Debugging PCL Transformations

I.   Determine compatibility of transformation into a <u>raster</u> format. Can you convert the PCL using PCLTool.exe with the <u>defaults</u> into TIFF Gp4 or raster PDF? Or, can you View PCL to bitmap view format?

Capture: If you don't have some other method of capturing your PCL, take the printer off-line and print to it. Then, go look in the .\WINDOWS\system32\spool\PRINTERS folder for the .spl file that is waiting for the spooler to release the PCL job to the printer.

The more advanced method is to go to our PCLTSDK_Vvv icon group – Print Capture – Port Monitor and run the PMSetup utility. It will create the PCL Transformation Driver, which prints to the PMon1: port monitor that also gets installed. You can either print from your application to the PCL Transformation Driver or print directly to the PMon1: port monitor. You may need to read the Port Monitor help to find out how the Port Monitor works and how to configure it.

PCLSplit: If the file is too big to work with, use PCLSplit –R to split-off the first "n" pages.

PCLTool: Use the program defaults and convert the PCL into TIFF/Gp IV format. Review the .LOG file generated and View Image the .TIF files created.

Investigate the probable causes of any errors that are generated. You may need to go back to the .TXT file produced by PCLCodes and search for commands that may cause problems. The most common search strings are: "bol Set", "ypeface", "Reset" and "macro".

If the TIFF image looks good, then print the PCL using the DOS command:

"COPY /B filename.PCL PRN"

Or, by using the View PCL, then the FastPrint feature, which uses the Windows SDK EscapePassThru function. Print from page 1 to however many pages you need to prove that the file is a good PCL file.

Also, print the TIFF images to compare to the printed PCL pages. If they match, then proceed to determining the index-ability of the text within the PCL file. If they do not match, then you will need to delve deeper into the PCLCodes disassembled PCL .TXT file to see where the error first occurs and look for the cause (ie. a font id for a font that is not in the file or resident?, a macro call for a missing macro? etc.).

You will want to make sure all the correct preferences are set in PCLTool for the nature of the PCL file. If you are missing a download font that is in the file, then you need the turn-on the "Convert Download Bitmap Fonts" preference.

PCLCodes: Disassemble the PCL file into readable english text with PCLCodes or PCLDasm.

Look into the .LOG file generated for warnings and error messages. The .LOG file error messages that are generated during the conversion of the PCL by PTC32.DLL (when using PCLTool.exe, PCLView.exe or PCLXForm.exe) are as a direct result of the transformation process undertook and bear more weight than the .LOG file generated by PCLCodes.

Investigate the probable causes of any errors that are generated.

If no errors are generated, take a look in the .TXT file produced to see the nature of the PCL file to determine what options need to be set for a trial conversion.

Look at the end of the disassembly text file generated for "invalid escapes", the PCL Escape Summary, Font Summary, Font Attribute Summary, Macro Summary, Palettes Referenced by Raster, Palettes Referenced by Color Entry and Page Summary.

If you need to delve deeper into the PCL disassembled detail, then set the other output options that are appropriate for your quest. Other output options include:
>Font Char. Descriptor data
>Bitmap Font Char. Raster Data
>Graphics Raster Data
>Summary of 24-bit Color Raster Values

II.   Determine Compatibility of Transformation Into a <u>Vector</u> Format. Can you convert the PCL using PCLTool.exe into EMF or vector PDF? Or, can you View PCL to vector view format?

PCLTool: Convert the PCL into destination file Type EMF.
Review the .LOG file generated and View Image the .EMF files created.
Investigate the probable causes of any errors that are generated.

If you see the wrong fonts or missing one, the most probable causes are that the PCL contains downloaded fonts, the PCL is referencing resident printer fonts or the PCL is referencing fonts, macros or other resources that are normally sent the printer prior to the print stream.

Switch to the Direct Driver PDF file type option to "as Vector" with "Embedded Fonts". Convert the PCL to PDF and view the PDF file in the Acrobat Reader. Debug any anomolies that may show-up.

If the ultimate (decollated) PDF file is going to be less than 5 pages in length, you may want to turn-off the "Embedded Fonts" option so as to see how small you may be able to get the PDF file size.

There are many workarounds available to get your PDF files looking 100% correct, small and with fast conversions. However, these workarounds are too numerous to document and are much better addressed by sending your zipped PCL, *.ini and .tpt files for analysis.

III. Check-out PCLXForm. exe and how easy it is to use our .TPT script programming language BEFORE you try to write any custom code with the API to our .dll's. For starters, the PCLTool SDK Demo disables the ability for any custom .exe to call our .dll's… even our sample custom programs. We want you to at least try the following sample script before you decide to write a custom .exe:

  RUN "pclxform .\scripts\convert\pdfport.tpt"

 Look in our .\out folder for the .pdf file created.
 Open the .tpt afterwards to see Job Parameters.
 Read the through to the end of the JobParams section of the PCLXForm help file.
 It will show you how to override the script JobParams on the command line.

 If you feel that PCLXForm with a custom .tpt script is not going to allow you to meet all your application requirements, then please allow us the opportunity to verify your assumption. If you provide us with your application, workflow, transformation, text extraction (for file splitting, output filenaming, indexing, routing, etc.), client/server implementation method, document type, maximum number of pages per file/per document and other requirements. We can provide you with a fixed quote for a custom .tpt script or at least confirm that writing your own .exe is going to meet all your requirements.

IV. If you get stuck – send us your PCL (zipped) and explain what you need done. Also, don't forget that we have TechSupport.exe that gathers everything we need to help you solve problems. For more information go to: http://www.pagetech.com/support/supportform.php.