



Tech Info Library

PostScript Language Tutorial and Cookbook: Correction To Manual

This article last reviewed: 18 May 1988

If you are trying to program in PostScript, you may find that THIS program runs perfectly well:

```
/Helvetica findfont
27 scalefont setfont
/rays
{0 1.5 179
{gsave rotate 0 0 moveto 108 0 lineto stroke grestore } for
} def
300 400 translate
.25 setlinewidth
newpath
0 0 moveto
(Pharmaceutical Research) true charpath
stroke
54 -15 translate
rays
showpage
```

Whereas this second program -- which is designed according to the Adobe PostScript programming manual -- does NOT. (The only difference: the word "stroke" was replaced with "clip".)

```
/Helvetica findfont
27 scalefont setfont
/rays
{0 1.5 179
{gsave rotate 0 0 moveto 108 0 lineto stroke grestore } for
} def
300 400 translate
.25 setlinewidth
newpath
0 0 moveto
(Pharmaceutical Research) true charpath
clip
54 -15 translate
rays
showpage
```

There are two reasons why this second program does not work:

1. The program shown is copied directly from page 103 of the PostScript Language Tutorial and Cookbook (ISBN #0-201-10179-3), except the name "StarLines" is replaced by "Pharmaceutical Research". The output should be the name "Pharmaceutical Research" in outlined letters with the ray pattern inside the letters.

Close examination of the program on page 103 shows that the user left out one critical line. The last five lines in the user's program read:

```
(Pharmaceutical Research) true charpath
clip
54 -15 translate
rays
showpage
```

These lines should read:

```
(Pharmaceutical Research) true charpath
clip
newpath      %-----new line added-----
54 -15 translate
rays
showpage
```

On page 128 of the PostScript Language Reference Manual (ISBN #0-201-10174-2) is a description of the `-clip-` operator, and the last paragraph states:

"Unlike `FILL` and `STROKE`, `CLIP` does not implicitly perform a `NEWPATH` after it has finished using the current path. Any subsequent path construction operators will append to the current path unless `NEWPATH` is executed explicitly. This can be a source of unexpected behavior."

2. On pages 260 and 261 of the PostScript Language Reference Manual, Appendix B (Implementation Limits), there is a list of limits that cannot be exceeded. The `PATH` limit (page 261) indicates the maximum number of points specified in all active path descriptions, including the current path, clip path, and paths saved by `SAVE` and `GSAVE`, cannot exceed 1500.

In this particular case, using the name "Pharmaceutical Research" exceeds the 1500-point limit; thus, the error "limitcheck Offending Command `-clip-`" when using the clip operator. If the name is reduced to "Pharmaceutical Rese", the program will execute properly. Adding even one letter to the name will generate the limitcheck error again.

There is a possible workaround for the output wanted. By doing a "gsave" and "grestore" to reset the clip region to its original size, it is possible to make the number of points on the clipping path small enough to avoid the error.

Here is a program that demonstrates one possible solution:

```
/Helvetica findfont
27 scalefont setfont
/rays
{0 1.5 179
{gsave rotate 0 0 moveto 108 0 lineto stroke grestore } for
} def
100 300 translate
.25 setlinewidth
gsave
newpath
0 0 moveto
(Pharmaceutical>true
charpath clip
newpath
100 -15 translate
rays
newpath
grestore
200 0 translate
0 0 moveto
(Research>true
charpath clip
newpath
60 -15 translate
rays
showpage
```

Copyright 1988 Apple Computer, Inc.

Keywords: <None>

=====

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 2954