

ABSTRACT

This report describes a Unix print filter designed to control an Hewlett Packard Laserjet or other printer that uses Hewlett Packard's Printer Control Language (HP-PCL). The filter gives users the ability to control print pitch, orientation, and indentation by using standard flags to the Unix `lpr` command or multiple entries in the `/etc/printcap` file and allows both ascii and binary (i.e., graphics and downloadable fonts) files to be printed. Additionally, the filter provides some accounting capability. The supported print pitch and orientation options are described, as are the different configuration options. The code for the filter is included in Appendix A and sample entries for the `/etc/printcap` file are included in Appendix B and C.

1. INTRODUCTION

As a part of the reconfiguration of computer resources in the Environmental Sciences Division (ESD) at Oak Ridge National Laboratory (ORNL), it was decided that an existing Digital Equipment Corporation (DEC) MicroVAX 2000 computer running Ultrix 4.0 would be used to provide printer and network support. An existing Hewlett Packard (HP) Laserjet 2000 printer was connected to the MicroVAX so that other computer systems in ESD and at ORNL, including mainframes, minicomputers, workstations, and personal computers, could generate output on the printer. Because the researchers in ESD need many different types of print styles and orientations, including the formatting of ascii files and the printing of binary HP-PCL graphics and typeset files, a Unix print filter was developed. The filter is written in C and should compile with either Kernighan & Ritchie C or ANSI C. The filter has been tested on an HP Laserjet 2000 and a QMS-PS 2000 with PCL4 emulation.

2. FILTER DESCRIPTION

Because ascii files contain only carriage returns that should be mapped to carriage return and line feed when sent to a printer and because binary files include line feeds explicitly, it is necessary for the filter to determine if the file being printed is ascii or binary. Because many commonly used word processing and typesetting programs (which output in binary format and may include downloaded fonts) send the reset command to the Laserjet (ESC-E) before anything else is sent, the filter was designed to detect this reset and, if it exists, to assume that a binary file is being printed. Binary files are passed directly to the printer. For ascii files, line feed characters are appended to carriage returns before they are passed to the printer. Because of the assumption made by the filter, ascii files (i.e., files that expect to have line feeds added to carriage returns) should not begin with the Laserjet reset string (ESC-E) and binary files must include the Laserjet reset string (ESC-E) as the first two characters of the file.

Accounting information is provided by the filter and will be written to the file specified in the `af=` line of the `/etc/printcap` entry discussed in the following. For each file processed, the filter will add a line to the accounting file containing the current date and time, the print width desired, the amount of indentation desired, the user's name, and the name of the computer system from which the user is printing. When used on a DEC Ultrix system, the accounting information also includes the job number, the job name, and the print queue to which the file was sent. Headers and trailers are also provided for in the filter implementation.

Because header and trailer pages are often difficult to identify when many printouts stack up in the output tray, the print filter is designed to use a different paper type (i.e., colored paper or thicker stock) for printing headers and trailers. The filter specifically requests the upper tray for printing header and trailer pages and then specifically requests the (bottom) paper deck for printing on $8\frac{1}{2} \times 11$ in. white paper. For this scheme to work, the upper tray should be kept full of $8\frac{1}{2} \times 11$ in. colored or special paper and the (bottom) paper deck should be kept full of $8\frac{1}{2} \times 11$ in. white paper. If the upper tray becomes empty, the printer will display a message asking for more paper in the upper tray. If paper is not added to the upper tray and the on-line button is pushed, the printer will begin printing header and trailer pages using the white $8\frac{1}{2} \times 11$ in. paper. If the (bottom) paper deck runs out of paper, the printer will automatically switch over to any other source of $8\frac{1}{2} \times 11$ in. paper; it will, therefore, print output on the colored or special paper. Care must be taken to ensure that paper trays and decks are continuously refilled with the correct type of paper. The filter assumes that the middle tray contains $8\frac{1}{2} \times 14$ in. paper.

2.1 USING THE FILTER

When using the filter, the default print style is the line printer font with 16.66 characters per inch (about 134 columns) on letter paper in a portrait orientation. This is the most versatile print style because it will accommodate line printer-type output from IBM mainframes and any other system that generates 132-column

output. Additional print styles are available with the filter. These print styles are shown in Table 1.

Table 1: Available Print Styles

Width field	Paper	Orientation	Font	Characters per inch
80	Letter	Portrait	Courier	10.0
96	Letter	Portrait	Courier	12.0
106	Letter	Landscape	Courier	10.0
127	Letter	Landscape	Courier	12.0
134	Letter	Portrait	Line Printer	16.66
136	Legal	Landscape	Courier	10.0
159	Letter	Landscape	Courier	15.0

These print styles are obtained by using the `-w` flag of the Unix `lpr` command from any host that supports it. Additionally, the `-i` flag of the Unix `lpr` command provides a method of indenting text to be printed. For example, the following command would print out text in 80 columns in the Courier font on letter paper in a portrait orientation:

```
% lpr -w80 filename
```

The following command would print out text in 96 columns with an indentation of 8 characters (leaving a total of 88 columns for the text) in the Courier font on letter paper in a portrait orientation:

```
% lpr -w96 -i8 filename
```

Because some systems support Berkeley `lpd` but their printing programs do not necessarily support the indent flag, additional styles, called “form styles” for the sake of clarity, have been added to the filter. These form styles are shown in Table 2.

Table 2: Available Form Styles

Width field	Width used (from Table 1)	Indentation
88	96	8
118	134	16
130	134	4
144	159	15

For example, the following command

```
% lpr -w88 filename
```

is equivalent to

```
% lpr -w96 -i8 filename
```

which is the second example command given in the preceding.

No indent or width flags should be used when printing binary files.

2.2 INSTALLING THE FILTER

2.2.1 Compiling the Filter

Set the following options in the `lj_if.c` code before compiling. If headers are desired, insert the line `#define HEADER` near the top of the file. If trailers are desired, insert the line `#define TRAILER` near the top of the file. If a form feed is to be appended to the end of every file, insert the line `#define PUT_FF_AT_EOF` near the top of the file. Replace the `#define` with `#undef` if the option is not desired. Because having both headers and trailers on printouts can lead to significant use of paper, the default option is to print headers but not trailers.

In addition, change the `route` variable to include an appropriate routing or other message for the header and trailer pages. After these options are set, compile the filter and then copy the executable file to an appropriate area. The example `/etc/printcap` entries discussed below expect the filter to be called `lj_if` and expect it to be located in the directory `/usr/local/lpdfilters`.

On most Unix-based computers, Berkeley lpd is used for printing. This system uses a file called `/etc/printcap` to hold configuration information about printers. If all the machines that will be accessing the printer have the `lpr -w` and `-i` flags available, the standard implementation with one `/etc/printcap` entry should be sufficient; however, if the Laserjet is to receive jobs from IBM mainframes or other machines that do not use conventional width and indentation flags, a special implementation with multiple `/etc/printcap` entries can be performed as needed to include all of the desired print and form styles.

2.2.2 Standard Implementation

To configure a machine for printing using the Laserjet print filter, add one of the entries shown in Appendix B to the `/etc/printcap` file on the machine connected to the Laserjet printer. The first entry shown in Appendix B is a general entry for all Unix-based machines, and the second entry is specifically for an Ultrix-based DEC machine. The Ultrix entry allows for some additional accounting information. Modify the `lp=` and `br=` lines as needed. See the Unix manual page for `printcap(5)` on the system.

After the appropriate `/etc/printcap` has been added on the machine, create the spool directory. This spool directory is called `/usr/spool/lpd/laserjet` in the example `printcap` entries. The directory should be owned by `daemon`. Next, use the `lpc` program to update the spooling information to avoid having to reboot the machine. Simply type `lpc stop laserjet` and then `lpc start laserjet`, and everything should be set to go.

This standard implementation is preferred; however, separate print queues can be established for the same printer to handle different print and form styles on print jobs from machines that do not support the `lpr -w` and `-i` flags. The special implementation is described in the following.

2.2.3 Special Implementation

To configure a machine for printing with the Laserjet print filter supporting queues for explicit print and form styles, use the entries shown in Appendix C. Appendix C includes entries for all the print and form styles described in the preceding. These queues can be used to obtain print and form styles without need for the `lpr -w` and `-i` flags; however, spool directories must be created for every entry added to the `/etc/printcap` file, and all print commands must include the name of the desired queue, usually by using the `lpr -P` flag. For example, to print text in 96 columns with an indentation of 8 characters (leaving a total of 88 columns for the text) in Courier font on letter paper in a portrait orientation, use the following command:

```
% lpr -Plj088 filename
```

It should be noted that the standard implementation with the use of the `lpr -w` and `-i` flags is preferred and that the special implementation exists only to support remote printing from hosts that cannot provide width and indent information in the print control file. Additionally, all the entries in Appendix C may not be necessary for every implementation, and the `lp=` and `br=` lines should be modified as needed. See the Unix manual page for `printcap(5)` on the system.

After the appropriate `/etc/printcap` entries have been added on the machine, create the necessary spool directories. These directories should be owned by `daemon`. Next, use the `lpc` program to update the spooling information to avoid having to reboot the machine. Simply type `lpc stop` and the names of the queues that were added to the `/etc/printcap` file. Finally, type `lpc start` and the names of the queues and everything should be set to go.

2.2.4 Obtaining the Filter Electronically

The print filter, associated documentation, and sample `/etc/printcap` entries can be obtained via anonymous ftp to `jupiter.esd.ornl.gov`. Use ftp to connect to jupiter, log in as `anonymous`, and use your electronic mail address as the password. Go to the `/pub/laserjet` directory, set the file transfer mode to binary, and get the desired files.

3. SUMMARY

This print filter gives users the ability to control print pitch, orientation, and indentation on an HP Laserjet printer by using the standard Berkeley lpd system via the Unix `lpr` program. Moreover, it stores accounting information useful to system managers for resource accounting, allocation, and planning. It is simple and easy to use and a necessity for a heterogeneous environment of networked computer systems.

4. APPENDIX A

The following is the code for `lj_if.c`.

8 Appendix A

```
1 #include <stdio.h>
2 #include <strings.h>
3 #include <time.h>
4 #define ESC '\033'
5 #define FF '\014'
6 #define HEADER
7 #undef TRAILER
8 #undef PUT_FF_AT_EOF
9
10 /* Put an appropriate message here */
11 char *routeto=" E.S.D. Building 1505 ";
12
13 /*****
14 * Input filter for use with Berkley lpd on Unix hosts. Controls an HP
15 * Laserjet printer and adds line feeds to carriage returns if an ascii
16 * file is printed. It assumes that if ESC and E (the Laserjet's reset
17 * command) are the first two characters in the file then the file is
18 * binary and line feeds are not added. Otherwise, it will add line
19 * feeds after carriage returns. The default width is 132 (actually
20 * 134) columns.
21 *
22 *****/
23 * Mon Apr 15 11:24:02 EDT 1991
24 *
25 * Forrest Hoffman
26 * Oak Ridge National Laboratory
27 * Environmental Sciences Division
28 * Building 1505, MS 6036
29 * Oak Ridge TN 37831-6036
30 * forrest@esdhof.esd.ornl.gov
31 * (615) 576-7680
32 * (615) 574-7314 message
33 *****/
34
35
36 /* hpreset routines */
37 /* Orientation */
38 #define PORTRAIT 0
39 #define LANDSCAPE 1
40
41 /* Symbol Set */
42 char *hp_math_7 = "0A";
43 char *hp_line_draw = "0B";
44 char *iso_60_norwegian_v1 = "0D";
45 char *iso_61_norwegian_v2 = "1D";
46 char *hp_roman_extension = "0E";
47 char *iso_4_united_kingdom = "1E";
48 char *iso_25_french = "0F";
49 char *iso_69_french = "1F";
50 char *hp_german = "0G";
51 char *iso_21_german = "1G";
52 char *hp_greek_8 = "8G";
53 char *iso_15_italian = "0I";
54 char *iso_14_jis_ascii = "0K";
55 char *iso_57_chinese = "2K";
56 char *technical_7 = "1M";
57 char *hp_math_8 = "8M";
58 char *iso_100_ecma_96 = "0N"; /* Latin 1 */
59 char *ocr_a = "00";
60 char *ocr_b = "10";
61 char *iso_11_swedish = "0S";
62 char *hp_spanish = "1S";
63 char *iso_17_spanish = "2S";
64 char *iso_10_swedish = "3S";
65 char *iso_16_portuguese = "4S";
66 char *iso_84_portuguese = "5S";
67 char *iso_85_spanish = "6S";
68 char *iso_6_ascii = "0U";
69 char *hp_legal = "1U";
70 char *iso_2_international_reference_version = "2U";
71 char *oem_1 = "7U";
72 char *hp_roman_8 = "8U";
```

```

73 char *pc_8 = "10U";
74 char *pc_8_dn = "11U";
75 char *hp_pi_font = "15U";
76
77 /* Spacing */
78 #define FIXED          0
79 #define PROPORTIONAL  1
80
81 /* Pitch */
82 #define PICA           10
83 #define ELITE         12
84 #define SMALL         15
85 #define COMPRESSED    16.66
86
87 /* Line Spacing */
88 #define LPI_1         1      /* Lines per inch */
89 #define LPI_2         2
90 #define LPI_3         3
91 #define LPI_4         4
92 #define LPI_6         6      /* Default */
93 #define LPI_8         8
94 #define LPI_12        12
95 #define LPI_16        16
96 #define LPI_24        24
97 #define LPI_48        48
98
99 /* Page Size */
100 #define EXECUTIVE     1
101 #define LETTER        2
102 #define LEGAL         3
103 #define LEDGER        6
104 #define A4            26
105 #define A3            27
106
107 /* Style */
108 #define UPRIGHT 0
109 #define ITALIC  1
110
111 /* Stroke Weight */
112 #define ULTRA_THIN  -7
113 #define THIN        -5
114 #define LIGHT       -3
115 #define MEDIUM      0
116 #define BOLD         3
117 #define BLACK        5
118 #define ULTRA_BLACK  7
119
120 /* Typeface */
121 #define LINE_PRINTER  0
122 #define COURIER       3
123 #define HELV          4
124 #define TMS_RMN      5
125 #define LETTER_GOTHIC 6
126 #define PRESTIGE     8
127 #define PRESENTATIONS 11
128 #define OPTIMA       17
129 #define GARAMOND     18
130 #define COOPER_BLACK 19
131 #define CORONET_BOLD 20
132 #define BROADWAY     21
133 #define BAUER_BODONI_BLACK_CONDENSED 22
134 #define CENTURY_SCHOOLBOOK 23
135 #define UNIVERSITY_ROMAN 24
136
137 /* End-of-Line Wrap */
138 #define ENABLE_EOL_WRAP 0
139 #define DISABLE_EOL_WRAP 1
140 /* The factory default is end-of-line wrap disabled. */
141
142 void widthfix(width, indent)
143 int *width, *indent;
144 {
145     switch(*(width)) {

```

10 *Appendix A*

```

146         case 88:
147             *width = 96;
148             *indent = 8;
149             break;
150         case 118:
151             *width = 134;
152             *indent = 16;
153             break;
154         case 130:
155             *width = 134;
156             *indent = 4;
157             break;
158         case 144:
159             *width = 159;
160             *indent = 15;
161             break;
162     }
163     return;
164 }
165
166 void hpreset(width)
167 int width;
168 {
169     /* Reset Printer */
170     printf("%cE", ESC);
171
172     switch(width) {
173         case 80:
174             printf("%c&l%dA", ESC, LETTER); /* Page Size */
175             printf("%c&l%dD", ESC, LPI_6); /* Line Spacing */
176             printf("%c&l%dO", ESC, PORTRAIT); /* Orientation */
177             printf("%c&l%dP", ESC, 66); /* Page Length */
178             printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
179             printf("%c(s%dH", ESC, PICA); /* Primary Pitch */
180             break;
181         case 96:
182             printf("%c&l%dA", ESC, LETTER); /* Page Size */
183             printf("%c&l%dD", ESC, LPI_6); /* Line Spacing */
184             printf("%c&l%dO", ESC, PORTRAIT); /* Orientation */
185             printf("%c&l%dP", ESC, 66); /* Page Length */
186             printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
187             printf("%c(s%dH", ESC, ELITE); /* Primary Pitch */
188             break;
189         case 106:
190             printf("%c&l%dA", ESC, LETTER); /* Page Size */
191             printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
192             printf("%c&l%dP", ESC, 51); /* Page Length */
193             printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
194             printf("%c&l%dD", ESC, LPI_8); /* Line Spacing */
195             printf("%c(s%dH", ESC, PICA); /* Primary Pitch */
196             break;
197         case 127:
198             printf("%c&l%dA", ESC, LETTER); /* Page Size */
199             printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
200             printf("%c&l%dP", ESC, 51); /* Page Length */
201             printf("%c(s%dT", ESC, COURIER);
202             /* Primary Typeface */
203             printf("%c&l%dD", ESC, LPI_8); /* Line Spacing */
204             printf("%c(s%dH", ESC, ELITE); /* Primary Pitch */
205             break;
206         case 136:
207             printf("%c&l%dA", ESC, LEGAL); /* Page Size */
208             /* have to trick the printer by swichting page length
209             and orientation */
210             printf("%c&l%dP", ESC, 84); /* Page Length */
211             printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
212             printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
213             printf("%c&l%dD", ESC, LPI_8); /* Line Spacing */
214             printf("%c(s%dH", ESC, PICA); /* Primary Pitch */
215             break;

```

```

216         case 159:
217             printf("%c&l%dA", ESC, LETTER); /* Page Size */
218             printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
219             printf("%c&l%dP", ESC, 51); /* Page Length */
220             printf("%c(s%dT", ESC, COURIER);
221             /* Primary Typeface */
222             printf("%c&l%dD", ESC, LPI_8); /* Line Spacing */
223             printf("%c(s%dH", ESC, SMALL); /* Primary Pitch */
224             break;
225         default:
226             printf("%c&l%dA", ESC, LETTER); /* Page Size */
227             printf("%c&l%dD", ESC, LPI_6); /* Line Spacing */
228             printf("%c&l%dO", ESC, PORTRAIT); /* Orientation */
229             printf("%c&l%dP", ESC, 66); /* Page Length */
230             printf("%c(s%dT", ESC, LINE_PRINTER);
231             /* Primary Typeface */
232             printf("%c(s%dH", ESC, COMPRESSED); /* Primary Pitch */
233     }
234
235     /* Primary Symbol Set */
236     printf("%c(%s", ESC, hp_roman_8);
237     /* Secondary Symbol Set */
238     printf("%c)%s", ESC, hp_roman_8);
239
240     /* Primary Spacing */
241     printf("%c(s%dP", ESC, FIXED);
242     /* Secondary Spacing */
243     printf("%c)s%dP", ESC, FIXED);
244
245     /* Secondary Pitch */
246     printf("%c)s%dH", ESC, COMPRESSED);
247
248     /* Primary Style */
249     printf("%c(s%dS", ESC, UPRIGHT);
250     /* Secondary Style */
251     printf("%c)s%dS", ESC, UPRIGHT);
252
253     /* Primary Stroke Weight */
254     printf("%c(s%dB", ESC, MEDIUM);
255     /* Secondary Stroke Weight */
256     printf("%c)s%dB", ESC, MEDIUM);
257
258     /* Secondary Typeface */
259     printf("%c)s%dT", ESC, LINE_PRINTER);
260
261     /* End-of-Line Wrap */
262     printf("%c&s%dc", ESC, DISABLE_EOL_WRAP);
263 }
264
265
266 void header(width, user, host, jobnum, jobname, printer)
267 int width, jobnum;
268 char *user, *host, *jobname, *printer;
269 {
270     int i, j, k;
271     long time(), ts;
272     char *t;
273
274     ts = time((long *)0);
275     t = ctime(&ts);
276     t[24] = '\0';
277
278     /* Print headers and trailers on colored paper in upper tray then
279        return to paper deck */
280     printf("%c&l1H", ESC);
281
282     for (i = 0; i < 3; ++i) {
283         for (j = 0; j < 5; ++j) {
284             printf("BEGINNING OF OUTPUT ");
285             printf("Printed on %s ", t);

```

12 *Appendix A*

```

286             printf("BEGINNING OF OUTPUT  ");
287             printf("\n");
288             putchar('\r');
289         }
290         printf("\n");
291         putchar('\r');
292         printf("                User: %s\n", user);
293         putchar('\r');
294         printf("                Host: %s\n", host);
295         putchar('\r');
296         printf("                Job #: %ld\n", jobnum);
297         putchar('\r');
298         printf("                Jobname: %s\n", jobname);
299         putchar('\r');
300         printf("                Printer: %s\n", printer);
301         putchar('\r');
302         printf("\n");
303         putchar('\r');
304         for ( j = 0; j < 5; ++j) {
305             for (k = 0; k < 3; ++k)
306                 printf("%s", routeto);
307             printf("\n");
308             putchar('\r');
309         }
310         printf("\n");
311         putchar('\r');
312     }
313     printf("%c",FF);
314
315     /* Return to paper deck */
316     printf("%c&15H", ESC);
317 }
318
319 void trailer(width, user, host, jobnum, jobname, printer)
320 int width, jobnum;
321 char *user, *host, *jobname, *printer;
322 {
323     int i, j, k;
324     long time(), ts;
325     char *t;
326
327     ts = time((long *)0);
328     t = ctime(&ts);
329     t[24] = '\0';
330
331     /* Print headers and trailers on colored paper in upper tray then
332        return to paper deck */
333     printf("%c&11H", ESC);
334
335     for (i = 0; i < 3; ++i) {
336         for (j = 0; j < 5; ++j) {
337             printf("        END OF OUTPUT  ");
338             printf("Printed on %s ",t);
339             printf("        END OF OUTPUT  ");
340             printf("\n");
341             putchar('\r');
342         }
343         printf("\n");
344         putchar('\r');
345         printf("                User: %s\n", user);
346         putchar('\r');
347         printf("                Host: %s\n", host);
348         putchar('\r');
349         printf("                Job #: %ld\n", jobnum);
350         putchar('\r');
351         printf("                Jobname: %s\n", jobname);
352         putchar('\r');
353         printf("                Printer: %s\n", printer);

```

```

354         putchar('\r');
355         printf("\n");
356         putchar('\r');
357         for ( j = 0; j < 5; ++j) {
358             for (k = 0; k < 3; ++k)
359                 printf("%s", routeto);
360             printf("\n");
361             putchar('\r');
362         }
363         printf("\n");
364         putchar('\r');
365     }
366     printf("%c",FF);
367
368     /* Return to paper deck */
369     printf("%c&l5H", ESC);
370 }
371
372 main(argc,argv)
373 int argc;
374 char **argv;
375 {
376     FILE *myfl;
377     int i, c1, c2, c, bump, width, indent, jobnum;
378     long time(), ts;
379 #ifdef PUT_FF_AT_EOF
380     int lastc;
381 #endif
382     static char *user, *host, *filename, *t, *jobname, *printer;
383
384     if (argc > 6) {
385         if (strcmp(argv[1], "-c") == 0)
386             bump = 1;
387         else
388             bump = 0;
389         sscanf(argv[1+bump], "-w%d", &width);
390         sscanf(argv[3+bump], "-i%d", &indent);
391         user = argv[5+bump];
392         host = argv[7+bump];
393         filename = argv[8+bump];
394         sscanf(argv[9+bump], "%ld", &jobnum);
395         jobname = argv[10+bump];
396         printer = argv[11+bump];
397     }
398     else {
399         width = 132;
400         indent = 0;
401         user = "unknown";
402         host = "unknown";
403     }
404
405     /* Preprocess width and indent information for special cases. */
406     widthfix(&width, &indent);
407
408     if ((myfl = fopen(filename, "a")) != NULL) {
409         ts = time((long *)0);
410         t = ctime(&ts);
411         t[24] = '\0';
412         fprintf(myfl, "%s %3d %3d %s@s %5d %s %s\n", t, width, indent,
413             user, host, jobnum, jobname, printer);
414         fclose(myfl);
415     }
416
417     hpreset(80);
418 #ifdef HEADER
419     header(width, user, host, jobnum, jobname, printer);
420 #endif
421     hpreset(width);
422
423     if ( (c1 = getchar()) == '\033' && (c2 = getchar()) == 'E' ) {

```

14 *Appendix A*

```

424             /* binary file so don't mess with it */
425             while ( (c = getchar()) != EOF)
426                 putchar(c);
427         }
428         else {
429             if (c1 == EOF)
430                 exit(0); /* premature EOF */
431             putchar('\r');
432             for (i = 0; i < indent; ++i)
433                 putchar(' ');
434             putchar(c1);
435             if (c1 == '\n') {
436                 putchar('\r');
437                 for (i = 0; i < indent; ++i)
438                     putchar(' ');
439             }
440             if (c2 != EOF) {
441                 putchar(c2);
442                 if (c2 == '\n') {
443                     putchar('\r');
444                     for (i = 0; i < indent; ++i)
445                         putchar(' ');
446                 }
447             }
448             while ( (c = getchar()) != EOF) {
449                 putchar(c);
450 #ifdef PUT_FF_AT_EOF
451                 lastc = c;
452 #endif
453                 if (c == '\n') {
454                     putchar('\r');
455                     for (i = 0; i < indent; ++i)
456                         putchar(' ');
457                 }
458             }
459 #ifdef PUT_FF_AT_EOF
460             if (lastc != '\f')
461                 putchar('\f');
462 #endif
463         }
464
465         hpreset(80);
466 #ifdef TRAILER
467         trailer(width, user, host, jobnum, jobname, printer);
468 #endif
469
470         putchar('\033'); /* reset printer */
471         putchar('E');
472
473         exit(0);
474 }

```

5. APPENDIX B

Add one of the following entries to the `/etc/printcap` file to use the print filter. Modify the `lp=` and `br=` lines as needed. See the Unix manual page for `printcap(5)` on the system.

16 *Appendix B*

```
1 # /etc/printcap entry for a non-Ultrix machine:
2 lp0|laserjet|lj|lj2000|HP Laserjet 2000:\
3     :af=/usr/adm/lpacct:\
4     :br#19200:\
5     :fc#0177777:\
6     :fs#03:\
7     :if=/usr/local/lpdfilters/lj_if:\
8     :lf=/usr/adm/lpd-errs:\
9     :lp=/dev/tty01:\
10    :mx#0:\
11    :pw#132:\
12    :pl#60:\
13    :sd=/usr/spool/lpd/laserjet:\
14    :sh:\
15    :xc#0177777:\
16    :xs#044000:
17 # /etc/printcap entry for an Ultrix machine:
18 lp0|laserjet|lj|lj2000|HP Laserjet 2000:\
19    :af=/usr/adm/lpacct:\
20    :br#19200:\
21    :fc#0177777:\
22    :fs#03:\
23    :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
24    :lf=/usr/adm/lpd-errs:\
25    :lp=/dev/tty01:\
26    :mx#0:\
27    :pw#132:\
28    :pl#60:\
29    :sd=/usr/spool/lpd/laserjet:\
30    :sh:\
31    :uv=psv1.0:\
32    :xc#0177777:\
33    :xs#044000:
```

6. APPENDIX C

Use the following entries to provide alternate print and form styles for machines that do not support the `lpr -w` and `-i` flags. These entries are for Ultrix-based machines; for non-Ultrix platforms remove the `uv=` line and remove the flags passed to `lj_if` on the `if=` line. See Appendix B for a comparison of generic and Ultrix-specific implementations of the `/etc/printcap` file.

18 Appendix C

```
1 lp0|lj134|esdlj134|ESDLJ134|laserjet|HP Laserjet 2000, Building 1505, Room 121:\
2 :af=/usr/adm/lpacct:\
3 :br#19200:\
4 :fc#0177777:\
5 :fs#03:\
6 :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
7 :lf=/usr/adm/lpd-errs:\
8 :lp=/dev/tty01:\
9 :mx#0:\
10 :pw#132:\
11 :pl#60:\
12 :sd=/usr/spool/lpd/laserjet:\
13 :sh:\
14 :uv=psv1.0:\
15 :xc#0177777:\
16 :xs#044000:
17 lj80|esdlj080|ESDLJ080|HP Laserjet 2000 with 80 columns:\
18 :af=/usr/adm/lpacct:\
19 :br#19200:\
20 :fc#0177777:\
21 :fs#03:\
22 :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
23 :lf=/usr/adm/lpd-errs:\
24 :lp=/dev/tty01:\
25 :mx#0:\
26 :pw#80:\
27 :pl#60:\
28 :sd=/usr/spool/lpd/lj80:\
29 :sh:\
30 :uv=psv1.0:\
31 :xc#0177777:\
32 :xs#044000:
33 lj106|esdlj106|ESDLJ106|HP Laserjet 2000 with 106 columns:\
34 :af=/usr/adm/lpacct:\
35 :br#19200:\
36 :fc#0177777:\
37 :fs#03:\
38 :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
39 :lf=/usr/adm/lpd-errs:\
40 :lp=/dev/tty01:\
41 :mx#0:\
42 :pw#106:\
43 :pl#60:\
44 :sd=/usr/spool/lpd/lj106:\
45 :sh:\
46 :uv=psv1.0:\
47 :xc#0177777:\
48 :xs#044000:
49 lj127|esdlj127|ESDLJ127|HP Laserjet 2000 with 127 columns:\
50 :af=/usr/adm/lpacct:\
51 :br#19200:\
52 :fc#0177777:\
53 :fs#03:\
54 :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
55 :lf=/usr/adm/lpd-errs:\
56 :lp=/dev/tty01:\
57 :mx#0:\
58 :pw#127:\
59 :pl#60:\
60 :sd=/usr/spool/lpd/lj127:\
61 :sh:\
62 :uv=psv1.0:\
63 :xc#0177777:\
64 :xs#044000:
65 lj136|esdlj136|ESDLJ136|HP Laserjet 2000 with 136 columns:\
66 :af=/usr/adm/lpacct:\
67 :br#19200:\
68 :fc#0177777:\
69 :fs#03:\
```

```

70      :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
71      :lf=/usr/adm/lpd-errs:\
72      :lp=/dev/tty01:\
73      :mx#0:\
74      :pw#136:\
75      :pl#60:\
76      :sd=/usr/spool/lpd/lj136:\
77      :sh:\
78      :uv=psv1.0:\
79      :xc#0177777:\
80      :xs#044000:
81 lj130|esdlj130|ESDLJ130|HP Laserjet 2000 with 130 columns:\
82      :af=/usr/adm/lpacct:\
83      :br#19200:\
84      :fc#0177777:\
85      :fs#03:\
86      :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
87      :lf=/usr/adm/lpd-errs:\
88      :lp=/dev/tty01:\
89      :mx#0:\
90      :pw#130:\
91      :pl#60:\
92      :sd=/usr/spool/lpd/lj130:\
93      :sh:\
94      :uv=psv1.0:\
95      :xc#0177777:\
96      :xs#044000:
97 lj159|esdlj159|ESDLJ159|HP Laserjet 2000 with 159 columns:\
98      :af=/usr/adm/lpacct:\
99      :br#19200:\
100     :fc#0177777:\
101     :fs#03:\
102     :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
103     :lf=/usr/adm/lpd-errs:\
104     :lp=/dev/tty01:\
105     :mx#0:\
106     :pw#159:\
107     :pl#60:\
108     :sd=/usr/spool/lpd/lj159:\
109     :sh:\
110     :uv=psv1.0:\
111     :xc#0177777:\
112     :xs#044000:
113 lj144|esdlj144|ESDLJ144|HP Laserjet 2000 with 144 columns:\
114     :af=/usr/adm/lpacct:\
115     :br#19200:\
116     :fc#0177777:\
117     :fs#03:\
118     :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
119     :lf=/usr/adm/lpd-errs:\
120     :lp=/dev/tty01:\
121     :mx#0:\
122     :pw#144:\
123     :pl#60:\
124     :sd=/usr/spool/lpd/lj144:\
125     :sh:\
126     :uv=psv1.0:\
127     :xc#0177777:\
128     :xs#044000:
129 lj118|esdlj118|ESDLJ118|HP Laserjet 2000 with 118 columns:\
130     :af=/usr/adm/lpacct:\
131     :br#19200:\
132     :fc#0177777:\
133     :fs#03:\
134     :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
135     :lf=/usr/adm/lpd-errs:\
136     :lp=/dev/tty01:\
137     :mx#0:\
138     :pw#118:\

```

20 *Appendix C*

```
139      :pl#60:\
140      :sd=/usr/spool/lpd/lj118:\
141      :sh:\
142      :uv=psv1.0:\
143      :xc#0177777:\
144      :xs#044000:
145 lj96|esdlj096|ESDLJ096|HP Laserjet 2000 with 96 columns:\
146      :af=/usr/adm/lpacct:\
147      :br#19200:\
148      :fc#0177777:\
149      :fs#03:\
150      :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%i -n %U -h %H %A %J %j %P:\
151      :lf=/usr/adm/lpd-errs:\
152      :lp=/dev/tty01:\
153      :mx#0:\
154      :pw#96:\
155      :pl#60:\
156      :sd=/usr/spool/lpd/lj96:\
157      :sh:\
158      :uv=psv1.0:\
159      :xc#0177777:\
160      :xs#044000:
161 lj88|esdlj088|ESDLJ088|HP Laserjet 2000 with 88 columns:\
162      :af=/usr/adm/lpacct:\
163      :br#19200:\
164      :fc#0177777:\
165      :fs#03:\
166      :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%i -n %U -h %H %A %J %j %P:\
167      :lf=/usr/adm/lpd-errs:\
168      :lp=/dev/tty01:\
169      :mx#0:\
170      :pw#88:\
171      :pl#60:\
172      :sd=/usr/spool/lpd/lj88:\
173      :sh:\
174      :uv=psv1.0:\
175      :xc#0177777:\
176      :xs#044000:
```

INTERNAL DISTRIBUTION

- | | |
|----------------------|---|
| 1. L. D. Bates | 28. W. M. Post |
| 2. R. N. Bolling | 29. D. E. Reichle |
| 3. J. B. Cannon | 30. F. E. Sharples |
| 4. J. H. Cushman | 31. D. S. Shriner |
| 5. T. H. Dunigan | 32. S. H. Stow |
| 6. W. R. Emanuel | 33. R. I. Van Hook |
| 7. D. E. Fowler | 34-48. ESD Library |
| 8. C. W. Gehrs | 49. ORNL Y-12 Technical
Library |
| 9. J. A. Green | 50-51. Laboratory Records
Department |
| 10. N. A. Griffith | 52. Laboratory Records,
ORNL-RC |
| 11. C. E. Hammons | 53. Document Reference
Section |
| 12. S. G. Hildebrand | 54. Central Research Library |
| 13. G. C. Hinkel | 55. ORNL Patent Section |
| 14-23. F. M. Hoffman | |
| 24. G. K. Jacobs | |
| 25. P. Kanciruk | |
| 26. A. W. King | |
| 27. L. W. Littleton | |

EXTERNAL DISTRIBUTION

56. W. E. Blass, Department of Physics and Astronomy, University of Tennessee, Nielsen Physics Building, Knoxville TN 37990
57. R. A. Crowe, Department of Physics and Astronomy, University of Hawaii at Hilo, 523 West Lanikaula St., Hilo, HI 96720-4091
58. J. F. Franklin, Bloedel Professor of Ecosystem Analysis, College of Forest Resources, University of Washington, Anderson Hall AR-10, Seattle, WA 98195
59. R. C. Harriss, Institute for the Study of Earth, Oceans, and Space, Science and Engineering Research Building, University of New Hampshire, Durham, NH 03824
60. G. Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
61. R. H. Olsen, Professor, Microbiology and Immunology Department, University of Michigan, Medical Sciences II, #5605, 1301 East Catherine Street, Ann Arbor, MI 48109-0620
62. A. Patrinos, Director, Environmental Sciences Division, Office of Health and Environmental Research, ER-74, U.S. Department of Energy, Washington, DC 20585
63. F. J. Wobber, Environmental Sciences Division, Office of Health and Environmental Research, ER-74, U.S. Department of Energy, Washington, DC 20585
64. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy Oak Ridge Field Office, P.O. Box 2001, Oak Ridge, TN 37831-8600
- 65-74. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831

ORNL/TM-12190

Environmental Sciences Division

**A Unix Print Filter for Controlling
an HP Laserjet Printer**

F. M. Hoffman

Environmental Sciences Division
Publication No. 3959

DATE PUBLISHED — September 1992

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6035
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615)576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

NTIS price codes—Printed Copy: \$0.00 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

“This submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.”

CONTENTS

ABSTRACT	v
1. INTRODUCTION	1
2. FILTER DESCRIPTION	2
2.1 USING THE FILTER	2
2.2 INSTALLING THE FILTER	4
2.2.1 Compiling the Filter	4
2.2.2 Standard Implementation	4
2.2.3 Special Implementation	5
2.2.4 Obtaining the Filter Electronically	5
3. SUMMARY	6
4. APPENDIX A	7
5. APPENDIX B	15
6. APPENDIX C	17