Chapter 1: PCL Goals and Guidelines

Contents of this Chapter

1.1 Introduction

Historically, printers were developed without a computer industry standard for feature access. Since features differed from device to device, applications for one printer had to be modified to work with another, requiring months or years of software development. Users were reluctant to upgrade to new technologies because applications required extensive modification to support a new printer.

To improve this situation, HP developed the PCL printer language to standardize access to printer features. The PCL language provides the highest level of communication between the system and the printer. It is independent of the host system, device drivers, I/O interface, and network communications. Its purpose is to bring all HP printers together under a common control structure. The feature compatibility provided by the PCL language protects the user's investment in applications and driver software and provides a vehicle for exploiting new features, capabilities, and technologies.

NOTE: "PCL" is a registered trademark of Hewlett-Packard. "PCL" should be used only as an adjective in literature (e.g., PCL language).

1.2 The PCL Committee

The purpose of the PCL Committee is to maintain a PCL language feature architecture that reduces product development effort, provides a system approach for PCL devices, and facilitates the development of robust device drivers. The objectives of the PCL committee are:

1. Review and approve proposals for extensions and changes to the PCL language:
    - Make recommendations for the future
    - Discuss backward compatibility implications
    - Create guidelines
    - Provide an expert forum for discussion and feedback on proposals for new features
2. Document and distribute extensions and changes to the PCL language.
3. Inform and influence development teams:
    - Document and distribute responses to proposals
    - Informally train implementors
    - Provide information for future product definition
4. Maintain the PCL language:
    - Obsolete unused features
    - Update feature partitions

Committee Members

Committee members as of 11/94 are:

Dean Anderson, SIO (Chairperson)
Ben Brezinski, VPR
Jordi Gonzalez, BCD
Melinda Grant, VCD
John Haney, SDD
Angela Hanson, BPR
Laura Mansfield, SPR
Claude Nichols, VPR
Bob Pentecost, NPR
Ted Podelnyk, ALO
Suzanne Richmond, SPO
Linda Rodda, ALO
Jerry Villone, SIO

1.3 The PCL Implementor's Guide

The PCL Implementor's Guide **should only be distributed internally**. It is not intended as a reference for Independent Software Vendors (ISVs) developing device drivers. Individual product teams will provide technical reference manuals and cookbooks to assist in driver development.

The PCL Implementor's Guide contains information for Hewlett-Packard design engineers and licensees of PCL technology. It includes all the control codes and escape sequences supported in Hewlett-Packard or licensed printers. By defining PCL structure and syntax, it describes how printer features may be controlled by user or system application programs.

The information in this document is subject to change without notice. For the latest revision of *The PCL Implementor's Guide*, contact a committee member. Comments concerning this specification can be sent to villone@sdd.hp.com. If you find parts of this specification to be contradictory, unclear, or incorrect, please notify the editor. Ideas on how to improve the specification are encouraged.

1.4 The Proposal Process

As technology or market direction changes, developers of PCL products may propose language modifications.

Pre-Proposal Planning

The first step in the process is to discuss the problem with a committee member.

Those presenting a proposal should make sure the committee understands the problem that the proposal is addressing. Committee deadlocks often occur because solutions are presented before the problem is understood. The committee may have encountered the same problem before and can recommend a preferred solution.

Proposal Presentation

Each proposal should have a PCL committee sponsor to assist with the form and content. The three forms of proposals and their respective procedures are:

1. OBVIOUS omissions, errors, or ambiguities in the *PCL Implementor's Guide*:

   The submitter should give a marked up copy of the erroneous page with the exact wording of the correction to a committee member. Approvals will be reflected in the next release of the *PCL Implementor's Guide*.

2. MINOR changes to the PCL language. The proposal must contain:

   - The exact wording of the change.
   - Possible effects on other parts of the PCL language.
   - The first products implementing the change and their time frames.

   Approvals will be reflected in the next release of the *PCL Implementor's Guide*.

3. MAJOR changes to the PCL language. The proposal must contain:

   - The exact wording of the change.
   - Possible effects on current and future products.
   - Possible effects on other parts of the PCL language.
   - The first products implementing the change.
   - The time frame for the implementation of the change.

   A hard copy of the proposal should be given to the PCL Committee chairperson. The submitter may need to attend committee reviews of the proposal. Submissions should be made early in a project, since the approval process takes two or three committee meetings (90 - 120 days) for complex proposals. Upon committee approval, the submitter should provide a hard and soft copy (preferably in Microsoft Word for Windows 6.0) to the Committee. Approvals will be reflected in the next release of *The PCL Implementor's Guide*.

Proposal Format

PCL command definitions should follow Implementor's Guide format and include the following in order:

1. Functional name of the command followed by the escape sequence.
2. Brief one-sentence functional description of command.
3. List of defined value field values.
4. Default value.
5. Range of possible values.
6. Detailed command description with possible interactions, exceptions, etc.

Feature Analysis

A feature analysis should be submitted along with the command definition. The feature analysis should describe all two-way interactions between the proposed command and other PCL, HP-GL/2, PJL, and PML commands. To obtain a templates in Microsoft Word, FrameMaker, and Interleaf, ask a PCL Committee member.

Process Review

To make the status of proposals brought to the PCL Committee more apparent, the following checkpoints will be tracked at the meetings and in the minutes.

State	Deliverables for next State
Investigation	1. Committee identifies changes, suggestions, and modifications needed to transition to the next phase. 2. Written proposal. 3. Schedule. 4. Concept approval required to transition to the next phase.
Feature Developement	1. Committee identifies changes, suggestions, and modifications needed to transition to the next phase. 2. Feature analysis completed.
Final Editing	1. Committee identifies changes, suggestions, and modifications needed to transition to the next phase. 2. If it has not been previously done, a file is delivered to Jerry Villone.
Approved	Implementor's Guide wording is approved.

Shelved Proposals go back to the Investigation phase. State transition can be bidirectional if significant new data comes to light.

1.5 Language Objectives

The ***primary*** objective of the PCL language is to supply customers with the right print solution at a competitive price.

In addition, internal HP objectives include:

- Reduce system support costs (driver development).
- Reduce R&D investment (leverage).
- Provide compatibility in PCL-based products.
- Provide automatic name familiarity for the PCL language in HP products.

1.6 Historical Perspective

The PCL language has evolved through five major levels of functionality driven by a combination of printer technology, user needs, and application software. Higher levels are supersets of lower levels.

PCL 1 *Print and Space* is the base function set for single-user output.

PCL 2 *Electronic Data Processing* adds general-purpose multi-user system printing.

PCL 3 *Office Word Processing* adds high-quality office documentation (DeskJet 5xx line of products).

PCL 4 *Page Formatting* adds advanced page printing capabilities (LaserJet II series).

PCL 5 *Office Publishing* includes cababilities such as font scaling and HP-GL/2 graphics (LaserJet III and 4).

The PCL model has been successful because

- HP printers are generally consistent in their implementation of the PCL language.
- HP printers implement the above language groups in cost-effective formatters.
- HP printers ignore unsupported commands.

1.7 Language Design Guidelines

A general guideline to remember is: *PCL is designed to provide "tools" for ISVs, not total solutions for customers.* Define new escape sequences only if they add new tools, not when they duplicate functionality available through other avenues. An example of such restraint is the scalable type functionality that was added to PCL without adding any new escape sequences.

1. **Escape sequences should have only a single function.** Conflict may occur when future devices attempt to support two or more features simultaneously. The Fill Rectangular Area command is an example that violates this rule. This command supports two very different types of area fill. Similarly, Print Mode Selection allows specification of several different features.
2. **Escape sequences should not duplicate functions.** Do not use two or more escape sequences to provide a single feature. An example that violates this rule is the availability of both PCL Unit moves and decipoint moves.
3. **Preserve data.** Given the option to print or discard, data should be preserved for the customer. An example is the Font Select algorithm.

   Another example is *Esc%-1B*, which switches to stand-alone HP-GL/2. On a DeskJet 1200C, the printed graphic may be clipped if it is in portrait rather than landscape orientation; however, data usable for debugging is still delivered to the customer.

4. **Conserve paper.** Implementors should avoid printing blank or useless pages that are not specifically requested by the data stream (e.g., multiple form feeds). If an *EscE* is received before a mark is made on the paper, the device should reset without ejecting the page. On special paper devices using expensive media, like typesetters and film recorders, it may even be appropriate to ignore multiple consecutive form feeds.

   One appropriate application is the clipping of raster data overflowing the bottom page boundary. Attempts to favor guideline 3 over guideline 4 resulted in a "joke page" containing the "leftover" raster graphics and nothing else. For example, on DeskJets prior to 540, data sent to the unprintable area with perforation skip off was printed on the next page, followed by a form feed. This implementation is not recommended because it wastes paper. Implementors should decide which guideline is most important for their project.

5. **Use a single sequence with a binary header for functionality requiring multiple interdependent sequences.** The Configure Image Data command *Esc*v#W[binary data]* follows this guideline. Font selection could benefit from consolidation.
6. **Avoid implicit actions.** When raster graphics capability was first added to PCL, an implicit start/end of raster graphics "mode" was defined. This has caused a great deal of difficulty as the definition of raster graphics has expanded.
7. **Obsolescence of functionality is a lengthy and expensive procedure.** First, the obsolete functionality must be moved to the obsolete section of the Implementor's Guide. Next, products that contain the obsolete functionality must stop documenting that functionality. Finally, after two or three product generations, when most software supporting the product line has stopped utilizing the obsolete functionality, it can be removed from future devices.
8. **Avoid undocumented underware escape sequences.** These sequences, which may expedite manufacturing processes, perform diagnostics, or provide functionality in a controlled environment, are very difficult to track and support. See Chapter 6 for the recommended underware escape sequence format.

9. **Handle error conditions consistently.** Extreme error conditions can be handled differently across a range of devices without jeopardizing the language standard, but consistency is recommended.
10. **Remember that previous implementations are not always perfect**. Language definition may need to change or be refined at the expense of backward compatibility. Two examples are the "joke page" and the new End Raster command (*Esc*rC*). Another example is the admission for LaserJet III that HMI should not have changed when print direction changed. There will always be creative tension because a software application base is built around the first products released.

Guidelines for Specific Escape Sequences

1. PCL parameter ranges were originally defined for a 16-bit environment. Future parsers may need to accept 32 bit values as they implement features whose value fields extend beyond 32,767. "Movement" is one place where limits may be encountered. Decipoint moves, which are limited to 45.5", may not be adequate for large format devices greater than E-size. Other examples are large graphics transfer, font downloads, and user-defined patterns.
2. When the contents of a value field identified by a state variable are out of range, the previous value should be retained rather than resetting to the default (e.g., Render Alorithm). When the contents of a value field identifying a physical range are exceeded, the value field should be clamped to the associated limit (e.g., Move CAP commands).
3. Use 0, if possible, as a default value. Default values should be determined partially by the impact of the escape sequence on devices that will not implement the functionality.
4. Signs encountered in an escape sequence should be considered and interpreted, not ignored. Examples are horizontal and vertical CAP positioning (e.g., *Esc&a#H* and *Esc&a#V*), and font stroke weight (*Esc(s#B* and *Esc)s#B*).
5. A "W" or "w" terminator on an escape sequence should be reserved for binary data transfers.
6. Group characters and terminating characters may be chosen for their mnemonic value. However, since this is often impossible, nothing can be inferred from the meaning of any specific selection.
7. The "Q" symbol sets should be used only for special character sets and not documented for broad use.
8. Metric compatibility for at least some portion of font products is an *excellent* idea from a customer standpoint. It should be an objective for all devices.

1.8 References

The HP-GL/2 Implementor's Guide (villone@sdd.hp.com)

The Hewlett-Packard Book of Characters (HP BPR; HP Publication No. 5091-5154E)

The PCL 5 Printer Language Technical Reference Manual (Copyright 1992, PN 5961-0509)

PCL 5 Comparison Guide (Copyright 1994, PN 5961-0702)

Printer Job Language Technical Reference Manual (Copyright 1994, PN 5961-0704)

Printer Job Language ERS (HP BPR)

PCL 5 Color Technical Reference Manual (Copyright 1994, PN 5961-0635)

PML Protocol Specification (pml-editor@hpbs987.boi.hp.com)