

Q. If I fill a circle using PostScript, then stroke it with a different color using the default linewidth of 1, not all of the pixels get covered at the edges. I thought a linewidth of 0 could cause this problem, but a linewidth of 1 shouldn't! Drawing applications such as Draw.app demonstrate the same problem: Create a circle filled with black and outlined with white, using a line thickness of 1. You'll notice stray black pixels around the edges. What's going on here? How can this problem be avoided?

A. PostScript scanning should normally cause any non-zero linewidth to eliminate the stray pixels in the above scenarios. However, PostScript also has a "stroke adjustment" feature which, when turned on, tries to create lines of uniform thickness on low-resolution output devices (such as displays). This causes the above problems to appear. A line of width 1 drawn on a path might fail to fully cover the pixels on the edges of a fill done on the same path. By default, stroke adjustment is enabled for displays and disabled for printers.

Given this, developers of drawing and other applications should look into turning off stroke adjustment when displaying graphic objects created/manipulated by users in documents. This can be accomplished with

```
false setstrokeadjust
```

The previous `strokeadjust` value should be restored once the graphics are drawn; this can be done via `gsave/grestore` or by explicitly remembering the previous value:

```
currentstrokeadjust  
false setstrokeadjust  
... drawing code ...  
setstrokeadjust
```

You can read more about the `setstrokeadjust` and `currentstrokeadjust` operators in section 6.5 of the PostScript Language Reference Manual 2nd edition

(the level 2 red book).

QA871

Valid for 2.0, 3.0