A Full Page Grid

This gives you a full portrait page gray 10X layout grid. Grids like this one are the starting point for most projects.

The Gonzo Grid

gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justify program

20 15 10 setgrid % create a full page ten point grid
57 76 showgrid % and show the grid

% the rest of your design goes in here. it will stay locked to
% the grid.

showpage % actually prints the page
quit % clears the printer for the next printout

Drawing a Rectangular Box

gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justify program

20 15 10 setgrid % create a full page ten point grid
57 76 showgrid % and show the grid
0.2 setlinewidth % determines how wide a line is used
5 25 moveto      % defines the starting point
10 20 lineto % defines the first side
35 20 lineto % defines the second side
35 10 lineto % defines the third side
10 10 lineto % defines the fourth side
stroke % draws the box

/Helvetica-Bold findfont 6 scalefont setfont
28 13 (MY NAME) cc % places some text in the box
% center

showpage % actually prints the page
quit % clears the printer for the
% next printout

PostScript Operators

begin red book pg 370
def red book pg 397
lineto red book pg 449
makefont red book pg 452
moveto red book pg 456
quit red book pg 468
rlineo red book pg 482
scalefont red book pg 488
setfont red book pg 503
setlinewidth red book pg 508
showpage red book pg 520
stroke red book pg 528

Gonzo Commands

cl callout left
callout center
cr callout right
MY NAME

MY NAME
On Your Own

Gather magazines, newspapers, and/or flyers for examples of type faces, layout and graphics. Pick ones you would not usually read.

Begin to collect illustrations (line art or drawings) of your choice for curvetracing.

### EXACT FONT SPELLINGS

#### Sans-Serif
- AvantGarde-Book
- AvantGarde-BookOblique
- AvantGarde-Demi
- AvantGarde-DemiOblique
- Helvetica
- Helvetica-Oblique
- Helvetica-Bold
- Helvetica-BoldOblique
- Helvetica-Narrow
- Helvetica-Narrow-Oblique
- Helvetica-Narrow-Bold
- Helvetica-Narrow-BoldOblique

#### Serif
- Bookman-Light
- Bookman-LightItalic
- Bookman-Demi
- Bookman-DemiItalic
- NewCenturySchlbk-Roman
- NewCenturySchlbk-Italic
- NewCenturySchlbk-Bold
- NewCenturySchlbk-BoldItalic
- Palatino-Roman
- Palatino-Italic
- Palatino-Bold
- Palatino-BoldItalic

#### Special Fonts
- Courier
- Courier-Oblique
- Courier-Bold
- Courier-BoldOblique

- ZapfChancery-MediumItalic
- ZapfDingbats

### FONT PUBLISHER RESOURCES

**Adobe Systems**
1585 Charleston Road
Mountain View, CA 94039
(800) 833-6687

**Agfa Corporation**
90 Industrial Way
Wilmington, MA 01887
(800) 424-8973

**Bitstream Inc**
Athenaeum House
215 1st Street
Cambridge, MA 02142
(800) 237-3335

**Fluent Fonts**
Casady & Green, Inc
PO Box 222770
Carmel, CA 93922

**The Font Company**
12629 N Tatum Blvd, #210
Phoenix, AZ 85032
(602) 998-9711

**FontHaus Inc**
15 Perry Avenue, #A8
Norwalk, CT 06850
(800) 942-9110

**Image Club**
729 24th Avenue SE
Calgary, AB
CANADA T2G5K8
(800) 661-9410

**Jonathan Macagba**
PO Box 14013
Philadelphia, PA 19122
(215) 634-0634

**Monotype**
53 W Jackson Blvd, #504
Chicago, IL 60604
(800) MONOTYPE

**Phil’s Fonts**
2380 Chaplain Street NS
Washington, DC 20009
(800) 424-2977

**Precision Type**
595 Old Willets Path
Hauppauge, NY 11788
(800) 248-FONT

**U-Design Type Foundry**
270 Farmington Avenue
Hartford, CT 06105
(203) 278-3648

**Font & Function**
Adobe Systems
1585 Charleston Road
Mountain View, CA 94039
(800) 833-6687

**TypeWorld**
One Technology Park Drive
Westford, MA 01886
(508) 392-2157

**U&lc**
2 Dag Hammarskjold Plaza
New York, NY 10017
(212) 371-0699
A Personal or Business Letterhead

Simple and clean business or personal stationary letterhead. The first is left justified with a simple bar, the second has a centered name and a fancier horizontal bar.

**The Left Justified Letterhead**

```plaintext
gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justification
/showthegrid true def % Do we want to see the layout grid?
  % true = yes; false = no
20 15 10 setgrid % create a full page ten point grid
/showthegrid
{57 76 showgrid} if % and maybe show the grid
1 setlinecap % make lines have rounded ends
0.3 setlinewidth % make all lines 3 points thick
1 69.5 mt 55 r % draw a single horizontal line
/cstretch 0.02 def % stretch spaces between characters
/ssstretch 0.1 def % stretch spaces between words
/font1 /Bookman-LightItalic 0.9 gonzofont % define a normal font
/font2 /Bookman-DemiItalic 0.9 gonzofont % define a bold font
/yinc 1 def % set the vertical line spacing
1 74 % create a left justified address
{[2Yodar Kritch]}
123 Main Street
Thatcher, AZ 85552
l
(602) 428-4073) cl
/showpage % and show the page
quit % finished!
```

**The Center Justified Letterhead**

```plaintext
gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justification
/showthegrid false def % Do we want to see the layout grid?
  % true = yes; false = no
20 15 10 setgrid % create a full page ten point grid
/showthegrid
{57 76 showgrid} if % and maybe show the grid
bestgray % pick a fine secret gray
1 setlinecap % make all lines have rounded ends
1 69 mt 55 pr % draw a horizontal line
[0.45 0.3 0.9] % [linewidth gray linewidth gray]
superstroke % actually stroke it
/cstretch 0.02 def % stretch spaces between characters
/ssstretch 0.1 def % stretch spaces between words slightly
/font1 /Bookman-LightItalic 0.9 gonzofont % define a normal font
/font2 /Bookman-DemiItalic 0.9 gonzofont % define a bold font
/yinc 1 def % set the vertical line spacing
28.5 74 % create a center justified address
{[2Yoderella Kritchette]}
123 Main Street
Thatcher, AZ 85552
l
(602) 428-4073) cc
/showpage % and show the page
quit % finished!
```

**ASSIGNMENT 2:**

**Design a personal or business letterhead**

- Left Justified Letterhead
- Center Justified Letterhead

Simple and clean business or personal stationary letterhead. The first is left justified with a simple bar, the second has a centered name and a fancier horizontal bar.

**The Left Justified Letterhead**

```plaintext
...%
```

**The Center Justified Letterhead**

```plaintext
...%
```

**ASSIGNMENT 2:**

**Design a personal or business letterhead**

- Left Justified Letterhead
- Center Justified Letterhead

Simple and clean business or personal stationary letterhead. The first is left justified with a simple bar, the second has a centered name and a fancier horizontal bar.

**The Left Justified Letterhead**

```plaintext
...%
```

**The Center Justified Letterhead**

```plaintext
...%
```
PostScript Notes

[ array ] generally numeric
( string ) whatever you want printed
{ procedure } user defined
% comment line

boolean – if/else – defined by true or false

PostScript Operators

setlinecap red book pg 506

Font Array Notations

/font0 /Helvetica [1.1 0 0 1.1 0 0] gonzofont

The first number in the font array is the width (x).
The second number is the rotation.
The third number is the slant.
The fourth number is the height (y).
The fifth number moves to the right or left (xshift).
The sixth number is the superscript or subscript (yshift).

Gonzo Commands

h does a half negative linefeed
mt moveto
rl rlineto
l lineto left
r lineto right
u lineto up
dl lineto down

pr path right
pl path left
pu path up
pd path down

bestgray best gray angle and arrangement
copies used for multiple copies
font1 thru font0 defines the gonzofonts
gonzofont calls the predefined fonts
cstretch stretches spaces between characters
ststretch stretches spaces between words
showthegrid turns on or off the layout grid
superstroke strokes complex black, gray, white
lines or boxes
yinc defines the vertical line spacing

On Your Own


Find examples of Business/Personal cards you would like to modify for your use.
Step-and-repeat Business Cards

This uses step-and-repeat to print twelve business cards on a single sheet of cover, parchment, or index stock.

**The 12-up Business Card**

```
gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justification

% Do we want to see the layout grid?
/showthegrid true def % true = yes; false = no

/repeatproc {
  0 0 10 setgrid              % get on the usual
  % maybe show the CARD-SIZED grid
  showthegrid {25 14 showgrid} if
  /yinc 1 def                 % set the vertical line spacing
  /cstretch 0.03 def          % stretch spaces between characters
  /sstretch 0.1 def           % stretch spaces between words
  /font1 /Bookman-LightItalic 0.9 gonzofont
  /font2 /Bookman-DemiItalic 0.9 gonzofont
  1 setlinecap               % round path ends
  bestgray                   % nice looking gray
  1 5 mt 23 pr               % draw a line to the right
  [0.45 0 0.3 0.9]           % [linewidth gray linewidth gray]
  superstroke                % and show it two-tone

  1 8 ([2Yodar Krich]1
    123 Main Street
    Thatcher, AZ 85552) cl

  1 3 ([1(602) 428-4073] cl) def

  (buscard) stepandrepeatedef % no showpage with stepandrepeatef
  quit

CardsNow Business Card

```
gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justification

% cards now form
stepnrptparams begin
/cardsnow [2 5 252 143.5 50 34 20 false false] def
end

% Do we want to see the layout grid?
/showthegrid true def % true = yes; false = no

/repeatproc {
  0 0 10 setgrid % get on the usual

  % maybe show the CARD-SIZED grid
  showthegrid {25 14 showgrid} if
  /yinc 1 def % set the vertical line spacing
  /cstretch 0.03 def % stretch spaces between characters
  /sstretch 0.1 def % stretch spaces between words
  /font1 /Bookman-LightItalic 0.9 gonzofont
  /font2 /Bookman-DemiItalic 0.9 gonzofont
  % define a bold font

  1 setlinecap % round path ends
  bestgray % nice looking gray
  1 5 mt 23 pr % draw a line to the right
  [0.45 0 0.3 0.9] % [linewidth gray linewidth gray]
  superstroke % and show it two-tone

  1 8 ([2Yodar Krich]1
    123 Main Street
    Thatcher, AZ 85552) cl

  1 3 ([1(602) 428-4073] cl) def

  (cardsnow) stepandrepeatedef
  quit

Rotary Card

This shows how to define your own step and repeat project.

```
gutility begin % turn on the gonzo utilities
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justification

ASSIGNMENT 3:
Design a personal or business card using step-and-repeat
Gonzo 17 STEPENDREPEAT

% Horizrpts - times proc repeats in horizontal direction
% Vertrpts - times proc repeats in vertical direction
% Hspacing - horizontal proc repeat spacing
% Vspacing - vertical proc repeat spacing
% Hstart - horizontal offset of first proc
% Vstart - vertical offset of first proc
% Ticklen - length of cropping ticks when and if used
% Use ticks? - show the ticks true/false boolean
% Landscape - landscape orientation true/false boolean

/stepnrptparams 40 dict def
stepnrptparams begin

/admitonetick [5 9 150 60 25 25 10 true true] def % 45 tickets
/babybumper [2 10 270 72 40 30 20 true false] def % 20 stickers
/badgeaminit [2 4 306 198 0 0 50 true false] def % 6 badges
/bigbumpstick [1 3 792 205 0 0 50 false false] def % 3 bumperstickers
/buscard [3 4 256 143 12 20 20 true true] def % 12 buscards
/busenvelope [1 1 685 305 0 150 25 false true] def % 1 envelope
/eightlabel [2 4 306 198 0 0 50 true false] def % 6 badges
/bigbumpstick [1 5 610 150 0 0 60 true false] def % 3 bumperstickers
/buscard [3 4 256 143 12 20 20 true true] def % 12 buscards
/fulllandpage [1 1 0 0 20 20 50 false true] def % entire landscape page
/fullportpage [1 1 0 0 20 20 50 false false] def % entire portrait page
/lilbumpstick [1 13 424 60.3 80 34 0 false false] def % 13 VHS splines
/3.5disklabel [2 3 216 226 100 60 false false] def % 6 disk labels 3.5
/5.25disklabel [1 7 316 110 275 35 false false] def % 7 disk labels 5.25
end

% /setrepeatparams is a working tool that extracts the selected repeat values from an entry in stepnprtparams.
/setrepeatparams {cvn stepnrptparams exch get aload pop /landscape1 exch def /ticktrue exch def /ticklen exch def
/vertstart exch def /horstart exch def /incvert exch def /inchoriz exch def /numvert exch def /numhoriz exch def} def

% /onetick optionally draws a single tick or crop mark, while /drawticks puts one at each corner of each form
/onetick { 0 ticklen 2 div rmoveto 0 ticklen neg rlineto ticklen 2 div neg dup neg rmoveto 0 ticklen 0 rlineto 0 setlinewidth
stroke} def
/drawticks {gsave ticktrue {0 0 moveto onetick inchoriz 0 moveto onetick 0 incvert moveto onetick inchoriz incvert moveto
onelapel} if grestore} def

% /srfile gets used for custom entries. Each entry can be a string or a proc, but ONLY ONE ENTRY is allowed per repeat.
/srfile [(You) (forgot) (to) (define) (srfile!)] def % default srfile

% /calcpages is an optional routine that decides how many pages are needed ONLY when you are using a custom srfile.
% This allows early exits when or if you run out of data.
/calcpages {dup cvn stepnprtparams exch get dup 0 get dup 1 get mul cvn srfile length exch div ceiling cvi /numpages
exch def} def

% /srexitproc () def % default short exit - don't do it.
/srexitproc () def % default short exit - don't do it.
/numpages 1 def % default number of sheets to print
/numpages 1 def % default number of sheets to print
/numstartnum def % initial ticket number
/numstartnum def % running pointer advances one per repeat
/repeatproc () def % artwork to get repeated default

% default variables used by stepnprt
/repeatproc () def % artwork to get repeated default

% /srfile gets used for custom entries. Each entry can be a string or a proc, but ONLY ONE ENTRY is allowed per repeat.
/srfile [(You) (forgot) (to) (define) (srfile!)] def % default srfile

% /calcpages is an optional routine that decides how many pages are needed ONLY when you are using a custom srfile.
% This allows early exits when or if you run out of data.
/calcpages {dup cvn stepnprtparams exch get dup 0 get dup 1 get mul cvn srfile length exch div ceiling cvi /numpages
exch def} def

% This is the main steandrepeat tool...

/steandrepeat () def % save /srntrap exch def mark exch /quicexit false def customdata /calcpages if setrepeatparams
numpages (gsave landscape1 {-90 rotate -792 0 translate} if horstart vertstart translate gsave numhoriz (gsave numvert
(drawtools save /rptsave1 exch def repeatproc rptsave1 restore /runnum runnum 1 add def 0 incvert translate srexitproc}
repeat quickexit {exit} if /getreore onelapel 0 translate) repeat quickexit {exit} if /getreore showpage /getreore restore repeat quickexit
{showpage} if /cleartomark srexitproc restore} def

% To force an early exit when you run out of names or reach a given ticket number, test suitably. Then conditionally make
% /quicexit true and exit repeatproc.
% Note that an ending showpage is NOT required and should NOT be used.
% define a rotary card form
stepnrptparams begin
/rotary [2 4 288 157 16 78 20 false false] def end

% Do we want to see the layout grid?
/showthegrid true def % true = yes; false = no
/repeatproc {
  0 0 10 setgrid % get on the usual
  % maybe show the CARD-SIZED grid
  showthegrid [28 15 showgrid] if
  /yinc 1 def % set the vertical line spacing
  /cstretch 0.03 def % stretch spaces between characters
  /sstretch 0.1 def % stretch spaces between words

  /font1 /Bookman-LightItalic 0.9 gonzofont % define a normal font
  /font2 /Bookman-DemiItalic 0.9 gonzofont % define a bold font

  1 setlinecap % round path ends
  bestgray % nice looking gray
  1.5 5 mt 25 pr % draw a line to the right
  [0.45 0 0.3 0.9] % [linewidth gray linewidth gray]
  superstroke % and show it two-tone

  1.5 8 ([2Yodar Kritch]1
  123 Main Street
  Thatcher, AZ 85552) cl

  1.5 3 ([1(602) 428-4073] cl) def

  (rotary) stepandrepeat
  quit

PostScript Operators
repeat red book pg 476

Gonzo Commands
manual manual feeds stock from top of tray
repeatproc actual material to be repeated
stepnrptparams step and repeat dictionary
stepandrepeat prints a step and repeat procedure
(replaces showpage)

On Your Own
Read: PS Reference Manual – Chapter 8: Operators, section 8.1
PS Tutorial – Chapter 4: Procedures & Variables
Two Personal Notes

C reates half size 2-up personalized stationary or note paper using the stepandrepeat procs.

**The Actual Program**

```postscript

gutility begin
nuisance begin
printerror
gonzo begin
/showthegrid true def

% define a new step and repeat proc for a notepad
stepnрrptпrпms begin
/twnote [2 1 395 0 20 20 10 false true] def end

% define ONE notepad the way you want it
/repeatproc {gsave 0 0 10 setgrid
showthegrid {35 57 showgrid} if
/cstretch 0.02 def
/ssstretch 0.1 def
/font0 /ZapfChancery-MediumItalic 7 gonzofont
/font1 /ZapfDingbats [40 0 0 35 0 0] gonzofont

0 0 35.5 57 1.75 roundbox
% set up a round box
[0.6 0 0.45 0.99] superstroke
% make it black with a gray stripe

gsave % a gray pencil, etc...
bestgray 0.8 setgray 1 setlinecap
line2 7.6 50.1 mt 17 r
3.8 55.4 translate -70 rotate
0 0 (|{/} cl grestore

3.53 (|2a note from:) cl % the blurb
10 50.5 (|1Yodar Kritch) cl
17.7 1.5 (|3(602) 428-4073) cc
grestore grestore} def

% call the proc, do it twice and print the page
1 copies % the number of copies
% no showpage needed
(twnote) stepandrepeat
quit

The Valentine

This gives two copies per page without using the step- andrepeat.

gutility begin
nuisance begin
printerror
gonzo begin

landscape % we want to print sideways
/showthegrid true def

% define a proc called valentine
/valueпeпп /gsave 20 15 10 setgrid
showthegrid {36 28 showgrid} if
/cstretch 0.02 def
/ssstretch 0.1 def
/font0 /ZapfChancery-MediumItalic 7 gonzofont
/font1 /ZapfDingbats [40 0 0 35 0 0] gonzofont

bestgray lightgray % a fine light gray
19 3 (|1\252) cc % and draw a big heart via dingbats
black % back to black
/yinc 7 def % seven block spacing

19 17 (|0Thinking of You) cc % the message
grestore grestore } def
% first grestore gets you off the grid

valentine % print the left valentine
gsave 396 0 translate
valentine % print the right valentine
grestore
```

** ASSIGNMENT 4:**

**Design a personal or business notepaper and a mug**
Thinking of You

Thinking of You
Incredibly Schmaltzy Glop Goes Here

A Desk-Top Valentine for You
Be My Tootsie-Wootsie

To Karren with ❤
from Kyle

Valentines Day 1991
showpage quit

**Inside of the Valentine**

gutility begin
nuisance begin
printerror
gonzo begin

landscape % we want to print sideways
/showthegrid true def
/invalentine {gsave
20 15 10 setgrid
showthegrid {36 28 showgrid} if
/cstretch 0.02 def
/sstretch 0.1 def

/font0 /ZapfChancery-MediumItalic 2 gonzofont
/font1 /ZapfDingbats .5 gonzofont

bestgray lightgray % a fine light gray
18 1.3 % smaller gray hearts
(|\252 |\252 |\252 |\252 |\252) cc
black % back to black
/yinc 2 def
18 19 % Your personal message
(|0Incredibly % inside message
Schmaltzy Glop
Goes Here) cc
18 3.5 (|0A Desk-Top
Valentine for You) cc
grestore grestore } def

/invalentine % print the left valentine
gsave 396 0 translate
invalentine % print the right valentine
grestore

showpage quit

**A Standard Valentine Mug**

gutility begin
nuisance begin
printerror
gonzo begin

landscape
/showthegrid true def
0 0 10 setgrid
/showthegrid {73 26 showgrid} if
/cstretch 0.02 def
/sstretch 0.1 def

/font1 /ZapfDingbats 4 0 0 4 0 -0.25 gonzofont
/font2 /Palatino-BoldItalic 4 0 0 4 0 0 gonzofont
/font3 /Palatino-BoldItalic 6 0 0 6 0 0 gonzofont
/font4 /Palatino-Italic 2 0 0 2 0 0 gonzofont

/yinc 7 def
21 17.5 (|3Be My
Tootsie-Wootsie) cc
/yinc 5 def
54 19 (|2To Karren
with |1\244|2
from Kyle
|4Valentines Day 1991) cc

showpage quit

**A Tall Mug**

gutility begin
nuisance begin
printerror
gonzo begin

landscape
/showthegrid true def
0 0 10 setgrid
/showthegrid {70 45 showgrid} if
/cstretch 0.02 def
/sstretch 0.1 def

%curvetrace the cap and scroll
/capscroll {save /snapc exch def
0.1 setlinewidth
%...........diploma
[32 9.8 10 37 11 10 43.5 13 10 43.5 13 -55 44 10.5
-120 44 10.5 -160
33 7 -150 33 7 80 33.5 110 32.9 145 31.5 9 -90
32.5 7.5 0
32.6 8.5 125 32.1 9 165 31.9 8.5 -90 32.4 8 0 32.1 8.7
135 32.1 8.7 10
32.3 8.8 10] curvetrace stroke
32.6 7.5 moveto 0.5 0.2 rlineto stroke
%..........ribbon
Luisa Vasquez

Congratulations!

Interior Design

NAU

1992
PostScript Operators

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>grestore</td>
<td>429</td>
</tr>
<tr>
<td>gsave</td>
<td>430</td>
</tr>
<tr>
<td>rotate</td>
<td>484</td>
</tr>
<tr>
<td>setgray</td>
<td>504</td>
</tr>
<tr>
<td>translate</td>
<td>533</td>
</tr>
</tbody>
</table>

Gonzo Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>returns printing to black from gray</td>
</tr>
<tr>
<td>landscape</td>
<td>sets the landscape mode (horizontal)</td>
</tr>
<tr>
<td>lightgray</td>
<td>lightest gray available</td>
</tr>
<tr>
<td>roundbox</td>
<td>defines a complex black, gray, white box</td>
</tr>
</tbody>
</table>

On Your Own

Read Blue Book Chapter 4: Procedures & Variables review
Red Book pg 32.
Find forms using horizontal and vertical columns.

COLOR RESOURCES

<table>
<thead>
<tr>
<th>Type</th>
<th>Contact Information</th>
</tr>
</thead>
</table>
| "EMBOSSING" FOILS | Kroy Color
14555 North Hayden Road
Scottsdale, AZ 85260
(602) 951-1593 |
| | LaserColor
Minds In Motion, Inc
14260 Garden Road #30B
Poway, CA 92064
(619) 748-5003 |
| | Transfer Print Foils
9 Cotters Lane
East Brunswick, NJ 08816
(201) 238-1800 |
| | Lewis Weiss
Box 370
Livingston Manor, NY 12758 |
| LAMINATING | Catalina Plastics
23901 Calabasas Road
Calabasas, CA 91302
(800) 333-3136 |
| | Kroy Color
14555 North Hayden Road
Scottsdale, AZ 85260
(602) 951-1593 |
| | Seal Products Inc
550 Spring St
Naugatuck, CT 06770
(203) 729-5201 |
| | USI
PO Box 644
33 Business Park Drive
Branford, CT 06405
(800) 282-9890 |
MY GROCERY LIST

<table>
<thead>
<tr>
<th>QUAN.</th>
<th>ITEM</th>
<th>STORE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Forms: A Grocery List & Quadpad

Uses graphics drawing and repeating line commands (xrpt, yrpt) to create a simple form.

The Grocery List

gutility begin
nuisance begin
printerror
gonzo begin
/showthegrid true def
20 15 10 setgrid
/showthegrid {57 76 showgrid} if
/cstretch 0.02 def
/sstretch 0.1 def
/font1 /NewCenturySchlbk-Bold 2.1 gonzofont
% define title letters
/font2 /NewCenturySchlbk-Bold 1.1 gonzofont
% define smaller letters
% set lower left corner of list
gsave 15 20 translate
% print a heavy box
line3 1 setlinecap
0 0 mt 34 u 30 r 34 d 30 l
% print group of horizontal lines
line1
[[0 0 mt 30 r] 2 17] yrpt
line2
6 0 mt 34 u
20 0 mt 34 u % print vertical dividers
line3
0 32 mt 30 r % and a title divider
/yinc 1 def % set the vertical line spacing

% show the title
15 36.5

(1MY GROCERY LIST) cc
% and subtitles
3 32.5 (2QUAN.) cc
13 32.5 (2ITEM) cc
25 32.5 (2STORE) cc
2 copies % make two copies
showpage % and print the page
quit

Quad Form Pads

% Creates 4-up back-to-back pads arranged so you can
% apply padding compound to the outside edges.

gutility begin
nuisance begin
printerror
gonzo begin
landscape % landscape orientation
/showthegrid true def
% Define a routine called
% quadpad that draws ONE
% pad the way you want it
/quadpad {
save /snap1 exch def
0 0 10 setgrid
/showthegrid {37 28 showgrid} if
/cstretch 0.02 def
/ssstretch 0.1 def
/font1 /AvantGarde-Demi 1.5 gonzofont
/yinc 2 def
font1
18.5 19 (Your stupendously great
padded form goes here) cc
6 15 25 7 1 roundbox line3 stroke
clear snap1 restore } def
% And these are the magic numbers to make two pads
% rightside up and two pads upside down...
gsave 16 316 translate quadpad grestore
gsave 406 316 translate quadpad grestore
% note that 11 x 72 = 792 and 8.5 x 72 = 612
% flip the whole world upside down
180 rotate -792 -612 translate
gsave 16 316 translate quadpad grestore
gsave 406 316 translate quadpad grestore
2 copies % make two copies
showpage % and print the page
quit

Four Vertical Pads

gutility begin
printerror
nuisance begin
gonzo begin

% defines the first pad
/hack {save /snapch exch def
310 390 translate -180 rotate
20 20 10 setgrid % 27 35 showgrid
% quickboxdraw needs: boxleft width boxtop height
% corner radius line width
0 27 30 0.15 quickboxdraw
0.15 setlinewidth 0 28 mt 27 r
0.09 setlinewidth
[[0 2 mt 27 r] 2 14] yrpt
11 0 mt 30 u
23 0 mt 30 u
25 0 mt 30 u
/cstretch 0.05 def /ssstretch 0.1 def
/font0 /Helvetica-Bold 1.3 gonzofont
/font1 /Helvetica-Narrow 0.9 gonzofont
/font2 /Helvetica 0.9 gonzofont
/yinc 1.3 def
13.5 34 (|0Hardware Hacker #_____
|2Brian Fenton, Electronics Now
|2transfer to IBM from __________________  ) cc
font1
5.5 28.5 (file) cc
17 28.5 (translate as) cc
24 28.5 (X) cc
26 28.5 (P) cc

/font2
1 26.5 (gonzo15a.ptl
Revue
readme_________
/resbn__________.gps
/resbn__________.ad) cl
12 26.5
(gonzo_________.ptl
Revue
readme_________
/resbn__________.gps
/resbn__________.ad) cl
snap restore} def

% defines the second pad
/resbin {save /snapch exch def
310 390 translate -180 rotate
20 20 10 setgrid % 27 35 showgrid
0 27 30 0.15 quickboxdraw
0.15 setlinewidth 0 28 mt 27 r
0.09 setlinewidth
[[0 2 mt 27 r] 2 14] yrpt
11 0 mt 30 u
23 0 mt 30 u
25 0 mt 30 u
/cstretch 0.05 def /ssstretch 0.1 def
/font0 /Helvetica-Bold 1.3 gonzofont
/font1 /Helvetica-Narrow 0.9 gonzofont
/font2 /Helvetica 0.9 gonzofont
/yinc 1.3 def
13.5 34 (|0Resource Bin #_____ 
|2Jack Lemeiux, Nuts & Volts 
|2transfer to IBM from _____________ ) cc
font2
1 26.5 (gonzo15a.ptl
Revue
readme_________
/resbn__________.gps
/resbn__________.ad) cl
12 26.5
(gonzo_________.ptl
Revue
readme_________
/resbn__________.gps
/resbn__________.ad) cl
snap restore} def

% defines the third pad
/blat {save /snapch exch def
20 20 10 setgrid % 27 35 showgrid
0 27 30 0.15 quickboxdraw
0.15 setlinewidth 0 28 mt 27 r
0.09 setlinewidth
[[0 2 mt 27 r] 2 14] yrpt
11 0 mt 30 u
23 0 mt 30 u

Beginning PostScript Projects
## Resource Bin

### Jack Lemeiux, Nuts & Volts

<table>
<thead>
<tr>
<th>file</th>
<th>translate as</th>
<th>X</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>agonzo17</td>
<td>gonzo17.ptl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revue</td>
<td>Revue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>readme_________</td>
<td>readme_________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resbn_________.gps</td>
<td>resbn_________.gps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c_________.12</td>
<td>resbn_________.a.ps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c_________.34</td>
<td>resbn_________.b.ps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resbn_________.txt</td>
<td>resbi_________.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resbn_________.ad</td>
<td>resbi_________.ad</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Hardware Hacker

### Brian Fenton, Electronics Now

<table>
<thead>
<tr>
<th>file</th>
<th>translate as</th>
<th>X</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>agonzo17</td>
<td>gonzo17.ptl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.tit.</td>
<td>hack_________.tit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.txt.</td>
<td>hack_________.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.fig. .1</td>
<td>hack_________.fg1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.fig. .2</td>
<td>hack_________.fg2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.fig. .3</td>
<td>hack_________.fg3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.fig. .4</td>
<td>hack_________.fg4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.fig. .5</td>
<td>hack_________.fg5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.num.</td>
<td>hack_________.num</td>
<td></td>
<td></td>
</tr>
<tr>
<td>___________.res.</td>
<td>___________.res</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.help</td>
<td>hack.hlp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hack.ad.</td>
<td>hack_________.ad</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classified</td>
<td>hack_________.cls</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PostScript Operators

clear red book pg 373
exch red book pg 408
restore red book pg 481
save red book pg 486

Gonzo Commands

gfont1 a way to call fonts for several strings
line1 sets the linewidth at .06 of a point or
grid square
line2 sets the linewidth at .12 of a point or
grid square
line3 sets the linewidth at .18 of a point or
grid square
quickboxdraw draws a rectangular box
snap name of a save object
xrpt name of a save object

On Your Own
Collect and bring in examples of report covers, catalog
covers, book covers etc.

POSTSCRIPT EMULATOR SOFTWARE RESOURCES

<table>
<thead>
<tr>
<th>Freedom of the Press</th>
<th>GoScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>900 Tech Park Drive #8</td>
<td>LaserGo</td>
</tr>
<tr>
<td>Billerica, MA 01821</td>
<td>9235 Trade Place Ste A</td>
</tr>
<tr>
<td>(800) 873-4367</td>
<td>San Diego, CA 92126</td>
</tr>
<tr>
<td>(800) 638-9636</td>
<td>(619) 530-2400</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GhostScript</th>
<th>UltraScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENie PSRT</td>
<td>QMS</td>
</tr>
<tr>
<td>401 N Washington Street</td>
<td>PO Box 81250</td>
</tr>
<tr>
<td>Rockville, MD 20850</td>
<td>Mobile, AL 36689</td>
</tr>
<tr>
<td>(800) 638-9636</td>
<td>(205) 693-4300</td>
</tr>
</tbody>
</table>
A REPORT
A simple report cover includes bold text and uses some simple sketching tools. Also shows use of two relative layout grids.

The Cover

gutility begin
nuisance begin
printerror
gonzo begin

% Do we want to see the main grid?
/showmaingrid false def

% Do we want to see the box grid?
/showboxgrid true def

% set the main page grid
20 15 10 setgrid

% and maybe show the grid
/showmaingrid {57 76 showgrid} if

% put down the main message ...

/cstretch 0.02 def
/ssstretch 0.02 def

/font1 /Palatino-Bold 8 gonzofont
/font2 /Palatino-Bold 2 gonzofont

% our title message
/font1
28.5 10 (A REPORT) cc

% and the author
/font2
28.5 6.5 (by Dramatis Personnae) cc

% draw a moveable box ...

% put bottom of box where wanted
gsave 10.5 20 translate

% create a box
0 0 36 50 1.5 roundbox

% and stroke the outline
0.6 setlinewidth stroke

% view the box grid - maybe
/showboxgrid {36 50 showgrid} if

% draw some stuff inside the box ...

1 setlinecap line1
1 30 mt 34 r % horizon line

% the mountain
gsave
newpath 2 29 mt 6 4.5 rlineto
2 -3 rlineto 4.5 3.5 rlineto
6 -5 rlineto closepath
gsave line3 stroke grestore
bestgray lightgray fill
grestore

% define a reusable generic cactus
/cactus
{1 setlinecap 3 setlinewidth
0 6 mt 18 u 0 setlinecap 0 0 mt 6 u
2 setlinewidth 1 setlinecap -4 22 mt
11 d 4 r 2 setlinewidth 1 setlinecap
3.5 20 mt 6 d 4 l} def

% the big cactus
gsave 28 8 translate cactus grestore

% medium reversed cactus
gsave 8 22 translate
0.2 dup neg exch
scale cactus grestore

% tiny frontwards cactus
gsave 18.5 30.5 translate
0.06 dup neg exch
scale cactus grestore

% add a little sunshine
% arc is defined by x y radius angle1 angle2
newpath 24 42 0 360 arc line3 stroke
grestore % get back out of the box

1 copies

showpage
quit
PostScript Operators

arc    red book pg 365
clear   red book pg 373
currentpoint red book pg 390
dup     red book pg 404
exch    red book pg 408
fill    red book pg 416
neg     red book pg 456
newpath red book pg 457
scale   red book pg 487

Gonzo Commands

roundbox used x y width height radius
showmaingrid used for main grid with additional grids on the same page
showboxgrid used to place a grid in a predefined box

On Your Own

Bring a report of several paragraphs length, preferably something you are currently working on. Or, bring a magazine or book article.
A Text Paragraph
Using Gonzo Justification

A

n introduction to gonzo justification for large blocks of text, using a layout grid

The Actual Program

gutility begin
nuisance begin
printerror
gonzo begin
/showthegrid false def
20 15 10 setgrid
/showthegrid [57 76 showgrid] if
% select a bunch of fonts to show special effects . . .
% Remember that we are on a 10X grid, so 1.4 is 14 points
/font0 /Times-Bold [6 0 0 6 0 -3.48] gonzofont
% special drop cap
/font1 /Times-Roman 1.4 gonzofont
/font2 /Times-Italic 1.4 gonzofont
/font3 /Times-Bold [1.5 0 1.5 0 -1.3] gonzofont
/font4 /Times-Roman 1.2 gonzofont
/font5 /Times-Italic 1.2 gonzofont
/font6 /Times-Bold 1.3 gonzofont
/font7 /Times-Bold 1.3 gonzofont
/font8 /Times-Roman [1.0 0 1.0 0 0.4] gonzofont % superscript
/font9 /Times-Roman [1.0 0 1.0 0 -0.3] gonzofont % subscript
/font- /ZapfDingbats 1.4 gonzofont
/font= /Symbol 1.4 gonzofont
% pick some template values
/txtwide 38 def % width of text
/yinc 1.7 def % line spacing
/estretch 0.02 def % extra space between characters
/ssstretch 0 def % extra space between spaces
/maxstretch 0.25 def % trip point for character stretching
/maxcstretch 0.1 def % maximum character stretching
/dropcount 3 def % number of lines the drop cap drops
/dropindent 5 def % indentation for the drop cap
/lastlinestretch 0.006 def
% slight stretch to last paragraph line
/overstrikechar (\320) def % overstrike character
/overstrikhe 0.7 def % vertical shift of overstrike character
/ytop 59 def % base of top text line
/ypos ytop def % select top text line
/xpos 10 def % left margin
/pm 1 def % paragraph margin
/kern 0.1 def % one point of custom kerning
% select two macros to simplify drop caps and centered drop text
/amacro {((Flzy0) stringmacro) def % start drop cap
/bmacro {((iFy1) stringmacro) def % finish drop cap
/cmacro {((zynC3) stringmacro) def % centered title
/dmacro {((pFl1) stringmacro) def % normal text
% and print the paragraph
% IMPORTANT NOTE: Some word processors may force
% extra carriage returns on you. The reverse slashes
% shown below avoid this problem, but may not be
% required. Try it both ways.

startgonzo
|a
|b
|c
|d
|e
|f
|g
|h

ere is a demonstration of a premium, gonzo fill justified paragraph that has lots of interesting features. A very fancy automatic fill justify is used which stretches both spaces and between-character spacing by selected and controlled amounts. You can also individually kern characters, as in the word

 ASSIGNMENT 7:
Do a text paragraph using super & subscripts and at least one heading

|4A|kw|kARD|1 here. See how the |4"W"|1 overlaps the
|4"A"|1.
|c
A Centered And Dropped Heading
|d
Note that only one single embedded command is needed to start either the drop cap or the auto centering headers. Note also that the above header is dropped slightly to improve its vibes.
|b
Naturally, |7bold|1, |2italic|1, |8superscript|1, and |9subscript|1 are very easy to in- clude, freely intermixed on any line. Fully automatic hanging punctuation is included for such things as periods, commas, and hyphens. This greatly improves the appearance, especially with small columns of |4300 DPI|1 text.
|h
Some other tricks include showing |4ALL CAPITAL LETTERS|1 in a slightly smaller font for improved

Beginning PostScript Projects
Here is a demonstration of a premium, gonzo fill justified paragraph that has lots of interesting features. A very fancy automatic fill justify is used which stretches both spaces and between-character spacing by selected and controlled amounts. You can also individually kern characters, as in the word AWARD here. See how the "W" overlaps the "A".

**A Centered And Dropped Heading**

Note that only one single embedded command is needed to start either the drop cap or the auto centering headers. Note also that the above header is dropped slightly to improve its vibes.

Naturally, **bold**, *italic*, superscript, and subscript are very easy to include, freely intermixed on any line. Fully automatic hanging punctuation is included for such things as periods, commas, and hyphens. This greatly improves the appearance, especially with small columns of 300 DPI text.

Some other tricks include showing ALL CAPITAL LETTERS in a slightly smaller font for improved balance, and in using a Greek alphabet should you need something electronic, such as a 220Ω resistor, a ℛΣΤ† complement, or an 0.1 µF capacitor. ♦
balance, and in using a Greek alphabet should you need something electronic, such as a 220|=W|= resistor, a 4R|=o|=T|= complement, or an 0.1 |=m|=F|= capacitor. |G|= 1

|x % end of gonzo marker % draw a box to prove we are out of gonzo 5 17 47 1.5 roundbox
[0.8 0 0.6 1 0.35 0] superinsidestroke
1 copies % make one copy
showpage % and show the page quit

Gonzo Commands
macro an executable array
startgonzo Begins gonzo scanning of text that follows. |g may also be used. Terminates on end of file or |x.

|? User defined embeddable escape commands and macro commands.

stringmacro A shorthand way of building macros. For instance /amacro { (Cyz3) stringmacro} def does an |C |y |z |3 all on a single |a embedded command.

Gonzo Variables
ytop Vertical distance to base of top column line.
ybot Vertical distance to base of bottom column line.
ypos Current vertical distance from bottom of page.
yinc Vertical distance between column lines.
xpos Left margin of current column, measured from left of page.
txtwide Width of current column in points.
yparendadj Extra vertical paragraph end leading (usually zero).
colcheck Checks for column bottom; link to page or column maker.
kern Adjusts the space between characters.
cstretch Minimum constant kerning of all characters (usually 0.2).
sstretch Minimum constant kerning of all spaces (usually zero).
lastlinestretch Slight stretch to last paragraph line.
maxxsstretch Maximum space stretch before character stretching begins.
maxcstretch Maximum allowable character stretching.
dropcount Number of character lines needed by initial drop cap.
dropindent Horizontal width reserved by initial drop cap.
overstrikechar Overstrike character.

overstrikeht Vertical shift of overstrike character.
headerx Header xpos as used by template (optional).
headery Header ypos as used by template (optional).
headerwide Header txtwide as used by template (optional).

Embeddable Gonzo Commands.
|0 Change to font0. Preferred use = initial drop cap.
|1 Change to font1. Preferred use = normal body text.
|2 Change to font2. Preferred use = italic body text.
|3 Change to font3. Preferred use = bold body text.
|4 Change to font4. Preferred use = all caps normal body text.
|5 Change to font5. Preferred use = all caps italic body text.
|6 Change to font6. Preferred use = all caps bold body text.
|7 Change to font7. Preferred use = bold and dropped headers.
|8 Change to font8. Preferred use = subscripts.
|9 Change to font9. Preferred use = superscripts.
|- Change to font-. Preferred use = ZapfDingbats.
|= Change to font=. Preferred use = Symbol in body text size.
|C Change to center justification on xpos+txtwide.
|F Change to fill justification between xpos and xpos+txtwide.
|L Change to left justification starting at xpos.
|R Change to right justification ending at xpos+txtwide.
|a Perform user defined Macro amacro.
|b Perform user defined Macro bmacro.
|c Perform user defined Macro cmacro.
|d Perform user defined Macro dmacro.
|e Perform user defined Macro emacro.
|f Perform user defined Macro fmacro.
|g Start gonzo justification. A PostScript alias for startgonzo.
|h Do a half of a negative linefeed.
|i Do an initial drop capital.
|j Do a positive kern.
|k Do a negative kern.
|l Do a full linefeed.
|n Nobreakconditional formfeed if within six lines of ybot.
|o Do an overstrike character.
|p Switch to normal paragraph margin, starting with next line.
| s | Do a showpage for end of scanned file only. |
| x | Exit gonzoo justify and switch to native PostScript. |
| y | Do a yinc up full negative linefeed. |
| z | Switch to zero paragraph margin, starting with next line. |

**On Your Own**


Collect personal data for your resume. Work on finding a simple silhouette or line art for curvetracing.

---

### DESK TOP RESOURCES

<table>
<thead>
<tr>
<th>Bove &amp; Rhodes Reports</th>
<th>Personal Publishing</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO Box 1289</td>
<td>245 South Parkside</td>
</tr>
<tr>
<td>Gualala, CA 95445</td>
<td>Itaska, IL 60143</td>
</tr>
<tr>
<td>(707) 884-4413</td>
<td>(312) 250-8900</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Desktop Communications</th>
<th>Presentation Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO Box 941745</td>
<td>513 Wilshire Blvd, Ste 344</td>
</tr>
<tr>
<td>Atlanta, GA 30341</td>
<td>Santa Monica, CA 90401</td>
</tr>
<tr>
<td>(800) 966-9052</td>
<td>(213) 455-1414</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Midnight Engineering</th>
<th>Publish!</th>
</tr>
</thead>
<tbody>
<tr>
<td>1700 Washington Avenue</td>
<td>501 Second Street</td>
</tr>
<tr>
<td>Rocky Ford, CO 81067</td>
<td>San Francisco, CA 94107</td>
</tr>
<tr>
<td>(719) 254-4558</td>
<td>(415) 243-0600</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nuts &amp; Volts</th>
<th>Recharger</th>
</tr>
</thead>
<tbody>
<tr>
<td>430 Princeland Court</td>
<td>3340 Sunrise Ave #102</td>
</tr>
<tr>
<td>Corona, CA 91719</td>
<td>La Vegas, NV 89101</td>
</tr>
<tr>
<td>(714) 371-8497</td>
<td>(702) 438-5557</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PC Publishing</th>
<th>Step-By-Step Elect Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>950 Lee Street</td>
<td>6000 North Forest Park Dr</td>
</tr>
<tr>
<td>Des Plaines, IL 60016</td>
<td>Peoria, IL 61656</td>
</tr>
<tr>
<td>(312) 296-0770</td>
<td>(800) 255-8800</td>
</tr>
</tbody>
</table>

---

### TRADITIONAL PRINTING RESOURCES

<table>
<thead>
<tr>
<th>American Printer</th>
<th>Printing Impressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>29 North Wacker Drive</td>
<td>401 North Broad Street</td>
</tr>
<tr>
<td>Chicago, IL 60606</td>
<td>Philadelphia, PA 19108</td>
</tr>
<tr>
<td>(312) 726-2802</td>
<td>(215) 238-5300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Converting Magazine</th>
<th>Printing Journal</th>
</tr>
</thead>
<tbody>
<tr>
<td>400 North Michigan Avenue</td>
<td>80 S Lake Avenue #550</td>
</tr>
<tr>
<td>Chicago, IL 60611</td>
<td>Pasadena, CA 91101</td>
</tr>
<tr>
<td>(312) 222-2000</td>
<td>(818) 793-7717</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>High Volume Printing</th>
<th>Printing News</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box 368</td>
<td>245 West 17th Street</td>
</tr>
<tr>
<td>Northbrook, IL 60065</td>
<td>New York, NY 10011</td>
</tr>
<tr>
<td>(312) 564-5940</td>
<td>(212) 463-6727</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>In-Plant Printer</th>
<th>Publishing Executive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box 368</td>
<td>401 North Broad Street</td>
</tr>
<tr>
<td>Northbrook, IL 60065</td>
<td>Philadelphia, PA 19108</td>
</tr>
<tr>
<td>(312) 564-5940</td>
<td>(215) 238-5300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>In-Plant Reproductions</th>
<th>Publishing Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>401 North Broad Street</td>
<td>401 North Broad Street</td>
</tr>
<tr>
<td>Philadelphia, PA 19108</td>
<td>Philadelphia, PA 19108</td>
</tr>
<tr>
<td>(215) 238-5300</td>
<td>(215) 238-5300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instant &amp; Small Printer</th>
<th>Quick Printing</th>
</tr>
</thead>
<tbody>
<tr>
<td>425 Huehl Road, Bldg 11</td>
<td>1680 SW Bayshore Blvd</td>
</tr>
<tr>
<td>Northbrook, IL 60065</td>
<td>Port St Lucie, FL 34984</td>
</tr>
<tr>
<td>(708) 564-5940</td>
<td>(407) 879-6666</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Paper &amp; Foil Converter</th>
<th>Screen Printing</th>
</tr>
</thead>
<tbody>
<tr>
<td>29 North Wacker Drive</td>
<td>407 Gilbert Avenue</td>
</tr>
<tr>
<td>Chicago, IL 60606</td>
<td>Cincinnati, OH 45202</td>
</tr>
<tr>
<td>(312) 726-2802</td>
<td>(513) 421-2050</td>
</tr>
</tbody>
</table>

---

### PRINTING EQUIPMENT RESOURCES

<table>
<thead>
<tr>
<th>California Printers</th>
<th>Print Equipment News</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO Box 11766</td>
<td>Box 5540</td>
</tr>
<tr>
<td>Santa Ana, CA 92711</td>
<td>Glendale, CA 91221</td>
</tr>
<tr>
<td>(714) 836-9653</td>
<td>(818) 954-9495</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Horsetrader</th>
<th>Printers Shopper</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO Box 11712</td>
<td>PO Drawer 1056</td>
</tr>
<tr>
<td>Santa Ana, CA 92711</td>
<td>Chula Vista, CA 91912</td>
</tr>
<tr>
<td>(712) 921-3120</td>
<td>(800) 854-2911</td>
</tr>
</tbody>
</table>

---

36 PostScript Beginning Projects
Resumes

Two forms of resumes. A standard resume done on a 10X grid and a chronological resume done without a grid.

The Standard Resume

gutility begin nuisance begin printerror gonzo begin
/showthegrid true def
20 15 10 setgrid
/showthegrid {57 76 showgrid} if
/estretch 0.02 def /sstretch 0.02 def

% FANCY TITLES VIA CUSTOM JUSTIFICATION
% There are two programmable justification modes in % gonzo, called justP and justQ. These can be entered at % the end of each line. At that time, you can measure the % actual width by doing a txtwide roomleft sub, you can % draw arbitrary art, or you can alter % stretch or positioning, etc. In this case, % we do a superfancy underliner that % automatically tracks the width of each % heading...

/superstroke { save /sssnap exch def
/sscmd exch def 0 2 sscmd length 2 div cvi
1 sub 2 mul (/aposn exch def gsave sscmd aposn 1 add get setgray stroke grestore) for sssnap restore newpath} def

/justQ {txtwide roomleft sub gsave xpos ypos 0.5 sub moveto 0.2 add 0 rlineto 1 setlinecap bestgray [0.35 0 0.2 0.9] superstroke grestore justL} def
/titshift 0 def
% pick some fonts
/font0 /Bookman-DemiItalic 1.1 gonzofont
/font1 /Bookman-Light 0.9 gonzofont
/font2 /Bookman-Light 0.9 gonzofont
/font3 /Bookman-DemiItalic 1.0 gonzofont
/font4 /Bookman-LightItalic 0.9 gonzofont
/font5 /Bookman-Demi 0.85 gonzofont

% amacro draws a justifying underline, % bmacro indents your text, % cmacro unindents you ...% amacro { (Q3) stringmacro /xpos xpos titshift sub def} def
/bmacro { (L2) stringmacro /xpos xpos 7 add def} def
/cmacro { /xpos xpos -7 add def} def

/yinc 1.1 def
53 74 ([0CRAIG FAULKNER]1
1847 Thatcher Blvd
Safford, AZ 85546
|h
(602) 428-1047) cr

/bestgray 1 setlinecap % a nice header line
2.5 69 mt 53 0 rline
[0.5 0 0.35 0.9] superstroke
3 1.5 mt 52 0 rline
[0.5 0 0.35 0.9] superstroke
/xpos 6.5 def % indent body 65 points
/ypos 66 def % top of body 680 points
/yinc 1.15 def % 11.5 point spacing
/txtwide 35 def % 350 point main text width

/startgonzo

|aObjective:|

I am seeking a position where I will utilize my extensive computer application skills.

|aSkills:|

5Hardware: Networking of Digital 8350 and MicroVax II; Texas Instruments DX110; IBM P/Ss, ATs, and XTs; Macintosches, Apple IIGs; and PostScript Laser Printers.

|aExperience:|

58/87 - 5/89 System Manager of MicroVax II|2
Eastern Arizona College, Thatcher, Arizona
Duties: Set up, develop programs for, and manage a network of mini- and micro-computers; manage user accounts and disk space; install and update software changes; work with faculty and administration in resolving...
Objective:
I am seeking a position where I will utilize my extensive computer application skills.

Skills:

Hardware: Networking of Digital 8350 and MicroVax II; Texas Instruments DX110; IBM P/Ss, ATs, and XTs; Macintoshes, Apple IIgs; and PostScript Laser Printers.

Software: Local area transport, communications, word processing, data base, and spreadsheet applications, desk top publishing, as well as inventory management.

Programming: COBOL, Pascal, C, DCL, PostScript, Machine Language, and BASIC.

Business: General business practices including accounting, inventory, sales, and marketing. Type 50 wpm.

Experience:

8/87 - 5/89 System Manager of MicroVax II
Eastern Arizona College, Thatcher, Arizona
Duties: Set up, develop programs for, and manage a network of mini- and micro-computers; manage user accounts and disk space; install and update software changes; work with faculty and administration in resolving on campus computer problems; assist students in editing and debugging programs. Do backups on systems;

3/86 - 5/89 Computer Consultant/Technician
Safford Builders Supply, Safford, Arizona
Duties: Installed and maintained communications equipment for data transmission between office and two stores; interpreted data for inventory programs; applied programs to receive and process daily sales information; translated software documentation into meaningful terms for office personnel.

4/88 - Present Computer Consultant/Technician
OmniTech, Safford, Arizona
Duties: Partner in a self started business designed to aid both business and individuals in computer related problems. Install and troubles shoot various types of computer systems.

Education:

Eastern Arizona College, Thatcher, Arizona.

Graduated May 1987. Nominated by faculty as Best All Around Senior; appeared on KTSP-TV for Academic All Stars; Student Body President. References available upon request.
on campus computer problems; assist students in editing and debugging programs. Do backups on systems.

Safford Builders Supply, Safford, Arizona
Duties: Installed and maintained communications equipment for data transmission between office and two stores; interpreted data for inventory programs; applied programs to receive and process daily sales information; translated software documentation into meaningful terms for office personnel.

[54/88 - Present] Computer Consultant/Technician
OmniTech, Safford, Arizona
Duties: Partner in a self started business designed to aid both business and individuals in computer related problems. Install and troubleshoot various types of computer systems.

Education:

New Life Christian School, Thatcher, Arizona. Graduated May 1987. Nominated by faculty as Best All Around Senior; appeared on KTSP-TV for Academic All Stars; Student Body President. References available upon request.

Chronological Resume

Old Mill — Glendale, Arizona
Position: First Assistant Manager
- Audit cash register/monies received
- Facilitate intra-company communications
- Submit daily deposits to bank
- Handle customer relations and correspondence
- Call in daily earnings
- Coordinate and conduct fashion shows
- Analysis of daily, weekly, monthly sales
- Transfer merchandise to manufacturer
- Compare sales figures to previous year
- Research bid/purchase orders for store maintenance
- Record/maintain petty cash fund
- Accept/check in shipments
- Assist in hiring sales personnel
- Cover floor for maximum sales potential
- Report payroll to parent company
- Activate/deactivate store alarm

J.C. Penney — Flagstaff, Arizona
April 1989 – January 1991
Position: Sales Associate
- Weekend supervisor
- Opened and closed cash registers
- Maintained inventory control
- Conducted department meetings
- Lock stored after closing store
CATHY LANCASTER

WORK EXPERIENCE

Old Mill — Glendale, Arizona
Position: First Assistant Manager
- audit cash register/monies received
- submit daily deposits to bank
- call in daily earnings
- analysis of daily, weekly, monthly sales
- compare sales figures to previous year
- record/maintain petty cash fund
- assist in hiring sales personnel
- report payroll to parent company
- handle customer relations and correspondence
- coordinate and conduct fashion shows
- transfer merchandise to manufacturer
- research bid/purchase orders for store maintenance
- accept/check in shipments
- cover floor for maximum sales potential
- activate/deactivate store alarm

J.C. Penney — Flagstaff, Arizona
Position: Sales Associate
April 1989 – January 1991
- weekend supervisor
- maintained inventory control
- wrote sales goals for associates
- conducted department meetings
- opened and closed cash registers
- completed reports of errors
- called in catalog orders for customers
- locked safe after closing store

Cloth World — Tempe, Arizona & Phoenix, Arizona
Position: Sales Associate
March 1987 – July 1988
- promoted sales/special events
- provided customer service
- handled custom orders
- created displays
- took inventory of stock
- put out new stock

SKILLS
Macintosh – MS Word Type 45 wpm
IBM – Word Perfect Point of Sale Terminal
Apple II – AppleWorks Ten Key Calculator/Sharp Cash Register

EDUCATION
Northern Arizona University
August 1988 – May 1991
Bachelor of Science: Fashion Merchandising and Marketing
Courses of Study:
- Business Statistics
- Presentation Techniques
- Market Research
- Merchandising Math

Phoenix College
January 1988 – May 1988

Arizona State University
August 1986 – December 1987

ACTIVITIES/AWARDS

J.C. Penney
- Monetary Award High Sales during Fashion Expo Sale September 1990
- Customer Service Award for Recognition of Service in Intimate Apparel November 1990

Northern Arizona University
- Associated Members of Fashion Merchandising/Historian September 1990 – May 1991
- European Fashion Study Tour June 1990

Arizona State University & Phoenix College
- Marching & Concert Bands August 1986 – May 1988
- Kappa Delta Sorority/Beta Psi Chapter August 1986 – December 1987
|Cloth World| — Tempe, Arizona & Phoenix, Arizona

- **March 1987 – July 1988**
  - **Position:** Sales Associate
  - **Tasks:**
    - Promoted sales/special events
    - Created displays
    - Provided customer service
    - Took inventory of stock
    - Handled custom orders
    - Put out new stock

**SKILLS**

- Macintosh – MS Word
- Type 45 wpm
- IBM – Word Perfect
- Point of Sale Terminal
- Apple II – AppleWorks
- Ten Key Calculator/Sharp Cash Register

**EDUCATION**

- Northern Arizona University
  - August 1988 – May 1991
  - Bachelor of Science: Fashion Merchandising and Marketing
  - Courses of Study:
    - Business Statistics
    - Market Research
    - Presentation Techniques
    - Merchandising Math

- Phoenix College
  - January 1988 – May 1988
  - Arizona State University
  - August 1986 – December 1987

**ACTIVITIES/AWARDS**

- J.C. Penney
  - Monetary Award High Sales during Fashion Expo Sale
  - September 1990
  - Customer Service Award for Recognition of Service in Intimate Apparel
  - November 1990

- Northern Arizona University
  - Associated Members of Fashion Merchandising/Historian
  - September 1990 – May 1991
  - European Fashion Study Tour
  - June 1990
  - CrossCurrents Fashion Show & Seminar/Set & Stage Producer
  - January 1991 – May 1991

- Arizona State University & Phoenix College
  - Marching & Concert Bands
  - August 1986 – May 1988
  - Kappa Delta Sorority/Beta Psi Chapter

---

**TONER REFILLING RESOURCES**

<table>
<thead>
<tr>
<th>Black Lightning</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR 1-87 Depot Road</td>
</tr>
<tr>
<td>Hartland, VT 05048</td>
</tr>
<tr>
<td>(800) BLACK99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Laser Products, Inc</th>
</tr>
</thead>
<tbody>
<tr>
<td>12900 SW 132nd Court</td>
</tr>
<tr>
<td>Miami, FL 33186</td>
</tr>
<tr>
<td>(305) 235-9544</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diversified TechniGraphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Morgan, Ste 112</td>
</tr>
<tr>
<td>Irvine, CA 92718</td>
</tr>
<tr>
<td>(714) 855-3838</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recharger Magazine</th>
</tr>
</thead>
<tbody>
<tr>
<td>3340 Sunrise L Avenue #102</td>
</tr>
<tr>
<td>Las Vegas, NV 89101</td>
</tr>
<tr>
<td>(702) 438-5557</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Imaging Specialties, Inc</th>
</tr>
</thead>
<tbody>
<tr>
<td>8475 Canoga Avenue</td>
</tr>
<tr>
<td>Canoga Park, CA 91304</td>
</tr>
<tr>
<td>(818) 709-4244</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Toner Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>206 A Street</td>
</tr>
<tr>
<td>Needham, MA 02194</td>
</tr>
<tr>
<td>(800) 288-6637</td>
</tr>
</tbody>
</table>

---

**Gonzo Commands**

- `titshift` horizontal shift in title
- `tabs` allows you to set left justified tabs or columns
- `justQ` user defined justification mode
- `justP` user defined justification mode
- `justL` left justification

**On Your Own**

Bring a simple line drawing or silhouette for curvetracing. Some sources: coloring books, clip art books, Dover books, cookie cutters, rubber stamps, newspapers, stencils, flyers, etc.
plain stroked curvetrace

plain filled curvetrace

showing points

curvetrace inside a path

curvetrace with a cusp

pipe or wire

double filled curvetrace

changing tension
Simple Curvetrace

Shows you several fundamental exercises in curvetracing that you can use to draw smooth continuous lines and curves.

Curvetrace Rules

% curvetrace rules: The curvetrace operator expects an array of form [ x0 y0 ang0 x1 y1 ang1 .... xn yn angn ]
% curvetrace, where:
% x0 is the horizontal position of your first point
% y0 is the vertical position of your first point
% ang0 is the direction you wish to LEAVE your first point
% with 0 = east 90 = north 180 = west -90 = south
% xn is the horizontal position of your last point
% yn is the vertical position of your last point
% angn is the direction you wish to ENTER your last point
% with 0 = east 90 = north 180 = west -90 = south
% if you want to see where you are, use /showtick true def.
% To remove all ticks, use /showtick false def.
% special rule: if x0 and y0 are both exactly zero, then % your curve APPENDS an existing path; if not, a new % path is begun.

Curvetrace Examples

gutility begin
nuisance begin
printerror
gonzo begin

/showfullgrid false def
20 20 10 setgrid
/showfullgrid [57 76 showgrid] if
/cstretch 0.04 def /strectch 0.02 def
/font2 /Helvetica-Narrow-Bold 0.8 gonzofont

% EXAMPLE #1: a plain old curvetrace
gsave 14 55 translate
10 10 showgrid

% EXAMPLE #2: a plain old curvetrace showing your % points for debugging
gsave 14 40 translate
10 10 showgrid
gsave 14 55 translate
10 10 showgrid

% EXAMPLE #3: a curvetrace with a cusp
% a cusp is done by repeating two x and y points having a % different entry and exit angle. Note how 4,2 repeats here.
gsave 14 25 translate

% EXAMPLE #4: variable linewidth curvetracing.
% for variable linewidths, use an "up" path and a "down" % path and then fill between them. WATCH THE PATH % SEQUENCE, OR % STRANGE THINGS % WILL HAPPEN!
% Your second path should % continue from where the % first one leaves you and % you end up back at % the beginning.
gsave 14 10 translate
10 10 showgrid

% EXAMPLE #5: a filled curvetrace
% All the previous examples have stroked the path. This % one fills it instead
gsave 34 55 translate
10 10 showgrid

ASSIGNMENT 9:
Change each example and draw a simple curvetrace object.

% EXAMPLE #6: curvetraceing inside a path

% The rule: If x0 and y0 are both exactly zero, then the
curvetrace APPENDS itself to an existing path. If not,
a new path is begun.

gsave 34 40 translate
10 10 showgrid
1 5 mt                  % start the path
3 pr [0 0 0 5 9 180 6 5 0] curvetrace
% add a curved portion
3 pr                      % add another straight part
0.3 setlinewidth 1 setlinecap stroke
5 -1 ([2curvetrace inside a path] cc
grestore

% EXAMPLE #7: pipes, braids or wire are done with
% multiple stroking.
% just use superstroke or superinsidestroke for fancy
drawing. With superstroke, an array of
% [fattestthick fattestgray .... thinnestthick thinnestgray]
% is expected.

gsave 34 25 translate
10 10 showgrid
[2 3 0 8 7 0] curvetrace
1 setlinecap bestgray
[0.6 0 0.45 0.9] superstroke
% black 0.6 wide and gray 0.45 wide
5 -1 ([2pipe or wire] cc
grestore

% EXAMPLE #8: changing the tension.
% an internal parameter called /tension decides how "gung
% ho" the curve leaves and enters the points. The "best"
% tension value is usually 2.8. Higher tension values make
% the curvetracing more "rigid" or straight-line-like.
% Negative tension values make the curvetracing "looser"
% or more loopy. Negative tensions go round the
% other way.

gsave 34 10 translate
10 10 showgrid
% define a simple curvetrace
0 setlinewidth
/ctproc {[3 3 90 7 7 0] curvetrace stroke} def
/tension 0.48 def ctproc % so loopy it overshoots
/tension 0.55 def ctproc % really gung ho
/tension 1.0 def ctproc
/tension 1.4 def ctproc
/tension 2.0 def ctproc
/tension 2.8 def ctproc % normal curve
/tension 6.0 def ctproc % fairly rigid
/tension 100 def ctproc % almost straight
/tension -0.8 def ctproc % round the other way.
5 -1 ([2changing tension] cc
grestore

showpage quit

PostScript Operators
curvyeto red book pg 393

Gonzo Commands
c㭎proc curvetrace proc – allows you to change tension on a curvetraced path
curvetrace calls the curvetraced path
showtick shows an arrow at each data point at angle entered
superinsidestroke allows an a-symmetric border

CLIP ART & CURVETRACE IDEAS

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Art Master</td>
<td>Box 8532, San Marino, CA 91108</td>
<td></td>
</tr>
<tr>
<td>California Freeware</td>
<td>1747 East Avenue Q #C-1</td>
<td>(805) 273-0300</td>
</tr>
<tr>
<td>Clipper</td>
<td>Dynamic Graphics, Peoria, IL 61614</td>
<td>(309) 688-8800</td>
</tr>
<tr>
<td>Dot Graphic Supply Co</td>
<td>1612 California St, Omaha, NE 68101</td>
<td>(800) 228-7272</td>
</tr>
<tr>
<td>Dover Publications</td>
<td>31 E 2nd St, Mineola, NY 11501</td>
<td>(516) 294-7000</td>
</tr>
<tr>
<td>Dream Maker Software</td>
<td>7217 Foothill Blvd, Tujunga, CA 91042</td>
<td>(800) 876-5665</td>
</tr>
<tr>
<td>GEnie BBS</td>
<td>PSRT, DTP, HSBS, ARTS</td>
<td></td>
</tr>
<tr>
<td>Graphic Products Corp</td>
<td>3601 Edison Place, Rolling Meadows, IL 60008</td>
<td>(312) 537-9300</td>
</tr>
<tr>
<td>Multi-Ad Services, Inc</td>
<td>PO Box 806, Peoria, IL 61652</td>
<td>(309) 692-1530</td>
</tr>
</tbody>
</table>

GRAPHICS PUBLICATIONS RESOURCES

<table>
<thead>
<tr>
<th>Publication</th>
<th>Address</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Imaging</td>
<td>210 Crossways Park Drive</td>
<td>(516) 496-8000</td>
</tr>
<tr>
<td>Graphic Arts Monthly</td>
<td>249 West 17th Street</td>
<td>(212) 463-6834</td>
</tr>
<tr>
<td>Art Material Trade News</td>
<td>6255 Barfield Road</td>
<td>(404) 256-9800</td>
</tr>
<tr>
<td>Graphic Arts Product News</td>
<td>29 North Wacker Drive</td>
<td>(312) 726-2802</td>
</tr>
<tr>
<td>Computer Graphics World</td>
<td>One Technology Park Drive</td>
<td>(508) 692-0700</td>
</tr>
<tr>
<td>Step-By-Step Graphics</td>
<td>6000 North Forest Park Drive</td>
<td>(800) 255-8800</td>
</tr>
<tr>
<td>Computer Pictures</td>
<td>701 Westchester Avenue</td>
<td>(914) 328-9157</td>
</tr>
</tbody>
</table>
tension changes the amount of curve or negative curve generated
ticklen defines the length of arrow shaft
tickhead defines the length of arrow head

On Your Own
Bring several names/addresses of friends or business to which you frequently address envelopes/letters/labels.
April 22, 1992

Aaron A. Aardvark
111 Ardmore Avenue
Artesia, AZ, 11111

This is a sample form letter with auto-changing addresses. The body of your form letter goes in this space.

Sincerely yours,

DTP Class, Spring 1995
Autoaddressing – Letters, Envelopes, Labels

An introduction to auto-addressing form letters, envelopes, and address labels. The same routines are easily used for autolabeling custom certificates, and awards.

Provisions are included for auto extracting custom information from a name or proc file. When using custom info, the routines are smart enough to handle partial sheets. Custom data can also be placed in variable form positions.

The actual stepandrepeat files being used are:

[1 1 685 305 0 150 20 true true] def  % defines #10 envelope
[1 1 0 20 20 50 false false] def  % entire portrait page for form letters
[1 4 290 180 65 40 true false] def  % 4 custom shipping labels
[1 1 0 20 20 50 false true] def  % entire landscape page for certificates or awards

% default variables used by stepnrpt
/srexitproc {} def     % default short exit - don’t do it.
/numpages 1 def        % default number to print
/startnum 0 def        % initial ticket number
/runnum startnum def   % running pointer
% advances one per repeat
/repeatproc {} def     % artwork to get
% repeated - default
/customdata false def  % autopaginate
% custom data?

% default srfile gets used for custom entries. Each entry can be a string or a proc, but ONLY ONE ENTRY % is allowed per final repeat.
/srfile [(You) (forgot) (to) (define) (srfile!)] def
% default srfile

% /calcpages is an optional routine that decides how many % pages are needed ONLY when you are using a custom % srfile. This allows early exits when or if you run out % of data.
/calcpages {dup cvn stepnrptparams exch get dup 0 get exch 1 get mul cvi srfile length exch div ceiling cvi /numpages exch def} def

% To force an early exit when you run out of names or % reach a given ticket number, test suitably. Then % conditionally make /quickexit true and exit repeatproc.

% Note that an ending showpage is NOT required and % should NOT be used.

Autoaddressing Form Letters
% This sends a form letter to everybody on the address list.
% Again, use the Gonzo routines for first rate quality
gutility begin
printerror
nuisance begin
gonzo begin
% your letterhead goes here

/letterhead {gsave
/font0 /Helvetica-Bold [12 0 0 12 0 0] gonzofont
296 730 (0Your letterhead goes here) cc
1 setlinecap 2 setlinewidth
50 720 moveto 530 0 rlineto stroke
restore} def

/font1 /Palatino-Roman 11 gonzofont
/font2 /Palatino-Italic 11 gonzofont
/font3 /Palatino-Bold 11 gonzofont
/font4 /Palatino-BoldItalic 11 gonzofont

/450 680 (March 30, 1994) cr
grestore} def

/body { gsave
font1
75 340
(|1This is a |2sample form letter|1 with |3auto- changing|1 addresses. The body of your form letter goes in this space.)
c1
grestore} def

/srfile length 1 sub runnum lt [/quickexit true def exit] if

DTP Class, Spring 1994) c1 grestore } def

/repeatproc { srfile length 1 sub runnum lt [/quickexit true def exit] if

letterhead
date
body
signature
200 10 moveto
75 400 srfile runnum get c1

ASSIGNMENT 10:
Do a personal or business envelope, form letter, and/or labels.
Zeus Zimmerman
9999 Zoraster Way
Zion, UT 9999

Aaron A. Aardvark
111 Ardmore Avenue
Artesia, AZ 11111
Auto Addressing Envelopes

% This example creates and then auto-addresses a business
% envelope. Once again, use the Gonzo utilities for best
% results.

gutility begin % open some dictionaries
printerror % print errors to paper
nuisance begin % print errors to paper
gonzo begin % open some dictionaries
% /striplines takes a multi-line string, extracts each
% individual line string, and then executes stripproc for
% each line. Useful for simple addressing and custom form
% filling-in.
/striplines {gsave () search {exch pop stripproc}
{stripproc exit} ifelse} loop grestore def % Some stripproc is needed by striplines. This stripproc
% does a left justified address, given the x and y position
% on the stack
/stripproc {currentpoint 3 -1 roll show yinc sub moveto} def % And this sits in for your logo and return address...

/returnaddress {gsave
(Zeus Zimmerman
9999 Zoraster Way
Zion, UT 9999) striplines
grestore} def

/yinc 14 def % spacing of both addresses

/font1 /Bookman-Demi [10 0 0 10 0 0] gonzofont
/font2 /Bookman-Demi [12 0 0 12 0 0] gonzofont

/repeatproc {
srfile length 1 sub runnum lt{/quickexit true def exit} if
20 280 moveto % positions return address
font1 setfont
returnaddress

270 135 moveto % positions main address/font2 setfont
srfile runnum striplines
} def

/srfile [
(Aaron A. Aardvark
111 Ardmore Avenue
Artesia, AZ, 11111)

(Boyd B. Benson
222 Baker Boulevard
Bismark, BB, 22222)

(Clyde C. Calhoun
333 Cumberland Circle
Chambers, CA, 3333)
] def

/runnum 0 def % open some dictionaries
/customdata true def
(fullportpage) stepandrepeat
quit

Auto Addressing Labels

% This example creates and then auto-addresses a strip of
% custom shipping labels. Once again, use the Gonzo
% justification and utilities for fast and convenient results.

gutility begin % open some dictionaries
printerror % print errors to paper
nuisance begin % print errors to paper
gonzo begin % open some dictionaries
% /striplines takes a multi-line string, extracts each
% individual line string, and then executes stripproc for
% each line. Useful for simple addressing and custom form
% filling-in.
/striplines {{() search {exch pop stripproc}
{stripproc exit} ifelse} loop } def % Some stripproc is needed by striplines. This stripproc
% does a left justified address...
/stripproc { font2 xpos ypos moveto show 0 yinc neg translate } def % And this sits in for your logo and return address...

/labelart {gsave 6 setlinewidth 0 setgray 1 setlinejoin
1 setlinecap 10 10 moveto 160 rlineto 270 0 rlineto
0 -160 rlineto closepath stroke 3 setlinewidth
10 112 moveto 270 0 rlineto stroke} def % a simple centering proc

censhow {currentpoint exch index stringwidth pop 2 div
sub exch moveto show} def % Here is some crude label art ...

/labelart {gsave 6 setlinewidth 0 setgray 1 setlinejoin
1 setlinecap 10 10 moveto 160 rlineto 270 0 rlineto
0 -160 rlineto closepath stroke 3 setlinewidth
10 112 moveto 270 0 rlineto stroke} def % And this sits in for your logo and return address...

/logoandreturnaddress {gsave
145 135
{{1LOGO
}}
}
\[2& \text{RETURN ADDRESS}) \text{ cc}\]
\}
def
/font1 /Bookman-Demi \[16 \text{ 0 20 0 0] gonzofont
/font2 /Bookman-Demi \[12 \text{ 0 12 0 0] gonzofont
/xpos \text{35 def} \% \text{left margin of main address}
/ypos \text{80 def} \% \text{base position of top address line}
/yinc \text{14 def} \% \text{spacing of main address}
/repeatproc {srfile length 1 sub runnum lt {/quickexit true def exit} if labelart logoandreturnaddress gsave srfile runnum get striplines grestore} def
/srfile [
(Aaron A. Aardvark
111 Ardmore Avenue
Artesia, AZ 11111)
(Boyd B. Benson
222 Baker Boulevard
Bismark, BC 2B2 2B2)
(Clyde C. Calhoun
333 Cumberland Circle
Chambers, CA 33333)
(Darryl. D. Doldrum
444 Dakota Drive
Dover, DE 44444)
(Eggbert E. Evans
555 Engle Easement
Euclid, Europe 55555)
(Frank F. Fairmont
666 Fogerty Flophouse
Fairview, FL 6666)
] def
/runnum \text{0 def}
/customdata true def
(shiplabel) stepandrepeat quit

**PostScript Operators**

- eq \text{red book page 407}
- ge \text{red book page 425}
- get \text{red book page 426}
- le \text{red book page 448}
- lt \text{red book page 451}
- ne \text{red book page 456}

**Gonzo Commands**

- calcpages \text{decides how many pages are needed when you are using a custom srfile.}

This allows early exits when or if you run out of data

censhow \text{a simple centering proc}
customdata \text{autopaginate custom data?}
numpages \text{default number to print}
quickexit \text{forces an early exit when you run out of names or reach a given ticket}
runnum \text{running pointer advances one per default short exit - don’t do it}
srfile \text{gets used for custom entries}
startnum \text{initial ticket number}
striplines \text{takes a multi-line string, extracts each individual line string, and then executes stripproc for each line. Useful for simple addressing and custom form filling-in.}
stripproc \text{needed by striplines}

**User Defined Procs**

- letterhead
- date
- body
- signature
- labelart
- logoandreturnaddress
- returnaddress

**On Your Own**

Collect flyers, graduation announcements, wedding announcements, cards, etc for quad- and tri-fold projects.

---

**SPINDLE/MUTILATE/TEAR RESOURCES**

<table>
<thead>
<tr>
<th><strong>ADI Machines</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI Machines</td>
</tr>
<tr>
<td>20505 E Valley Blvd, #104</td>
</tr>
<tr>
<td>Walnut, CA 91789</td>
</tr>
<tr>
<td>(714) 594-0097</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Panasonic</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Panasonic</td>
</tr>
<tr>
<td>Two Panasonic Way</td>
</tr>
<tr>
<td>Secaucus, NJ 07094</td>
</tr>
<tr>
<td>(201) 348-7000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Alvin</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alvin</td>
</tr>
<tr>
<td>PO Box 188</td>
</tr>
<tr>
<td>Windsor, CT 06095</td>
</tr>
<tr>
<td>(800) 243-0197</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Stanley-Bostitch</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanley-Bostitch</td>
</tr>
<tr>
<td>PO Box 188</td>
</tr>
<tr>
<td>Windsor, CT 02818</td>
</tr>
<tr>
<td>(401) 884-2500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Falcon Sales</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Falcon Sales</td>
</tr>
<tr>
<td>35420 Stanley Dr</td>
</tr>
<tr>
<td>Sterling Heights, MI 48077</td>
</tr>
<tr>
<td>(800) 243-1141</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Staplex</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Staplex</td>
</tr>
<tr>
<td>777 Fifth Avenue</td>
</tr>
<tr>
<td>Brooklyn, NY 11232</td>
</tr>
<tr>
<td>(800) 221-0822</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Maruzen</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Maruzen</td>
</tr>
<tr>
<td>917D N Plum Grove Rd</td>
</tr>
<tr>
<td>Schaumburg, IL 60173</td>
</tr>
<tr>
<td>(708) 240-1128</td>
</tr>
</tbody>
</table>

---

Beginning PostScript Projects
Thank you for the giant sea tortoise.
Quadfold and Trifold
Announcements or Flyers

A sample of three possible folding techniques – quadfolds, vertical trifolds, and horizontal trifolds – for announcements, flyers, brochures, greeting cards, thank you notes, etc using layout grids.

Quadfold Card
% Four panel foldover for announcements or thank you % notes
gutility begin
nuisance begin
printerror
gonzo begin
/showallgrids true def
% This magic proc takes four individual panels called % firstpanelproc through fourthpanelproc and arranges % them properly on the page ....
/quadfoldproc {
save /p1snap exch def 20 11 72 mul 2 div 20 add 10 setgrid
upside downright
showallgrids {27 36 showgrid} if
firstpanelproc clear p1snap restore
save /p1snap exch def 20 20 3 sub 10 setgrid
showallgrids {27 36 showgrid} if
secondpanelproc clear p1snap restore
save /p1snap exch def 8.5 72 mul 2 div 20 add 20 3 sub 10 setgrid
showallgrids {27 36 showgrid} if
thirdpanelproc clear p1snap restore
save /p1snap exch def 8.5 72 mul 2 div 20 add 11 72 mul 2 div 20 add 10 setgrid
upside downright
showallgrids {27 36 showgrid} if
fourthpanelproc clear p1snap restore
} def
/upside downright { 27 36 translate 180 rotate } def
% used by quadfoldproc
% here is a quick and simple border ...
/niceborder { 2 [0 20 0 36 27 36 27 0 0 0 20] roundpath
[0.8 0.6 1 0.4 0] superinsidestroke } def
% Here are four example panels ...
/firstpanelproc {
cstretch 0.02 def
/ssstretch 0.02 def
/font1 /Helvetica 0.85 gonzofont
/font2 /ZapfChancery-MediumItalic 3 gonzofont
niceborder 5 5 ( |1first panel ) cl
13.5 20 (|2Thank you ) cc
} def
/secondpanelproc {
cstretch 0.02 def
/ssstretch 0.02 def
/font1 /Helvetica 0.85 gonzofont
/font2 /ZapfChancery-MediumItalic 3 gonzofont
5 5 ( |1second panel ) cl } def
/thirdpanelproc {
cstretch 0.02 def
/ssstretch 0.02 def
/font1 /Helvetica 0.85 gonzofont
/font2 /ZapfChancery-MediumItalic 3 gonzofont
5 5 ( |1third panel ) cl
/yinc 4 def
13.5 20 (|2for the giant sea tortoise.) cc
} def
/fourthpanelproc {
cstretch 0.02 def
/ssstretch 0.02 def
/font1 /Helvetica 0.85 gonzofont
/font2 /ZapfChancery-MediumItalic 3 gonzofont
5 5 ( |1fourth panel ) cl } def
% and this does the whole job automatically
quadfoldproc
showpage
quit

Vertical Trifold Brochure
% Three or four panel info brochure
gutility begin
nuisance begin
printerror
gonzo begin
landscape % because we want vertical panels
/showallgrids true def
Horizontal Trifold Flyer
% Three panel brochure with center panel inverted ...
gutility begin
nuisance begin
printerror
gonzo begin
/showallgrids true def
/quadpixel {} def % cures a bug of mine
/showallgrids {gsave 0 setlinewidth 0 11 72 mul 2 mul 3 div moveto 1000 0 rlineto stroke 0 11 72 mul 1 mul 3 div moveto 1000 0 rlineto stroke grestore} if
/upsideuppanel { 57 23 translate 180 rotate } def % used by middle panel
% here are your three panels. Each is done separately and % positions itself automatically ...
/toppanelproc {
/cstretch 0.02 def
/sstretch 0.02 def
/font0 /Helvetica-Bold 1.2 gonzofont
/font1 /Helvetica 0.85 gonzofont
/font2 /Helvetica-Oblique 0.85 gonzofont
} def
/midpanelproc {
/cstretch 0.02 def
/sstretch 0.02 def
/font0 /Helvetica-Bold 1.2 gonzofont
/font1 /Helvetica 0.85 gonzofont
/font2 /Helvetica-Oblique 0.85 gonzofont
} def
/botpanelproc {
/cstretch 0.02 def
/sstretch 0.02 def
/font0 /Helvetica-Bold 1.2 gonzofont
/font1 /Helvetica 0.85 gonzofont
/font2 /Helvetica-Oblique 0.85 gonzofont
} def
/triplepanelbrooch {
save /p1snap exch def 20 11 72 mul 2 mul 3 div 20 add 3 sub 10 setgrid
/showallgrids {gsave 0 setlinewidth 57 23 showgrid} if
toppanelproc clear p1snap restore
save /p1snap exch def 20 11 72 mul 3 div 20 add 3 sub 10 setgrid
/upsideuppanel
/showallgrids {gsave 57 23 showgrid} if
midpanelproc clear p1snap restore
save /p1snap exch def 20 17 10 setgrid
showallgrids {57 23 showgrid} if
botpanelproc clear p1snap restore
} def

% this does all the work ....

triplepanelbrooch

showpage quit

Gonzo Commands

colcheck
panelset positions each panel in the vertical
tifold borchure trifold borchure
quadpixel refines the grid
upside downpanel inverts a panel

User Defined Procs

paneltex each panel of the vertical trifold
toppanelproc horizontal top panel
midpanelproc horizontal middle panel
botpanelproc horizontal bottom panel

On Your Own

Read Blue Book Chapter 5: Printing Text.

Collect and bring in short sayings for bumperstickers, T-shirts, and badges. Bring in a simple 3 - 4 line address, etc for a rubberstamp.
Arcjustify Badges

& Bumperstickers

Text can be made to follow any predetermined path. We are using a simple arc justification routine to create circle text for use in T-Shirts and badges. Astrobright peel and stick papers can be used to create custom bumperstickers in very small quantities. There are three different examples given using step and repeat routines.

Arcjustify & Badge

gutility begin
nuisance begin
printerror
gonzo begin
landscape % this one horizontal
/showthegrid false def
20 15 10 setgrid
/showthegrid {76 57 showgrid} if
% THE FIRE JACKET
% NOTE: in an arcjustify, the first number is the xcenter,
% the second is the ycenter, and the third is the radius.
% Use a negative radius if you want to be
% concave instead of convex.
/str ( ) def
% define a kerning character
/customkernchar (~) def
% special arc justify kern between
% characters
/customkern -0.25 def
/arckern 0.15 def % stretch characters
/font9 /Times-Bold 7 gonzofont
38 6 42 (THATCHER FIRE DEPT.) karctjustify

%THE BADGE
% Define a badge proc
/badge {gsave translate % move to desired center location
0 setlinewidth % draw an (optional) outside circle
0 0 17.2 0 360 arc stroke % for the Badge-A-Minit cutter
0 0 9.5 0 360 arc stroke % the actual badge diameter
line3
0 0 7.5 0 360 arc stroke % the outside printed circle
line1
0 0 5.7 0 360 arc % this is the inside gray
gsave bestgray lightgray % circular fill area
fill grestore
stroke
/arckern 0.03 def % slight character kerning
% and show it convex - top arc
/font0 /Helvetica-Bold 1.2 gonzofont
font0
0 0 6.2 (Schizophrenic’s International) karctjustify
% show it concave - bottom arc
/arckern 0.08 def
0 0 -7 (Convention 1989 - 1990) karctjustify
/cstretch 0.02 def /ssstretch 0.02 def
/font1 /Helvetica-Narrow-Bold [7 0 0 10 0 0] gonzofont
/font3 /Helvetica-Narrow-Bold 1.2 gonzofont
/font2 /Helvetica-Narrow 1 gonzofont
/yinc 2 def
0 setgray
0 3.1 (|3HI AND HELLO) cc
0 1.9 (|3OUR NAMES ARE) cc
0 -1.4 (|2and) cc
line1 % the two write-your-name lines

Assignments

12: Design a badge or other arcjustify project and at least one bumpersticker

Auto-name Badge Printing

gutility begin
nuisance begin
printerror
gonzo begin
% define a step and repeat
% proc for 12 badges per page
/stepnrptparams 40 dict def
stepnrptparams begin
/12badges [3 4 192.5 192.5 92.5 90 20 false false] def
end
% note that gonzo utilities dictionaries were opened before
% the stepandrepeat dictionary on the template file - be
% sure to append the auto label file to the end of the
% step and repeat template
% Here is some badge art...
/badge { save /snapb exch def
/showthegrid false def
20 15 10 setgrid
showthegrid {57 76 showgrid} if
0 setlinewidth
0 0 9.5 0 360 arc stroke % the actual badge diameter
line3
0 0 7.5 0 360 arc stroke % the outside printed circle
line1
0 0 5.7 0 360 arc % this is the inside area
stroke
% show it convex
/arckern 0.03 def % slight character kerning
/font0 /Helvetica-Bold 1.2 gonzofont
defont0
0 0 6.2 (Schizophrenic’s International) karcjustify
% show it concave
/arckern 0.08 def
0 0 -7 (Convention 1989 - 1990) karcjustify
/estretch 0.02 def /sstretch 0.02 def
/font1 /Helvetica-Narrow-Bold [7 0 0 10 0 0] gonzofont
defont1
/font3 /Helvetica-Narrow-Bold 1.2 gonzofont
defont3
/font2 /Helvetica-Narrow 1 gonzofont
defont2
/yinc 2 def
0 setgray
0 3.1 ([3HI AND HELLO) cc
0 1.9 ([3OUR NAMES ARE) cc
0 -1.4 (2and) cc
line1 % the two write-your-name lines
-4 -0.5 mt 8 r
-3 -3.5 mt 6 r
snapb restore } def
/font4 /Helvetica-Narrow-Bold 12 gonzofont
defont4
/xpos 20 def % left margin of main address
/ypos 11 def % base position of top address line
/yinc 14 def % spacing of main address
/repeatproc {
srfile length 1 sub runnum lt (/quickexit true def exit) if
  badge
gsave
  srfile runnum get striplines
  grestore
} def
/srfile [
(Aaron A. Aardvark)
(Boyd B. Benson)
(Clyde C. Calhoun)
(Darryl. D. Doldrum)
(Eggbert E. Evans)
]
(/Frank F. Fairmont)
(Gertrude G. Gregg)
(Harriet H. Helespont)
(Isolda I. Igg)
} def
/runnum 0 def
/customdata true def
(12badges) stepandrepeat
quit

Three Large Bumperstickers

UTILITY BEGIN
printerror
nuisance begin
gonzo begin
/showbumpgrid false def
% This makes sure your bumpersticker formats are in step
% and repeat ...
/stepnrptparams 40 dict def
stepnrptparams begin
/smallbumper [2 10 270 72 40 30 20 true false] def
/medbumper [1 5 585 144 20 30 20 true false] def
/bigbumpstick [1 3 792 195 0 15 40 true true] def
defend
/mybumper {
  20 7 10 setgrid
  showbumpgrid {75 18 showgrid} if
  /estretch 0.02 def /sstretch 0.2 def
  /kern 0.7 def % note |k gives 7 points of kerning
  /font1 /Palatino-BoldItalic [7.5 0 0 12 0 0] gonzofont
defont1
  37.5 5.5 (This T[koo Shall P[kass) cc
  0 0 75 18 1 roundbox line3 stroke } def
/repeatproc {mybumper} def
(bigbumpstick) stepandrepeat
quit

Five Medium Bumperstickers

UTILITY BEGIN
printerror
nuisance begin
gonzo begin
/showbumpgrid false def
/stepnrptparams 40 dict def
stepnrptparams begin
/smallbumper [2 10 270 72 40 30 20 true false] def
/medbumper [1 5 585 144 20 30 20 true false] def
/bigbumpstick [1 3 792 205 0 0 40 true true] def
defend
% here is your bumpersticker proc
This Too Shall Pass
This Too Shall Pass
This Too Shall Pass
I Hate BUMPERSTICKERS

I Hate BUMPERSTICKERS

I Hate BUMPERSTICKERS

I Hate BUMPERSTICKERS

I Hate BUMPERSTICKERS
/mybumber {} 0 1.5 10 setgrid showbumpgrid {57 14 showgrid} if
/cstretch 0.02 def /sstretch 0.02 def 
yinc 5 def
/font1 /Helvetica-Narrow-Bold 4 gonzofont
/font2 /AvantGarde-Demi 5 gonzofont

28 8 (|I Hate
[2BUMPERSTICKERS]) cc
bestgray
0.5 0.5 56 13 0.5 roundbox
[1 0 0.55 0.8] superstroke
} def
/repeatproc {mybumber} def
(medbumper) stepandrepeat
quit

Ten Small Bumperstickers

gutility begin
nuisance begin
printerror
gonzo begin
/showbumpgrid false def

/stepnruptparams 40 dict def
stepnruptparams begin
/babybumper [2 10 270 72 20 35 20 true false] def
/smallbumper [1 10 585 72 20 30 20 true false] def
/medbumper [1 5 585 144 20 30 20 true false] def
/bigbumpstick [1 3 792 205 0 0 40 true true] def
end

% here is your bumpersticker proc
/mybumber {} 0 0 10 setgrid
/showbumpgrid {56 6 showgrid} if
/cstretch 0.02 def /sstretch 0.02 def
/font1 /Helvetica-Narrow-Bold 4 gonzofont
/font2 /AvantGarde-Demi [4 0 0 5 0 0] gonzofont
font1
28 1.4 (Do Not Read This Sign!) cc
line1 0 0 mt 6 u 56 r 6 d 56 l
} def
/repeatproc {mybumber} def
(smallbumper) stepandrepeat
quit

New PostScript Commands

Gonzo Commands
arckern defines the amount of kern between arced characters
customkern defines additional kern needed between specific arced characters
customkernchar defines the character used to call the custom kern

carcjustify calls the arc justified text

New Stepandrepeat Commands
babybumper does 20 very small bumperstickers
bigbumpstick does 3 large bumperstickers
medbumper does 5 medium bumperstickers
smallbumper does 10 small bumperstickers
12badges does 12 badges per page

On Your Own
Do Blue Book page 163, Program 10/Circular Text and page 167, Program 11/Text Along an Arbitrary Path.
Kodar Kritch
Reverse Printing

Shows examples of font matrix change to print upside down, reversed for t-shirts and window decals, and negative (white on black) for rubberstamps and signs.

**Printing Upside Down**

posium begin
gonzo begin
oftware begin
 shine begin
tlinefalse def
landscape
20 20 10 setgrid
showthegrid {57 78 showgrid} if
/cstretch 0.02 def /sstretch 0.02 def
/font1 /ZapfChancery-MediumItalic 15 gonzofont
% a negative height upsidedowns the characters
/font2 /ZapfChancery-MediumItalic [15 0 0 -15 0 0] gonzofont

37.5 30 (|1Yodar Kritch) cc
37.5 30 (|2Yodar Kritch) cc
showpage quit

**Reverse Printing**

posium begin
gonzo begin
oftware begin
 shine begin
/shearfalse def
landscape
20 20 10 setgrid
showthegrid {57 78 showgrid} if
% a negative width reverses the lettering.
/cstretch 0.02 def /sstretch 0.02 def
/font0 /Helvetica-Narrow-Bold [-9 0 0 9 0 0] gonzofont
/yinc 12 def
font0
38.50 (MY OTHER BODY IS IN THE SHOP) cc
showpage quit

**Negative Printing**

% Makes up a master negative for rubber stamp production

185 470 10 setgrid
25 30 showgrid
% define and fill a black block
-1 -1 mt 0 32 rlineto 27 0 rlineto 0 -32 rlineto closepath
black fill
nuisance begin
negative % this makes black = white and vice versa
% a border
0.5 setlinewidth 1 setlinecap
0 0 mt 30 u 25 r 30 d 25 l stroke
/cstretch 0.05 def /sstretch 0.05 def
/font1 /Helvetica-Narrow-Bold 1.2 gonzofont
/font2 /ZapfChancery-MediumItalic 1.6 gonzofont
/font3 /ZapfChancery-MediumItalic 1.6 gonzofont
/font4 /Palatino-BoldItalic 0.8 gonzofont
/font5 /Palatino-BoldItalic 1 gonzofont
/font6 /Helvetica-Bold 1.4 gonzofont

ASSIGNMENT 13:
Design a T-shirt and rubberstamp or window decal.
My other body is in the shop.
T-SHIRT TRANSFER RESOURCES

**Black Lightning**  
Riddle Pond Road  
West Topsham, VT 05086  
(800) BLACK99

**Sax Arts & Crafts**  
PO Box 51710  
New Berlin WI 53151  
(800) 323-0338

**Norton Products**  
Box 2012  
New Rochelle, NY 10802

**Transfer Magic**  
130 Grand Street  
Carlstadt, NJ 07072  
(201) 935-9005

T-SHIRT TRANSFER RESOURCES

**Black Lightning**  
Riddle Pond Road  
West Topsham, VT 05086  
(800) BLACK99

**Sax Arts & Crafts**  
PO Box 51710  
New Berlin WI 53151  
(800) 323-0338

**Norton Products**  
Box 2012  
New Rochelle, NY 10802

**Transfer Magic**  
130 Grand Street  
Carlstadt, NJ 07072  
(201) 935-9005

TRADITIONAL PRINTING RESOURCES

**Grantham/Polly-Stamp**  
418 Central Avenue NE  
East Grand Forks, MI 56721  
(218) 773-0331

**Merigraph/RA Stewart**  
641 South Palm Unit H  
La Habra, CA 90631  
(201) 272-2000

**JLS Rubber Stamps**  
PO Box 4418  
Redding CA 96099  
(916) 246-4393

**Niji StampLand**  
235 Valley Drive  
Brisbane, CA 94005  
(415) 468-7066

**Justrite/Louis Melind**  
5240 Alhambra Avenue  
Los Angeles, CA 90032  
(213) 225-5683

**Stamp Magic Systems**  
6012 Jet Port Industrial Blvd  
Tampa, FL 33623  
(800) 237-9106

**M&R Marking Systems**  
69 Myrtle Street  
Cranford, NJ 07016  
(201) 272-2000

Gonzo Commands

- **backwards**: automatically prints backwards
- **black**: sets a black area or line
- **blackflash**: prints an entire black page
- **negative**: makes black = white and vice versa
- **white**: sets a white area or line

On Your Own

Read Blue Book Chapter 9: More Fonts  
Collect or design awards, certificates or flyers and simple borders.
Jocelyn Susan Nelson
Dorothea B Riley
2504 16th Street
Wittford, AZ 85964

photo by
Diane Dobbins

Econobuy Food and Drug

Without prejudice U.C.C. 1-207

Without prejudice AZ 47-2214

Without prejudice U.C.C. 1-207

Without prejudice AZ 47-2214
Awards, Certificates, Flyers and Other Borders

Awards, certificates, coupons, flyers are just a few of the places where fancy borders are used. Here are a few we have come up with.

**Scroll Bordered Certificate**

% A scroll bordered certificate that looks good on parchment
gutility begin
nuisance begin
printerror
gonzo begin
landscape
/showfullgrid false def
20 20 10 setgrid
/showfullgrid {76 57 showgrid} if
% here are some routines that draw a full sized looping % scroll border
/bline {gsave -0.5 0 moveto 0.3 -0.6 0.7 -0.6 1 0 rcurveto stroke grestore} def
/bloop {gsave -0.5 0 moveto -1 3 1 3 0 rcurveto stroke grestore} def
/bcorner {gsave 0 -0.8 moveto 0.4 0.15 0.1 0.6 0.8 rcurveto -3.8 1.8 -1.8 3.8 0 0 rcurveto 0.2 -0.1 0.65 -0.4 0.8 0 rcurveto stroke grestore} def
/drawscrollborder {borderlinewidth borderthickness div 0.37 div setlinewidth gsave translate borderthickness .37 mul dup scale 2.25 hloops 1 sub 0.0 mul add 3.853 vloops 1 sub 1 mul add translate bcorner 1.3 0 translate hloops 1 sub {bloop bline 1 0 translate} repeat bloop 0.3 0 translate -90 rotate bcorner 1.3 0 translate vloops 1 sub {bloop bline 1 0 translate} repeat bloop 0.3 0 translate -90 rotate bcorner 1.3 0 translate hloops 1 sub {bloop bline 1 0 translate} repeat bloop 0.3 0 translate -90 rotate bcorner 1.3 0 translate vloops 1 sub {bloop bline 1 0 translate} repeat bloop grestore} def
% and here is how you use the border
/borderthickness 5.0 def /borderlinewidth 0.1 def
/hloops 34 def /vloops 25 def
1.5 0 drawscrollborder
% some text
/ssstretch 0.3 def /cstretch 0.02 def
/font0 /Palatino-BoldItalic 3 gonzofont
/font1 /ZapfChancery-MediumItalic [5 0 0 5 0 0] gonzofont
/yinc 4 def
font0
37.5 40 (This certificate of appreciation is reluctantly awarded to …) cc
font1
37.5 28 (Joni L. Greebleknocker) cc
% this is the "sign here" line
line1
17 26 mt 42 r stroke
% and here’s the rest of the message ...
font0
37.5 20 (for no apparent reason at all. She certainly did nothing whatsoever to deserve it.) cc
/showpage quit

A Simple Motto

% A simple notice or motto % having a single line art % deco border.
gutility begin
nuisance begin
printerror
gonzo begin
landscape
/showfullgrid true def
20 20 10 setgrid
/showfullgrid {76 57 showgrid} if
/font1 /Bookman-Demi [5.5 0 0 7 0 0] gonzofont % fairly compressed font
/yinc 10 def
38 36.5 (|You may touch the dust … but please don’t write in it!) cc
% a \274 is a three dot ... ellipsis
% The fancy border just consists of two rotated and % repeated graphic elements ...
decoleft {gsave black 0 0 mt 3 0 rlineto 0 3 rlineto -3 0 rlineto closepath fill

**ASSIGNMENT 14:**

Use any border to create an award, flyer, coupon or other project.
This certificate of appreciation is reluctantly awarded to ... 

Joni L. Greebleknocker

for no apparent reason at all. She certainly did nothing whatsoever to deserve it.
Rope-a-Dope
% Does a fancy rope border that actually ties itself into
% a knot!

SIGNs, DISPLAYs, ARChIVAL RESOURCES

Fombeords
2211 North Elston Avenue
Chicago, IL 60614
(800) 362-6267

Gaylord
Box 4901
Syracuse, NY 13221
(800) 448-6160

Graphic Dimensions Ltd
41-23 Haight Street
Flushing, NY 11355
(800) 221-0262

Light Impressions
439 Monroe Avenue
Rochester, NY 14607
(800) 828-6216

Rope-a-Dope
% Does a fancy rope border that actually ties itself into
% a knot!
You may touch the dust ... but please don’t write in it!
hsquareknot {save /squaresnap exch def currentpoint /yhold exch def /xhold exch def rsqloop1 drawarope save /sssnap exch def xhold 31.5 pdis mul add yhold moveto lsqloop1 drawarope clear ssnap restore /showsegrule {aposn 1 add dup 8 ge exch 10 le and aposn 1 add dup 23 ge exch 29 le and or aposn 1 add dup 42 ge exch 45 le and or} def xhold yhold moveto rsqloop1 drawarope clear ssnap restore} def

% border with square knot demo - convert for use.

106 45 {dup mul exch dup mul add 1.0 exch sub} setscreen
bestgray
0.8 setgray % only for subtle fuzzy effect shown here; % otherwise 0 setgray
-1.5 -1.7 translate % fudge factor (temporary)
% special adjustment to position
10 10 translate 0.6 dup scale
/pdis 1.3 def
/zdis 2.6 def
10 80 moveto hsquareknot
% to change the number of twists in the rope, the number % before {-90} repeat is down the left side. The number % before {0} repeat is to the bottom right. The number % before {90} repeat is up the right side. The number % before {180} repeat closes the top from the right.
% the other unrepeated numbers in the array are simply the % direction the next turn in the rope is to go.
10 pdis 2.5 mul add 80 moveto [180 180 180 180 -165 -150 -135 -120 -105 -90 50 [-90] repeat -90 -75 -60 -45 -30 -15 0 0 0 40 {0} repeat 15 30 45 60 75 90 50 {90} repeat 90 105 120 135 150 165 180 12 {180} repeat] drawarope

%%% END ROPE BORDER %%%

showpage quit

A Coupon
% A small art deco border for a coupon or gift certificate.

gonzo begin
gutility begin
nuisance begin
/logo {save
/snap1 exch def
% your curvetraced logo goes here
snap1 restore } def
%///// COUPON /////
200 200 10 setgrid
% 36 29 showgrid
gsave
2 12 translate
0.4 setlinewidth 2 setlinecap 0 setlinejoin
0 0 mt 32 r 15 u 32 l 15 d
0.2 setlinewidth 2 setlinecap 0 setlinejoin
-2 -2 mt 36 r 19 u 36 l 19 d
-2 -2.2 mt 9.8 d 36 r 9.8 u
/corner {
gsave
newpath 0.3 setlinewidth 0 setlinejoin 2 setlinecap
-1.9 3.6 mt 1.5 1.5 rl 0 -2.5 rl 3 0 rl -3 rl 2.5 0 rl
-1.5 -1.5 rl -2.5 0 rl 0 3 rl -3 0 rl 0 2.5 rl closepath
gsave white stroke grestore
black fill
grestore
} def
corner
gsave -1 1 scale -32 0 translate corner grestore
gsave 1 -1 scale 0 -15 translate corner grestore
gsave -1 -1 scale -32 -15 translate corner grestore
grestore
null
Bring this $5.00 coupon to COTTON CLOUDS, booth #1302 & save $5.00 on any yarn purchase.

- Cotton Yarns • Rayon Chenille •
- Cotton/Rayon Variegated Bouclés •

1-800-322-7888
Cotton Clouds 5176 S. 14th Ave, Safford, AZ 85546, USA
<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knockwurst Ala King</td>
<td>$3.27</td>
</tr>
<tr>
<td>Hot Clam Sundae</td>
<td>$1.15</td>
</tr>
<tr>
<td>Coconut Anchovy Pizza</td>
<td>$8.35</td>
</tr>
<tr>
<td>Butterscotch Pudding <strong>au gratin</strong></td>
<td>$1.25</td>
</tr>
<tr>
<td>Pastrami &amp; Kiwi Blintz</td>
<td>$4.50</td>
</tr>
<tr>
<td>Sopapillas, Enchilada Style</td>
<td>$2.46</td>
</tr>
<tr>
<td>Grape Chili <strong>au jus</strong></td>
<td>$4.45</td>
</tr>
<tr>
<td>Avocado &amp; Liver Surprise</td>
<td>.15</td>
</tr>
<tr>
<td>Chocolate Sushi</td>
<td>$25.50</td>
</tr>
</tbody>
</table>
Menu Justification, Supertabbing

 Fancy justification is needed for menu justifying with simple dotted lines, and for more complex justification/font changes in columnar layouts.

**Menu Justify**

% Gives you a fast and accurate menu ............ justify
% Also includes a simple but stunning border.
% Here are two needed routines for the menu justify below

%%% WIDTH CALCULATIONS %%%

% A new curwide variable lets you calculate the gonzo
% string width independent of printing it. Handy for
% menu justify, etc..
% Use has to be bracketed with /oktoprint false def
% /oktoadvance false def --- stuff --- and then true.
% Works with any justify mode.
% make this change to main code so lengths can be
% extracted. Note that roomleft gets flushed at line end.

gonzo begin
/endtheline (/curwide txtwide roomleft sub
def justx cvx exec oktoprint {printline} if) bind def
% calloutwidth calculates only the width
% of the string, returning it to the top of
% the stack. For consistency with other
% callouts, use 0 0 (string) cw. This
% INCLUDES all stretching and font
% changes!
/cw {save /snapc1 exch def /oktoprint false def
/oktoadvance false def /linestring linestring2 def /justx
(justL) def 3 1 roll /ypos exch def /xpos exch def
stringgonzo curwide snapc1 restore} def end

%%% MENU JUSTIFY %%%

% The cm routine takes a leading string and a trailing string
% separated by a delimiter and places a row of
% CONSTANT WIDTH, PRECISELY SPACED,
% VERTICALLY ALIGNED and WHOLE dots (or
% whatever) between the two. It is based on putting down
% the dots first and ERASING only whole dots that are
% not wanted.
% This has been modified specially for a 10X grid

 ASSIGNMENT 15: Design a menu or other form using either supertab or menu justification routines.
printerror
nuisance begin
gonzo begin
/showthegrid false def
20 20 10 setgrid
showthegrid \{50 50 showgrid\} if
% title fill goes UNDER border
9 41 mt 5 pu 29 pr 5 pd closepath
bestgray lightgray fill black
% This is the "magic" border. For other effects, change the
% roundpath radius to 0.01, 0.8, 1, 1.5, 2.5, 3.5, 5 or 8. Or
% -2 or -6 (!) DO NOT USE 3.0. Use 2.99 instead.
1 setlinejoin 1 setlinecap
2.99 \[20 12 6 12 6 9 9 49 6 49 6 46 41 46 41 49 38
49 38 9 41 9 41 12 20 12\] roundpath closepath
[0.8 0 0.5 1 0.28 0 0.1 1] superstroke
% this draws the title bar and autoerases the ends
0 setlinejoin 0 setlinecap
0.01 \[ 9 42 9 41 38 41 38 42 38 40 38 41 9 41 9 40\] roundpath closepath
[0.8 0 0.5 1 0.28 0 0.1 1] superstroke
% set up the menu stuff ...
/txtwide 22 def % textwidth
/cstretch 0.01 def
/sstretch 0.05 def
/yinc 2.6 def
/font1 /Palatino-Roman 1.2 gonzofont
/font2 /Palatino-Italic 1.2 gonzofont
/font3 /ZapfChancery-MediumItalic 2.5 gonzofont
% Trick to cure Mitzi’s anorexism ...
font3
23.5 42.7 (Mitzi’s Yuppy Fare) cc
23.5 42.75 (Mitzi’s Yuppy Fare) cc
23.5 42.8 (Mitzi’s Yuppy Fare) cc
font1
12.5 36.5 (Knockwurst Ala King                   $3.27
Hot Clam Sundae $1.15
Coconut Anchovy Pizza $8.35
Butterscotch Pudding au gratin $1.25
Pastrami & Kiwi Blintz $4.50
Sopapillas, Enchilada Style $2.46
Grape Chili au jus $4.45
Avocado & Liver Surprise $1.15
Chocolate Sushi $25.50 cm

showpage quit

**Supertabs**
% Lets you do fancy columns of data in which you can mix
% or match fonts and justification modes.
% supertab (or cst) for short is an enhancement to the
% gonzo justification routines that let you mix and
% match justification modes and fonts in tabular data.
% For instance, your first column could be left justified
% bold Helvetica. Your second column can be centered
% Revue with extra kerning. Your third column could be
% right justified dollars and cents, and so on. You can
% even have multiple lines within each individual tab
% position, and ordinary multiple-spaced data can be used
% as tab delimiters.
% One MAJOR use is to convert arbitrary fixed pitch tables
% from ANY source into fully justified text. Quickly,
% invisibly and automatically.
% The format is as follows:
% xpos ypos (-col1- delim -col2- delim .... -coln-
% -col1- delim -col2- delim .... -coln-
% .................
% -col1- delim -col2- delim .... -coln- ) cst
% A supertab list has to be created first, of form
% /stabs \[ [-xoffset1- {justmode1} {customproc1}]\]
% \[ [-xoffset2- {justmode2} {customproc2}]\]
% \[ .................\]
% \[ [-xoffsetn- {justmoden} {customprocn}] \] def
% The xoffset value is how far over from the current xpos
% left margin the justify action is to take place. Note that
% this is the LEFT edge for left or fill justify, the CENTER
% for center justify, or the RIGHT edge for right justify.
% The justmode value is the callout justification mode that
% is to be used for the tabbed substring. Use cl = callout
% left, cc = callout center, cr = callout right, cf = callout
% fill, etc.
% The customproc is anything you want done immediately
% before showing the tab substring. The most common use
% would be changing a font as in \{font6\}, etc.
% Any number of EXTRA spaces can exist between input
% tab entries if the -cropleadingspaces- boolean shown
% below is true.
% Note: For intentionally blank lines, be sure to include a
% few spaces. If your -stabdelimiter- is two spaces, be sure
% to ALWAYS have at least two spaces between columns
% and NEVER double space within a column.
% stabdelimiter decides when to move you over to the next
% column. Popular values would be two or more spaces for
% dealing with imported listings, or an actual [I] tab
/%stabdelimiter ( ) def % two spaces means to tab over
% stabnewline decides when to move you down to the next
% row. While a simple carriage return works well,
% something fancier is needed if the individual tabs also
% have carriage returns in them.
/%stabnewline (in) def% return means start next line
% cropleadingspaces is handy whenever you are adapting
% preformatted fixed pitch data to full proportional
% PostScript ...
% inactive as default
/cropleadingspaces false def

% ytabadjust provides for extra room between lines on
% multiple line tab entries when needed ...
/ytabadjust 0 def % default

% cst is the main callout supertab routine. It grabs a long
% string and breaks that string down into individual tab
% elements and then formats them.
/cst {/cstabstring exch def /ypos exch def /xpos exch def
  mark {cstabstring stabnewline search {exch pop exch
  /cstabstring exch def dup length 1 le {pop} if} {dup length
  0 le {pop} if exit} ifelse} loop} save /cstsnap exch def
/cstline exch def 0 1 cstline length 1 sub {cstline exch get
/cstchomp exch def mark {cstchomp stabdelimiter search
  {exch pop exch /cstchomp exch def dup length 0 le {pop}
  if} {dup length 1 le {pop} if exit} ifelse} loop} /cstlarray
  exch def 0 1 cstlarray length 1 sub { /posn exch def save
  /ctsnap exch def stabs dup length 1 sub dup posn ge {pop
  posn} if get 0 get xpos add ypos stabs posn get 2 get cvx
  exec cstarray possn get spchomp stabs posn get 1 get cvx
cvexec ctsnap restore} for /ypos ypos yinc sub ytabadjust sub
def} for ypos cstsnap restore /ypos exch def}

% spchomp takes a string and conditionally removes all
% leading spaces, returning the truncated string
/spchomp {cropleadingspaces {dup 0 exch {32 eq {1
  add}{exit} ifelse} forall exch dup length 2 index sub 3 -1
  roll exch getinterval} if} def

% ///// demos - remove or alter before reuse. /////
gutility begin
printerror
nuisance begin
gonzo begin

/font1 /Helvetica 9 gonzofont
/font2 /Helvetica-BoldOblique 9 gonzofont
/font3 /Helvetica-Bold 9 gonzofont

% this first demo combines left, center and right
% justifications...

% create a supertab dictionary. each element of form
% [ -xpos- {justmode} {otherstuff} ]
/stabs
[
[0 {cl} {font1}]
[140 {cc} {font2}]
[250 {cr} {font3}]
] def

% marker to move to next tab position
/stabdelimiter ( ) def
% marker to move to next line
/stabnewline (\n) def
% get rid of leading spaces?
cropleadingspaces true def

% The data below is of form  xpos ypos (tabbed data) cst.
% Be sure to use several spaces on any intentionally blank
% lines and to provide a minimum of TWO spaces between
% columns...

200 400 (Aaron A Aardvark member $123.45
Boyd B. Bismark associate $92384.22
Clarence C. Calhoun director $5.83
Doyle D. Dogg student $29.86) cst

% this second demo shows how to use multi row tabular
% callouts ...

85 110 330 111 4 roundbox % define a rounded border
2 setlinewidth stroke % and draw it
0.5 setlinewidth % add a double hairline
0 92 mt 330 r
0 94 mt 330 r
/stabs % define your supertabs array
[
[0 {cl} {font1}]
[150 {cc} {font2}]
[300 {cr} {font3 /yinc 15 def}] % pros do lots of stuff
] def

/stabdelimiter ($$\n) def % goes between entries
/stabnewline (!!!\n) def % goes between lines
cropleadingspaces true def % get rid of leading spaces
/ytabadjust 30 def % extra spacing adjust

15 75 (This first multiple
line [?submessage?] is
flush left and font1
$$
centered stuff #1
$$
Right #1 has been
spaced weirdly
!!
This second multiple
line [?submessage?] is
flush left and font1
$$
centered stuff #2
$$
Right #2 has been
spaced weirdly) cst

% This one shows you how to center all of the data in all of
% the columns and gives you one way to handle any
% headers ...
Example 1 – left, center & right justify

<table>
<thead>
<tr>
<th>WHOOPIE</th>
<th>GOLLY</th>
<th>WOWIEGEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>This first multiple line <strong>submessage</strong> is flush left and font1</td>
<td><em>centered stuff #1</em></td>
<td>Right #1 has been spaced weirdly</td>
</tr>
<tr>
<td>This second multiple line <strong>submessage</strong> is flush left and font1</td>
<td><em>centered stuff #2</em></td>
<td>Right #2 has been spaced weirdly</td>
</tr>
</tbody>
</table>
/stabdelimiter () def
/stabnewline (\n) def

font1
15 98 (WHOOPIE GOLLY WOWIEGEE) cst

showpage quit

Gonzo Commands

MENU JUSTIFY

cm menujustify - accepts tabbed string pairs, converts into dotted menu listings
cropleadingspaces boolean - useful when adapting preformatted fixed pitch data to full proportional PostScript
curwide calculates the gonzo string width independent of printing it
cw calloutwidth - calculates only the width of the string, returning it to the top of the stack.
drawmdots draws a line of menu dots, using show command
mdoteht height of dot used
menudots character used for dots
menufont default dot font
menudelim delimiter used for dots
mlineproc processes the mline string into a leading string, the dot row and a trailing string, erases unused dots.
spchomp takes a string and conditionally removes all leading spaces, returning the truncated string

SUPERTABS

cst main callout supertab routine. grabs a long string and breaks it into individual tab elements and formats them.
stabdelimiter decides when to move over to the next column. Popular values would be two or more spaces for dealing with imported listings or an actual [I] tab

On Your Own

Collect and bring examples of outlined fonts or areas filled with words, other characters, or graphics.

<table>
<thead>
<tr>
<th>MENU COVER RESOURCES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>American Thermoplastic Co</strong></td>
</tr>
<tr>
<td>622 Second Avenue</td>
</tr>
<tr>
<td>Pittsburgh, PA 15219</td>
</tr>
<tr>
<td>(800) 245-6600</td>
</tr>
<tr>
<td><strong>Menu Graphics</strong></td>
</tr>
<tr>
<td>3133 Chester Avenue</td>
</tr>
<tr>
<td>Cleveland, OH 44114</td>
</tr>
<tr>
<td>(80) 321-9482</td>
</tr>
<tr>
<td><strong>H. Risch, Inc</strong></td>
</tr>
<tr>
<td>PO Box 567</td>
</tr>
<tr>
<td>Rochester, NY 14602</td>
</tr>
<tr>
<td>(716) 232-6938</td>
</tr>
</tbody>
</table>
Large Letters, Clipping, and Font Path Capturing

There are times when single letters are needed for stencils, posters, etc., or when character paths can be used for logos, letterheads, or as outlined letters to be used as a path for a repeated word or group of words. Patterns can also be repeated and clipped inside a curvetraced element of design and dashes can follow a line or curvetraced path for cutout characters.

**Big Stencils**

% Prints a message as one-per-page large outline letters for % use in stencils, floats, storefront signs, houseletters, etc.

```
gutility begin             % turn on the gonzo utilities
nuisance begin             % turn on the nuisance commands
printerror                 % turn on the printing error trapper

% This strange and wondrous proc does all the work ...
/makebigletters {fonttouse findfont [letterheight 0 0
letterwidth 0 0] makefont setfont {str exch 0 exch put str
dup stringwidth pop 8.5 72 mul exch sub 2 div 11 72 mul
letterheight 0.85 mul sub moveto false charpath stroke
showpage} forall} def

/str (X) def   % a single character string
% From top of "W" to bottom of "g"
/letterheight 9 inches def % Usually same as letterheight
/letterwidth 9 inches def

% watch the spelling!
/fonttouse {/NewCenturySchlbk-Bold} def
(BIG) makebigletters        % your message
quit
```

**Capturing a Font Character Path**

% There are lots of exciting and unique new features to % PostScript Level 2 that makes this a "must have" % upgrade. But by far the most useful (and long overdue) % feature of Level 2 is that nearly all font paths are fully % open and unlocked. And thus easily captured.

% Font paths are simply and quickly grabbable on the fly. % Besides the obvious use of creating outline characters, % you can use captured font paths for composite % characters, to eliminate the need for font downloads on a % logo or letterhead, to do the non-linear transformations

% needed for perspective, 3-D, star wars and other % distorted lettering, and to capture letters for vinyl sign % cutters and wood routers.

% Because of the **invalidaccess** lockout on earlier % versions of Post Script, earlier font path grabbing was % complex and tricky. With Level 2, you simply grab the % path on the fly.

% Here are the bare fundamentals of font path grabbing; % see the GEnie PSRT library files for detailed examples % of the incredible stuff you can do with grabbed paths.

% THE ROUTINES SHOWN HERE WORK ONLY ON % POSTSCRIPT LEVEL TWO. But you can easily take % the resultant paths and use them on any genuine % Adobe PostScript printer of any level.

% First, define your font. If you are going to work with the % usual font dictionaries, you may want to make it 1000 % points high. Otherwise, pick your final size...

/NewCenturySchlbk-BoldItalic findfont [500 0 0 450 0 0] makefont setfont

% Then select a letter and just % grab your fontpath...
0 0 moveto (K) false charpath

% This now creates an % unlocked outline path. If % you want to, you can % combine the path with % other design elements.

% Here is how to convert the path to an array. An array is % chosen because it is easier to report back to your host.
mark
{/moveto cvx}
{/lineto cvx}
{/curveto cvx}
{/closepath cvx}
pathforall
] /outlinepath exch def

% And it is that simple! If you want to do non-linear % transformations or otherwise leap tall buildings in a % single bound, you just alter the four procs needed by % pathforall.

% Here is how you report your path back to your host for % permanent recording. Naturally, you ARE using full % two-way host recordable comm or you absolutely,
CAPTURING FONT PATHS ON THE MAC

1 open Word or Works
   open fnptpath2.ps
   make changes to font and desired letter(s)
   saveas fnptpath2.ps?
   close file
   (it is not necessary to close program on System 7)

2 open LaserWriter Utility
   pull down on Utilities to:
   Download PostScript File
   open (double click) fnptpath2.ps?
   save as PostScript Log.1

3 open PostScript Log.1 in Word/Works
   "change" control characters (squares) to spaces
   define your file
   call your file
   add gonzo opening stuff and showpage quit
   saveas yournewfilename
   add gonzo opening stuff and showpage quit
   saveas yournewfilename
   print using font downloader
   (it is not necessary, emphatically should not be here.

% positively, emphatically should not be here.
/stall {37 sin pop repeat} def % useful delay proc
2500 stall % optional host turnaround delay
outlinepath {== flush 50 stall } forall % optional internal character delay
% Here is your expected host returned data for our example...
% 160.0 -0.119995 moveto
% 165.5  23.73 lineto
% 155.5  23.73 lineto
% 140.0  23.73 118.0 20.58 118.0 40.83 curveto
% 118.0  46.23 121.0 53.43 122.5 58.38 curveto
% 137.5 109.23 lineto
% 181.0  145.23 lineto
% 232.999 45.78 lineto
% 234.499 43.08 237.999 35.88 237.999 33.18 curveto
% 237.999 30.03 233.999 25.08 230.499 24.18 curveto
% 217.999 20.50 22.83 191.5 23.73 curveto
% 187.0 -0.119995 lineto
% 373.999 -0.119995 lineto
% 378.499  23.73 lineto
% 369.499  23.73 lineto
% 342.499  23.73 332.999  25.53 321.499  48.48 curveto
% 244.999 196.98 lineto
% 332.999 268.98 lineto
% 346.999 280.23 353.999 286.08 371.499 292.83 curveto
% 383.999 297.78 397.999 300.93 411.499 300.93 curveto
% 415.999 324.78 lineto
% 267.999 324.78 lineto
% 263.499 300.93 lineto
% 271.499 300.93 lineto
% 281.499 301.38 301.999 303.18 301.999 289.68 curveto
% 301.999 283.83 300.499 289.68 curveto
% 287.499 269.88 lineto
% 152.0  155.58 lineto
% 151.0  156.48 lineto

% Here is your expected host returned data for our example...
% 161.0 -0.119995 moveto
% 165.5  23.73 lineto
% 155.5  23.73 lineto
% 140.0  23.73 118.0 20.58 118.0 40.83 curveto
% 118.0  46.23 121.0 53.43 122.5 58.38 curveto
% 137.5 109.23 lineto
% 181.0  145.23 lineto
% 232.999 45.78 lineto
% 234.499 43.08 237.999 35.88 237.999 33.18 curveto
% 237.999 30.03 233.999 25.08 230.499 24.18 curveto
% 217.999 20.50 22.83 191.5 23.73 curveto
% 187.0 -0.119995 lineto
% 373.999 -0.119995 lineto
% 378.499  23.73 lineto
% 369.499  23.73 lineto
% 342.499  23.73 332.999  25.53 321.499  48.48 curveto
% 244.999 196.98 lineto
% 332.999 268.98 lineto
% 346.999 280.23 353.999 286.08 371.499 292.83 curveto
% 383.999 297.78 397.999 300.93 411.499 300.93 curveto
% 415.999 324.78 lineto
% 267.999 324.78 lineto
% 263.499 300.93 lineto
% 271.499 300.93 lineto
% 281.499 301.38 301.999 303.18 301.999 289.68 curveto
% 301.999 283.83 300.499 289.68 curveto
% 287.499 269.88 lineto
% 152.0  155.58 lineto
% 151.0  156.48 lineto
%

% CLIPPING IN A CHARACTER PATH
% Uses the captured character path, strokes and clips % around it leaving word(s) inside the outline.
Clipping in a Curvetrace

% To place a simple pattern inside a portion of curvetraced art draw your path then clip it. Note the path and the clip ALWAYS come before the pattern.
letterheight  height in inches for big stencil letters
letterwidth  width in inches for big stencil letters -
            usually the same as the height
makebigletters command to actually create separate
            letters on separate sheets of paper
outlinepath  internal delay character
stall        delay procedure
str           single character string

On Your Own
Find samples of sequentially numbered tickets, tags, etc.
Do: Filling an Area with a Pattern, Program 15, pg 187;
    Making a Poster, Program 13, pg 179; and Making an
    Outline Font, Program 16, pg 199 in the Blue Book.
DO NOT DISTURB!

DO NOT DISTURB!
Tags, Tear-offs, Tents, & Tickets

There is frequent need for autonumbered tickets, tags, tents and a way to tear-off a phone number from a notice or poster. The doorknob hanger prints a pair of identical tags using the BLANKS USA die cut forms. The tear-off does a notice for a supermarket bulletin board with "ripoff" phone numbers to take home. The tent can be used either on a restaurant table or, as here, for the top of a display package. The autonumbered tickets also use a BLANKS USA die cut form of their 4-up standard ticket.

Doorknob hanger

```postscript
590 790 translate -180 rotate /showthegrid false def /stepnrptparams 10 dict def
stepnrptparams begin /twodoorhangers [2 1 285 750 22 18 10 false false] def end

% here is the message we want repeated
/repeatproc { gsave 0 0 10 setgrid showthegrid {25 52 showgrid} if
/font0 /AvantGarde-Demi [6 0 0 5.5 0 0] gonzofont /yinc 6.1 def
12.5 42 (DO NOT DISTURB!) cc

% fonts for faces
/font1 /Symbol [2 0 0 2 0 0] gonzofont
/font2 /Symbol [2.25 0 0 2 0 0] gonzofont
/font3 /ZapfDingbats [22 0 0 22 0 0] gonzofont
/font4 /Symbol [2 -2 0 2.5 0 0] gonzofont
```

% note two spaces between eyes
% two returns between eyes and mouth
% six returns between mouth and outside sun
% necessary for correct positioning

```
sadsun {
/yinc 0.9 def
0 0 {1\267 \267 |2\307 |3b} cc
} def
```

```
gsave 12.5 18.5 translate sadsun grestore
```

Tear-off Notice

```postscript
590 790 translate -180 rotate /showthegrid false def % print feet first
/showthegrid false def % show the layout grid?
% manual % optional manual feed
(twodoorhangers) stepandrepeat quit
```

ASSIGNMENT 17:
Design a tent, ticket, tag or tearoff notice.

```postscript
nuisance begin % turn on the nuisance commands
printerror % turn on the printing error trapper
gonzo begin % turn on the gonzo justification
```

% note that message is INSIDE the box grid

```
gfont 6.1 def
12.5 42 (DO NOT DISTURB!) cc
```

% fonts for faces

```
/font1 /Helvetica-Bold 4 gonzofont
/font2 /Helvetica-Bold 2.5 gonzofont
```

```
0 20 55 55 2 roundbox
[1.5 0 1.2 1 0.7 0] superinsidestroke
```

```postscript
```
```
Beginning PostScript Projects 95
TIRED OF DORM LIFE?

Non-Smoking Roommate Wanted

Share an apartment with a working WMNU student

Three miles from campus

$250 PER MONTH (utilities, cable included)

Available January 1

call Cindy 797-4244
% trick to remember centering

font1
cen 44 (TIRED OF DORM LIFE?) cc

font2
cen 37 (Non-Smoking Roommate Wanted)

Share an apartment with a working WMNU student
Three miles from campus

|1$250  PER MONTH|2
| utilities, cable included
Available January 1
call Cindy 797-4244) cc

% get out of box for rest of sheet
grestore

line1
0 18 [3 2] 1.5 setdash mt 58 r % a dashed cross line
% x y [#on [#off] dashed line length
/ripoff { save /snaptik exch def

% We have to modify the step and repeat dictionary for the
% ripoff entry...

/stepnrptparams 10 dict def
stepnrptparams begin
/12tabs [1 1 5 0 1.5 3 10 false false] def
end

/repeatproc {gsave
-90 rotate -18 0 translate
3 3.5 [2 1] 0 setdash mt 21 r

/font3 /Helvetica-Bold 1.5 gonzofont
/yinc 2 def
4.5 1.5 (|Share an apartment Cindy 779-4424) cl
grestore} def

(12tabs) stepandrepeat
snaptik
restore} def

ripoff % make the tearoff tabs and print the page
quit

Tent
gonzo begin
gutility begin
printerror
nuisance begin

% tentproc - simplifies layout of back-to-back
% lettering as used on an announcement, point

% of purchase ad, or sealing product label.
% predefine a frontproc and a rearproc. Then enter
% tentproc with -lowerleft- -lowerbot- -gridresolution-
% -xgridwidth- -ygridheight- -showtick- -showgrid-
% tentproc

/tentproc { /showtheticks exch def /showthegrid exch def
/ygridheight exch def /xgridwidth exch def /gridresolution exch def
/lowerbot exch def /lowerleft exch def save /snaptik exch def
gsave lowerleft lowerbot translate 0 0 moveto gridresolution dup dup setgrid
showthegrid { xgridwidth ygridheight showgrid} if
showtheticks { gsave 0 setlinewidth -1 -2 moveto 0 2 rlineto
-2 -1 moveto 2 0 rlineto stroke
xgridwidth 1 add -2 moveto 0 2 rlineto stroke
xgridwidth -1 moveto 2 0 rlineto stroke
-2 ygridheight 1 add moveto 2 0 rlineto stroke
-1 ygridheight moveto 0 2 rlineto stroke
xgridwidth ygridheight 1 add moveto 2 0 rlineto stroke
xgridwidth 1 add ygridheight moveto 0 2 rlineto stroke
-2 ygridheight 2 mul 3 add moveto 2 0 rlineto stroke
-1 ygridheight 2 mul 2 add moveto 0 2 rlineto stroke
xgridwidth ygridheight 2 mul 3 add moveto 2 0 rlineto stroke
xgridwidth 1 add ygridheight 2 mul 2 add moveto 0 2 rlineto stroke } if
save /frontsave exch def
frontsideproc
frontsave restore

180 rotate xgridwidth neg ygridheight 1 add 2 mul neg translate
showthegrid { xgridwidth ygridheight showgrid} if
save /rearsave exch def
rearsideproc
rearsave restore} def

% demo -- remove before use

/lowerleft 100 def
/lowerbot 50 def
/gridresolution 10 def
/xgridwidth 20 def
/ygridheight 22.5 def
/showthegrid false def
/showtheticks true def

% define a step-and-repeat
/stepnrptparams 40 dict def
stepnrptparams begin
/threetentsplit [3 1 250 500 25 25 10 true true] def
end

0.9 1 scale % make it smaller in the x direction only

/frontsideproc {
Natural Beeswax Candles

~ 100% dripless ~
~ hand-rolled ~
~ natural honey scent ~

eden candles

...we'd love to hear from you

eden candles

p.o.box 299
cotton, ak 99543
903-485-2017
Natural Beeswax Candles

100% dripless ~ hand-rolled ~ natural honey scent ~

eden candles

3...we'd love to hear from you

p.o.box 299 cotton, ak 99543 903-485-2017

This is the main stepandrepeat tool...

customdata false def % autopaginate custom data?

%% define some artwork to be used on the tickets

/block1 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt 5 5 rlineto -5 5 rlineto -5 5 rlineto -10 10 rlineto 10 10 rlineto 5 5 rlineto 5 5 rlineto 10 20 22.5 false false tentproc} def
/block2 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt 5 5 rlineto 5 5 rlineto -5 5 rlineto -5 5 rlineto -5 5 rlineto -10 10 rlineto 10 10 rlineto 5 5 rlineto 5 5 rlineto 5 5 rlineto 10 20 22.5 false false tentproc} def
/block3 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt 5 5 rlineto 5 5 rlineto 5 5 rlineto 5 5 rlineto -10 10 rlineto 10 10 rlineto 5 5 rlineto 5 5 rlineto 5 5 rlineto 10 20 22.5 false false tentproc} def
/block4 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt 5 5 rlineto 5 5 rlineto 5 5 rlineto 5 5 rlineto 10 20 22.5 false false tentproc} def

% define some artwork to be used on the tickets

/block1 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt -5 -5 rlineto 5 -5 rlineto -5 -5 rlineto -10 10 rlineto 10 10 rlineto 5 -5 rlineto 5 -5 rlineto 5 -5 rlineto 10 20 22.5 false false tentproc} def
/block2 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt -5 -5 rlineto 5 -5 rlineto -5 -5 rlineto -10 10 rlineto 10 10 rlineto 5 -5 rlineto 5 -5 rlineto 5 -5 rlineto 10 20 22.5 false false tentproc} def
/block3 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt -5 -5 rlineto 5 -5 rlineto -5 -5 rlineto -10 10 rlineto 10 10 rlineto 5 -5 rlineto 5 -5 rlineto 5 -5 rlineto 10 20 22.5 false false tentproc} def
/block4 { save /snapbl exch def 0.2 setlinewidth newpath 15 15 mt -5 -5 rlineto 5 -5 rlineto -5 -5 rlineto -10 10 rlineto 10 10 rlineto 5 -5 rlineto 5 -5 rlineto 5 -5 rlineto 10 20 22.5 false false tentproc} def
Valley Quilters

Annual Quilt Raffle

$1.00/each – $5/six

1995 - 1001

name

address
city/state/zip

phone

1995 - 1002

name

address
city/state/zip

phone

1995 - 1003

name

address
city/state/zip

phone

1995 - 1004

name

address
city/state/zip

phone
Beginning PostScript Projects

-10 10 rlineto 10 10 rlineto 5 -5 rlineto gsave bestgray 0.5 setgray fill grestore stroke
snapbl restore} def

/cardtrick{
  0.2 setlinewidth
-0.2 -0.2 mt 30.4 0 rlineto 0 30.4 rlineto -30.4 0 rlineto
  closepath stroke
gsave block1 grestore
gsave 30 0 translate 90 rotate block2 grestore
gsave 30 30 translate 180 rotate block3 grestore
gsave 0 30 translate 270 rotate block4 grestore
} def

% To force an early exit when you run out of names or
% reach a given ticket number, test suitably. Then
% conditionally make /quickexit true and exit repeatproc.

% Note that an ending showpage is NOT required and
% should NOT be used.

% default variables used by stepnrpt - make changes here in
% starting number and number of pages
/srexitproc {} def     % default short exit - don’t do it.
/startnum 1001 def /runnum startnum def /numstring 15
/string def
/numpages 3 def        % default number of sheets to print

/repeatproc {save /snapt exch def
20 20 10 setgrid
% 47 15 showgrid
%
% this is what remains the same on each group of tickets

/cstretch 0 def /sstretch 0.01 def
/font0 /ZapfChancery-MediumItal 3 gonzofont
/font1 /Helvetica 1 gonzofont
/font2 /Helvetica-Narrow 0.7 gonzofont
/font3 /ZapfChancery-MediumItalic 2 gonzofont

/yinc 2.3 def
37.5 10 ((Valley Quilters
3Annual
Quilt Raffle) cc
37.5 2 ($1.00/each – $5/six ) cc

%place some artwork on the ticket
gsave 19.5 3 translate 0.3 dup scale cardtrick grestore

% rotate and translate for the ticket stub text placement
gsave .90 rotate 0 -15 translate
/yinc 2.5 def
I 9 (2name
address
city/state/zip
phone) cl
0.1 setlinewidth

{{1 2.2 mt r} 2.5 4}yrpt

% calls the font for the changing numbers and changes
% numbers remove the "A" from the string for numbers
% only change the numbers in the font array for point size

/Helvetica findfont [1.0 0 0 1.2 0 0] makefont setfont
4.8 12 moveto (1995 - ) show runnum numstring cvs show
4.8 -2 moveto (1995 - ) show runnum numstring cvs show

grestore
snapt restore } def

(fourtickets) stepandrepeat
quit

PostScript Operators

TEAR-OFF

setdash OFF

AUTONUMBERING TICKETS

ceiling red book pg 372
cvi red book pg 394
cvn red book pg 394
cvs red book pg 396
div red book pg 404
dup red book pg 404
get red book pg 426
mark red book pg 454
mul red book pg 456
show red book pg 520

Gonzo Commands

TENT

frontsideproc user defined tent front
gridresolution size of grid used
lowerleft horizontal starting point
lowerbot vertical starting point
rearsideproc user defined tent rear
tentproc actually does the work
xgridwidth width of grid used
ygridheight height of grid used

AUTONUMBERING TICKETS

drawticks puts tick marks at corners
horstart horizontal starting point
inchoriz width of each ticket
incvert height of each ticket
numhoriz number of tickets horizontally on page
numstring string of numbers to be printed
numvert number of tickets vertically on page
quickexit allows exit in middle of page
runnum current ticket number
vertstart vertical starting point

New Step and Repeat Commands

/fourtickets [1 4 470 150 0 0 10 false false] def
prints 4 autonumbered tickets per
sheet based on Blanks-USA regular size
/threetentsplit [3 1 250 500 25 25 10 true true] def
prints three tents
/12tabs [11 1 5 0 1.5 3 10 false false] def
repeats a short message/phone number
for tear-off/take-away purposes
/twodoorhangers [2 1 285 750 22 18 10 false false] def
prints two doorhangers feet first

On Your Own
Review and alter the examples given in the setdash
operator definitions – red book pg 500.
Write a pulp novel or newsletter to be set-up using the
pulpnovel template. Or find a paperback novel you would
not normally read to typeset for the template.
This pulp novel template is a useful starting point for creating your own custom gonzo book templates. It is purposely kept VERY simple. As usual, if you don’t like it, rearrange the scenery to suit yourself.

Note that any template MUST have an internal proc named templatesstartup and that your template MUST NOT call gonzofont before run time.

VERY IMPORTANT NOTE: Some word processors insist on adding extra carriage returns between paragraphs or to enforce certain line length limits. MAKE CERTAIN YOURS DOES NOT! If all else fails, insert a <cr> ahead of time and if needed.

**Pulp Novel Template and Demo**

```plaintext
300 dict /pulpnovel exch def
pulpnovel begin

% All fonts MUST defer till run time!
/templatesfonts {
/font0 /Bookman-Demi 36 gonzofont % initial raised cap
/font1 /Bookman-Light 9 gonzofont % main text
/font2 /Bookman-LightItalic 9 gonzofont % italics
/font3 /Bookman-Demi 10 gonzofont % headers
} def

% values used by gonzo justify ...

/cstretch 0.2 def % minimum character kerning
/laststretch 0.03 def % stretch last paragraph line
/pm 10 def % use paragraph indents
/ssstretch -0.1 def % minimum space kerning
/txtwide 255 def % width of text
/limpos 30 def % left margin
/ybot 62 def % bottom column reference
/yinc 11 def % vertical line spacing
/ytop ybot yinc 39 mul add def % BASE of top line
/pagenum 1 def % running pagnumber

% colcheck does all the work in a pagemaker. It is % automatically called at the start of each new printable % line. It can make room for figures, etc.

/colcheck {ypos ybot ytop} if} def
```

% This header does nothing; the footer centers an % unadorned page number

```
/header { } def
/footer {save /footsnap exch def
/ybot1 ybot def /ybot
-9999 def rpagenum ( ) cvs dup stringwidth pop 2
div txtwide 2 div xpos add sub neg exch ybot1 15 sub exch
def} def
```

% templatesstartup sets the margins and gets the header and % footer going on the first page

```
/templatesstartup { templatefonts font1 /xpos limpos def
/ypos ytop def /gonzojust true def /rslashok true def
/pagenum pagenum def header footer} def
```

```
end % simplepages template

%%%%%%
%%%%%% change to your novel %%%%
```

---

**ASSIGNMENT 18:**

Typeset your pulpnovel or one column newsletter

---

0FD]larcie didn’t expect to live. With the hand she could move enough to reach them, she tore unit and command patches from her uniform shirt\261left only her nametag, rank and flight surgeon’s insignia. She drew out the chain from around her neck, yanked off the two crystal pendants hanging with her ID tags, and shoved them into the corner behind her.[p

"Mama?" The child stirred on her lap, trying to push himself back. "Why are\261" She put a finger to his lips, her other hand cupping his head to prevent its bump-ing the metal bulkhead. "Hush, Tris."

She could barely whisper; she sat on the bottom of a locker meant only for a pressure suit\261one of four in the maintenance compartment\261with the toddler held snug between her body and her drawn-up knees.
Darcie didn’t expect to live. With the hand she could move enough to reach them, she tore unit and command patches from her uniform shirt—left only her nametag, rank and flight surgeon’s insignia. She drew out the chain from around her neck, yanked off the two crystal pendants hanging with her ID tags, and shoved them into the corner behind her.

"Mama?" The child stirred on her lap, trying to push himself back. "Why are—" She put a finger to his lips, her other hand cupping his head to prevent its bumping the metal bulkhead. "Hush, Tris."

She could barely whisper; she sat on the bottom of a locker meant only for a pressure suit—one of four in the maintenance compartment—with the toddler held snug between her body and her drawn-up knees.

Outside ship noises reached her: the roar of engines crescendoing toward thrust into lightskip—the fourth attempt. She braced her head well back in the corner behind the pressure suit, hugged Tristan to her breast and locked her teeth.

REST OF STORY GOES HERE . . .

She lunged left into the cross corridor, her right fist punching the manual trigger on its bulkhead. The shield door dropped behind her with a whoosh that was lost in a crunch and her pursuer’s garbled scream. She pressed Tristan’s face to her shoulder and forced herself not to look back.

The cross passage opened on one parallel to the concourse she had
sealed. It led to the lifepods—if they hadn’t already been jettisoned.

She pressed herself to the bulkhead to listen for pursuers and to peer into the concourse. It was empty, up and down, except for the smoke-obscured shapes of bodies on the deck. She tried to set the child on his feet, to rest her arm, but he only clung to her, wide-eyed with the recognition of her fear.

Darcie smoothed his hair and kissed his forehead. “Come on then, little soldier,” she said, collecting him again, and slipped on into the passage. Smoke from screen grenades stung her eyes, making them run and blurring her vision. She stumbled over a body and paused, panting. One of the surface troopers, a young man she didn’t know. Flung beyond his hand lay an energy pistol.
Outside ship noises reached her: the roar of engines crescendoing toward thrust into lightskip the fourth attempt. She braced her head well back in the corner behind the pressure suit, hugged Tristan to her breast and locked her teeth.

REST OF STORY GOES HERE . . .

She lunged left into the cross corridor, her right fist punching the manual trigger on its bulkhead. The shield door dropped behind her with a whoosh that was lost in a crunch and her pursuer’s garbled scream. She pressed Tristan’s face to her shoulder and forced herself not to look back.

The cross passage opened on one parallel to the concourse she had sealed. It led to the lifepods if they hadn’t already been jettisoned.

She pressed herself to the bulkhead to listen for pursuers and to peer into the concourse. It was empty, up and down, except for the smoke-obscured shapes of bodies on the deck. She tried to set the child on his feet, to rest her arm, but he only clung to her, wide- eyed with the recognition of her fear.

Darcie smoothed his hair and kissed his forehead. “Come on then, little soldier,” she said, collecting him again, and slipped on into the passage. Smoke from screen grenades stung her eyes, making them run and blurring her vision. She stumbled over a body and paused, panting. One of the surface troopers, a young man she didn’t know. Flung beyond his hand lay an energy pistol.

The “IDEAL” BOOK-ON-DEMAND PRINTER

**Genuine Adobe PostScript**
Level II or higher.

**Bolt-on Duplexer**
Give you double sided printing with fewer hassles and less scrap.

**Internal or External Hard Drive**
Allow unattended operation and speed up printing time.
They also free the operator and host computer for other tasks.

**600+ DPI Resolution**
Also photo enhancements. Optimize your artwork for the printer you are now using.

**12 PPM**
Minimum page count needed for fastest make-up and printing times.

**Two-way Comm**
Proper use of PostScript demands on-screen error messages and host recordable comm.


**PostScript Operators**

- dict   red book pg 402
- pop    red book pg 463
- stringwidth  red book pg 528

**Gonzo Commands**

- colcheck  what actually does the pagemakeup
- footer   what goes at the bottom of the page
- gonzojust turns on the gonzo justification
- header   what goes at the top of the page
- lmpos    left margin
- pagenum  starting page number
- pulpnum  template definition
- rpagenum running pagenumber adds one to each page
- rslashok boolean: prints the reverse slash
- templatefonts fonts to be called in the pulpnum
- templatetemplate template
- templatestartup sets margins, gets header & footer

**On Your Own**

Collect two column magazine articles, or write your own article or newsletter.
Two Column Newsletters

This routine extends the gonzo justify to allow multiple columns over multiple pages. The text will automatically "flow" around included figures. This is a two column version.

**Two Column Newsletter**

gonzo begin
gutility begin
printrerror
nuisance begin

% /// TWO COLUMN TEMPLATE ///

% NOTE: using a grid gets tricky on multi-page documents. To keep things simple, we will work full size here. Individual figures can be on their own 10X grid.

% Select your fonts. Font0 is the drop cap. Font1 is normal text. Font2 is italic text. Font3 is bold text. Font= is the end bullet. Font- is used in the figures. The first font number is the width; the third is the height, and the sixth is the amount of drop for a header ...

/font0 /Palatino-Bold [42 0 0 40 0 -22] gonzofont
/font1 /Palatino-Roman [10 0 0 10 0 0] gonzofont
/font2 /Palatino-Italic [10 0 0 10 0 0] gonzofont
/font3 /Palatino-Bold [10 0 0 10 0 -6] gonzofont
/font= /ZapfDingbats [10 0 0 10 0 0] gonzofont
/font- /ZapfDingbats [160 0 0 160 0 0] gonzofont

% pick some template values
/txtwide 240 def % width of text columns
/yinc 11 def % line spacing
/cstretch 0.2 def % minimum extra space between characters
/ssstretch 0 def % minimum extra space between spaces
/maxsstretch 0.25 def % trip point for character stretching
/maxcstretch 0.1 def % maximum allowable character stretching
/dropcount 3 def % number of lines the drop cap drops
/dropindent 29 def % indentation for the drop cap
/lastlinestretch 0.3 def % slight stretch to last paragraph line

 ASSIGNMENT 19: Write and set up a two column newsletter with text and figures.

/end

% makefigroom lets the text flow around the figure spaces

/makefigroom {figdict {/figvalues exch def pop pagenum figvalues 0 get eq {colnum figvalues 1 get eq {ypos figvalues 3 get cvx exec figvalues 2 get cvx exec add le ypos figvalues 3 get cvx exec ge and {/ypos ypos figvalues 2 get cvx exec yinc div cvi 1 add yinc mul sub def} if} if} forall} def}

% showfigs goes through the dictionary and shows the needed figures on the required pages

/showfigs {figdict {/figvalues exch def pop pagenum figvalues 0 get eq {figvalues 4 get {/hold2 exch def /hold1 exch def save /figsnapx exch def /colcheck {} def colleft

% decide when to change columns. This particular colcheck % also makes room for figures and prints all but the last % page, so it is your pagemaker.

//colcheck { makefigroom ypos colbot lt {colnum numcolumns ge {showpage /pagenum pagenum 1 add def showfigs /ypos coltop yinc sub def /xpos colleft def /colnum 1 def leftrightpage exch def header footer makefigroom} /colnum colnum 1 add def /ypos coltop yinc sub def /xpos xpos colspacing add def makefigroom} ifelse} if def

% setup your figures by using a dictionary. The Format is % [pagenum colnum figheight figbot drawiftrue figname]

/figdict 10 dict def figdict
begin
/fig1 [1 2 140 {colbot 101 add} true (clubs)] def
/fig2 [2 1 140 {colbot} true (spades)] def
/fig3a [4 1 140 {coltop 144 sub} true (shears)] def
/fig3b [4 2 140 {coltop 144 sub} false (shears)] def
end

% set up your figures by using a dictionary. The Format is % [pagenum colnum figheight figbot drawiftrue figname]
This fairly simple code routine does show you how you can do a two column layout which includes built-in figures and also can include headers and footers that are left-right sensitive. While there are several fine layout programs available today, such as 2Ventura Publisher1 and 2PageMaker1, this text-only code shown here gives a very high quality result, is totally flexible, device independent, fast, and economical. More importantly, you can easily switch in to or out of PostScript anytime, so there is no practical limit to what goes where on the page.

Centered & Dropped Headings

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated
This fairly simple code routine does show you how you can do a multi column layout which includes built-in figures and also can include headers and footers that are left-right sensitive.

While there are several fine layout programs available today, such as Ventura Publisher and PageMaker, this text-only code shown here gives a very high quality result, is totally flexible, device independent, fast, and economical.

More importantly, you can easily switch in to or out of PostScript anytime, so there is no practical limit to what goes where on the page.

Centered & Dropped Headings

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.
here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.
over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.
over and over again so we can simplify all of your typing
on this newsletter demo.
We can throw in a bunch of dummy paragraphs here that do
not say much of anything, yet could get easily repeated
over and over again so we can simplify all of your typing
on this newsletter demo.
We can throw in a bunch of dummy paragraphs here that do
not say much of anything, yet could get easily repeated
over and over again so we can simplify all of your typing
on this newsletter demo.
We can throw in a bunch of dummy paragraphs here that do
not say much of anything, yet could get easily repeated
over and over again so we can simplify all of your typing
on this newsletter demo.[=F

% and show the final page. Don’t skip the |x gonzo ender
% above!

showpage

PostScript Operators

| clear | red book pg 373 |
| exec | red book pg 409 |
| forall | red book pg 423 |

Gonzo Commands

colbot | column bottom position

colcheck | also makes room for figures and prints

all but the last page, it is the

pagemaker.

colleft | first column left margin

colsspacing | width of columns

coltop | column top position

figdict | a dictionary used to call the figures

needs: [pagenum colnum figheight

figbot drawiftrue figname]

firstpagefooter | boolean - turn on for a special first

page footer

firstpageheader | boolean - turn on for a special first

page footer

footer | defines where the footer is placed

header | defines where the header is placed

leftfooter | information for the left page footer

letheader | information for the left page header

leftpage | boolean - picks up first page header

and footer

makefigroom | lets the text flow around the figure

spaces

numcolumns | number of columns

pagenum | starting page number

pagenumcheck | finds the ‘pagenum’ and replaces it

with actual digits

pmnorm | paragraph margin

righthead | information for the right page header

righfooter | information for the right page footer

setup | defines and calls the page making

routine, includes colnum, xpos,

figures, header, footer, firstpage info.

showfigs | goes through the figdict dictionary and

shows the needed figures on the

required pages

On Your Own

Redesign the templates for the two and three column

newsletters changing the number of columns (1, 4, 5),

column width, and the column top and column height.

Replace the figures with your own artwork or curvetraces.
Three Column Newsletters

This routine extends the gonzo justify to allow multiple columns over multiple pages. The text will automatically "flow" around included figures. This is a three column version.

Three Column Newsletter

gonzo begin
gutility begin
printerror
nuisance begin

% /// THREE COLUMN TEMPLATE ///

% NOTE: using a grid gets tricky on multi-page % documents. To keep things simple, we will work full size % here. Individual figures can be on their own 10X grid.

% Select your fonts. Font0 is the drop cap. Font1 is normal % text. Font2 is italic text. Font3 is bold text. Font= is the % end bullet. Font- is used in the figures. The first font % number is the width; the third is the height, and the sixth % is the amount of drop for a header ...

/font0 /Palatino-Bold [42 0 0 40 0 -22] gonzofont
/font1 /Palatino-Roman [10 0 0 10 0 0] gonzofont
/font2 /Palatino-Italic [10 0 0 10 0 0] gonzofont
/font3 /Palatino-Bold [10 0 0 10 0 -6] gonzofont
/font= /ZapfDingbats [10 0 0 10 0 0] gonzofont
/font- /ZapfDingbats [160 0 0 160 0 0] gonzofont

% pick some template values

/txtwide 140 def % width of text columns
/yinc 11 def % line spacing
/cstretch 0.2 def % minimum extra space between % characters
/ssstretch 0 def % minimum extra space between % spaces
/maxsstretch 0.25 def % trip point for character stretching
/maxcstretch 0.1 def % maximum allowable character % stretching
/dropcount 3 def % number of lines the drop cap drops
/dropindent 29 def % indentation for the drop cap
/lastlinestretch 0.3 def % slight stretch to last paragraph line

/colleft 60 def % first column left margin
/colbot 200 def % column bottom position
/coltop 500 def % column top position
/numcolumns 3 def % number of columns
/colspacing 160 def % width of columns
/ypos coltop def % select top text line
/pmnorm 10 def % paragraph margin

% decide when to change columns. This particular colcheck % also makes room for figures and prints all but the last % page, so it is your pagemaker.

/colcheck { makefigroom ypos colbot lt {colnum numcolumns ge {showpage /pagenum pagenum 1 add def
showfigs /ypos coltop yinc sub def /xpos colleft def
/colnum 1 def leftpage {false} {true} ifelse /leftpage exch def
def footer makefigroom} [/colnum colnum 1 add def
/ypos coltop yinc sub def /xpos xos colspacing add def
makefigroom] ifelse} if} def

% setup your figures by using a dictionary. The Format is % [pagenum colnum figheight figbot drawiftrue figname]

/figdict 10 dict def
figdict
begin
/fig1 [1 2 140 {colbot 101 add} true (clubs)] def
/fig2 [2 1 140 {colbot} true (spades)] def
/fig3a [4 1 140 {coltop 144 sub} true (shears)] def
/fig3b [4 2 140 {coltop 144 sub} false (shears)] def
end

% makefigroom lets the text flow around the figure spaces

/makefigroom {figdict {} /figvalues exch def pop pagenum figvalues 0 get eq {colnum figvalues 1 get eq {ypos figvalues 3 get cvx exec figvalues 2 get cvx exec add le ypos figvalues 3 get cvx exec ge and {/ypos yos figvalues 2 get cvx exec yinc div cvi 1 add yinc mul sub def} if} if} forall} def

% showfigs goes through the dictionary and shows the % needed figures on the required pages

/showfigs {figdict {} /figvalues exch def pop pagenum figvalues 0 get eq {figvalues 4 get /hold2 exch def /hold1 exch def save /figsnapx exch def /colcheck {} def colleft}
his fairly simple code routine does show you how you can do a three column layout which includes built-in figures and also can include headers and footers that are left-right sensitive. |p
While there are several fine layout programs available today, such as |2Ventura Publisher|1 and |2PageMaker|1, this text-only code shown here gives a very high quality result, is totally flexible, device independent, fast, and economical.

More importantly, you can easily switch in to or out of PostScript anytime, so there is no practical limit to what goes where on the page.

Centered & Dropped Headings

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing.
This fairly simple code routine does show you how you can do a three column layout which includes built-in figures and also can include headers and footers that are left-right sensitive.

While there are several fine layout programs available today, such as Ventura Publisher and PageMaker, this text-only code shown here gives a very high quality result, is totally flexible, device independent, fast, and economical.

More importantly, you can easily switch in to or out of PostScript anytime, so there is no practical limit to what goes where on the page.

**Centered & Dropped Headings**

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.
yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of
dummy paragraphs here that
do not say much of anything,
yet could get easily repeated
over and over again so we can
simplify all of your typing on
this newsletter demo.

We can throw in a bunch of
dummy paragraphs here that
do not say much of anything,
yet could get easily repeated
over and over again so we can
simplify all of your typing on
this newsletter demo.

We can throw in a bunch of
dummy paragraphs here that
do not say much of anything,
yet could get easily repeated
over and over again so we can
simplify all of your typing on
this newsletter demo.

We can throw in a bunch of
dummy paragraphs here that
do not say much of anything,
yet could get easily repeated
over and over again so we can
simplify all of your typing on
this newsletter demo.
simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.
on this newsletter demo. We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.
on this newsletter demo.
We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo. We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo. We can throw in a bunch of dummy paragraphs here that do not say much of anything, yet could get easily repeated over and over again so we can simplify all of your typing on this newsletter demo.

|x

% and show the final page. Don’t skip the |x gonzendor
% above!

showpage

**On Your Own**

Do further work on 1,2,3,4,5 column layouts. 8.5 x 11 portrait layouts with 4 or more columns should be left, right, or center justified and not fill justified for the best spacing. Compare your layouts with those in quality magazines and newsletters.
# CHARACTER CODES

<table>
<thead>
<tr>
<th>A ✡</th>
<th>a ✡</th>
<th>1 ✡</th>
<th>[ ✡</th>
<th>i ✡</th>
</tr>
</thead>
<tbody>
<tr>
<td>B ✢</td>
<td>b ✢</td>
<td>2 ✢</td>
<td>] ✢</td>
<td>e ✢</td>
</tr>
<tr>
<td>C ✣</td>
<td>c ✣</td>
<td>3 ✣</td>
<td>{ ✣</td>
<td>o ✣</td>
</tr>
<tr>
<td>D ✤</td>
<td>d ✤</td>
<td>4 ✤</td>
<td>} ✤</td>
<td>£ ✤</td>
</tr>
<tr>
<td>E ✥</td>
<td>e ✥</td>
<td>5 ✥</td>
<td>' ✥</td>
<td>/ ✥</td>
</tr>
<tr>
<td>F ✦</td>
<td>f ✦</td>
<td>6 ✦</td>
<td>&quot; ✦</td>
<td>¥ ✦</td>
</tr>
<tr>
<td>G ✧</td>
<td>g ✧</td>
<td>7 ✧</td>
<td>, ✧</td>
<td>¥ ✧</td>
</tr>
<tr>
<td>H ✨</td>
<td>h ✨</td>
<td>8 ✨</td>
<td>. ✨</td>
<td>§ ✨</td>
</tr>
<tr>
<td>I ✩</td>
<td>i ✩</td>
<td>9 ✩</td>
<td>/ ✩</td>
<td>¶ ✩</td>
</tr>
<tr>
<td>J ✪</td>
<td>j ✪</td>
<td>0 ✪</td>
<td>&lt; ✪</td>
<td>♆ ✪</td>
</tr>
<tr>
<td>K ✫</td>
<td>k ✫</td>
<td>! ✫</td>
<td>&gt; ✫</td>
<td>&quot; ✫</td>
</tr>
<tr>
<td>L ✬</td>
<td>l ✬</td>
<td>@ ✬</td>
<td>? ✬</td>
<td>« ✬</td>
</tr>
<tr>
<td>M ✭</td>
<td>m ✭</td>
<td># ✭</td>
<td>' ✭</td>
<td>« ✭</td>
</tr>
<tr>
<td>N ✮</td>
<td>n ✮</td>
<td>$ ✮</td>
<td>~ ✮</td>
<td>» ✮</td>
</tr>
<tr>
<td>O ✯</td>
<td>o ✯</td>
<td>% ✯</td>
<td>fi ✯</td>
<td>fi ✯</td>
</tr>
<tr>
<td>P ✰</td>
<td>p ✰</td>
<td>^ ✰</td>
<td>fl ✰</td>
<td>fl ✰</td>
</tr>
<tr>
<td>Q ✱</td>
<td>q ✱</td>
<td>&amp; ✱</td>
<td>&amp; ✱</td>
<td>&amp; ✱</td>
</tr>
<tr>
<td>R ✲</td>
<td>r ✲</td>
<td>* ✲</td>
<td>* ✲</td>
<td>* ✲</td>
</tr>
<tr>
<td>S ✳</td>
<td>s ✳</td>
<td>( ✳</td>
<td>( ✳</td>
<td>( ✳</td>
</tr>
<tr>
<td>T ✴</td>
<td>t ✴</td>
<td>) ✴</td>
<td>) ✴</td>
<td>) ✴</td>
</tr>
<tr>
<td>U ✵</td>
<td>u ✵</td>
<td>- ✵</td>
<td>- ✵</td>
<td>- ✵</td>
</tr>
<tr>
<td>V ✶</td>
<td>v ✶</td>
<td>= ✶</td>
<td>= ✶</td>
<td>= ✶</td>
</tr>
<tr>
<td>W ✷</td>
<td>w ✷</td>
<td>- ✷</td>
<td>- ✷</td>
<td>- ✷</td>
</tr>
<tr>
<td>X ✸</td>
<td>x ✸</td>
<td>+ ✸</td>
<td>+ ✸</td>
<td>+ ✸</td>
</tr>
<tr>
<td>Y ✹</td>
<td>y ✹</td>
<td>; ✹</td>
<td>; ✹</td>
<td>; ✹</td>
</tr>
<tr>
<td>Z ✺</td>
<td>z ✺</td>
<td>: ✺</td>
<td>: ✺</td>
<td>: ✺</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th></th>
<th>—</th>
<th>7</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
PostScript Operators: Red Book Chapter 8.2

Project 1
begin          red book pg 370
def            red book pg 397
lineto         red book pg 449
makefont       red book pg 452
moveto         red book pg 456
quit           red book pg 468
repeat         red book pg 476
rlineto        red book pg 482
scalefont      red book pg 488
setfont        red book pg 503
setlinewidth    red book pg 508
showpage       red book pg 520
stroke         red book pg 528

Project 2
setlinecap     red book pg 506

Project 3
repeat         red book pg 476

Project 4
grestore       red book pg 429
gsave          red book pg 430
rotate         red book pg 484
setgray        red book pg 504
translate      red book pg 533

Project 5
clear          red book pg 373
exch           red book pg 408
restore        red book pg 481
save           red book pg 486

Project 6
arc            red book pg 365
clear          red book pg 373
closepath      red book pg 375
currentpoint   red book pg 390
dup            red book pg 404
exch           red book pg 408
fill           red book pg 416
neg            red book pg 456
newpath        red book pg 457
scale          red book pg 487

Project 9
curveto        red book pg 393

Project 10
eq             red book pg 407
ge             red book pg 425
gt             red book pg 426
le             red book pg 448
lt             red book pg 451
ne             red book pg 456
**Project 16**

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add</code></td>
<td>363</td>
</tr>
<tr>
<td><code>charpath</code></td>
<td>372</td>
</tr>
<tr>
<td><code>clip</code></td>
<td>374</td>
</tr>
<tr>
<td><code>clippath</code></td>
<td>374</td>
</tr>
<tr>
<td><code>cvx</code></td>
<td>396</td>
</tr>
<tr>
<td><code>dup</code></td>
<td>404</td>
</tr>
<tr>
<td><code>eoclip</code></td>
<td>406 - (rb section 4.5)</td>
</tr>
<tr>
<td><code>findfont</code></td>
<td>418</td>
</tr>
<tr>
<td><code>forall</code></td>
<td>423</td>
</tr>
<tr>
<td><code>initclip</code></td>
<td>437</td>
</tr>
<tr>
<td><code>makefont</code></td>
<td>452</td>
</tr>
<tr>
<td><code>mark</code></td>
<td>454</td>
</tr>
<tr>
<td><code>mul</code></td>
<td>456</td>
</tr>
<tr>
<td><code>pathforall</code></td>
<td>462</td>
</tr>
<tr>
<td><code>pop</code></td>
<td>463</td>
</tr>
<tr>
<td><code>put</code></td>
<td>466</td>
</tr>
<tr>
<td><code>scalefont</code></td>
<td>488</td>
</tr>
<tr>
<td><code>setdash</code></td>
<td>500</td>
</tr>
<tr>
<td><code>setfont</code></td>
<td>520</td>
</tr>
<tr>
<td><code>show</code></td>
<td>528</td>
</tr>
<tr>
<td><code>sin</code></td>
<td>529</td>
</tr>
<tr>
<td><code>stringwidth</code></td>
<td>528</td>
</tr>
<tr>
<td><code>sub</code></td>
<td>529</td>
</tr>
</tbody>
</table>

**Project 17**

**TEAR-OFF**

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>setdash</code></td>
<td>500</td>
</tr>
</tbody>
</table>

**AUTONUMBERING TICKETS**

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ceiling</code></td>
<td>372</td>
</tr>
<tr>
<td><code>cvi</code></td>
<td>394</td>
</tr>
<tr>
<td><code>cvn</code></td>
<td>394</td>
</tr>
<tr>
<td><code>cvs</code></td>
<td>396</td>
</tr>
<tr>
<td><code>div</code></td>
<td>404</td>
</tr>
<tr>
<td><code>dup</code></td>
<td>404</td>
</tr>
<tr>
<td><code>get</code></td>
<td>426</td>
</tr>
<tr>
<td><code>mark</code></td>
<td>454</td>
</tr>
<tr>
<td><code>mul</code></td>
<td>456</td>
</tr>
<tr>
<td><code>show</code></td>
<td>520</td>
</tr>
</tbody>
</table>

**Project 18**

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dict</code></td>
<td>402</td>
</tr>
<tr>
<td><code>pop</code></td>
<td>463</td>
</tr>
<tr>
<td><code>stringwidth</code></td>
<td>528</td>
</tr>
</tbody>
</table>

**Project 19**

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clear</code></td>
<td>373</td>
</tr>
<tr>
<td><code>exec</code></td>
<td>409</td>
</tr>
<tr>
<td><code>forall</code></td>
<td>423</td>
</tr>
</tbody>
</table>

Gonzo Step and Repeats

Horizontal and vertical are defined AFTER selecting portrait or landscape. Parameters (in points or true/false) are as follows...

#horizrpts - times proc repeats in horizontal direction
#vertrpts - times proc repeats in vertical direction
hspaceing - horizontal proc repeat spacing
vspacing - vertical proc repeat spacing
hstart - horizontal offset of first proc
vstart - vertical offset of first proc
ticklen - length of cropping ticks when and if used
useticks? - show the ticks true/false boolean
landscape - landscape orientation true/false boolean

Steandrepeat Definitions

/admitonetick [5 9 150 60 25 25 10 true true] def % 45 tickets
/babybumper [2 10 270 72 40 30 20 true false] def % 20 stickers
/badgeaminit [2 3 220 220 90 60 250 false false] def % 6 badges
/bigbumpstick [1 3 792 205 0 0 40 true true] def % 3 bumperstickers
/buscard [3 4 256 143 12 20 20 true true] def % 12 buscards
/busenvelope [1 1 685 305 0 150 20 true true] def % 1 envelope
/eightlabel [2 4 306 198 0 0 50 true false] def % 1/8 page
/fourtickets [1 4 470 150 0 0 10 false false] def % 4 autonumbered tickets
/fulllandpage [1 1 0 20 20 50 false true] def % entire landscape page
/fullportpage [1 1 0 20 20 50 false true] def % entire portrait page
/medbumper [1 5 585 144 20 30 20 true false] def % 5 bumperstickers
/quadsplit [2 2 396 306 0 0 50 true true] def % 1/4 page
/readerserv [12 25 25 -15 120 450 0 false false] def % 300 RS numabers
/shiplabel [1 4 290 180 160 65 40 true false] def % 4 custom labels
/sixlabel [1 4 306 264 0 0 50 true false] def % 1/6 page
/smallbumper [2 10 270 72 40 30 20 true false] def % 20 stickers
/stdlabel [1 11 254 74 185 5 20 true false] def % 11 stock dp labels
/tenlabel [2 5 305 144 0 45 50 true false] def % 1/10 page
/threetentsplit [3 1 250 500 25 25 10 true true] def % 3 tents
/twodoorhangers [2 1 285 750 22 18 10 false false] def % 2 door hangers
/videospline [1 13 424 60.3 80 34 0 false false] def % 13 VHS splines
/3.5disklabel [2 3 216 226 100 60 20 false false] def % 6 disk labels 3.5
/5.25disklabel [1 7 316 110 275 35 0 false false] def % 7 disk labels 5.25
/12tabs [11 1 5 0 1.5 3 10 false false] def % 12 rip-off tabs on bottom of flyer
Gonzo Commands Used in Projects

**Project 1**
- cl     callout left
- cc     callout center
- cr     callout right
- cf     callout fill
- gutility         gutility dictionary
- gonzo            gonzo dictionary
- nuisance         nuisance dictionary
- printerror       returns error to host and prints page
- setgrid          defines the xstart ystart points of each grid square
- showfont   prints the font
- showgrid         prints the grid

**Project 2**
- mt    moveto
- rl    rlineto
- h     does a half negative linefeed
- l     lineto left
- r     lineto right
- u     lineto up
- d     lineto down
- pr    path right
- pl    path left
- pu    path up
- pd    path down
- bestgray         best gray angle and arrangement
- copies   used for multiple copies
- font1 thru font0  defines the gonzofonts
- gonzofont    calls the predefined fonts
- cstretch         stretches spaces between characters
- sstretch         stretches spaces between words
- showthegrid   turns on or off the layout grid
- superstroke      strokes complex black, gray, white lines or boxes
- yinc             defines the vertical line spacing

**Project 3**
- manual   manual feeds stock from top of tray
- repeatproc    actual text and graphics to be repeated
- stepnrptparams   step and repeat dictionary
- stepandrepeat    prints a step and procedure (replaces showpage)

**Project 4**
- black   returns printing to black from gray
- landscape        sets the landscape mode (horizontal)
- lightgray        lightest gray available
- roundbox         defines a complex black, gray, white box

**Project 5**
- font1   a way to call fonts for several strings
- line1   sets the linewidth at .06 of a point or grid square
- line2   sets the linewidth at .12 of a point or grid square
- line3   sets the linewidth at .18 of a point or grid square
xrpt defines vertical lines repeated in the horizontal direction
yrpt defines horizontal lines repeated in the vertical direction

**Project 6**
roundbox uses x y width height radius
showmaingrid used for main grid with additional grids on the same page
showboxgrid used to place a grid in a predefined box

**Project 7**
macro an executable array
startgonzo Begins gonzo scanning of text that follows. |g may also be used. Terminates on end of file or |x.
|? User defined embeddable escape commands and macro commands.
stringmacro A shorthand way of building macros. For instance
/amacro {(Cyz3) stringmacro} def does an |C |y |z |3 all on a single |a embedded command.

Also see Gonzo Justification Variables and Commands Appendix

**Project 8**
titshift horizontal shift in title
tabs allows you to set left justifid tabs or columns
justQ user defined justification mode
justP user defined justification mode
justL user defined left justification

**Project 9**
ctproc curvetrace proc – allows you to change tension on a curvetraced path
curvetrace calls the curvetraced path
showtick shows an arrow at each data point at angle entered
superinsidestroke allows an a-symmetric border
tension changes the amount of curve or negative curve generated
ticklen defines the length of arrow shaft
tickhead defines the length of arrow head

**Project 10**
body user defined proc
calcpages decides how many pages are needed when you are using a custom srfile. This allows early exits when or if you run out of data.
censhow a simple centering proc
customdata autopaginate custom data?
date user defined proc
datat user defined proc
letterhead user defined proc
logoandreturnaddress user defined proc

numpages default number to print
quickexit forces an early exit when you run out of names or reach a given ticket
returnaddress user defined proc
runnum running pointer advances one per
signature user defined proc
srexitproc default short exit - don’t do it
srfile gets used for custom entries
startnum initial ticket number
striplines takes a multi-line string, extracts each individual line string, and then executes stripproc for each line. Useful for simple addressing and custom form filling-in.
strip(proc) needed by striplines

**Project 11**

- **botpanelproc**
  user defined: horizontal bottom panel
- **colcheck**
- **midpanelproc**
  user defined: horizontal middle panel
- **panelset**
  positions each panel in the vertical trifold brochure
- **paneltext**
  user defined: each panel of the vertical trifold
- **quadpixel**
  refines the grid
- **toppanelproc**
  user defined: horizontal top panel
- **upside downpanel**
  inverts a panel

**Project 12**

- **arckern**
  defines the amount of kern between arced characters
- **customkern**
  defines additional kern needed between specific arced characters
- **customkernchar**
  defines the character used to call the custom kern
- **karcjustify**
  calls the arc justified text

**Project 13**

- **backwards**
  automatically prints backwards
- **black**
  sets a black area or line
- **black flash**
  prints an entire black page
- **negative**
  makes black = white and vice versa
- **white**
  sets a white area or line

**Project 14**

**SCROLL BORDER**

- **drawscrollborder**
  defines and calls the border
- **borderthickness**
  how high the border is
- **border linewidth**
  how heavy the scrolls are
- **hloops**
  horizontal loops in scroll border
- **vloops**
  vertical loops in scroll border

**ROPE-A-DOPE**

- **drawarope**
  drawarope automates it rope1 printing
- **drawelement**
  drawelement grabs three edgepoint sets and passes them on to drawproc
- **hsquareknot**
  draws a horizontal square knot, aided by rsqloop1 and lsqloop1
- **lsqloop1**
  left square knot loop
- **path to step array**
  converts a path \{ang0 ang1 ... angn\} into an array
  \{ \{xa0 y0 xb0 yb0\} \{xa1 y0 xb1 yb1\} \ldots\} The "a"
  points are zdis normal "above" path center, while "b"
  points are zdis normal "below" path center.
- **pdis**
  width of each rope element
- **rope1**
  rope1 builds a rope turn of one-half pdis, 500 max.
- **rsqloop1**
  right square knot loop
- **show segrule**
  boolean
- **zdis**
  height of each rope element

**CHARACTER BORDER**

- **stringup**
  moves character repeat up left side
- **stringright**
  moves character repeat across top
- **stringdown**
  moves character repeat down right side
- **stringleft**
  moves character repeat across bottom

**QUADFOLD BORDER**

- **roundpath**
  defines a path with rounded corners from \(-radius- \{x1 y1 \ldots\}\)
x2 y2...xn yn] note it must begin and end in the center of a line not at a corner

superinsidestroke defines a non-symmetrical border or line [-widest-
-gray-...-narrowest- -gray-] each segment starts from the left edge

Project 15
MENU JUSTIFY

cm menujustify - accepts tabbed string pairs, converts into
dotted menu listings
cropleadingspaces boolean - useful when adapting preformatted fixed pitch
data to full proportional PostScript
curwide calculates the gonzo string width independent of printing it
cw calloutwidth - calculates only the width of the string,
returning it to the top of the stack.
drawmdots draws a line of menu dots, using show command
mdoteht height of dot used
menudots character used for dots
menufont default dot font
menudelim delimiter to separate columns
mlineproc processes the mline string into a leading string, the dot row and a trailing string, erases unused dots.
spchomp takes a string and conditionally removes all leading spaces, returning the truncated string

SUPERTABS
cst main callout supertab routine. grabs a long string and
breaks it into individual tab elements and formats them.
stabdelimiter decides when to move over to the next column. Popular values would be two or more spaces for dealing with imported listings or an actual [I] tab
stabnewline decides when to move down to the next line
stabs defines the supertab array
ytabadjust provides for extra room between lines on multiple line tab entries when needed

Project 16
cen center - use when you are doing several groups of center justified text
fonttouse font used for big stencil letters
letterheight height in inches for big stencil letters
letterwidth width in inches for big stencil letters - usually the same as the height
makebigletters command to actually create separate letters on separate sheets of paper
outlinetpath internal delay character
stall delay procedure
str single character string

Project 17
TENT

frontsideproc user defined tent front
gridresolution size of grid used
lowerleft horizontal starting point
lowerbot vertical starting point
rearsideproc user defined tent rear
tentproc actually does the work
xgridwidth width of grid used
ygridheight    height of grid used

AUTONUMBERING TICKETS

drawticks    puts tick marks at corners
horstart     horizontal starting point
inchoriz     width of each ticket
incvert      height of each ticket
numhoriz     number of tickets horizontally on page
numstring    string of numbers to be printed
numvert      number of tickets vertically on page
quickexit    allows exit in middle of page
runnum       current ticket number
vertstart    vertical starting point

Project 18

colcheck     what actually does the pagemakeup
footer       what goes at the bottom of the page
gonzojust    turns on the gonzo justification
header       what goes at the top of the page
lmpos        left margin
pagenum      starting page number
pulpnovel    template definition
rpagenum     running pagenumber adds one to each page
rslashok     boolean: prints the reverse slash
templatefonts fonts to be called in the pulpnovel template
templatestartup sets margins, gets header & footer

Project 19

colbot       column bottom position
colcheck     also makes room for figures and prints all but the last
page, it is the pagemaker.
colleft      first column left margin
colspacing   width of columns
coltop       column top position
figdict       a dictionary used to call the figures needs: [pagenum
colnum figheight figbot drawiftrue figname]
firstpagefooter boolean - turn on for a special first page footer
firstpageheader boolean - turn on for a special first page footer
footer       defines where the footer is placed
header       defines where the header is placed
leftfooter    information for the left page footer
leftheader    information for the left page header
leftpage      boolean - picks up firts page header and footer
makefigroom   lets the text flow around the figure spaces
numcolumns    number of columns
pagenum       starting page number
pagenumcheck  finds the 'pagenum' and replaces it with actual digits
pmnorm        paragraph margin
rightheader   information for the right page header
rightfooter   information for the right page footer
setup         defines and calls the page making routine, includes
colnum, xpos, figures, header, footer, firstpage info.
showfigs      goes through the figdict dictionary and shows the needed
figures on the required pages
Gonzo Justification Variables & Commands

Gonzo Variables

- **colcheck** Checks for column bottom; link to page or column maker.
- **cstretch** Minimum constant kerning of all characters (usually 0.2).
- **dropcount** Number of character lines needed by initial drop cap.
- **dropindent** Horizontal width reserved by initial drop cap.
- **headerx** Header xpos as used by template (optional).
- **headery** Header ypos as used by template (optional).
- **headerwide** Header txwide as used by template (optional).
- **kern** Adjusts the space between characters.
- **lastlinestretch** Slight stretch to last paragraph line.
- **maxcstretch** Maximum allowable character stretching.
- **maxsstretch** Maximum space stretch before character stretching begins.
- **overstrikechar** Overstrike character.
- **overstrikeht** Vertical shift of overstrike character.
- **pm** Normal paragraph indentation when used.
- **sstretch** Minimum constant kerning of all spaces (usually zero).
- **txtwide** Width of current column in points.
- **xpos** Left margin of current column, measured from left of page.
- **ybot** Vertical distance to base of bottom column line.
- **yinc** Vertical distance between column lines.
- **yparendadj** Extra vertical paragraph end ledding (usually zero).
- **ypos** Current vertical distance from bottom of page.
- **ytop** Vertical distance to base of top column line.

**Embeddable Gonzo Commands.**

- **0** Change to font0. Preferred use = initial drop cap.
- **1** Change to font1. Preferred use = normal body text.
- **2** Change to font2. Preferred use = italic body text.
- **3** Change to font3. Preferred use = bold body text.
- **4** Change to font4. Preferred use = all caps normal body text.
- **5** Change to font5. Preferred use = all caps italic body text.
- **6** Change to font6. Preferred use = all caps bold body text.
- **7** Change to font7. Preferred use = bold and dropped headers.
- **8** Change to font8. Preferred use = subscripts.
- **9** Change to font9. Preferred use = superscripts.
- **-** Change to font-. Preferred use = ZapfDingbats.
- **=** Change to font=. Preferred use = Symbol in body text size.

- **C** Change to center justification on xpos+txtwide.
- **F** Change to fill justification between xpos and xpos+txtwide.
- **L** Change to left justification starting at xpos.
- **R** Change to right justification ending at xpos+txtwide.

- **a** Perform user defined Macro amacro.
- **b** Perform user defined Macro bmacro.
- **c** Perform user defined Macro cmacro.
- **d** Perform user defined Macro dmacro.
| e | Perform user defined Macro `emacro`. |
| f | Perform user defined Macro `fmacro`. |
| g | Start gonzo justification. A PostScript alias for `startgonzo`. |
| h | Do a half of a negative linefeed. |
| i | Do an initial drop capital. |
| j | Do a positive kern. |
| k | Do a negative kern. |
| l | Do a full linefeed. |
| n | Nobreakconditional formfeed if within six lines of `ybot`. |
| o | Do an overstrike character. |
| p | Switch to normal paragraph margin, starting with next line. |
| s | Do a `showpage` for end of scanned file only. |
| x | Exit gonzo justify and switch to native PostScript. |
| y | Do a `yinc` up full negative linefeed. |
| z | Switch to zero paragraph margin, starting with next line. |