outside transparency regions.
- Use a PDF working space that closely matches the color space defined by the Print Media in the RIP (CMYK on CMYK print modes, RGB on RGB print modes).
- If EPS, PS, or PDF files (without transparency blend operations that contain pre-blended transparency image regions) are used it is advised to:
  - Turn OFF Spot Color Replacement
  - Make Color Management settings the same (both Profiles and Rendering Intents) for both Vector and Graphic elements
- If all else fails the following approaches might be taken, but are not generally advised:
  - Export the print mode's ICC profile and use it directly in the creative application. The output can hopefully then be color managed by the creative application to get consistent results between inside and outside transparent regions when the creative application performs transparency flattening. If this method is used then both Spot Color Replacement and ICC profiles should be disabled (implying that all color management has been performed).
  - First render the document as a TIFF file in the creative application. This is not the best solution because:
- The resulting image can not easily be scaled without pixelization.
- Spot Color Replacement will not be optimal for the print mode.
- BIG image files will be created.
- This is a slow process.

Conclusions

In this article a brief description of the operations that are performed both inside and outside transparency regions have been discussed. Being aware of the differences in color conversions required for transparency blending can be extremely helpful in identifying as well as preventing undesirable color differences.