

Introduction

MKPCL is a program to convert a bitmap image to PCL format. It should be of interest to anyone who needs to print an image such as a logo on a form or a signature on a check etc. It takes a source bitmap file as input, and outputs a monochrome PCL file that can be printed directly on an HP compatible laser printer.

Using a simple command line user interface, the program functions comfortably on any character based terminal or console screen. **MKPCL** is available for DOS, Linux, freeBSD, and SCO UNIX systems and can directly convert a BMP to PCL, JPG to PCL, or PCX to PCL. Printer macros are easily created that can be used as an automatic overlay if desired. The new PCL file can then be used to embed the graphics into a print stream that is not processed by a printer driver that would normally do the conversion. No external programs or drivers are needed to print the resulting PCL file.

Users on AIX, HP-UX, as well as mainframe systems have quite successfully used **MKPCL** on a DOS/Windows machine to create the necessary PCL files, and transferred the output over to their production machine. Regardless of the OS running **MKPCL**, the output is identical..

The command line interface lets you use the program as part of a UNIX shell script or DOS batch file to convert to PCL hundreds of image files with a single execution of the script that loops through a list of source image files. Installation requirements are minimal. Install the executable anywhere in the PATH that is convenient. On UNIX systems set the permissions to suit. On DOS systems you must create a directory \TMP at the root of the file system where the program will look to write temporary files. Otherwise, the program does not interfere with the existing setup.

Running the Program

The program determines the input file type by examining the file's contents, so its filename is not important for proper operation. A typical invocation of the program would be something like...

```
mkpcl source.bmp output.pcl
```

On completion, the file **output.pcl** would contain the program's output for later use by your application. There is no restriction on the naming of the output file.

The same, but enlarged to 142%...

```
mkpcl -s 142 source.bmp output.pcl
```

As above but implemented as permanent macro #245...

```
mkpcl -s 142 -m 245 -p source.bmp output.pcl
```

The resulting **output.pcl** can then be used by your application to insert the image into the print stream.

Program Options

MKPCL offers some 20+ command line option flags that can be used when the program is executed to modify the output; e.g., resize, lighten, darken, rotate, etc. The program is quite fast, and on relatively small images such as a logo it will only take a second or two to run. You can very effectively fine tune the image by experimenting with the parameters and running the program again.

Below is a copy of the **MKPCL** help screen, and we will cover in greater detail some of the options. Some flags require an associated value, indicated with ###, while others only require the flag to be present on the command line.

```
USAGE: ./mkpcl [ -options ] inputfile outputfile
       ./mkpcl without argument for help
```

M K P C L P R O G R A M O P T I O N S

```
-----
-b          Put border around image          -m ###    Output as macro ###
-B ###     Brightness adjustment           -p ###    Make macro permanent
-c #       Compress 0, 2, 3, 4              -r ###    Resolution in dots per inch

-C #,#,#,# Crop image                       -R ###    Rotate image 0, 90, 180, or 270
-d         Save/Restore cursor(CAP)         -s ###    Scale to ###% of original size

-f         Flop image left to right          -S ###    Shadow threshold
-g ###     Set gamma value                  -t-T###   Trim overscan, Trim Threshold
-h ###     Image height in user units       -u ??     User units px,in,mm,pt,cm,pc

-H ###     Highlight threshold              -v #      Limit runtime display messages
-i         Invert to negative image         -w ###    Image width in user units

-I ##     Color Index (0-99) See Sample    -W ###    Watermark Tone (1-254)
-l #      Rotate to page orientation        -x ###    X axis positioning in dots
-L #,#    Limit output size                -y ###    Y axis positioning in dots
-----
```

-c Compress File compression is almost always worthwhile. Depending on the image, the output file size can be reduced as much as 80 - 90%.

-L Limit Size This option allows a bounding box size to be defined, and **MKPCL** will size the image so it will fit within the box.

-h, -w Size Specifies height or width to an absolute dimension, stated in user units.

-u Units Specifies the measuring scale used when sizing to absolute dimensions. Available options are pixels, inches, millimeters, points, centimeters, and picas.

- H, -S** The highlight and shadow threshold parameters are used to "clean up" a dirty background or smoky shadows.

- t, T Trim** If you have a scan with extra white around the image, the trim flag will discard solid white pixels surrounding the image. The **-T** flag is used in conjunction with **-t** to redefine white.

- W** The watermark flag re-defines black to a gray level. It is useful for "ghosting" solid black images.

- x, -y Locate** Specifies the X,Y location in the PCL coordinate system where the image is to be located on the page.

All parameters are discussed in detail in the user manual.

Working with the Application

The capabilities of your application to make timely downloads and the insertion of printer commands into the output stream will determine how the output file is used. Often, it is practical to implement this using a printer macro. The process is twofold. First the macro is downloaded to the printer and then later executed by the application.

Ideally, your application would have the flexibility to allow you to inject the macro file into the job stream whenever you wanted. If this is possible, then you can download the macro(s) at the beginning of the print operation. Otherwise, you will have copy the file to the printer as a permanent macro before the application is started. This could happen immediately before the application is run or even first thing in the morning. As long as the printer is not powered down, the permanent macros will remain resident waiting to be executed. Another possibility is storing the images on a hard disk or FLASH memory inside the printer itself.

In any case, at runtime the application has to issue the PCL code to run the macro that prints the image. It can be set up to automatically print the image on every page, or conditionally based on some runtime criterion. The PCL command will be similar to `<esc>&f245y4X`. The `<esc>` is the single ASCII char 27 common to all PCL commands. The **245** is the macro ID that was assigned by **MKPCL** above but would change with your assignment of the ID. The **4X** indicates that the macro is to be enabled as an automatic overlay and will be run on every page unless the printer receives a disable macro command.

In a database application, you would typically put the PCL enabling code into a variable on the format. So if you do not want the macro to print on any given page, simply clear the variable, or in the case of an overlay, issue the disable overlay command, and the graphic is gone. Consider that you could download several different macros together, and then based on current data, print one or another on a given page. There are many possibilities.

You may find it necessary from time to time to manually edit a PCL file. This can be a problem with many editors, particularly if the file contains raster data. There is salvation in a public domain text editor named **vim** that is widely available over Internet for both DOS and UNIX systems and is standard on Linux systems. It has a binary mode, among other features, that makes it quite suitable for this purpose. It is highly recommended.

DeskJet Printers

MKPCL was written for use with black and white laser printers; consequently, the output is a monochrome image. If you are starting with a full color original, the program will reduce it to black and white in the conversion to PCL.

Further, the majority of contemporary DeskJet printers speak PCL3+, which is not nearly as conducive to printing raster graphics as the PCL5 LaserJets. In the HP line, any of the 6xx, 8xx, 9xx, 1120, 1220, 2000 etc. are reasonably easy to work with.

Where a LaserJet processes the entire page before printing begins, the PCL3+ DeskJet is more akin to a line printer as the code is interpreted as the job is processed. These relatively newer printers do, however, process ALL the text before printing begins, and this provides the opportunity to introduce raster graphics into the print stream. If you have an inkjet printer other than HP, request from us a test PCL file to see if your printer is suitable.

Price and Availability

You can download a free demo version of **MKPCL** that is identical to the regular version except that it marks the image with alternating black and white lines. You will find the download links on the sidebar. Select the file that is appropriate for your OS. The demo package is 2MB and can be emailed if it is not a problem for your email account. When requesting a copy of the demo, be sure to specify the target operating system; i.e., DOS, Linux, SCO Unix, or FreeBSD. . See the email address below. You can also download the user manual from the link on the sidebar. The manual is also included in each of the demo packages, so you needn't get a separate copy if you are getting one of the demos.

As of April 1, 2016, the regular program is priced at \$125 CDN (about \$95-100 US) for the DOS, Linux, SCO UNIX, or freeBSD version. If you have any questions, call Jim Asman at (604) 584-0977. If you want to order the program, follow the "How to Order" link on the sidebar. Email queries should be directed to jim@spectralservices.com. If you have navigated to this page directly via a search engine, the home page is located at <http://www.spectralservices.com> .

