PCL Page Formatting

Introduction

If you have any inclination to create a PCL overlay of any kind to print a business form on a PCL laser printer, it is essential that you understand the PCL page formatting commands. This not only has an impact on the form but also has an impact on your application as well. Often it is desirable to format the overlay with one margin setup, and format the data with another. A laser printer permits landscape and portrait orientations, varying paper sizes, endless margin and line spacing arrangements, all potentially on the same report! You may want to pull the first sheet from one paper source, a letterhead maybe, and the rest of the job from another tray. Perhaps the report or part of the report needs to be printed duplex.

While an impact printer gives the operator some general control over the placement of the print on the page by virtue of how the paper is loaded into the printer, a page printer mechanically moves the paper through the printer the same regardless of the page's contents. Therefore, all formatting instructions must be sent to the laser printer via printer codes. Hewlett Packard's PCL 5 printer language, available in most LaserJets (and compatibles) since 1990, provides a very comprehensive set of tools to format output exactly as you want.

With the exception of the "business inkjets" such as HP's 2250, 2280, et. al., which are PCL 5 printers, HP DeskJets are PCL 3+ printers. The PCL 3 Language is a subset of PCL 5. Most of what applies to a LaserJet will also apply to the DeskJet as well, but there are some differences that will be addressed as necessary. Note that there a number of printers out there that are NOT compatible with PCL3 or PCL 5 and are not suitable for our purpose here. Be aware that PCL 6 printers are NOT backwards compatible with PCL 5 printers. Still others require a windows driver and don't speak PCL at all.

If you are operating in an environment where there is not a "printer driver" operating between the application and the printer, your application will have to supply printer control codes to properly format the data on the printer. At minimum you need to specify page layout so page breaks coincide will the paper being ejected from the printer. This may amount to a simple init code at the beginning or PCL code being injected throughout the document.

While a typical PCL printer driver will specifically position each character or word on the page, a "print and space" job stream where the text location is determined by its position in the file and by the line spacing and page margin parameters. Printer throughput will probably be much better than it might be otherwise if run through a typical printer driver.

The PCL Language

With few exceptions, PCL commands are straight ASCII text preceded by the escape character (ascii 27) or hex 0x1B. This easily permits the programmer to insert printer code into the print stream. In this discussion the escape character will be represented by the two character sequence **^[** , but do keep in mind that it is a single character in the output stream.

PCL commands are generally grouped in families of related codes, and each family will begin with the same sequence of characters, usually three. While there are a great number of PCL commands, it is not nearly the obstacle to learn that you may think because they are grouped the way that they are. For example, the commands used for job setup and page format begin with **^[&l** (that's a lower case ell), and anything to do with the primary font selection will begin **^[(s** . Typically, the command is completed with a value for the parameter being modified and then an ending character that specifies the parameter itself. The complete command to specify letter sized paper is **^[&l2A** where the **A** indicates that paper size is being specified and the **2** is the PCL value for letter size. Other values are used to define other paper sizes. PCL commands are terminated with a capital letter.

What you have to recognize is that page layout parameters are dependent on each other's values, so it is possible, if not probable, that just changing one parameter will not give the results you expected. This brings us to the crux of this article, which is to highlight the salient page layout parameters and how they interact with each other.

Page Layout

When a new page layout is sent to the printer, at minimum the following page parameters should be specified.

1. Paper Size
2. Orientation (Portrait or Landscape)
3. Line Spacing
4. Top Margin (Stated in # of lines with above spacing value)
5. Text Length (# of print lines before printer ejects page)

Additionally it may be necessary to include …

6. Paper Source (specific paper tray)
7. Duplexing options (ignored if printer is not duplex equipped)
8. Number of copies

Now we will look at each of these parameters in greater detail. Give this material a quick study for the moment and refer back to it later when complete examples are given.

Paper Size ^[&l##A

NAME	PCL VALUE	SIZE
Executive	1	7¼ x 10½ in.
Letter	2	8½ x 11
Legal	3	8½ x 14
Ledger	6	11 x 17
A4	26	210 x 297 mm
A3	27	297 x 420

Simply substitute the ## from the command above with the appropriate PCL value to get the complete PCL code to set the desired paper size. When the printer receives a paper size command, the page will be printed immediately if there is any data in the print buffer. Further, all user margin and line spacing defaults will be restored from the control panel. This underlines the need to send all relevant page settings together.

Orientation ^[&l#O

NAME	PCL VALUE
Portrait	0
Landscape	1
Reverse Portrait	2
Reverse Landscape	3

Orientation is straightforward. Substitute the PCL value needed for the desired orientation. Like paper size, an orientation command will cause the current page to be printed immediately if there is any printable text in the buffer. It is not possible to print in two orientations on the same side of a page. There is a text rotate command, quite apart from orientation to turn text sideways. The reverse orientations can be quite useful when duplex printing.

Line Spacing ^[&l##D

This version of the line spacing command takes selected PCL values as the “lines per inch” value(LPI). Legal values are 0,1, 2, 3, 4, 6, 8, 12, 16, 24, 48. If you want 6 lpi, **^[&l6D** will do the trick. Note that all legal values are a factor of 48. If you require spacing values other than those just listed, there is a different command to set the “vertical motion index” (VMI).

VMI ^[&l##.####C

Where the previous command defined the line spacing in terms of lines per inch, the VMI command defines the line spacing in terms of the depth of a single line in 1/48 th’s of an inch with precision to four decimal places. You can obviously obtain very precise line spacing values. From time to time you may want several lines of text to exactly fill a given depth. **^[&l8C** and **^[&l6D** produce identical results.

Top Margin ^[&l##E

The top margin is stated in ## of lines at the current line spacing. Your typical page that has 60 print lines at 6 lpi would have a top margin of three lines and a bottom margin of three lines, all totaling to 66 lines for the page.

Text Length ^[&l###F

The text length is the number of print lines below the top margin wanted before the printer ejects the page. Although the text length is defined in terms of lines, the printer does NOT count lines as such. The text length value is used to compute a position down the page that is effectively the bottom margin. Any linefeed or half linefeed that moves the print position past this “bottom margin” will cause the page to be printed. Once the bottom margin has been established with the text length command, subsequent changes to the line spacing value, do NOT change the bottom margin.

Paper Source ^[&l#H

NAME	PCL VALUE
Standard Tray	1
Manual Feed	2
Lower Tray	4
Optional Tray	5

Most of the time a paper source command is probably unnecessary. The most obvious reason would be that the printer is only equipped with a single paper source. This is particularly true for very inexpensive laser printers, and I’m not certain if any inkjets offer multiple input trays. Also, if you have two paper trays, one may be loaded with letter size and the other with legal size, in which case the printer will pull from the tray with the paper size specified. If, however, you have two trays both loaded with letter size paper and one is your letterhead that you only want to print on infrequently, then a tray select command will be necessary to insure that the letterhead is only used as desired.

Like other commands, a paper source command will cause the page to be printed(from the current tray) if there is any printable data on the page before the new command takes effect. When changing trays, then, always put the source command as the first thing on the new page. If the entire job is being printed form the same tray then the source command can be placed in the form init on the options page of “define output”; otherwise, the print code will have to be on the format itself.

Duplex Options ^[&l#S

NAME	PCL VALUE
Simplex(no duplexing)	0
Duplex(long edge binding)	1
Duplex(short edge binding)	2

Double sided printing creates few problems on a duplex equipped laser printer. If you are printing duplex, you just choose whether the binding edge is along the short or long edge of the paper. For people that produce lengthy accounting reports, duplexing not only cuts down on the number of sheets of paper in the printout, but also when viewing the report you always have two pages in sight at the same time.

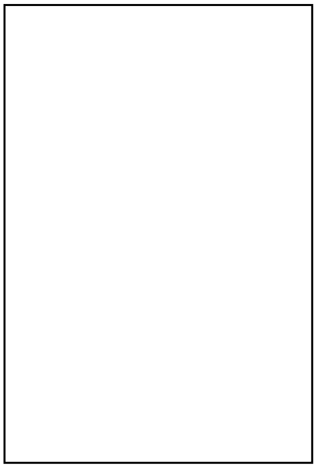
If you are printing in duplex mode and want to allow some extra space on the left hand side of the page for binding, keep the text centered on your output format. If you push the text to the right a little bit, things will be fine on the odd numbered pages, but text on the even numbered pages on the back side of each sheet will be pushed the wrong way. If the text is centered on the output format, you can use the PCL sheet offset command to push the text to the right. When printing the back side of the sheet, the offset will be applied to the right side of the text which puts the binding on the same physical side of the paper as the front. The offset command is **^[&l###U** where ### is an integer value in decipoints or 1/720'ths of an inch. For example, to allow an extra ¼ inch to the binding edge send **^&l180U** at the beginning of the first page.

Number of Copies ^[&l##X

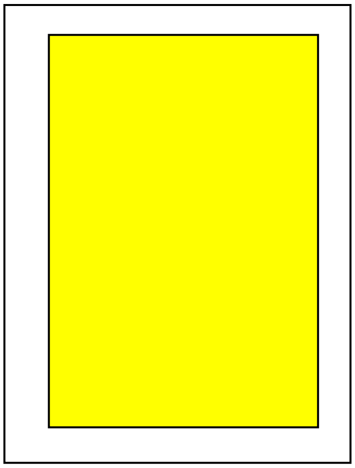
When using a laser printer, don't forget that if you require more than one copy of a page you can simply issue the command where ## is the number of copies wanted. Often you find multi-part forms that have two identical pages and one or two others that are different. You only need to define the duplicated page once on the output format so that it includes **^[&l2X** and then **^[&l1X** on the next page. PCL3 inkjet printers cannot print multiple copies internally.

PCL Page Definitions

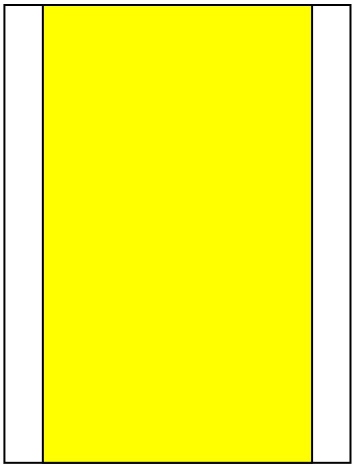
There are three page concepts within the PCL language. The first is the **physical** page and is simply the sheet of paper as we would always have thought of it. In the drawings below, on a letter size sheet the white area is about 1/4 " left and right and about 1/6" on top and bottom.



Physical Page



Printable Page



Logical Page

Next consider the **printable** page. The shaded area represents the area of the paper where the printer can actually mark the paper, or conversely the white area represents an unprintable region on the sheet. The unprintable area is required to ensure that toner particles don't find there way around to the back side of the sheet which would most certainly happen if printing were permitted out to the edge. The size of the unprintable area varies slightly from portrait to landscape formats. I've never found a way to beat this.

Finally, we have the concept of the **logical** page. The logical page represents all of the addressable locations on the page where the printer can be directed print. You see that the logical page overlaps parts of the unprintable region. While you can position the printer's cursor in the unprintable region, you still cannot mark the sheet there. The reason to have the logical page to extend over the unprintable area is to permit addressing schemes that have their origin at the physical edge of the paper, at least on the top and bottom. This is extremely useful, as you will see in a moment. The logical page on an inkjet printer is the same as its printable page; i.e., the bottom ½" is neither addressable nor printable.

Assembling the Codes

Given that the paper size and orientation have been decided, the line spacing value must then be established, as it is the basis of the margin and text length parameters. While you can pick most any line spacing arrangement you desire, 6 lpi or 8 lpi are likely on almost all of your output formats. If you are methodical this is all very easy.

First calculate, at the desired line spacing, the number of lines it takes to fill the entire depth of the physical page. This would be (depth in inches * lpi). The page format then is a division of these lines into three parts: the top margin, the text, and the bottom margin. The latter is defined by the first two. There is no reason that margins must be equal. Often you may start out knowing how many print lines are wanted, in which case your task is to split the remaining lines between the top and bottom margins. For example, say we want to print 55 text lines on letter size portrait at 6 lpi. The page has a total of 66 lines (11 inches * 6 lpi). If we subtract out the 55 text lines, we're left with 11 lines total for margins. I would normally split that into a 5 line top with a 6 line bottom. An aside, if there aren't any other considerations, put the extra line at the bottom if you have an odd number of lines for margins. Very often equal margins will appear a bit "top heavy" and certainly the addition of another would make it a certainty. In this example the code would read …

^[&l2A ^[&l0O ^[&l6d ^[&l5E ^[&l55F

The other scenario is that you want a specific margin size in which case you subtract the line count of the combined margins from the page depth to arrive at the text length. The default page is typically letter size portrait at 6 lpi with ½" inch margins. At 6 lpi the margins are each 3 lines leaving 60 lines of text (66 – 3 – 3). The code to format the page then is …

^[&l2A ^[&l0O ^[&l6D ^[&l3E ^[&l60F

Combining Codes

Escape sequences in PCL can be combined if the codes are all from the same family with the same beginning sequence to each code, **^[&l** in this case. When combined, the above 5 codes reduce to

^[&l2a0o6d3e60F

To combine the codes, send the beginning sequence just once, and then follow it with the PCL values and their respective terminating letters. When codes are combined all letters are lower case except for the last letter of the combined sequence which is capitalized. In the above sequence, there is nothing special about the “F” other than it happens to be the last command. The capital letter signifies the end of the PCL sequence. If another command were appended then its terminating letter would be the one capitalized. If we add another command to the end we simply make the “F” lower case and capitalize the new terminating letter. If we set the printer to make 2 copies we get …

^[&l2a0o6d3e60f2X

Other than being the default, there is nothing particularly magic about a ½“ margin. If you prefer to have a longer text length, simply reduce the size of the top margin and expand the text length. Two line margins would give you a text length of 62 lines, and if the margins were reduced to one line, 64 print lines are available.

^[&l2a0o6d2e62F (62 print lines)
^[&l2a0o6d1e64F (64 print lines)

Similarly, you can specify larger margins and a shorter text length. Do whatever is to your advantage, and if you study all of the examples here and understand the parameter values you will be in good shape. If you specify a layout that is too large to fit on the page you have indicated; e.g., a letter size page with 80 lines at 6 lpi, the printer will prompt you for a larger sized sheet of paper.

Now consider a landscape letter size page at 6 lpi with ½ inch margins. In landscape mode the page depth is 8½ inches giving us 51 lpi lines. Allowing 6 lines total for margins we are left with 45 print lines. The same specifications on legal size only change the paper size command as the page depth is the same on both.

^[&l2a1o6d3e45F (letter)
^[&l3a1o6d3e45F (legal)

At 8 lpi the print depth is 68 lines and the ½ inch margins are 4 lines each leaving 60 print lines on the landscape page.

^[&l2a1o8d4e60F (letter)
^[&l3a1o8d4e60F (legal)

A legal size portrait page at 6 lpi with ½" margins has a page depth of 84 lines (14 * 6) and a text length of 78 lines (84 –3 –3). The code is …

^[&l3a0o6d3e78F

Suppose that for some reason you require an extra 2 inches in the bottom margin, perhaps for a signature or something else that is going to be subsequently filled in by hand, all you need to do is change the text length. Continuing with the previous example, we only need to reduce the text length by 12 lines to gain the 2 inches of space, leaving an aggregate bottom margin of 2½ inches. Aside from the text length the two sequences are otherwise identical.

^[&l3a0o6d3e66F

Obviously you could leave the 78 line format and just make sure that nothing is printed on the output format on the bottom 12 lines. And that would be probably the most reasonable approach. I've raised the issue here to point out that from time to time the space on a filePro output format space is at a premium.

Metric Page Formats

Should you find yourself with an A4 page to format, the principles are the same, it's just that the page is defined in millimeters, and doesn't translate evenly to inches. Not that big of a deal though. Consider an A4 portrait page at 6 lpi with 3 line margins. Assuming 25.4mm to the inch, the 297mm A4 depth translates to about 11.69 inches, very close to 70 6 lpi lines. We get a text length of 64 lines compared to the 60 we see on a similarly configured letter size page. At 8 lpi the page depth is 93 lines, and with margins of 4 lines at 8 lpi we get a text length of 85 lines. The PCL code …

^[&l26a0o6d3e64F (6 lpi)

^[&l26a0o8d4e85F (8 lpi)

In landscape format, the 210mm depth comes in at 8.26 inches, giving a page depth of 49 lines at 6 lpi and 66 lines at 8 lpi. Keeping the ½ inch margins, our landscape codes for A4 are …

^[&l26a1o6d3e43F (6 lpi)

^[&l26a1o8d4e58F (8 lpi)

You might want to change the border configuration to give a different text length, or maybe change the line spacing using the VMI code fine tune the line spacing to exactly fill the page. I'm not aware of the typical line spacing used in metric environments. If for nothing more than an exercise, we will format an A4 page that has a depth of 66 lines and 60 print lines making it compatible with the letter page similarly formatted.

The key to this of course is to get the line spacing the makes 66 lines exactly fill an A4 page. The VMI command for line spacing if you recall is stated in 1/48ths of an inch. The 11.69 inch depth of the A4 page in 1/48ths inch is 561 (11.69 * 48). If we divide this by our 66 lines we get the VMI value of 8.5000. Coincidentally, the VMI value here worked out to need only one decimal point of precision, but as stated previously the PCL recognizes up to four decimal places. Our code to format an A4 page to conform to a letter formatted page is …

^[&l26a0o8.5c3e60F

If you had to run the same reports in Europe and the US, you could easily have an environmental variable that triggered either a letter size or A4 size format to be activated.

Specialized Formats

There are occasions where you simply must be able to define a page format that does not have a top or bottom margin. An example that comes to mind is the laser version of the IRS Form 1099. This is really a page that contains 3 identical forms, each being 22 lines in length, very analogous to a page of laser labels.

As 3 different records are going to be printed on each sheet, the output format must be defined as 22 lines in length at 6 lpi. Recall that the logical page does extend from the very top of the sheet to the very bottom of the sheet. While part of that overlays the unprintable region, we don’t need to print there, but need the space to exist for alignment of all 3 records that are going to be printed on the sheet. What we need is a form that has a text length of 66 lines without any margins at all. That’s easy!

^[&l2a0o6d0e66F

Stick that code into the form init, and when the output is generated, the application can keep on outputting 22 line forms which the printer will accumulate until is has received 66 lines when it will print the page and then start processing the next page.

The IRS W2 forms for a laser are 2 forms per sheet. Use the same print code here as for the 1099 except define the output format as being 33 lines in length. The process works just the same. The application pumps out the 33 line forms and the laser pages them.

The above code can be used to format the laser printer to be mostly compatible with reports that have been formatted for a dot matrix printer that is set up with a 66 line page and 60 print lines. The only concern you have is to make sure that you don’t place any print data on line 1 or line 66, otherwise, the output should be identical. I have never seen nor heard of an inkjet printer, besides some Hewlett-Packard models, that will let you print 66 lines at 6 lpi on a letter sized page! HP models in the 600 series, 800 series, and later models, will permit 66 line forms as long as there is nothing printable in the bottom 3 lines. Certainly, do a test to verify the operation.

Here in Canada there was a 6 lpi T5 form that has three forms on a legal sheet, each of them 4 inches high, with a 2 inch “throw away” piece at the bottom. This requires the output format to be 24 lines and the PCL format to have no top margin and 72 print lines. The code needed is …

```
^[&l3a0o6d0e72F
```

Sometimes you can save some aggravation in positioning text on preprinted forms if you double or triple the size of the format and correspondingly decrease the line spacing so you are printing on every second or third line. Going back to the 1099, define the output format as being 66 lines rather than 22, and change the print code to print at 18 lpi with a text length of 198 lines (11 * 18). Because 18 is not an even factor of 48, we need a VMI code to set 18 lpi. The page length in 1/48ths is 528 high (11 * 48) , and when we divide through by our 198 lines on the page, we get a VMI of 2.6666 (528 / 198). Our code is then

```
^[&l2a0o2.6666c0e198F
```

In theory, you would only put printing variables on every third line, but should you find that a line is a bit high or low, you can try printing on an adjacent line. Of course, if you are creating your own PCL overlay, you can design it so all data will “land” perfectly at 6 lpi or whatever.

Unix Note

One last thing, if you are using a PCL interface with a Unix variant, make sure that your **lp** command includes the raw option. For example…

```
lp –s –o raw –d laser
```

In cooked (as opposed to raw) mode, there is some post processing of the print data that takes place, aimed at paging raw text. This will wreak havoc on binary data such as graphics like a logo or signature.

Please report any errors or omissions that you find.