2 PJL Command Syntax and Format

Introduction

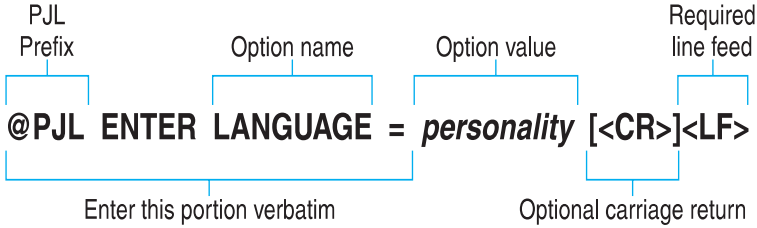
This chapter explains the conventions used to describe PJL command syntax. This chapter also describes the several different formats that PJL commands may take, giving examples of each. The chapter provides an explanation of how PJL-compatible printers handle illegal commands.

Syntax Conventions

The following syntax conventions are used to describe the PJL commands in this manual:

variables	Items in italics indicate names of variables.
COMMANDS	Items in uppercase letters indicate PJL command names and words that you type verbatim. PJL command names referred to in text are also in uppercase.
[ ]	Items in brackets [. . .] indicate optional parameters. The brackets themselves are not typed.
< >	Identifies a control code character, such as <CR> for carriage return, or a special defined identifier. The table on the next page lists the control codes and special identifiers used in the PJL syntax. (The < and > symbols themselves are not typed, but are replaced with the control codes or special identifiers they represent. For example, replace <FF> with the form feed character [ASCII 12].)
↩	This character indicates that the current line of code is a continuation of the previous line. For example, “All of this text belongs on the ↩ same line.”
|	A vertical bar indicates there is more than one optional parameter, such as LPARM and IPARM: [LPARM : *personality* | IPARM : *port*].

The following illustration is an example of a PJL command line containing the ENTER command:



The table below lists the control codes and special identifiers used in this manual:

<HT>	Horizontal tab character (ASCII 9).
<LF>	Line feed character (ASCII 10).
<CR>	Carriage return character (ASCII 13).
<SP>	Space character (ASCII 32).
<ESC>	Escape character (ASCII 27).
<FF>	Form feed character (ASCII 12).
<WS>	White space, a result of one or more <SP> or <HT>.
<words>	Printable characters (Roman-8 characters 33 through 255) and <WS>, starting with a printable character.
^D	PostScript end-of-file indication. It is not part of PJL, but is used to end PostScript examples.

Format of PJL Commands

All PJL command lines follow one of the following four formats. Each format defines how commands using that format are structured.

Format #1

```
<ESC>%-12345X
```

The only command that uses this format is the Universal Exit Language (UEL) command.

Format #2

```
@PJL [<CR>]<LF>
```

This format allows a PJL line with no command, and is used to add clarity to PJL command listings. You can use one or more of these commands to visually break up several lines of PJL commands.

Format #3

```
@PJL command [<words>] [<CR>]<LF>
```

The COMMENT and ECHO commands currently are the only PJL commands using format number 3.

Format #4

```
@PJL command [command modifier : value]
↪ [option name [= value]] [<CR>]<LF>
```

This format is used for all of the other PJL commands and is described in more detail below.

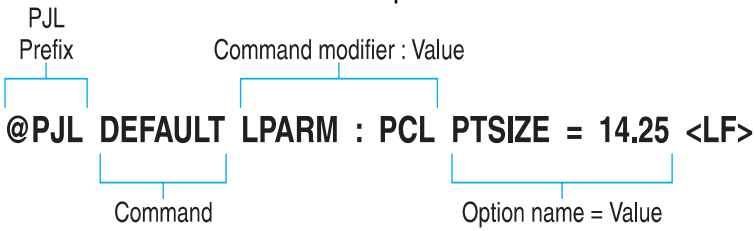
command — command is one of the assigned PJL command names, such as ENTER, RDYMSG, or RESET.

[command modifier : value] — The command modifier enables the user to specify what is effected by the command. For example, with the command modifier LPARM you can specify language-specific variables. A PJL command with a command modifier of LPARM : PCL only affects PCL-specific settings. A PJL command can contain only one command modifier. For example, in the command: @PJL SET

[LPARM : *personality*]|[IPARM : *port*] *variable* = *value* [<CR>] <LF>, you can use either the LPARM command modifier or the IPARM command modifier, but not both.

[*option name* [= *value*]] — The option parameter specifies an option or sets a command option to a certain value. Examples include "@PJL INQUIRE COPIES" and "@PJL ENTER LANGUAGE = *personality*." The range of values varies with each specific command and each printer model. A PJL command may have no options, or one or more options (an unlimited number).

The DEFAULT command shown below illustrates format number 4. The command sets the default PCL point size to 14.25.



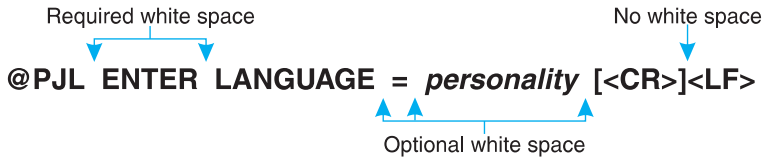
PJL Syntax Rules

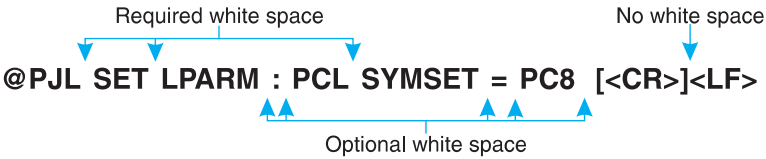
Following are the rules governing the use of PJL commands:

- The PJL prefix "@PJL" always must be uppercase. The remainder of the PJL command is not case-sensitive. For clarity, however, this manual shows other portions of PJL commands in uppercase to indicate portions that are not variable and that should be typed as shown (such as the first three words in the following command):
  `@PJL STMSG DISPLAY = "message" [<CR>]<LF>`
- Spacing between characters, or "white space," is comprised of one or more of either the space character (ASCII 32) or the horizontal tab character (ASCII 9). For clarity and consistency, this manual shows all white space as one blank space.

- The placement of white space in PJL commands depends on its location within the command. Some white space is required and some is optional:
- White space is required between the @PJL prefix and the PJL command name, and between the PJL command name and command modifiers. For example:
  `@PJL OPMSG DISPLAY` or
  `@PJL ENTER LANGUAGE = personality`
- If white space is shown in any other place in the command, it is optional (see the examples below).
- If white space is not shown between two portions of a command, white space is not allowed. An example is between the optional carriage return and required line feed character that terminate most commands.

For clarity, this manual consistently shows white space as one blank space. The PJL language requirements are shown in both examples below:





Types of Variables

PJL uses alphanumeric variables, numeric variables, and strings. The following explains the three types of variables and their ranges.

- **Alphanumeric variables.** Any combination of letters and digits, with the stipulation that the first character always must be a letter. Letters consist of the uppercase letters (Roman-8 characters 65 through 90) and lowercase letters (97 through 122). Digits consist of numbers 0 through 9 (characters 48 through 57).

  Examples of valid alphanumeric variables include:
  LaserJet279
  J1953

  Examples of invalid alphanumeric variables include:
  279LaserJet
  (Alphanumeric variables must begin with a letter)

  J 1953
  (Space characters [ASCII 32] are not allowed in alphanumeric variables)

- **Numeric variables.** Any number consisting of digits, with one optional decimal point and an optional + or – sign preceding the first digit. Only one decimal point may be used, and it must be placed somewhere after the first digit. Digits are not required after the decimal point.

  Examples of valid numeric variables include:
  0.123456
  –123.456
  +657000
  2468.

  Examples of invalid numeric variables include:
  .123456
  (The decimal point must be preceded by at least 1 digit)

  –123.45.6
  (Only 1 decimal point is allowed in a numeric variable)

  +657,000
  (Commas are not allowed in numeric variables)

- **Strings.** Enclosed in quotation marks, strings consist of any combination of characters from Roman-8 character 32 through 255, plus character 9 (horizontal tab), excluding character 34 (quotation marks).

  Examples of valid strings include:

  `"<HT>This is a valid string."`
  (Tabs are allowed in strings)

  `"Print job #4655"`

  Examples of invalid strings include:
  `"This is not a valid" string."`
  (Strings cannot contain quotation marks)

  `"This is also not<CR>valid."`
  (<CR> is not within the valid range of characters for a string.)

Note Strings displayed on the printer control panel are generally displayed using the Roman-8 symbol set. On HP LaserJet 4PJ, 4V, and 4MV printers, when the language is set to Japanese, strings which correspond to a control panel message are displayed on the control panel using the JIS X0201-76 character set.

This is also true for some other printers/languages. See the note on page 8-3.

Processing Invalid Commands

There are two general types of invalid commands: those commands with syntax errors, and those that have syntax or semantic warnings. Each type is handled differently.

- Syntax errors cause the printer to ignore the entire PJL command, and include errors such as unrecognized commands and command modifiers, strings missing closing double-quotes, numeric values missing digits before the decimal point, and numeric values encountered when alphanumeric values are expected. When the printer receives commands with syntax errors, it ignores the entire command.

  For example, the value portion of the JOB command's NAME option is a string and requires double quotes around the value (as shown below). In the second example below, the JOB command is ignored since the string (April Paychecks) contains the opening but not the required closing quotes.

  Valid command:
  `@PJL JOB NAME = "April Paychecks" <LF>`

  Invalid command:
  `@PJL JOB NAME = "April Paychecks <LF>`
- Syntax warnings and semantic warnings are issued for commands such as those having unsupported options, values that are out of range, values that are the wrong type or missing, or values that are included when none are allowed. When the printer receives commands with syntax or semantic warnings, it executes as much of the command as possible, but the portion of the command containing the warning is ignored.
- For example, in the following two sample PJL commands, START is a valid option for the JOB command, but FINISH is not a valid option (the END option should be used). The START option is executed, but the FINISH option is ignored.

  Valid command:
  `@PJL JOB START = 1 <LF>`

  Invalid command:
  `@PJL JOB START = 1 FINISH = HOME <LF>`

Note Any errors that occur during PJL parsing can be received by enabling device status as described in Chapter 7 (send the @PJL USTATUS DEVICE = VERBOSE command). Appendix D lists the status codes that are received by the host when an invalid command is received and unsolicited verbose device status is enabled.