PostScript Language Level 3 Extensions for Trapping

Technical Note #5622

ADOBE SYSTEMS INCORPORATED
Corporate Headquarters
345 Park Avenue
San Jose, CA 95110-2704
(408) 536-6000

13 June 2002

Date/Rev #	Comments
27 Feb 02	Describes extensions introduced in 2000.
13 June 02	Corrected inconsistent key naming that affected the PostScript version of this note, and clarified "TrapWidth" in trap join style diagrams.

Contents

Contents

Trapping Extensions

1 Overview

This technical note describes extensions to the PostScript language. These extensions consist of the following four new keys that have been added to the trap parameter dictionary:

- **TrapJoinStyle**
- **TrapEndStyle**
- **ImageToImageTrapping**
- **MinimumBlackWidth**

These keys control new features in the Adobe® Trap Engine and are extensions to the current trapping language specification. The current trapping language specification can be found in section 6.3 in the third edition of the PostScript Language Reference. Subsection 6.3.3, beginning on page 446, titled “Trapping Parameters” is most relevant.

NOTE: The new parameters reside in the trapping parameters dictionary and can only be changed at the trap zone level. The trap parameters dictionary is a member of the trapping details dictionary.

2 New TrapJoinStyle and TrapEndStyle Keys

The two new keys **TrapJoinStyle** and **TrapEndStyle** specify how traps are drawn. **TrapJoinStyle** controls the shape of the join between traps of the same color with a common end point. **TrapEndStyle** controls how to shape the end of a trap that touches a third object.

2.1 TrapJoinStyle

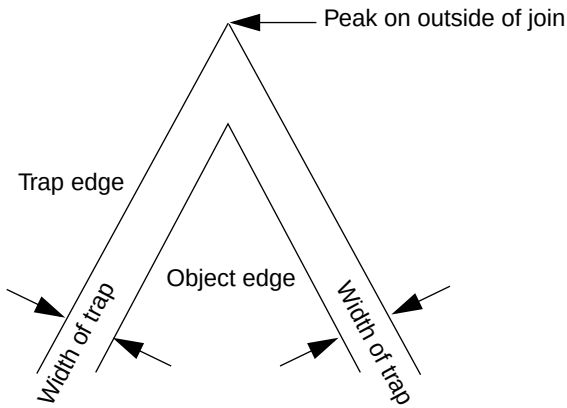
TrapJoinStyle is a key of type name that provides three options for the style of a trap join:

- Miter
- Round
- Bevel.

The default value is Miter to retain compatibility with previous versions of the Adobe Trapping Engine.

The join is where two trap edges meet at a common endpoint. Without applying a trap join style, the outside of the join would appear as a peak, as shown in Figure 2.2.

FIGURE 2.2 Hypothetical trap join without a join style applied



NOTE: This diagram is for illustrative purposes. The Adobe Trapping Engine will always apply a join style.

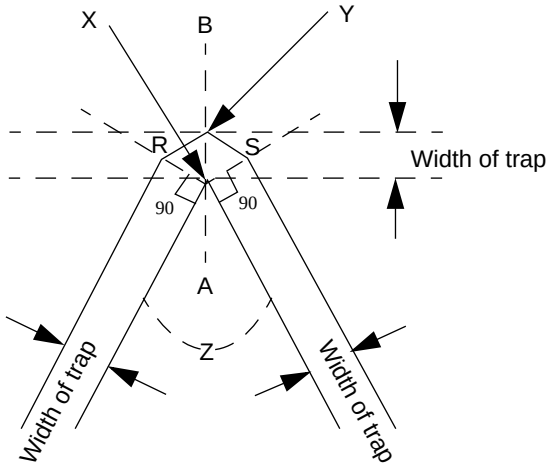
The style of a trap join is only visible on the outside of the join between two traps. The following sections illustrate and explain the construct each of the trap join styles.

Miter Join

Figure 2.3 illustrates a miter join. Construct a miter join using the following procedure:

1. Draw line AB through intersection point X of the object edges, such that it bisects the angle Z between the two line segments.
2. Choose a point R to the left of line AB and a point S on the right, where each point has the following characteristics:
   - Distance between the point and X is the width of the trap for that part of the object. The width of the trap is derived by the trapping engine, based on trapping parameters and the color and shape properties of the area being trapped.
   - Line between the point and X is perpendicular to the adjacent object edges.
3. Locate a point Y along line AB, where the distance between Y and X is the width of the trap.
4. Draw the outside of the miter trap join as the line segments connecting point R to point Y and point S to point Y, as shown in Figure 2.3.

FIGURE 2.3 Miter join

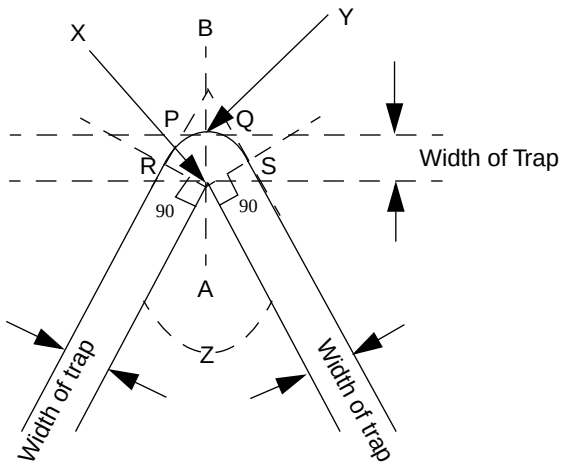


Round Join

Figure 2.4 illustrates a round join. Construct a round join using the following procedure:

1. Draw line AB through intersection point X of the object edges, such that it bisects the angle Z between the two line segments.
2. Choose a point R to the left of line AB and a point S on the right, where each point has the following characteristics:
   – Distance between the point and X is the width of the trap for that part of the object. The width of the trap is derived by the trapping engine, based on trapping parameters and the color and shape properties of the area being trapped.
   – Line between the point and X is perpendicular to the adjacent object edges.
3. Locate a point Y along line AB, where the distance between Y and X is the width of the trap.
4. Draw a line through Y perpendicular to AB, intersecting with the trap's edge at points P and Q.
5. Draw the outside of the round trap join as the Bézier curve segment, connecting point R to point S, using points R, P, Q, and S as control points.

FIGURE 2.4 Round join

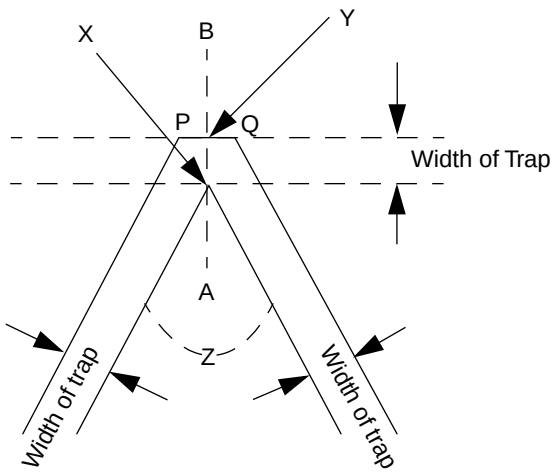


Bevel Join

Figure 2.5 illustrates a bevel join. Construct a bevel join using the following procedure:

1. Draw line AB through intersection point X of the object edges, such that it bisects the angle Z between the two line segments.
2. Locate a point Y along line AB, where the distance between Y and X is the width of the trap.
3. Draw a line through Y perpendicular to AB, intersecting with the trap edges at points P and Q.
4. Bevel the outside of the trap join at the line created between P and Q. The distance of the bevel from the edges being trapped should be the width of the trap on that side of the edges.

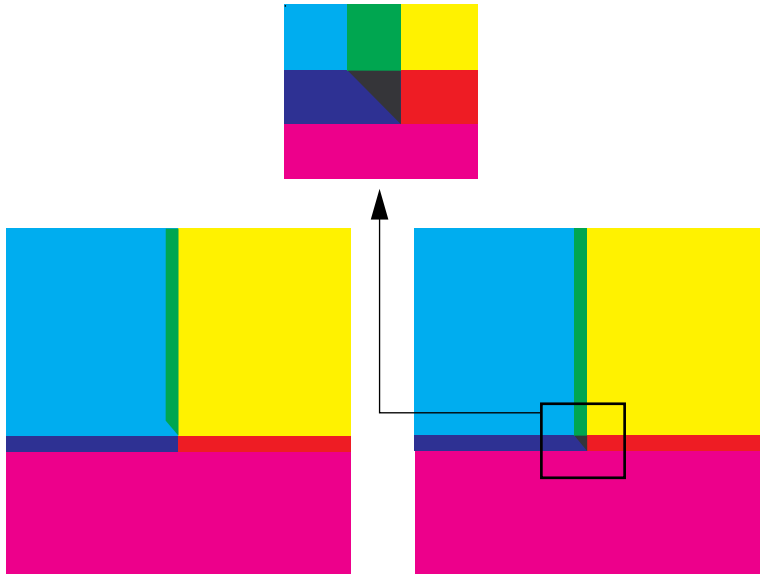
FIGURE 2.5 Bevel join



2.6 TrapEndStyle

TrapEndStyle is a key of type name that specifies how the Adobe Trapping Engine forms the end of a trap that touches another object. The **TrapEndStyle** values are Miter and Overlap. The default is Miter to retain compatibility with previous versions of the Adobe Trapping Engine. Figure 2.7 illustrates the two trap end styles.

FIGURE 2.7 Trap end styles



The Miter setting shapes the end of the trap to reduce trap risk with the intersecting object.

The Overlap setting affects the shape of the trap generated by the lightest neutral density object that intersects with two darker objects. The end of the lightest trap is wrapped around the point where the three objects intersect.

NOTE: Choosing the Overlap or Miter setting is mainly based on your preference of trap shaping, and the type of press being used for printing.

3 New Key for Image-to-Image Trapping

The new key **ImageToImageTrapping** of type boolean specifies whether to generate traps along an image boundary where the adjacent object is another image. The addition of this key means that trapping between two images is not affected only by the **ImageInternalTrapping** key as was the case in previous versions of the Adobe Trapping Engine. The default value for **ImageToImageTrapping** is *false*.

For image-to-image trapping (**ImageToImageTrapping** is *true*), the only value of **ImageTrapPlacement** that has a distinct effect is Normal, which will result in the Normal trapping rules being applied. All other values of **ImageTrapPlacement** will result in a centerline trap between images when **ImageToImageTrapping** is *true*, with the following exception: **ImageInternalTrapping** is also *true*, in which case image trap placement is always Normal between images. The following table summarizes the behavior of image-to-image trap placement based on the values of the keys for image trapping.

ImageToObjectTrapping	ImageToImageTrapping	ImageInternalTrapping	Image-To-Image trap placement
false	false	false	none
false	false	true	none
false	true	false	Normal if **ImageTrapPlacement** key is Normal; otherwise, CenterLine.
false	true	true	Normal
true	false	false	none
true	false	true	none
true	true	false	Normal if **ImageTrapPlacement** key is Normal; otherwise, CenterLine
true	true	true	Normal

4 Changes and Additions for Black Traps

A new key **MinimumBlackWidth**, which is a number type of key, has been added for controlling the width of black traps. Also the minimum values for the **TrapWidth** and the **BlackWidth** keys is 0.0.

To determine the width of a black trap, the Adobe Trapping Engine takes into consideration the **TrapWidth** value and two other values:

- **BlackWidth**. A unitless, trapping parameter that is a factor in determining the width of a black trap.
- **MinimumBlackWidth**. A trapping parameter, introduced with Adobe Trapping Engine version 305, whose value is the minimum width in points of a black trap.

NOTE: On Adobe Trapping Engines prior to version 305, **BlackWidth** multiplied by **TrapWidth** is the maximum width of a black trap.

When **BlackWidth** and **MinimumBlackWidth** are each greater than 0, the Adobe Trapping Engine determines the width of a black trap to be the greater of two values:

- **TrapWidth** multiplied by **BlackWidth**
- **MinimumBlackWidth**

With Adobe Trapping Engine versions prior to 305, it was not possible to set the normal **TrapWidth** value to 0. With Adobe Trapping Engine version 305 and higher, you can set **TrapWidth** to 0 and use the **MinimumBlackWidth** parameter to set the width for black traps. In this case, the Adobe Trapping Engine will only produce black traps. Also, prior to the version 305, **BlackWidth** could not be set to 0. With Adobe Trapping Engine 305 and higher, you can set both **BlackWidth** and **MinimumBlackWidth** to 0 to avoid creating black traps.

A commercial printer typically prefers the width of black traps to be 1.5 to 2 times the normal **TrapWidth** value. Because of this preference, the default value of **BlackWidth** was increased from 1 to 2 with Adobe Trapping Engine version 305 and higher. This additional margin ensures that any support colorants in a rich black are sufficiently choked back. Also, the color abutting the black spreads by the additional amount, safely ensuring that extraordinary misregistration does not leave an unsightly gap.

5 Backward Compatibility

When the **ImageToImageTrapping** key is set to *true*, **ImageInternalTrapping** key trapping affects the placement of traps between two images. When **ImageInternalTrapping** is set to *true*, a trap between two image will be positioned as a “normal” trap that is based on the two relative neutral densities and the sliding trap limit. When **ImageInternalTrapping** is set to *true* and **ImageToImageTrapping** is set to *false*, the Adobe Trap Engine to will create traps between appropriate pixels within an image. This behavior is not backward compatible with previous versions of the engine.

A zero setting for **TrapWidth** or **BlackWidth** will cause an error in an older interpreter. This problem will occur when an updated plug-in emits a zero value that is sent to an older interpreter. However, this failure is the desired behavior as an older product will not know how to make proper use of values less than 0 for these two parameters.