
Recycled METAFONT

Pierre A. MacKay

It is increasingly the fashion for publications to advertise themselves as ecologically exemplary and archivally antacid. Academic journals are especially under scrutiny in such matters, and we can expect that in the future almost all of them will feel the necessity of including something like

```
This journal is printed on recycled
paper meeting the minimum requirements
of American National Standard
for Information Sciences---Permanence
for Paper for Printed Library
Materials, ANSI Z39, 48--1984.
\goodpaper\recycle
```

on the inner cover or one of the other front-matter pages.

At Humanist Typesetting and Graphics, the ☉ `\goodpaper` symbol has been in use for quite a while. Theodora MacKay worked out the following some years ago, when the *Middle East Studies Bulletin* converted to non-acid paper, and it has proved serviceable ever since.

```
\def\goodpaper{%
\oalign{\hfil
\raise.25ex\hbox{%
$\scriptstyle\mathchar"231$}%
\hfil\crcr
\mathhexbox20D}}}
```

Something similar might be done with the Type-1 Symbol font, but since this font lacks the large open circle here set by `\mathhexbox20D` the effort would be painful.

The image for recycled paper is more of a problem since it cannot be generated from any existing glyphs in the METAFONT repertory. Nor is it as yet common among Type-1 fonts. So far, I have seen it only in a Linotype-Hell font, but since none of the other characters in that font was of interest, I decided to work out a new METAFONT version.

The symbol takes a number of forms, but all with a consistent theme. Three arrows, each of which is rolled back on itself over a 60° angle, are arranged to form an equilateral triangle. The arrows may be broad, (paper bags, boxes, publications) or narrow (several types of plastic). The broad form is usually shown in outline, or in white on a dark circular background. The point of the arrow in its broad forms is quite blunt, though in the narrow stick form it often appears in the conventional sharp style, with swept-back wings. The publication

which first asked for this symbol had previously used a photoreduction of the light image on a dark background, but it was clear that this took ink poorly, so we decided to follow the general model of the Linotype-Hell symbol, and draw the arrows in outline.

The construction starts out simply. The basic arrow (in this case the lower left) starts out heading straight to the left, then rolls up towards the viewer and doubles back at an angle to meet the tail of the second arrow. Sixty degree angles are easy to handle in METAFONT and the basic framework can quickly be constructed by treating the tail as if it were tightly folded over. To get the rolled effect, half of the folded area is cut away (on a line which can be seen in schematic form in the source code listing, lines 6–12).

This leaves far too many sharp corners, which need to be rounded. Here it is necessary to make some aesthetic choices. The obvious approach, which could be achieved by a geometric construction, would be to make a smooth curve carry round from the tail of the arrow to its eventual heading in the direction of the point. This, unfortunately, makes the tail end of the arrow look grossly unsymmetrical. In the dozen or so examples I have seen of this symbol, there is clearly an attempt to retain as symmetrical a shape for the tail as can be managed, consistent with the illusion that the arrow is rolling up out of the surface toward the viewer. Because the line from `a10` to `a9` is necessarily straight, the curve from `a8` to `a10` needs to have as large a straight component as can be managed. The values used as terminal control points for `a8`, `a10` and `a11` are fudges, arrived at by trial and error (lines 30–33). The curve that runs by point `a3` is a precise match for the curve that runs by point `a7`.

The criteria used for the shape of the head are that it should be as broad and blunt as possible, so long as it retains some sort of point even at 300 dpi. If the tail were thinner, the arrowhead could be designed a bit sharper, as it is in the rendition of this symbol on recent imprints of the O'Reilly & Associates handbooks. For our purposes, the broader arrow seemed preferable.

It would be nice to be able to rotate the completed arrow through two transforms and have the three arrows set in place that way, but the basic picture is far too complex, and produces METAFONT's plaintive message "That transformation is too hard." So, with careful attention to the model on page 138 of the METAFONTbook, all the points are transformed individually (lines 39–47), and the picture is twice redrawn.

```

1. cmchar "Recycle";
2. beginchar(Recycle,18u#,asc_height#,0);
3. italcarr 0;
4. adjust_fit(0,0);
5. numeric tail; tail=2.25cap_stem;
6. % * /
7. % / / Fold the tail of
8. % / /----- the arrow over
9. % \ / | itself, at an
10. % \ / | angle of 60 degrees
11. % \ /-----| Then cut design along
12. % * line *-----*
13. penpos1(tail,90); penpos2(tail,90); penpos4(tail,150);
14. % wings and point of arrowhead
15. penpos5(3.0cap_stem,150); penpos6(2.25cap_hair,60);
16. %
17. x1=.5w-.25u; x2=3u; bot y1l = bot y2l = 0;
18. x2'=x2-(tail / (sqrt 3)); % extend upper edge of tail by
19. % 1/2 base of equilateral triangle
20. y1'=y9=y2'=y1r; % z9 is point of acute angle made by this fold
21. x1-x1'=1.5(x2-x2'); % x1' is arbitrary point to set
22. % length of oblique arm
23. z4r=z1' rotatedaround(z2',60); % rotate to find location of pen4
24. z6=z5=z4; % all three pens on same center.
25. %
26. % Cut through construction on line *-----*
27. z3=whatever[z2',z4r]; x3=.5[x2',x2]; % top end of line *----*
28. z7=whatever[z2l,z4l]; y7=y2; % middle of line *----*
29. x8'-x2=x2-x2'; y8'=0; % bottom of line *----*
30. % Fudge start and end of curves.
31. x8=x8'+.35(x2-x2'); y8=0;
32. z10=whatever[z7,z4l]; z11=whatever[z7,z8'];x10=x11=x7+.80crisp;
33. z12=whatever[z3,z4r]; z13=whatever[z3,z7];x12=x13=x3+.80crisp;
34. %
35. % Find point of angle between two parts of tail and bisect it
36. z9=whatever[z2l,z4l]; % point of acute angle
37. z9-z0'=whatever*dir 30; y0'=h; % bisect it.
38. z0=whatever[z9,z0']; x0=.5w; % point for rotation on bisector
39. forsuffices $=a,b,c: transform $; endfor
40. a=identity;
41. b=identity rotatedaround(z0,120);
42. c=identity rotatedaround(z0,240);
43. pickup crisp.nib;
44. for n=1,2,3,4,5,6,7,8,9,10,11,12,13: forsuffices e=l,,r: forsuffices $=a,b,c:
45. z$[n]e = z[n]e transformed $; endfor endfor endfor
46. forsuffices $=a,b,c:
47. z$8' = z8' transformed $;
48. draw z$9--z$1r--z$1l--z$8{z$8-z$1l}..z$11{z$7-z$8'}..{z$10-z$7}z$10;
49. draw z$10--z$4l--z$5l--z$6r--z$5r--z$4r--
50. z$12{z$3-z$4r}..{z$7-z$3}z$13--z$7{z$8'-z$7}..{z$11-z$8}z$8;
51. labels($1,$1l,$1r,$2,$3,$4,$5l,$5r,$6r,$7,$8,$9,$10); endfor
52. labels(2',8'); endchar;
53.

```

“Recycle” is the first and, as yet, the only character in the HTG pi font. It is mapped to the uppercase “R”.

```
\font\htgpi=htgpi10 at 12pt
\def\recycle{\htgpi R}
```

The parameter file `htgpi10.mf` is a straight steal from `cmsy10` with only the initial comment lines and the final line changed. The driver file `pifont.mf` is modeled on `symbol.mf`, even to the point of retaining the early half of the file for slanted characters.

```
1. % pifont.mf Driver file for new
2. % characters. Parameters based on cmsy
3. font_coding_scheme="Font dependent";
4. mode_setup; font_setup;
5. font_slant slant;
6. font_x_height x_height#;
7. font_quad 18u#
8. if not monospace:+4letter_fit# fi;
9. % Slanted symbols here as needed
10. % Remaining characters unslanted
11. slant:=mono_charic#:=0;
12. currenttransform:=identity
13. yscaled aspect_ratio
14. scaled granularity;
15. Recycle:=ASCII"R";
16. input recycle
17. bye.
```

Who knows what other characters may be added in the future.

◇ Pierre A. MacKay
 Department of Classics DH-10
 Department of Near Eastern
 Languages and Civilization
 (DH-20)
 University of Washington
 Seattle, WA 98195 U.S.A.
 Phone: 206-543-2268;
 FAX:206-543-2267
 mackay@cs.washington.edu