

*Ghostscript, PostScript and PDF*

Peter Samuel

*Peter.Samuel@uniq.com.au*

Presented to the Sydney Linux Users Group



26 March 1999

## What are they?

*PostScript* — A device independent page description language developed by Adobe in the 1980s. *PostScript* is a true programming language with a look and feel very much like Forth.

*PDF* — A device independent portable document format developed by Adobe in the 1990s. *PDF* is not a programming language.

*Ghostscript* — An interpreter for the *PostScript* language and *PDF* files, with the ability to convert *PostScript* and *PDF* files to many raster formats, view them on displays, and print them on printers that don't have *PostScript* language capability built in. It is also a set of C procedures (the *Ghostscript* library) that implement the graphics capabilities that appear as primitive operations in the *PostScript* language.

## **Versions of *Ghostscript***

### **GNU *Ghostscript***

Stable versions of *Ghostscript* released under the GNU General Public License. These versions usually lag the current development versions of *Ghostscript* by one or more release levels. The GNU release is freely distributable in binary form. Current release is GNU *Ghostscript* 5.10.

### **Aladdin *Ghostscript***

Mostly stable versions of *Ghostscript* released under the Aladdin Free Public License. This version contains the most up to date non-beta release of *Ghostscript*. Source code is freely distributable, but binary versions are not. Current release is Aladdin *Ghostscript* 5.50. There are also a number of beta releases, the latest of which is Aladdin *Ghostscript* 5.73.

## Where do you get GNU *Ghostscript*?

GNU *Ghostscript* is available from the master GNU ftp server

`ftp://ftp.gnu.org/gnu/ghostscript/`

or from a number of Australian mirror sites:

`ftp://archie.au/gnu/ghostscript`

`ftp://ftp.progsoc.uts.edu.au/pub/gnu/ghostscript`

`ftp://mirror.aarnet.edu.au/pub/gnu/ghostscript`

GNU *Ghostscript* is also shipped with a number of Linux distributions, however the version shipped may not be the latest available GNU version. For example, RedHat ships GNU *Ghostscript* 4.03 with their 5.2 distribution.

## **Where do you get *Aladdin Ghostscript*?**

*Aladdin Ghostscript* is available from the master *Ghostscript* site

<http://www.cs.wisc.edu/~ghost/index.html>

This site also contains a wealth of other *PostScript* and *Ghostscript* related resources and links. It is well worth a visit.

## **How do you build Aladdin *Ghostscript*?**

Obtain the following files from the *Ghostscript* web site

ghostscript-5.50.tar.gz  
ghostscript-5.50gnu.tar.gz  
ghostscript-5.50jpeg.tar.gz  
ghostscript-5.50libpng.tar.gz  
ghostscript-5.50zlib.tar.gz  
ghostscript-fonts-other-5.50.tar.gz  
ghostscript-fonts-std-5.50.tar.gz

Also down load any relevant patches from

<http://www.cs.wisc.edu/~ghost/aladdin/relnotes/gs550/index.html>

## **How do you build Aladdin *Ghostscript*?**

Do not unpack the font archives at this stage.

After unpacking the other archives, you'll have the following directories

```
gs5.50  
jpeg-6b  
libpng-1.0.2  
zlib-1.1.3
```

Create symbolic links to the library directories

```
cd gs5.50  
ln -s ../jpeg-6b jpeg; ln -s ../libpng-1.0.2 libpng; ln -s ../zlib-1.1.3 zlib
```

## **How do you build Aladdin *Ghostscript*?**

Create a *makefile* from the appropriate *makefile* template

```
cp unix-gcc.mak makefile
chmod +w makefile
```

Edit the makefile and make changes relevant to your local setup. For RedHat Linux 5.2, I've used the following entries:

```
prefix = /pkgs/ghostscript-5.50
datadir = $(prefix)/lib
gsdir = $(datadir)
gsdatadir = $(gsdir)
GS_LIB_DEFAULT=$(gsdatadir):$(gsdir)/fonts:/pkgs/fonts
GENOPT=-DA4
```



## **How do you build *Aladdin Ghostscript*?**

```
XINCLUDE=-I/usr/X11R6/include  
XLIBDIRS=-L/usr/X11R6/lib  
XLIBDIR=/usr/X11R6/lib  
XLIBS=Xt SM ICE Xext X11
```

Decide which device drivers and features you wish to include. I like to include everything

```
FEATURE_DEVS=psl3.dev pdf.dev dpsnext.dev pipe.dev rasterop.dev  
DEVICE_DEVS=x11.dev x11alpha.dev x11cmyk.dev x11cmyk2.dev x11cmyk4.dev  
x11cmyk8.dev x11gray2.dev x11gray4.dev x11mono.dev lvga256.dev vgalib.dev
```

## How do you build Aladdin Ghostscript?

```
DEVICE_DEVS1=cif.dev inferno.dev mgr4.dev mgr8.dev mgrgray2.dev mgrgray4.dev  
mgrgray8.dev mgrmono.dev pkm.dev pkmraw.dev plan9bm.dev sgirgb.dev  
DEVICE_DEVS2=ap3250.dev appledmp.dev ccr.dev cp50.dev declj250.dev eps9high.dev  
eps9mid.dev epson.dev epsonc.dev ibmpro.dev imagen.dev iwhi.dev iwlo.dev  
iwlq.dev la50.dev la70.dev la75.dev la75plus.dev lj250.dev ln03.dev  
lq850.dev m8510.dev necp6.dev oce9050.dev oki182.dev okiibm.dev  
DEVICE_DEVS3=cdjmono.dev deskjet.dev djet500.dev laserjet.dev lj4dith.dev lj5gray.dev  
lj5mono.dev ljet2p.dev ljet3.dev ljet3d.dev ljet4.dev ljetplus.dev  
DEVICE_DEVS4=cdeskjet.dev cdj500.dev cdj550.dev cdjcolor.dev cljet5.dev djet500c.dev  
dnj650c.dev paintjet.dev pj.dev pjetxl.dev pjxl.dev pjxl300.dev  
DEVICE_DEVS5=lp2563.dev  
DEVICE_DEVS6=bj10e.dev bj200.dev bjc600.dev bjc800.dev hl7x0.dev jetp3852.dev  
lbp8.dev lips3.dev lp8000.dev r4081.dev sj48.dev st800.dev stcolor.dev  
t4693d2.dev t4693d4.dev t4693d8.dev tek4696.dev uniprint.dev xes.dev
```

## **How do you build Aladdin *Ghostscript*?**

DEVICE\_DEVS7=dfaxhigh.dev dfaxlow.dev faxg3.dev faxg32d.dev faxg4.dev  
DEVICE\_DEVS8=pcx16.dev pcx24b.dev pcx256.dev pcxcmyk.dev pcxgray.dev pcxmono.dev  
DEVICE\_DEVS9=pbm.dev pbmraw.dev pgm.dev pgmraw.dev pgnm.dev pgnmraw.dev  
pnm.dev pnmraw.dev ppm.dev ppmraw.dev  
DEVICE\_DEVS10=tiffcrle.dev tiffg3.dev tiffg32d.dev tiffg4.dev tiffzw.dev tiffpack.dev  
DEVICE\_DEVS11=bmp16.dev bmp16m.dev bmp256.dev bmpmono.dev bmpamono.dev  
tiff12nc.dev tiff24nc.dev  
DEVICE\_DEVS12=bit.dev bitcmyk.dev bitrgb.dev psgray.dev psmono.dev psrgb.dev  
DEVICE\_DEVS13=png16.dev png16m.dev png256.dev pnggray.dev pngmono.dev  
DEVICE\_DEVS14=cgm24.dev cgm8.dev cgmmono.dev jpeg.dev jpeggray.dev miff24.dev  
DEVICE\_DEVS15=epswrite.dev pdfwrite.dev pswrite.dev pxlmono.dev pxlcolor.dev

## **How do you build Aladdin *Ghostscript*?**

Build and install the software. Depending on your particular hardware configuration, and the load on your machine, this could take from 5 minutes to several hours.

```
make install
```

Install the fonts

```
cd ..  
S='pwd'  
cd /pkgs/ghostscript-5.50/lib  
gunzip -c $S/ghostscript-fonts-std-5.50.tar.gz | tar xvf -  
cd fonts  
gunzip -c $S/ghostscript-fonts-other-5.50.tar.gz | tar xvf -
```

## Using *Ghostscript*

Use the -h option to see rudimentary usage options and to see what drivers are available:

Aladdin Ghostscript 5.50 (1998-9-16)

Copyright (C) 1998 Aladdin Enterprises, Menlo Park, CA. All rights reserved.

Usage: gs [switches] [file1.ps file2.ps ...]

Most frequently used switches: (you can use # in place of =)

-dNOPAUSE           no pause after page   | -q     'quiet', fewer messages  
-g<width>x<height>   page size in pixels   | -r<res> pixels/inch resolution  
-sDEVICE=<devname>   select device           | -dBATCH exit after last file  
-sOutputFile=<file>   select output file: - for stdout, |command for pipe,  
                          embed %d or %ld for page #

Input formats: PostScript PostScriptLevel1 PostScriptLevel2 PDF

## **Using *Ghostscript***

### Available devices:

x11 x11alpha x11cmyk x11cmyk2 x11cmyk4 x11cmyk8 x11gray2 x11gray4 x11mono  
cif inferno mgr4 mgr8 mgrgray2 mgrgray4 mgrgray8 mgrmono pkm pkmraw  
plan9bm sgirgb ap3250 appledmp ccr cp50 declj250 eps9high eps9mid epson  
epsonc ibmpro imagen iwhi iwlo iwlq la50 la70 la75 la75plus lj250 ln03  
lq850 m8510 necp6 oce9050 oki182 okiibm cdjmono deskjet djet500 laserjet  
lj4dith lj5gray lj5mono ljet2p ljet3 ljet3d ljet4 ljetplus cdeskjet  
cdj500 cdj550 cdjcolor cljet5 djet500c dnj650c paintjet pj pjetxl pjxl  
pjxl300 lp2563 bj10e bj200 bjc600 bjc800 hl7x0 jetp3852 lbp8 lips3 lp8000  
r4081 sj48 sparc st800 stcolor t4693d2 t4693d4 t4693d8 tek4696 uniprint  
xes dfaxhigh dfaxlow faxg3 faxg32d faxg4 pcx16 pcx24b pcx256 pcxcmyk  
pcxgray pcxmono pbm pbmraw pgm pgmraw pgnm pgnmraw pnm pnmraw ppm ppmraw  
tiffcrle tiffg3 tiffg32d tiffg4 tiffzw tiffpack bmp16 bmp16m bmp256  
bmpmono bmpamono tiff12nc tiff24nc bit bitcmyk bitrgb psgray psmono psrgb

## Using *Ghostscript*

png16 png16m png256 pnggray pngmono cgm24 cgm8 cgmmmono jpeg jpeggray  
miff24 epswrite bbox pdfwrite pswrite pxlmono pxlcolor nullpage

Search path:

```
. : /pkgs/ghostscript-5.50/lib : /pkgs/ghostscript-5.50/lib/fonts :  
/pkgs/fonts
```

For more information, see `/pkgs/ghostscript-5.50/lib/doc/Use.htm`.

Report bugs to `ghost@aladdin.com`, using the form in `Bug-form.htm`.

*Ghostscript's* default device is the first device listed in the *makefile*. On Unix like systems, this is usually the X11 device. To display the contents of a *PostScript* file on your X11 screen, use

```
gs /pkgs/ghostscript-5.50/lib/examples/golfer.ps
```

## Using *Ghostscript*



To specify a different output device driver, use the `-sDEVICE` option.

```
gs -sDEVICE=djet500 /pkgs/ghostscript-5.50/lib/examples/golfer.ps
```



## Using *Ghostscript*

To redirect the output to a file, use the `-sOutputFile` option.

```
gs -sDEVICE=djet500 -sOutputFile=/tmp/golfer.dj \  
  /pkgs/ghostscript-5.50/lib/examples/golfer.ps
```

To redirect the output to standard output, use the `-sOutputFile` option.

```
gs -sDEVICE=djet500 -sOutputFile=- \  
  /pkgs/ghostscript-5.50/lib/examples/golfer.ps
```

To redirect the output to a process, use the `-sOutputFile` option.

```
gs -sDEVICE=djet500 -sOutputFile='| lpr -Pdj' \  
  /pkgs/ghostscript-5.50/lib/examples/golfer.ps
```

## Using *Ghostscript*

To convert a *PostScript* image to a different image format, use one of the image device drivers.

```
gs -sDEVICE=jpeg -sOutputFile=golfer.jpg -dJPEGQ=95 \  
  /pkgs/ghostscript-5.50/lib/examples/golfer.ps
```

All of the above examples place you in an interactive *Ghostscript* session. To terminate the session you must enter quit at the GS> prompt. To run *Ghostscript* in batch mode, use the following options

```
gs -sDEVICE=jpeg -sOutputFile=golfer.jpg -dJPEGQ=95 \  
  -q -dNOPAUSE -dSAFER \  
  /pkgs/ghostscript-5.50/lib/examples/golfer.ps \  
  -c quit
```

## *PostScript* **Previewers**

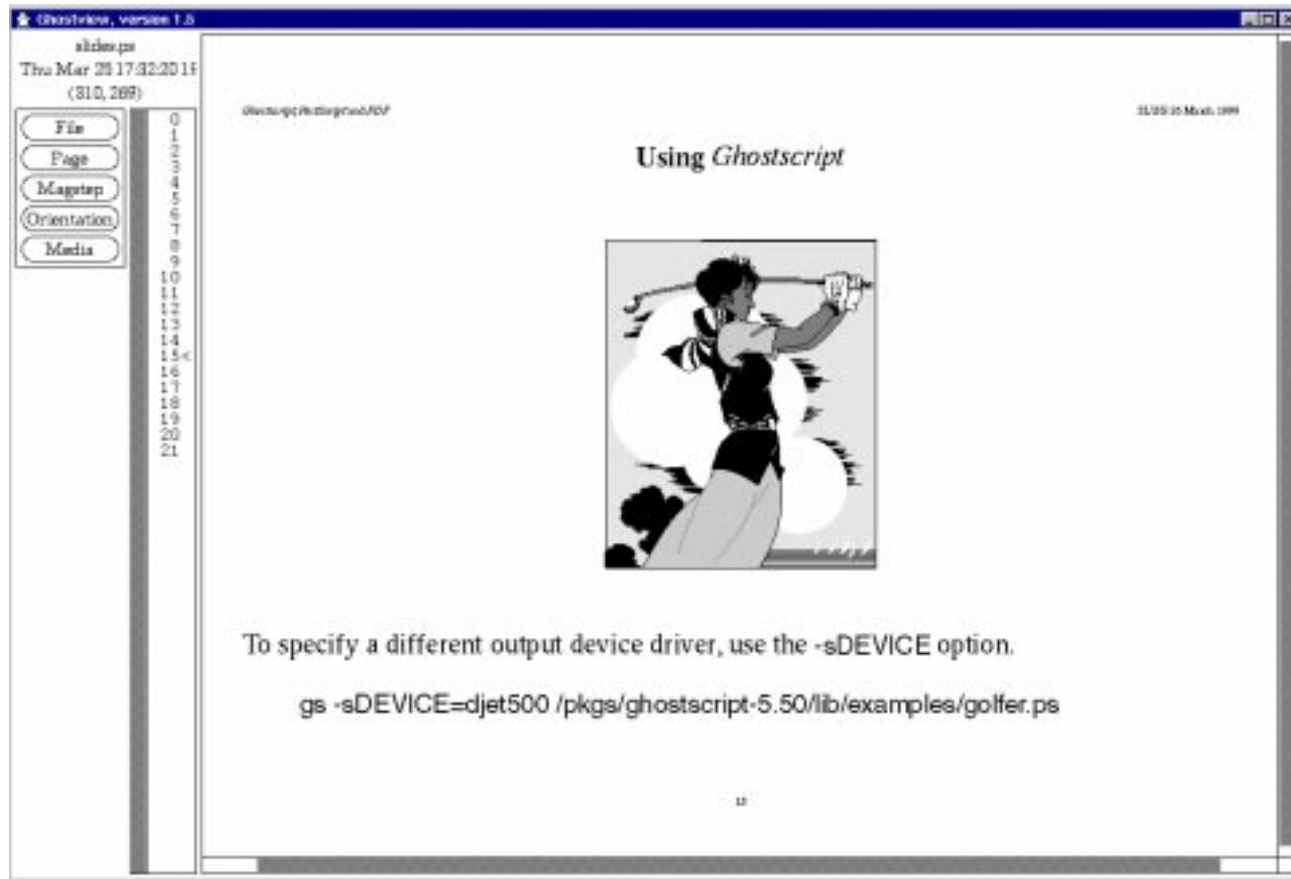
The interactive interface provided with *Ghostscript* is a little primitive. Better interfaces are available by using either *ghostview*, or its successor, *gv*.

Neither *ghostview*, nor *gv*, understand *PostScript*. They both require *Ghostscript* to write to their X11 windows to display the *PostScript* image(s).

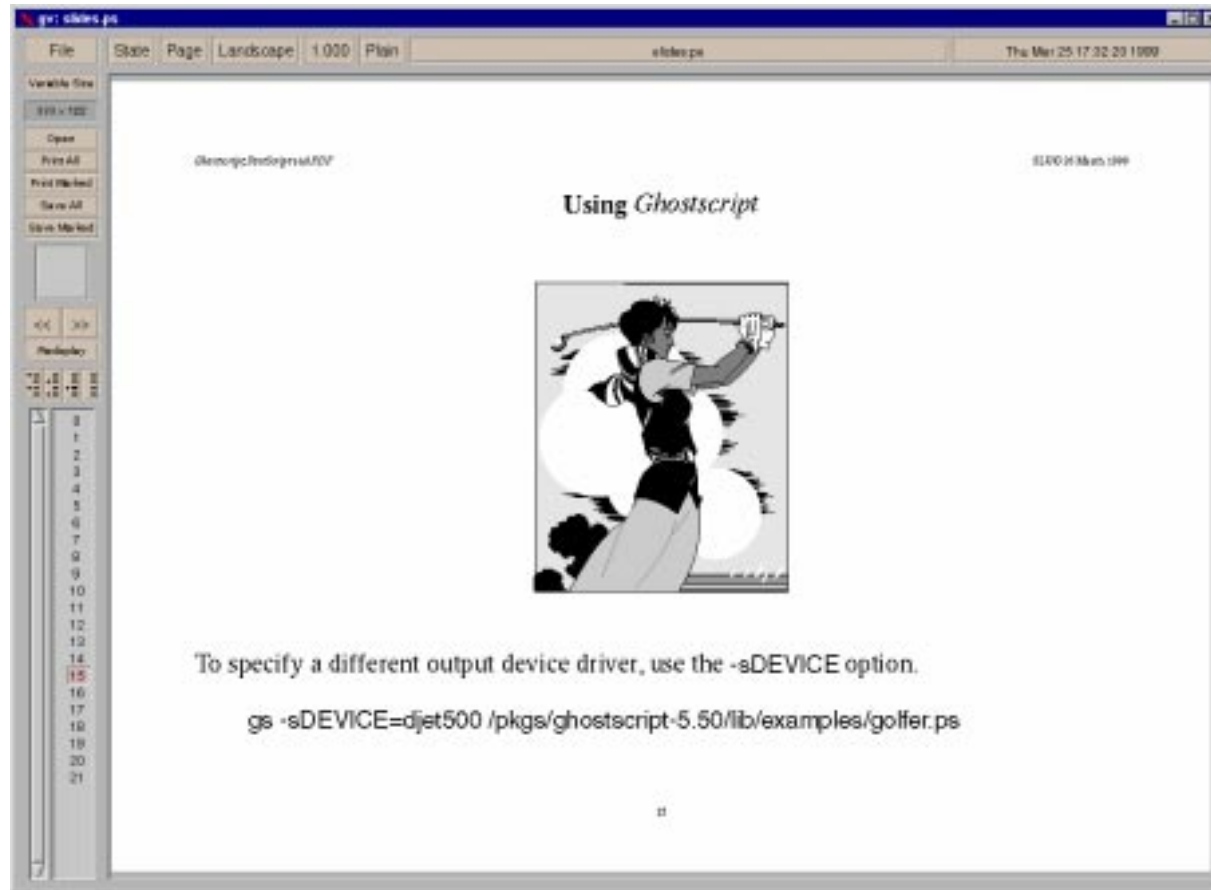
Both programs understand the Adobe Document Structuring Conventions (DSC). If the *PostScript* file contains DCS code, both *ghostview* and *gv* can be used to navigate through the document.

Because *Ghostscript* understands *PDF*, both *ghostview* and *gv* may be used to view *PDF* files. This is especially useful for operating systems for which Adobe has not provided an Acrobat *PDF* reader.

## Ghostview Screendump



## GV Screendump



## Using *Ghostscript* to debug *PostScript*

Using *Ghostscript* in interactive mode allows you to see what's happening on both the page (X11 window) and internally.

```
Aladdin Ghostscript 5.50 (1998-9-16)
```

```
Copyright (C) 1998 Aladdin Enterprises, Menlo Park, CA. All rights reserved.
```

```
This software comes with NO WARRANTY: see the file PUBLIC for details.
```

```
Loading NimbusRomNo9L-Regu font from
```

```
  /pkgs/ghostscript-5.50/lib/fonts/n021003l.pfb... 2324992 936291
```

```
  1329180 40089 0 done.
```

```
Loading NimbusRomNo9L-Medi font from
```

```
  /pkgs/ghostscript-5.50/lib/fonts/n021004l.pfb... 2365176 978092
```

```
  1329180 42336 0 done.
```

```
>>showpage, press <return> to continue<<
```

## Using *Ghostscript* to debug *PostScript*

*PostScript* errors are displayed in all their gory details.

Aladdin Ghostscript 5.50 (1998-9-16)

Copyright (C) 1998 Aladdin Enterprises, Menlo Park, CA. All rights reserved.

This software comes with NO WARRANTY: see the file PUBLIC for details.

Loading NimbusRomNo9L-Regu font from

```
/pkgs/ghostscript-5.50/lib/fonts/n021003l.pfb... 2324992 936291  
1329180 40053 0 done.
```

Loading NimbusRomNo9L-Medi font from

```
/pkgs/ghostscript-5.50/lib/fonts/n021004l.pfb... 2365176 978092  
1329180 42300 0 done.
```

Error: /undefined in tranlsate

Operand stack:

```
false --nostringval-- false --nostringval-- false  
--nostringval-- 0 16840
```

## Using Ghostscript to debug PostScript

Execution stack:

```
%interp_exit .runexec2 --nostringval-- --nostringval--  
--nostringval-- 2 %stopped_push --nostringval-- 2 3  
%oparray_pop --nostringval-- --nostringval-- false 1  
%stopped_push 1 3 %oparray_pop .runexec2 --nostringval--  
--nostringval-- --nostringval-- 2 %stopped_push  
--nostringval-- --nostringval-- --nostringval--
```

Dictionary stack:

```
--dict:909/941(G)-- --dict:0/20(G)-- --dict:88/200(L)--
```

Current allocation mode is local

Current file position is 9609

GS<1>

The incorrect spelling of `translate` caused the error. The error occurred 9609 bytes into the file, which is the first word *after* the typo. The contents of each stack is also shown.



## Using *PostScript* to debug *PostScript*

You can also use a *PostScript* printer to debug *PostScript* code. You'll need to download some error handling code to the printer first. Adobe provide an error handler at

<ftp://ftp.adobe.com/pub/adobe/framemaker/win/all/ehandler.zip>

After unpacking the archive, simply print `EHANDLER.PS` to your printer. If there are any *PostScript* errors in subsequent files sent to the printer, the printer will print a description of these errors in a similar manner to the *Ghostscript* output shown above.

You can also use a *PostScript* printer in interactive mode to debug small code fragments. After connecting to the printer, using a serial port or telnet or whatever is appropriate for the printer, type `executive` and you'll be placed into an interactive session with the printer. Type `quit` at the prompt to terminate the session.

## Using *PostScript* to debug *PostScript*

```
telnet oak 9100
Trying 149.135.120.97...
Connected to oak.ind.tansu.com.au.
Escape character is '^]'.
executive
Copyright (c) Hewlett-Packard Company, 1997
Version 2014.108
PS>serverdict {} forall pstack
{}
/initprinter
...
-filestream-
/stdin
PS>quit
```

## Fonts

*Ghostscript* version 4.x and above, ships with good quality fonts. *Ghostscript* 5.50 ships with 84 different fonts, including good imitations of the 35 "standard" Adobe fonts.

*Ghostscript* 5.x will also understand TrueType fonts.

Adding fonts is easy. Just install the file into one of *Ghostscript*'s search path directories and create a Fontmap entry for that font.

```
cp Godzilla.ttf /pkgs/fonts
echo "/TT_Godzilla- (Godzilla.ttf) ;" >> /pkgs/fonts/Fontmap
```

**OH NO! THERE GOES TOKYO! GO GO GODZILLA!**

## Other Tools

There are a number of very useful *PostScript* utilities available:

- **a2ps** – an ASCII to *PostScript* converter. Does 1 or 2 up pages as well as pretty printing — <http://www-inf.enst.fr/~demaille/a2ps/>
- **mpage** – another ASCII to *PostScript* converter. Does 1 to 8 up pages, will also do  $n$  up printing of *PostScript* input — <ftp://tsx-11.mit.edu/pub/linux/sources/usr.bin/>
- **psutils** – a suite of applications that do all sorts of manipulations on *PostScript* input, including page re-ordering and  $n$  up printing — <http://www.dcs.ed.ac.uk/home/ajcd/psutils/>
- **pstoedit** – converts *PostScript* to many different formats including xfig, pic, MIF, gnuplot, idraw and *PDF* — <http://sunsite.doc.ic.ac.uk/public/packages/X11/contrib/applications/pstoedit/>

## Using *Ghostscript* as a print filter

Setup an appropriate `/etc/printcap` entry

```
lp:\
    :sd=/var/spool/lpd/lp:\
    :mx#0:\
    :sh:\
    :rm=oak:\
    :rp=lp:\
    :if=/var/spool/lpd/lp/filter:
```

The filter `/var/spool/lpd/lp/filter` decides how to convert the input into the appropriate output. Once it has converted the input into *PostScript* it calls *Ghostscript*

```
.... | gs -q -sDEVICE=bj10e -dNOPAUSE -dSAFER -sOutputFile=- -
```

## Using *Ghostscript* as a print filter

RedHat Linux 5*x* ships with *rhs-printfilters* which use a number of file to *PostScript* conversion utilities in combination with *Ghostscript* to support a large number of printers. They also ship the *PrintTool* graphical front end as part of their *ControlPanel* package. For most setups, these packages make the task of setting up local and remote printers relatively painless.

System V Unix systems are a little more arcane to configure, but the principle is the same. You end up with an interface program that converts its input into *PostScript* and then calls *Ghostscript* to convert the *PostScript* to the relevant printer format.