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 `routeto` 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 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`.

```
#include <stdio.h>
#include <strings.h>
#include <time.h>
#define ESC '\033'
#define FF '\014'
#define HEADER
#undef TRAILER
#undef PUT_FF_AT_EOF

/* Put an appropriate message here */
char *routeto="  E.S.D. Building 1505  ";

/***************************************************************************
 * Input filter for use with Berkley lpd on Unix hosts.  Controls an HP    *
 * Laserjet printer and adds line feeds to carriage returns if an ascii    *
 * file is printed.  It assumes that if ESC and E (the Laserjet's reset    *
 * command) are the first two characters in the file then the file is     *
 * binary and line feeds are not added.  Otherwise, it will add line       *
 * feeds after carriage returns.  The default width is 132 (actually       *
 * 134) columns.                                                           *
 *                                                                         *
 ***************************************************************************
 * Mon Apr 15 11:24:02 EDT 1991                                            *
 *                                                                         *
 * Forrest Hoffman                                                         *
 * Oak Ridge National Laboratory                                           *
 * Environmental Sciences Division                                         *
 * Building 1505, MS 6036                                                  *
 * Oak Ridge TN 37831-6036                                                 *
 * forrest@esdhof.esd.ornl.gov                                             *
 * (615) 576-7680                                                          *
 * (615) 574-7314 message                                                  *
 ***************************************************************************/


/* hpreset routines */
/* Orientation */
#define PORTRAIT        0
#define LANDSCAPE       1

/* Symbol Set */
char *hp_math_7 = "0A";
char *hp_line_draw = "0B";
char *iso_60_norwegian_v1 = "0D";
char *iso_61_norwegian_v2 = "1D";
char *hp_roman_extension = "0E";
char *iso_4_united_kingdom = "1E";
char *iso_25_french = "0F";
char *iso_69_french = "1F";
char *hp_german = "0G";
char *iso_21_german = "1G";
char *hp_greek_8 = "8G";
char *iso_15_italian = "0I";
char *iso_14_jis_ascii = "0K";
char *iso_57_chinese = "2K";
char *technical_7 = "1M";
char *hp_math_8 = "8M";
char *iso_100_ecma_96 = "0N";   /* Latin 1 */
char *ocr_a = "0O";
char *ocr_b = "1O";
char *iso_11_swedish = "0S";
char *hp_spanish = "1S";
char *iso_17_spanish = "2S";
char *iso_10_swedish = "3S";
char *iso_16_portuguese = "4S";
char *iso_84_portuguese = "5S";
char *iso_85_spanish = "6S";
char *iso_6_ascii = "0U";
char *hp_legal = "1U";
char *iso_2_international_reference_version = "2U";
char *oem_1 = "7U";
char *hp_roman_8 = "8U";
```

```
char *pc_8 = "10U";
char *pc_8_dn = "11U";
char *hp_pi_font = "15U";

/* Spacing */
#define FIXED           0
#define PROPORTIONAL    1

/* Pitch */
#define PICA            10
#define ELITE           12
#define SMALL           15
#define COMPRESSED      16.66

/* Line Spacing */
#define LPI_1           1       /* Lines per inch */
#define LPI_2           2
#define LPI_3           3
#define LPI_4           4
#define LPI_6           6       /* Default */
#define LPI_8           8
#define LPI_12          12
#define LPI_16          16
#define LPI_24          24
#define LPI_48          48

/* Page Size */
#define EXECUTIVE       1
#define LETTER          2
#define LEGAL           3
#define LEDGER          6
#define A4              26
#define A3              27

/* Style */
#define UPRIGHT 0
#define ITALIC  1

/* Stroke Weight */
#define ULTRA_THIN      -7
#define THIN            -5
#define LIGHT           -3
#define MEDIUM          0
#define BOLD            3
#define BLACK           5
#define ULTRA_BLACK     7

/* Typeface */
#define LINE_PRINTER    0
#define COURIER         3
#define HELV            4
#define TMS_RMN         5
#define LETTER_GOTHIC   6
#define PRESTIGE        8
#define PRESENTATIONS   11
#define OPTIMA          17
#define GARAMOND        18
#define COOPER_BLACK    19
#define CORONET_BOLD    20
#define BROADWAY        21
#define BAUER_BODONI_BLACK_CONDENSED    22
#define CENTURY_SCHOOLBOOK      23
#define UNIVERSITY_ROMAN        24

/* End-of-Line Wrap */
#define ENABLE_EOL_WRAP         0
#define DISABLE_EOL_WRAP        1
/* The factory default is end-of-line wrap disabled. */

void widthfix(width, indent)
int *width, *indent;
{
        switch(*(width)) {
```

```
                case 88:
                        *width = 96;
                        *indent = 8;
                        break;
                case 118:
                        *width = 134;
                        *indent = 16;
                        break;
                case 130:
                        *width = 134;
                        *indent = 4;
                        break;
                case 144:
                        *width = 159;
                        *indent = 15;
                        break;
        }
        return;
}

void hpreset(width)
int width;
{
        /* Reset Printer */
        printf("%cE", ESC);

        switch(width) {
                case 80:
                        printf("%c&l%dA", ESC, LETTER); /* Page Size */
                        printf("%c&l%dD", ESC, LPI_6);  /* Line Spacing */
                        printf("%c&l%dO", ESC, PORTRAIT); /* Orientation */
                        printf("%c&l%dP", ESC, 66);     /* Page Length */
                        printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
                        printf("%c(s%dH", ESC, PICA);   /* Primary Pitch */
                        break;
                case 96:
                        printf("%c&l%dA", ESC, LETTER); /* Page Size */
                        printf("%c&l%dD", ESC, LPI_6);  /* Line Spacing */
                        printf("%c&l%dO", ESC, PORTRAIT); /* Orientation */
                        printf("%c&l%dP", ESC, 66);     /* Page Length */
                        printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
                        printf("%c(s%dH", ESC, ELITE);  /* Primary Pitch */
                        break;
                case 106:
                        printf("%c&l%dA", ESC, LETTER); /* Page Size */
                        printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
                        printf("%c&l%dP", ESC, 51);     /* Page Length */
                        printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
                        printf("%c&l%dD", ESC, LPI_8);  /* Line Spacing */
                        printf("%c(s%dH", ESC, PICA);   /* Primary Pitch */
                        break;
                case 127:
                        printf("%c&l%dA", ESC, LETTER); /* Page Size */
                        printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
                        printf("%c&l%dP", ESC, 51);     /* Page Length */
                        printf("%c(s%dT", ESC, COURIER);
                                /* Primary Typeface */
                        printf("%c&l%dD", ESC, LPI_8);  /* Line Spacing */
                        printf("%c(s%dH", ESC, ELITE);  /* Primary Pitch */
                        break;
                case 136:
                        printf("%c&l%dA", ESC, LEGAL);  /* Page Size */
                        /* have to trick the printer by swichting page length
                           and orientation */
                        printf("%c&l%dP", ESC, 84);     /* Page Length */
                        printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
                        printf("%c(s%dT", ESC, COURIER); /* Primary Typeface */
                        printf("%c&l%dD", ESC, LPI_8);  /* Line Spacing */
                        printf("%c(s%dH", ESC, PICA);   /* Primary Pitch */
                        break;
```

```
                case 159:
                        printf("%c&l%dA", ESC, LETTER); /* Page Size */
                        printf("%c&l%dO", ESC, LANDSCAPE); /* Orientation */
                        printf("%c&l%dP", ESC, 51);     /* Page Length */
                        printf("%c(s%dT", ESC, COURIER);
                                /* Primary Typeface */
                        printf("%c&l%dD", ESC, LPI_8);  /* Line Spacing */
                        printf("%c(s%dH", ESC, SMALL);  /* Primary Pitch */
                        break;
                default:
                        printf("%c&l%dA", ESC, LETTER); /* Page Size */
                        printf("%c&l%dD", ESC, LPI_6);  /* Line Spacing */
                        printf("%c&l%dO", ESC, PORTRAIT); /* Orientation */
                        printf("%c&l%dP", ESC, 66);     /* Page Length */
                        printf("%c(s%dT", ESC, LINE_PRINTER);
                                /* Primary Typeface */
                        printf("%c(s%dH", ESC, COMPRESSED); /* Primary Pitch */
        }

        /* Primary Symbol Set */
        printf("%c(%s", ESC, hp_roman_8);
        /* Secondary Symbol Set */
        printf("%c)%s", ESC, hp_roman_8);

        /* Primary Spacing */
        printf("%c(s%dP", ESC, FIXED);
        /* Secondary Spacing */
        printf("%c)s%dP", ESC, FIXED);

        /* Secondary Pitch */
        printf("%c)s%dH", ESC, COMPRESSED);

        /* Primary Style */
        printf("%c(s%dS", ESC, UPRIGHT);
        /* Secondary Style */
        printf("%c)s%dS", ESC, UPRIGHT);

        /* Primary Stroke Weight */
        printf("%c(s%dB", ESC, MEDIUM);
        /* Secondary Stroke Weight */
        printf("%c)s%dB", ESC, MEDIUM);

        /* Secondary Typeface */
        printf("%c)s%dT", ESC, LINE_PRINTER);

        /* End-of-Line Wrap */
        printf("%c&s%dC", ESC, DISABLE_EOL_WRAP);
}


void header(width, user, host, jobnum, jobname, printer)
int width, jobnum;
char *user, *host, *jobname, *printer;
{
        int i, j, k;
        long time(), ts;
        char *t;

        ts = time((long *)0);
        t = ctime(&ts);
        t[24] = '\0';

        /* Print headers and trailers on colored paper in upper tray then
           return to paper deck */
        printf("%c&l1H", ESC);

        for (i = 0; i < 3; ++i) {
                for (j = 0; j < 5; ++j) {
                        printf("BEGINNING OF OUTPUT  ");
                        printf("Printed on %s  ",t);
```

```
                        printf("BEGINNING OF OUTPUT  ");
                        printf("\n");
                        putchar('\r');
                }
                printf("\n");
                putchar('\r');
                printf("                                 User: %s\n", user);
                putchar('\r');
                printf("                                 Host: %s\n", host);
                putchar('\r');
                printf("                                Job #: %ld\n", jobnum);
                putchar('\r');
                printf("                              Jobname: %s\n", jobname);
                putchar('\r');
                printf("                              Printer: %s\n", printer);
                putchar('\r');
                printf("\n");
                putchar('\r');
                for ( j = 0; j < 5; ++j) {
                        for (k = 0; k < 3; ++k)
                                printf("%s", routeto);
                        printf("\n");
                        putchar('\r');
                }
                printf("\n");
                putchar('\r');
        }
        printf("%c",FF);

        /* Return to paper deck */
        printf("%c&l5H", ESC);
}

void trailer(width, user, host, jobnum, jobname, printer)
int width, jobnum;
char *user, *host, *jobname, *printer;
{
        int i, j, k;
        long time(), ts;
        char *t;

        ts = time((long *)0);
        t = ctime(&ts);
        t[24] = '\0';

        /* Print headers and trailers on colored paper in upper tray then
           return to paper deck */
        printf("%c&l1H", ESC);

        for (i = 0; i < 3; ++i) {
                for (j = 0; j < 5; ++j) {
                        printf("   END OF OUTPUT     ");
                        printf("Printed on %s  ",t);
                        printf("   END OF OUTPUT     ");
                        printf("\n");
                        putchar('\r');
                }
                printf("\n");
                putchar('\r');
                printf("                                 User: %s\n", user);
                putchar('\r');
                printf("                                 Host: %s\n", host);
                putchar('\r');
                printf("                                Job #: %ld\n", jobnum);
                putchar('\r');
                printf("                              Jobname: %s\n", jobname);
                putchar('\r');
                printf("                              Printer: %s\n", printer);
```

```
                putchar('\r');
                printf("\n");
                putchar('\r');
                for ( j = 0; j < 5; ++j) {
                        for (k = 0; k < 3; ++k)
                                printf("%s", routeto);
                        printf("\n");
                        putchar('\r');
                }
                printf("\n");
                putchar('\r');
        }
        printf("%c",FF);

        /* Return to paper deck */
        printf("%c&l5H", ESC);
}

main(argc,argv)
int argc;
char **argv;
{
        FILE *myfl;
        int i, c1, c2, c, bump, width, indent, jobnum;
        long time(), ts;
#ifdef PUT_FF_AT_EOF
        int lastc;
#endif
        static char *user, *host, *filename, *t, *jobname, *printer;

        if (argc > 6) {
                if (strcmp(argv[1], "-c") == 0)
                        bump = 1;
                else
                        bump = 0;
                sscanf(argv[1+bump], "-w%d", &width);
                sscanf(argv[3+bump], "-i%d", &indent);
                user = argv[5+bump];
                host = argv[7+bump];
                filename = argv[8+bump];
                sscanf(argv[9+bump], "%ld", &jobnum);
                jobname = argv[10+bump];
                printer = argv[11+bump];
        }
        else {
                width = 132;
                indent = 0;
                user = "unknown";
                host = "unknown";
        }

        /* Preprocess width and indent information for special cases. */
        widthfix(&width, &indent);

        if ((myfl = fopen(filename,  "a")) != NULL) {
                ts = time((long *)0);
                t = ctime(&ts);
                t[24] = '\0';
                fprintf(myfl, "%s %3d %3d %s@%s %5d %s %s\n", t, width, indent,
                        user, host, jobnum, jobname, printer);
                fclose(myfl);
        }

        hpreset(80);
#ifdef HEADER
        header(width, user, host, jobnum, jobname, printer);
#endif
        hpreset(width);

        if ( (c1 = getchar()) == '\033' && (c2 = getchar()) == 'E' ) {
```

```
		/* binary file so don't mess with it */
		while ( (c = getchar()) != EOF)
			putchar(c);
	}
	else {
		if (c1 == EOF)
			exit(0); /* premature EOF */
		putchar('\r');
		for (i = 0; i < indent; ++i)
			putchar(' ');
		putchar(c1);
		if (c1 == '\n') {
			putchar('\r');
			for (i = 0; i < indent; ++i)
				putchar(' ');
		}
		if (c2 != EOF) {
			putchar(c2);
			if (c2 == '\n') {
				putchar('\r');
				for (i = 0; i < indent; ++i)
					putchar(' ');
			}
		}
		while ( (c = getchar()) != EOF) {
			putchar(c);
#ifdef PUT_FF_AT_EOF
			lastc = c;
#endif
			if (c == '\n') {
				putchar('\r');
				for (i = 0; i < indent; ++i)
					putchar(' ');
			}
		}
#ifdef PUT_FF_AT_EOF
		if (lastc != '\f')
			putchar('\f');
#endif
	}

	hpreset(80);
#ifdef TRAILER
	trailer(width, user, host, jobnum, jobname, printer);
#endif

	putchar('\033'); /* reset printer */
	putchar('E');

	exit(0);
}
```

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.

```
# /etc/printcap entry for a non-Ultrix machine:
lp0|laserjet|lj|lj2000|HP Laserjet 2000:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#132:\
        :pl#60:\
        :sd=/usr/spool/lpd/laserjet:\
        :sh:\
        :xc#0177777:\
        :xs#044000:
# /etc/printcap entry for an Ultrix machine:
lp0|laserjet|lj|lj2000|HP Laserjet 2000:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#132:\
        :pl#60:\
        :sd=/usr/spool/lpd/laserjet:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :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.

```
lp0|lj134|esdlj134|ESDLJ134|laserjet|HP Laserjet 2000, Building 1505, Room 121:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#132:\
        :pl#60:\
        :sd=/usr/spool/lpd/laserjet:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj80|esdlj080|ESDLJ080|HP Laserjet 2000 with 80 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#80:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj80:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj106|esdlj106|ESDLJ106|HP Laserjet 2000 with 106 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#106:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj106:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj127|esdlj127|ESDLJ127|HP Laserjet 2000 with 127 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#127:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj127:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj136|esdlj136|ESDLJ136|HP Laserjet 2000 with 136 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
```

```
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#136:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj136:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj130|esdlj130|ESDLJ130|HP Laserjet 2000 with 130 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#130:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj130:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj159|esdlj159|ESDLJ159|HP Laserjet 2000 with 159 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#159:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj159:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj144|esdlj144|ESDLJ144|HP Laserjet 2000 with 144 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#144:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj144:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj118|esdlj118|ESDLJ118|HP Laserjet 2000 with 118 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#118:\
```

```
        :pl#60:\
        :sd=/usr/spool/lpd/lj118:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj96|esdlj096|ESDLJ096|HP Laserjet 2000 with 96 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#96:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj96:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
lj88|esdlj088|ESDLJ088|HP Laserjet 2000 with 88 columns:\
        :af=/usr/adm/lpacct:\
        :br#19200:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/local/lpdfilters/lj_if -w%W -l%L -i%I -n %U -h %H %A %J %j %P:\
        :lf=/usr/adm/lpd-errs:\
        :lp=/dev/tty01:\
        :mx#0:\
        :pw#88:\
        :pl#60:\
        :sd=/usr/spool/lpd/lj88:\
        :sh:\
        :uv=psv1.0:\
        :xc#0177777:\
        :xs#044000:
```

INTERNAL DISTRIBUTION

1. L. D. Bates
2. R. N. Bolling
3. J. B. Cannon
4. J. H. Cushman
5. T. H. Dunigan
6. W. R. Emanuel
7. D. E. Fowler
8. C. W. Gehrs
9. J. A. Green
10. N. A. Griffith
11. C. E. Hammons
12. S. G. Hildebrand
13. G. C. Hinkel

14–23. F. M. Hoffman

24. G. K. Jacobs
25. P. Kanciruk
26. A. W. King
27. L. W. Littleton
28. W. M. Post
29. D. E. Reichle
30. F. E. Sharples
31. D. S. Shriner
32. S. H. Stow
33. R. I. Van Hook

34–48. ESD Library

49. ORNL Y–12 Technical Library

50–51. Laboratory Records Department

52. Laboratory Records, ORNL–RC
53. Document Reference Section
54. Central Research Library
55. ORNL Patent Section

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

CONTENTS