

U.S. DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

**Programs to Print Postscript Files on
Precision Image Plotter**

by

Robert K. Mark

Open-File Report 92-421

This report is preliminary and has not been reviewed for conformity
with U.S. Geological Survey editorial standards.

Any use of trade, firm, or product names is for descriptive purposes only
and does not imply endorsement by the U. S. Government.

Menlo Park
California 94025

Programs to Print Postscript Files on Precision Image Plotter
U.S. Geological Survey Open-file Report 92-421

Robert Mark
U.S. Geological Survey
345 Middlefield Road MS-975
Menlo Park, CA 94025
(415) 329-4917
rmark@isdmnl.wr.usgs.gov

Requirements: Freedom of Press® Professional software and Precision Image CGI c-language subroutine library on Sun computer (SunOS 4.1.2).

Postscript printer files must be rasterized and converted to CGI file format to be plotted on the Precision Image 636 electrostatic plotter. Freedom of Press® (Custom Applications Inc.) is used to generate either a black-and-white Raster-BW-1bit (20:0) file or a color Raster-CMYK-1bit-sep (20:8) file. Shell scripts fop1 or fop4 can be used to run the appropriate Freedom of Press application. The raster files can be quite large (up to about 100 megabytes). Shell scripts ras1plt or ras4plt can then be used to run zeh1 or zeh4 to generate the compressed CGI plotter files, which are much smaller. The color plot files (0sep.plt, 1sep.plt, 2sep.plt, 3sep.plt) must be plotted in that order, as the first loads the paper and the last ejects it. Plotter should be in CGI mode.

Although these programs have been used by the U.S. Geological Survey, no warranty, expressed or implied, is made by the USGS as to the accuracy and functioning of the programs and related program materials, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

```

#!/bin/sh
# Sample script fop1
#
# -zx<>:<y> page size in inches if not specified in postscript file
# -zr rotate image 90 degrees
#
# change paths below to local path for Freedom of Press
#

case $# in
0)    echo 'Usage fop1 postscript_file [-zx<>:<y>] [-zr]' 1>&2; exit 2
esac
if test -f output.001
then
    echo Sorry, output.001 already exists
    exit 1
fi
if test -f $1
then
    /usr/local/freedom/bin/freedom -e1 -g20:0 -r400 $*
    ls -l *.001
    cat info.001
    exit 0
fi
echo file does not exist

```

```

#!/bin/sh
# Sample script fop4
#
# -zx<>:<y> page size in inches if not specified in postscript file
# -zr rotate image 90 degrees
#
# change paths below to local path for Freedom of Press
#

case $# in
0)    echo 'Usage fop4 postscript_file [-zx<>:<y>] [-zr]' 1>&2; exit 2
esac
if test -f output.001
then
    echo Sorry, output.001 already exists
    exit 1
fi
if test -f $1
then
    /usr/local/freedom/bin/freedom -e1 -g20:8 -r400 $*
    ls -l *.001
    cat info.001
    exit 0
fi
echo file does not exist

```

```

#!/bin/sh
# Sample script ras1plt
# /home/pi636 is spooling directory for PI636 plotter,
# change as required

if test -f blk.plt
then
    echo Sorry, 'plt file blk.plt already exist' 1>&2
    exit 2
fi
if test -f output.001
then
    zeh1
    echo 'plot file blk.plt have been generated'
    ls -l blk.plt
    echo 'move it to /home/pi636 (mv blk.plt /home/pi636)'
    echo 'remember to remove *.001 (rm *.001)'
    exit 0
fi
echo output.001 does not exist

```

```

#!/bin/sh
# Sample script ras4plt
# /home/pi636 is spooling directory for PI636 plotter,
# change as required

if test -f 0sep.plt
then
    echo Sorry, 'plt files *sep.plt already exist' 1>&2
    exit 2
fi
if test -f output.001
then
    zeh4
    echo 'plot files *sep.plt have been generated'
    ls -l *sep.plt
    echo 'move them to /home/pi636 (mv *sep.plt /home/pi636)'
    echo 'remember to remove *.001 (rm *.001)'
    exit 0
fi
echo output.001 does not exist

```

```
/*
zeh1.c
U.S. Geological Survey Open-file Report 92-???
Robert Mark
U.S. Geological Survey
345 Middlefield Road MS-975
Menlo Park, CA 94025
(415) 329-4917
rmark@isdmnl.wr.usgs.gov
```

Although this program has been used by the U.S. Geological Survey, no warranty, expressed or implied, is made by the USGS as to the accuracy and functioning of the program and related program materials, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

```
compile with: cc zeh1.c libhcgi.a -lm -o zeh1
```

```
*/
#include "hcgi.h"
#include "names.h"
#include <stdio.h>
#define FILE_NAME "output.001"
#define INFO_NAME "info.001"
unsigned char ZEH = '\272';

unsigned char render[2300];
long irow;
short tot_cols;

main ( argc, argv )
short argc;
char **argv;
{
    short x, y, para[10];
    void zeh_comp();
    void swap();
    short i,j;
    FILE *in_file, *info_file;
    int ncol, nrow;
    char line[100];

    info_file = fopen(INFO_NAME, "r");
    if(info_file == NULL)
    {
        printf("Can not open file %s\n", INFO_NAME);
        exit(8);
    }

    for(i=0;i<8;i++)
    {
        fgets(line, sizeof(line), info_file);
```

```

        printf("%s", line);
        if(i==5) sscanf(line, "%d", &ncol);
        if(i==7) sscanf(line, "%d", &nrow);
    }
    printf("%d cols, %d rows\n", ncol*8, nrow);

    in_file = fopen(FILE_NAME, "r");
    if(in_file == NULL)
    {
        printf("Can not open file %s\n", FILE_NAME);
        exit(8);
    }

    flush_flag = FALSE;
    machine_init = CLEAR;
    outfd = -1;
    outfd = creat("blk.plt", 0644);
    state.paper_size = E_SIZE;
    machine_init |= PSIZE_SET;
    para[0] = 0;
    para[1] = 1;
    para[2] = ncol;
    para[3] = 1;
    para[4] = nrow;
    para[5] = 1;
    para[6] = ncol;
    para[7] = 186;
    para[8] = 0;
    para[9] = 1;
    x = 0;
    y = 13600;
    CCINIT();
    CGRMOD( x, y, para);
    tot_cols = ncol;
    for(j=0;j < nrow;j++)
    {
        fread(render, ncol, 1, in_file);
        zeh_comp();
    }
    raster_term();
    CCTERM();
    exit(0);
}

void zeh_comp()
{
    unsigned char renout[2300];
    short i;
    short renpos;
    short count;
    unsigned char ibyte, temp;
    union {
        long lcnt;

```

```

        unsigned char ccnt[4];
    } udat;

    i = 0;
    renpos = 0;
    ibyte = render[i++];
    count = 1;
    while ( i < tot_cols )
    {
        if( render[i] != ibyte)
        {
            if( count == 1 )
            {
                if( ibyte == ZEH )
                {
                    renout[ renpos++ ] = ZEH;
                    renout[ renpos++ ] = 1;
                    renout[ renpos++ ] = ibyte;
                }
                else
                {
                    renout[ renpos++ ] = ibyte;
                }
            }
            else
            {
                renout[ renpos++ ] = ZEH;
                renout[ renpos++ ] = count;
                renout[ renpos++ ] = ibyte;
            }
            ibyte = render[i];
            count = 1;
        }
        else
        {
            count++;
            if (count == 256 )
            {
                count--;
                renout[ renpos++ ] = ZEH;
                renout[ renpos++ ] = count;
                renout[ renpos++ ] = ibyte;
                count = 1;
            }
            i++;
        }
        renout[ renpos++ ] = ZEH;
        renout[ renpos++ ] = count;
        renout[ renpos++ ] = ibyte;

        udat.lcnt = irow;
        renout[ renpos++ ] = udat.ccnt[0];
        renout[ renpos++ ] = udat.ccnt[1];
        renout[ renpos++ ] = udat.ccnt[2];
        renout[ renpos++ ] = udat.ccnt[3];
    }

```

```

        renout[ renpos] = '\0';
/*
    printf("renpos: %d\n",renpos);
    for( i = 0; i < renpos; i++)
        printf("%d  %d\n",i,renout[i]);
*/
... for(i = 0; i < (renpos - 1); i= i + 2)
    {
        temp = renout[i+1];
        renout[i+1] = renout[i];
        renout[i] = temp;
    }

    CGRAST( renout, renpos);
}

```



```
/*
zeh4.c
U.S. Geological Survey Open-file Report 92-???
Robert Mark
U.S. Geological Survey
345 Middlefield Road MS-975
Menlo Park, CA 94025
(415) 329-4917
rmark@isdmnl.wr.usgs.gov
```

Although this program has been used by the U.S. Geological Survey, no warranty, expressed or implied, is made by the USGS as to the accuracy and functioning of the program and related program materials, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

```
compile with: cc zeh4.c libhcgi.a -lm -o zeh4
```

```
*/
#include "hcgi.h"
#include "names.h"
#include <stdio.h>
#define FILE_NAME "output.001"
#define INFO_NAME "info.001"
#define BLACK 0
#define CYAN 1
#define MAGENTA 2
#define YELLOW 3
unsigned char ZEH = '\272';
```

```
unsigned char render[2300];
long irow;
short tot_cols;
```

```
main ( argc, argv )
short argc;
char **argv;
{
    short color;
    short x, y, para[10];
    void zeh_comp();
    short i,j,k;
    FILE *in_file, *info_file;
    long offset;
    int ncol, nrow;
    char line[100];

    info_file = fopen(INFO_NAME, "r");
    if(info_file == NULL)
    {
        printf("Can not open file %s\n", INFO_NAME);
```

```

        exit(8);
    }

    for(i=0;i<8;i++)
    {
        fgets(line, sizeof(line), info_file);
        printf("%s", line);
        if(i==5) sscanf(line, "%d", &ncol);
        if(i==7) sscanf(line, "%d", &nrow);
    }
    printf("%d cols, %d rows\n", ncol*8, nrow);
    in_file = fopen(FILE_NAME, "r");
    if(in_file == NULL)
    {
        printf("Can not open file %s\n", FILE_NAME);
        exit(8);
    }

color = BLACK;
while( color <= YELLOW )
{
    printf("Color %d\n", color);
    flush_flag = FALSE;
    machine_init = CLEAR;
    outfd = -1;
    if( color == BLACK) outfd = creat("0sep.plt", 0644);
    if( color == CYAN ) outfd = creat("1sep.plt", 0644);
    if( color == MAGENTA) outfd = creat("2sep.plt", 0664);
    if( color == YELLOW ) outfd = creat("3sep.plt", 0664);

    state.paper_size = E_SIZE;
    machine_init |= PSIZE_SET;
    if( color < YELLOW ) state.paper_size |= NO_EJECT;
    if( color > BLACK ) state.paper_size |= NO_PAPER;
    if( color == BLACK ) offset = 3 * ncol * nrow;
    if( color == CYAN ) offset = 0;
    if( color == MAGENTA ) offset = ncol * nrow;
    if( color == YELLOW ) offset = 2 * ncol * nrow;
    k = fseek(in_file, offset, 0);
    printf("Offset %d\n", offset);
    para[0] = color;
    para[1] = 1;
    para[2] = ncol;
    para[3] = 1;
    para[4] = nrow;
    para[5] = 1;
    para[6] = ncol;
    para[7] = 186;
    para[8] = 0;
    para[9] = 1;
    x = 0;
    y = 13600;
    CCINIT();
}

```

```

        CGRMOD( x, y, para);
        tot_cols = ncol;
        for(j=0;j < nrow;j++)
        {
            fread(render, ncol, 1, in_file);
            zeh_comp();
        }
        /* raster_term();          */
        CCTERM();
        color++;
    }
    exit(0);
}

void zeh_comp()
{
    unsigned char renout[2300];
    short i;
    short renpos;
    short count;
    unsigned char  ibyte, temp;
    union {
        long lcnt;
        unsigned char ccnt[4];
    } udat;

    i = 0;
    renpos = 0;
    ibyte = render[i++];
    count = 1;
    while ( i < tot_cols )
    {
        if( render[i] != ibyte)
        {
            if( count == 1 )
            {
                if( ibyte == ZEH )
                {
                    renout[ renpos++ ] = ZEH;
                    renout[ renpos++ ] = 1;
                    renout[ renpos++ ] = ibyte;
                }
                else
                {
                    renout[ renpos++ ] = ibyte;
                }
            }
            else
            {
                renout[ renpos++ ] = ZEH;
                renout[ renpos++ ] = count;
                renout[ renpos++ ] = ibyte;
            }
            ibyte = render[i];

```

```

        count = 1;
    }
    else
        count++;
    if (count == 256 )
    {
        count--;
        renout[ renpos++] = ZEH;
        renout[ renpos++] = count;
        renout[ renpos++] = ibyte;
        count = 1;
    }
    i++;
}
renout[ renpos++] = ZEH;
renout[ renpos++] = count;
renout[ renpos++] = ibyte;

udat.lcnt = irow;
renout[ renpos++] = udat.ccnt[0];
renout[ renpos++] = udat.ccnt[1];
renout[ renpos++] = udat.ccnt[2];
renout[ renpos++] = udat.ccnt[3];
renout[ renpos] = '\0';
/*
printf("renpos: %d\n",renpos);
for( i = 0; i < renpos; i++)
    printf("%d  %d\n",i,renout[i]);
*/
for(i = 0; i < (renpos - 1); i= i + 2)
{
    temp = renout[i+1];
    renout[i+1] = renout[i];
    renout[i] = temp;
}

CGRAST( renout, renpos);
}

```