Start up options
/AFC makes editing of automatic font changes easier (PRS Files).

/SUB makes editing of substitute fonts easier (PRS Files).

/UNITS=i ─ inches
    " ─ inches
    c ─ centimeters
    p ─ points (printer units)
    w ─ wp-units (1200ths)

/MONO ─ Use this option if you have a plasma, LCD, or composite display monitor.

/UPDATE <ALL FILENAME WITH FONTS ADDED> <NEW ALL FILE FROM WORDPERFECT>

/CONVERT <ALL FILENAME> ─ Convert the file and exits.

/CP=437 ─ Standard English code page
  850 ─ PC Multi-lingual code page
  851 ─ Greek
  8510 ─ Greek (alternative)
  860 ─ Portugal
  8600 ─ Portugal (Brazil alternative)
     861 ─ Iceland
  863 ─ Canada-French
  865 ─ Norway
  895 ─ Czech-Kamenicky
  899 ─ Cyrillic
  8990 ─ Cyrillic (Accented russian vowels)

/AUTOFONT <ALL FILENAME> - Add 3rd-party font information to the .ALL file. If the .ALL file is not specified on the command line, you will be asked to select from a list of the printers from the .ALL files in the directory from where PTR was run. Autofont support in PTR is currently implemented for HP Laser printers. Only those printers will show up in the list. If there aren't any .ALL files in the PTR.EXE directory, PTR will prompt you for the directory containing .ALL files.

The list of fonts to install created by 3rd-party font programs are usually in a file called \autofont\glue.txt. If this file is not found, PTR will prompt for it. The glue.txt file references other files which contain the font metric information (.TFM).

After running /autofont you will need to select the fonts from the autofont font libraries in WP.

/SYMSET - Used with /autofont to install a specific symbol set. /symset will bring up a menu of common symbol sets. The selected symbol set will only be installed if the glue.txt and .TFM file contain a font that supports that symbol set.

Characters or printer codes can be entered in an ASCII (""), decimal ([]), hexadecimal (<>), octal ({}), or binary (::) format. Each format is identified by delimiters which must enclose the value string. Only values [32] through [126] can be sent in ASCII format.

Character or code	ASCII	Decimal	Hex	Octal	Binary	
Escape		[27]	<1B>	{33}	:11011:	
A	"A"	[65]	<41>	{101}	:1000001:	

Multi-byte printer strings can be represented by separating byte values with commas within numeric format delimiters. No commas are needed within ASCII strings. An additional <u>ASCII text</u> format (delimited by BEGINTEXT and ENDTEXT) is like the ASCII format, but allows values from [0] to [127] in the string. Any new line in the text is sent to the printer as [10]. Delimiters BEGINTEXT and ENDTEXT must each appear alone on a line. The text format is useful for long command strings that control PostScript printers.

String	ASCII	Decimal	Hex	Octal	Binary	
Escape A		[27,65]	<1B,41>	{33,101}	:11011,1000001:	
AB	"AB"	[65,66]	<41,42>	{101,102}	:1000001,1000010:	

It is more space-efficient (but sometimes less readable) to represent all values in a WPDL string in the same format than it is to use a mixed format. For example, Escape A can be stored in 3 bytes as decimal [27,65] or hex <1B,41>, but requires 4 bytes to represent the mixed-format [27]"A".

Constants in expressions can be represented in ASCII ("), decimal, hexadecimal (trailing-h), octal (trailing-q), or binary (trailing-b) formats. Only values 32 through 126 can be represented in ASCII format.

ASCII	Decimal	Hex	Octal	Binary	
	10	0ah	12q	1010b	
'('	40	28h	50q	101000b	
'd'	100	64h	144q	1100100b	

Variables

Built-In Variables
WordPerfect puts information into built-in variables for the printer strings (WPDL expressions) to use. See

below for a description of each of the built-in variables. Built-in variables and other WPDL codes are displayed in uppercase.

Global Variables
Variables that may be accessed by the printer, font, group, map or sheet feeder strings of more than one definition. Global variables are accessed by **Alt-F8**. Pressing **Alt-F8** while on global variables will switch to the next type of variables (string, integer, functions).

Printer and Map Variables
Variables that are local to the current definition. **F9**, **Shift-F9** or **Alt-F9** will bring up local string, function, or integer variables unless the string does not allow local variables (font, group and sheet feeder strings) in which case they bring up global variables.

String Variables
WPDL expression that send codes to the printer.

Function Variables
WPDL expression that can send codes to the printer and return a value to an integer variable. The value is the last expression executed.

String and Function variables may not be assigned.

Integer Variables
A place where a number (double word value) is stored.

User-defined variables are lowercase ASCII, at least two characters long, do not begin with a numeric digit, and contain no spaces or brackets ("", [], <>, {}, ::).

General Built-In Variables

NO   Logical false. [0]

YES Logical true. [1]

REM Remainder of the last divide (/) or divide-and-round (//) operation. Not set by the modulo operation (%).

Job Variables
Set at the beginning of each new job.

COPIES   Number of copies being printed. Useful in Auto Multiple Copies string (e.g., [27] "&l" ASCII(COPIES) "X" for the HP LaserJet).

QUALITY Print quality. [0=draft, 1=medium, 2=high]

PAGE Physical page number in current job. Set at each new page.

GRAYSCALE   Darkness of rules or shaded boxes. [0-100, % of black] Useful in printing multiple-intensity, shaded boxes.

Form Variables
Set at each form change.

BIN   Number of the bin that is currently in use.

FORMTYPE Code for form type. [0=other, 1=blank, 2=bond, 3=letterhead, 4=envelope, 5=label, 6=transparency, 7=cardstock]

FORMLENGTH   Physical length of paper side (edge not inserted). [1200ths inch] At 6 LPI, lines/page = FORMLENGTH/200 (e.g., [27] "&l" ASCII(FORMLENGTH/200) "P" on the HP LaserJet).

FORMWIDTH   Physical length of paper top (edge inserted). [1200ths inch]

DUPLEX   Duplex type. [0=none, 1=long edge, 2=short edge]

Orientation Variable
Set at each orientation change.

ORIENTATION   Current font orientation. [0=portrait, 1=landscape, 2=reverse portrait, 3=reverse landscape] Useful in string that selects the font (e.g., [27] "&l" IF (ORIENTATION==1) "1" ENDIF "O" for the HP LaserJet).

> When Set Paper Length string is sent to printer, ORIENTATION is set for the next page, so it may not contain the current page's orientation. If you need the current page's orientation when setting the paper length for the next page, a user-defined variable (rather than ORIENTATION) should be used, and that variable should be set to the value of ORIENTATION as part of the Change Orientation string.

Graphics Variables
Set for each block of graphics data. A figure may be divided into multiple blocks of data to print.

GRAPHICS Flag specifying that graphics are being printed. [0-1]

GRAPHDENSITY Byte specifying current figure's graphics density. [Byte from printer's Graphics Resolution table]

GRAPHCOUNT   Number of graphic data bytes for current figure. Raster graphics only.

XRESOLUTION   Horizontal resolution of current figure. [dots/inch] Useful in string to transfer graphics.

YRESOLUTION   Vertical resolution of current figure. [dots/inch] Useful in string to transfer graphics.

GRAPHWIDTH   Width of figure. [XRESOLUTION units]

GRAPHHEIGHT   Height of figure. [YRESOLUTION units]

GRAPHDEPTH   (Currently not available.)

GRAPHBYTE   Contains the value that was found in the list of bytes that require special handling so that the special handling string can send something other than just the byte (substitute another value or send the value twice...).

Color Variables
Set at each color change. The values are determined by information in the printer's RGB (red-green-blue) table.

COLOR   Current color ID. [Number from printer's RGB table]

RED   Current red value. [0-RGB max value]

GREEN   Current green value. [0-RGB max value]

BLUE Current blue value. [0-RGB max value]

Attribute Variable
Set at each attribute change. Flags specifying which attributes are currently on due to use of a driver type.

Attribute variables are not set by manual or automatic font changes, but are set only when the printer is using a driver type to produce the attribute. To learn about current attributes caused by font changes, WPDL strings should test font (not attribute) variables. For example, if a printer may use an automatic font change to print italics rather than using the italics driver type, WPDL strings should test the value of SLANT (not ITALICS) to determine whether italic text is being printed.

ATTRIBUTES & 0001h Extra large
ATTRIBUTES & 0002h Very large
ATTRIBUTES & 0004h Large
ATTRIBUTES & 0008h Small print
ATTRIBUTES & 0010h Fine print
ATTRIBUTES & 0020h Superscript
ATTRIBUTES & 0040h Subscript
ATTRIBUTES & 0080h Outline
ATTRIBUTES & 0100h Italics
ATTRIBUTES & 0200h Shadow
ATTRIBUTES & 0400h Redline
ATTRIBUTES & 0800h Double underline
ATTRIBUTES & 1000h Bold
ATTRIBUTES & 2000h Strikeout

ATTRIBUTES & 4000h Underline
ATTRIBUTES & 8000h Small caps

<u>Font Variables</u>
Set at each font change.

RESOURCE ID of resource being used for printing in the current font. [Number from printer's Resource table]

FNTNUM Unique font identifier. [0-number of fonts in PRS file] Useful in font selection string.

GRPNUM Unique font group identifier. [0-number of groups in PRS file] Useful in group selection string.

SLANT Signed value specifying distance the top of a capital letter is moved from upright. [1200ths inch]

WEIGHT50 Font stroke weight (5.0 method). [0=extra light, 2=normal, 4=extra bold]

WEIGHT Font stroke weight (5.1 method). [0=extra light, 2=normal, 3=demi−bold, 4=bold, 6=black]

PTSIZE Font point size. [3600ths inch]

CAPHEIGHT Height of capital letters (C) in current font from baseline to top. [1200ths inch]

XHEIGHT Height of lowercase letters (x) in current font. [1200ths inch]

DHEIGHT Height of descender tails (gjpqy) in current font from baseline to bottom. [1200ths inch]

NOMWIDTH Nominal (average) width of font characters. [1200ths inch]

PITCH Average printing width of font characters (including letter spacing adjustments). [WUNITS] May differ from NOMWIDTH when font is being expanded or compressed for printing.

HUNITS Horizontal motion units (denominator). [units/inch]

HUNITSN Horizontal motion units (numerator). [HUNITSN/HUNITS = fractional units of motion in inches]

HS Current horizontal spacing amount. [HUNITS] Useful for Set HMI or Relative Horizontal Microspace commands (e.g., [27] "*p○" ASCII(HS) "X" for the HP LaserJet+).

WUNITS Character width units. [units/inch] Usually the same as HUNITS.

VUNITS Vertical motion units (denominator). [units/inch]

VUNITSN   Vertical motion units (numerator). [VUNITSN/VUNITS = fractional units of motion in inches]

VS   Current vertical spacing amount. [VUNITS] Useful for Set VMI or Relative Vertical Spacing commands (e.g., [27] "*p○" ASCII(VS) "Y" for the HP LaserJet+).

Character Variables
Set for each character.

XPOS Current horizontal position. [HUNITS] Useful in Absolute Horizontal Position commands.

YPOS Current vertical position. [VUNITS] Useful in Absolute Vertical Position commands.

<div align="center">

Arithmetic Operators
</div>

Operators allowed in any WPDL expression, listed in order of precedence. Parentheses can be used to enforce a different order of evaluation.

OPERATION	SYMBOL	FUNCTION
Negate	-	Change the sign of the value.
Bitwise Not	~	Invert value (e.g., 1101 to 0010).
Logical Not	!	Logical negation (all values treated as 0 or 1). Logical Not of non-0 is 0. Logical Not of 0 is 1.
Multiply	*	Product of two values.
Divide	/	Ratio of two values.
Rounded Divide	//	Ratio of two values, rounded.
Modulo	%	Remainder of a division. (e.g., 256 % 156 = 100, 40 % 15 = 10)
Add	+	Sum of two values.
Subtract	-	Difference of two values.
Shift left	<<	Multiply by power of 2. (e.g., A << 1 = A*2, A << 2 = A*4, A << 3 = A*8)
Shift right	>>	Divide by power of 2. (e.g., A >> 1 = A/2, A >> 2 = A/4, A >> 3 = A/8)
Less than	<	1 if first value is less than second, otherwise 0.
Less than or equal	<=	1 if first value is less than or equal to second, otherwise 0.
Greater than	>	1 if first value is greater than

		second, otherwise 0.	‖
Greater than or equal	>=	1 if first value is greater than	‖
		or equal to second, otherwise 0.	‖
Equal	==	1 if first value equals second,	‖
		otherwise 0.	‖
Not equal	!=	1 if first and second values not	‖
		equal, otherwise 0.	‖
Bitwise and	&	1100 & 1010 = 1000 (binary return).	‖
Bitwise or	|	1100 | 1010 = 1110 (binary return).	‖
Bitwise xor	^	1100 ^ 1010 = 0110 (binary return).	‖
Logical and	&&	101 && 000 = 0 (decimal return).	‖
Logical or	||	101 || 000 = 1 (decimal return).	‖
Assignment	:=	Set first value to that of second.	‖
Add & assign	+=	Set value of first to sum of first	‖
		and second.	‖
Subtract & assign	-=	Set value of first to difference	‖
		of second from first.	‖

String Output Functions

WPDL functions which send string values to the printer. Each function acts on its parameter's low-order word. To output the high-order word, the value must be manipulated (value >> 16 or value / 65536) to move it into the low word position.

SENDHI(var) Send var's high byte. [binary]

SENDLO(var) Send var's low byte. [binary]

SENDHILO(var) Send var's bytes; high, then low. [binary]

SENDLOHI(var) Send var's bytes; low, then high. [binary]

DOWNLOAD("file",start,count)
Send contents of file (path not given). The path is specified with printer selection in WP by the user. A starting file position can be specified after the filename. If present, the download starts at the file position instead of at the beginning of the file. After the start position, a byte count can be specified. If present, the download will start at the starting position and proceed until the byte count is sent. Replaces <X> in WP 4.2.

ASCII(expr,cnt) Send the value of expression as cnt characters (or more if necessary). Parameter cnt is optional. If cnt is larger than the output string, the string is left-padded with '0's.

Special Operators

FILESIZE("File")   Returns the size of a file.

READFILE("File",pos,count)

> Returns the contents of a file at a specified position. It requires as parameters a filename, a position, and the number of bytes to be read. Since the value of the bytes are being returned as the value of the function and values in WPDL are double words, a maximum of 4 bytes can be read from the file at one time. The first byte is the value of the low byte in the returned value, and so forth until the fourth byte in the file is the value of the highest byte of the returned value.

REPOSITION   Cause printhead to return to previous horizontal position. Useful for printers that have no backspace capabilities, or after printer functions which have uncertain effects on the horizontal position. Replaces <D> in WP 4.2.

;   Designates comment field. Anything between the semicolon and the end of the line is ignored.

PROMPT(msg, msg, ...)   Display messages on the message line of WP's printer control. Up to 66 characters are displayed.

CLEARPROMPT   Clear message lines of WP's printer control.

BEEP Cause WP to beep.

WAIT Cause WP to pause printing and wait for a GO in printer control.

SETMETHOD(driver,type)

> Selects the driver type to be used for the functions which implement more than one driver (e.g., Horizontal Motion Control: whether HMI, HMI with Microspacing, Relative, or Absolute). If SETMETHOD is not used to specify a driver type, the default driver type (marked in PTR with '*') is used. The following parameters are valid:

Driver	Type		Driver	Type
HORIZONTAL	HMI		DOUBLEUNDER	AUTOSAME
	MSHMI			AUTOSEPARATE
	RELATIVE			WPSEPARATE
	ABSOLUTE			AUTOUNDER2
VERTICAL	VMI			WPUNDER2
	RELATIVE			SINGLEUNDER

```
           | ABSOLUTE                |  AFCONLY
BOLD       | AUTOSAME      STRIKEOUT  | AUTOSAME
           | DOUBLEOVER               | AUTOSEPARATE
           TRIPLEOVER                 | WPSEPARATE
           MULTIPLEPASS               | BACKSPACE
           BACKSPACE                  | AFCONLY
           AFCONLY        REDLINE     | AUTOSAME
UNDERLINE  | AUTOSAME                 | AUTOSEPARATE
           AUTOSEPARATE               | WPSEPARATE
           WPSEPARATE                 | BACKSPACE
           BACKSPACE                  | AFCONLY
           AFCONLY        SHADOW      | AUTOSAME
ITALIC     | AUTOSAME                 | MULTIPLEPASS
           SINGLEUNDER                | AFCONLY
           AFCONLY        OUTLINE     | AUTOSAME
                                      | AFCONLY
SUBSCRIPT  | AUTOSAME      SUPERSCRIPT | AUTOSAME
           | NORMAL                   | NORMAL
```

Conditional Expressions

WPDL functions which control the flow of execution within expressions. These functions are useful for specifying conditional or repeating information in printer strings.

For example, if a printer's font deselect string turns off the bold attribute as a side effect, this side effect could be counteracted by conditionally turning bold back on in the same string (e.g., IF (BOLD) [27] "G" ENDIF on the IBM Graphics printer).

IF(expr)   Evaluate expr. If true (non-0) continue execution, skipping any following ELSEIF or ELSE clauses. If false (0) skip to next ELSEIF, ELSE, or ENDIF. IF must have a corresponding ENDIF.

ELSEIF(expr)   Optional operator that behaves identically to the IF(expr) function described above, but may not appear outside of an IF-ENDIF block.

ELSE Optional operator that marks the beginning of code executed if all previous expressions in the IF-ENDIF block are false.

ENDIF   Marks the end of the IF conditional.

WHILE(expr)Evaluate expr. If false (0) skip past ENDWHILE.

ENDWHILE Loops back to the WHILE to reevaluate expr.

|* Printers

This screen displays names for all printers defined in the current file. If no file is loaded, you can create a new file by adding names here and defining them on subsequent screens, or you can retrieve a file from disk. WordPerfect creates ".PRS" from definitions in a ".ALL" file.

Copy
If the destination file does not exist, it is created with the current printer definition and is marked with an ALL file type. If the file already exists, the definition is appended to the file, but any printer definition, font libraries, character maps, PS tables, and typefaces by the same names are overwritten by the incoming definition. Global variables are checked to see if an identical variable exists and uses it if it does. If a variable is found with the same name, but the expression does not match, the name of the variable will receive a "." followed by a "s", "i", or "f" (string, integer, function) and the index number of the variable tacked on to the end of the name.

|** Printer: (printer name)

A "standard" printer is one that uses escape sequences for printer control. PostScript printers use text command strings (defined by Adobe Systems Incorporated) rather than escape sequences for printer control.

Initialize and Reset
Strings for initializing and resetting the printer.

Printer Commands
Strings that control miscellaneous printer capabilities (changing orientation, duplex printing, centering printhead, manual paper feeding, setting page length).

Horizontal Motion (non-PostScript printers only)
Methods for controlling the printhead's horizontal position.

Vertical Motion (non-PostScript printers only)
Methods for controlling the printhead's vertical position on the page.

Attribute Methods (non-PostScript printers only)
Methods of performing attributes such as bold, underline, etc. when there is not an automatic font change for that attribute for the font.

Fonts
In WordPerfect 5.1 .ALL files, fonts are grouped into libraries based on the type of resource (memory, cartridge slot, etc.) they use and which printers share the fonts or by some other classification, such as manufacturer. In .PRS files, no font libraries are displayed, rather all of the fonts in the .PRS file are displayed in one list in order to make changing automatic font changes or substitute fonts easier. Fonts can only be added or deleted in a .ALL file. PTR will create a .ALL file if the printer in a .PRS file is copied to a new file. In order to select the fonts in WordPerfect, the file must have a .ALL extension.

Forms

List of form types available on the printer (e.g., blank, letterhead, envelopes, cardstock).

Graphics
Graphics resolutions and methods of printing graphics (raster or column data) and rules and shaded boxes.

Resources
This category lists the font resources available on the printer for loading and unloading fonts or groups. A resource is anything that limits the number of fonts or groups which can be loaded at a given time (e.g., memory for soft fonts, slots for font cartridges).

Miscellaneous Numbers
This category lists restrictions the printer places on minimum margins and the number of fonts that can be printed on each page. With this information WP enforces the printer's margin restrictions.

Helps and Hints about Printer
Explanatory notes about the printer. These notes are displayed in WP at printer selection time. They should explain any dip switch or other assumptions and any idiosyncracies the printer has.

Miscellaneous Questions
This category lists capabilities for printing super/subscripts, page advances, and graphics figures.

Soft Font Format Type
This category lists general printer classifications, identifying the downloadable font format supported by this printer.

|*** Printer Commands

Initialize Printer (download fonts, etc.)
Commands to initialize the printer to its normal state for printing in WordPerfect (e.g., to download initially present soft fonts, to set emulation mode). Time consuming commands that preferably you would only have to perform if the printer had been turned off or some other program had been run which changes the state of the printer.

Initialize at Start of Print Job
Commands that initialize printer for each job (e.g., shift into a particular emulation, disable perforation skip).

Reset at End of Print Job
Commands that restore printer to default parameters after print job. Should not contain general reset commands that may destroy downloaded soft fonts and overlays.

Initialize at Start of Page
Commands that set conditions cleared by a form feed.

Reset/Advance at End of Page
Commands that reset the printer for the next page (e.g., form feed). Sent regardless whether or not a sheet feeder is being used.

Printer Auto Page Advance?
Whether the Reset/Advance at End of Page string advances printhead to next page (YES or NO). If NO, WP advances to the next page using the printer's vertical motion driver.

Eject Page (when not using sheet feeder)
Commands to load the next page when no sheet feeder is being used. If a sheet feeder is being used, the expression in the sheet feeder definition will take the place of this command.

Set Page Length
Commands that set physical length of non-inserted edge of page. WP stores the print-time page length in 1200ths of an inch in variable FORMLENGTH. The HP LaserJet printing at 6 LPI could use [27] "&l" ASCII(FORMLENGTH/200) "P".

Center Printhead (for sheet feeder)
Commands that move printhead to center of platen.

Change Orientation
For printers that can rotate any font (e.g., Mannesmann Tally MT910, HP LaserJet 2000) commands to set the printing orientation. WP stores the print-time orientation in variable ORIENTATION.

Manual Feed On
Commands that turn on manual paper feeding or select the manual feed tray.

Manual Feed Off
Commands that turn off manual paper feeding or select the default paper tray.

Printhead Bias (Left)
Distance (in 1200ths of an inch) between left edge of page and left-most printhead position.

Printhead Bias (Top)
Distance (in 1200ths of an inch) between top edge of page and top-most printhead position.

|* Horizontal Motion**

This screen lists methods (driver types) for controlling the printhead's horizontal position. The default method for the current printer is marked (*).

Horizontal printing <u>resolution</u> is set in each font's Spacing Information screen for text printing, and in the printer's Graphics Resolution screen for graphics printing.

HMI (only)
The printer uses Horizontal Motion Indexing (HMI) and cannot microspace. Under HMI the printer is given a horizontal motion index (character width), after which all characters sent to the printer are printed at that width. Text may be printed at any pitch by using the appropriate index. In order to vary character spacing within text, the index must be changed for each width change.

HMI (with microspacing)
The printer prints characters using HMI (described above) and also supports microspacing (described below). (Such printers were called MSHMI printers in earlier versions of WP.)

Relative Horizontal Spacing (microspace)
The printer has relative microspacing capabilities. Under relative microspacing, every character is printed at an assigned width, but the printhead can be moved (spaced) anywhere relative to its current position. Graphics motion commands can usually be used to perform relative horizontal spacing. This driver type is used if no absolute positioning is available, and if microspacing is not slow or jumpy.

Absolute Horizontal Spacing (microspace)
The printer has absolute microspacing capabilities. Under absolute microspacing, every character is printed at an assigned width, but the printhead can be moved (spaced) anywhere as an offset from the left edge of the page. If it is available, this is usually the best horizontal driver type to use.

|**** HMI (only)

This screen lists printer control strings for performing horizontal motion indexing (HMI). Under HMI the printer is given a horizontal motion index (character width), after which all characters sent to the printer are printed at that width. Text may be printed at any pitch by using the appropriate index. In order to vary character spacing within text, the index is changed for each width change.

Set HMI (Horizontal Motion Index)
Commands that set the character width (index). The horizontal spacing variable, HS, should be used for the print-time width. At print-time WP sets HS at every width change. Diablo 630 could use [27,31] LO(HS+1).

Maximum Units
Maximum HMI value the printer supports.

Perform Backspace
Commands that move the printhead back (left) the width of the previously printed character.

Perform Carriage Return
Commands that move the printhead to left-most printing position.

Auto Line Feed on Carriage Return?
Whether printer automatically performs a line feed with each carriage return (YES or NO).

Set Forward Printing
For daisy wheel printers that have no internal buffering, commands that set printing direction left-to-right. Useful along with Set Reverse Printing string to allow WP to print forward and backward on alternate printhead passes.

Set Reverse Printing
For daisy wheel printers that have no internal buffering, commands that set printing direction right-to-left.

Useful along with Set Forward Printing string to allow WP to print forward and backward on alternate printhead passes.

|**** HMI (with microspacing)

This screen lists printer control strings for performing horizontal motion indexing (HMI) for characters and relative horizontal movement (microspacing) for spaces and width adjustments. (In earlier versions of WP, printers with both microspace and HMI capabilities were called MSHMI printers.)

Set HMI (Horizontal Motion Index)
Commands that set the character width (index). The horizontal spacing variable, HS, should be used for the print-time width. At print-time WP sets HS at every width change. Diablo 630 could use [27,31] LO(HS+1).

Maximum Units
Maximum HMI value the printer supports.

Set HMI for Each Character?
Whether HMI value should be sent for each character sent to the printer (YES or NO).

Perform Backspace
Commands that move the printhead back (left) the width of the previously printed character.

Perform Carriage Return
Commands that move the printhead to left-most printing position.

Auto Line Feed on Carriage Return?
Whether printer automatically performs a line feed with each carriage return (YES or NO).

|**** Relative Horizontal Spacing (microspacing)

This screen lists printer control strings for performing Relative Horizontal Spacing (microspacing). Under relative microspacing, every character is printed at an assigned width, but the printhead can be moved (spaced) anywhere relative to its current position. Graphics motion commands can usually be used to perform relative horizontal spacing. The motion is in increments of the printer's horizontal motion units.

Move Forward
Commands that move the printhead forward (right) a specified number of microspace units. At print-time WP stores in variable HS the number of units the printer should move.

Move Backward
Commands that move the printhead backward (left) a specified number of microspace units. At print-time WP stores in variable HS the number of units the printer should move.

Maximum Units
Maximum number of microspace units for a single relative horizontal movement supported by the printer.

Perform Carriage Return
Commands that move the printhead to left-most printing position.

Auto Line Feed on Carriage Return?
Whether printer automatically performs a line feed with each carriage return (YES or NO).

Send Attribute Byte
For printers that require an attribute byte with each character sent, commands specifying the attribute information. This information is found in WPDL attribute and font variables set by WP at print-time. The attribute byte may specify how hard to strike the character, how far to move the printhead, how far to move the ribbon.

|** Absolute Horizontal Spacing (microspacing)**

This screen lists printer control strings for performing absolute horizontal microspacing. Under absolute microspacing, every character is printed at an assigned width, but the printhead can be moved (spaced) anywhere as an offset from the left edge of the page. The motion is in increments of the printer's horizontal motion units.

Move to Horizontal Position
Commands that move the printhead to the specified horizontal position. At print-time WP stores in variable XPOS the position to which the printer should move. The HP LaserJet+ could use [27] "*p" ASCII(XPOS) "X".

|* Vertical Motion**

This screen lists methods (driver types) for controlling the printhead's vertical position. The default method for the current printer is marked (*).

Vertical printing <u>resolution</u> is set in each font's Spacing Information screen for text printing, and in the printer's Graphics Resolution screen for graphics printing.

VMI (Vertical Motion Index)
The printer uses Vertical Motion Indexing (VMI). Under VMI the printer is given a line height (vertical motion index), after which each line feed advances the printhead by that height. Text may be printed with any number of lines per inch by using the appropriate index. The index may be explicitly changed from line to line.

Relative Vertical Motion
The printhead advances vertically relative to its current position. The movement is in multiples of the printer's Units of Vertical Motion.

Absolute Vertical Motion
The printhead advances vertically to an offset from the top edge of the page. The position is in multiples of the printer's Units of Vertical Motion.

|** VMI (Vertical Motion Index)**

This screen lists printer control strings for performing Vertical Motion Indexing (VMI). Under VMI the printer is given a line height (vertical motion index), after which each line feed advances the printhead by that height. Text may be printed with any number of lines per inch by using the appropriate index. The index may be explicitly changed from line to line.

Set VMI (Vertical Motion Index)
Commands that set the line height (index). The vertical spacing variable, VS, should be used for the print-time line height. At print-time WP sets VS at every line height change. Diablo would use [27,30] LO(VS+1).

Maximum Units
Maximum VMI value the printer supports.

Perform Line Feed
Commands that move the printhead down (by the current VMI) to the next line.

Auto Carriage Return on Line Feed?
Whether printer automatically performs a carriage return with each line feed (YES or NO).

Perform Reverse Line Feed
Commands that move the printhead up (by the current VMI) to the next line.

Auto Carriage Return on Rev Line Feed?
Whether printer automatically performs a carriage return with each reverse line feed (YES or NO).

|** Relative Vertical Motion**

This screen lists printer control strings for performing vertical motion relative to the printhead position in multiples of the printer's Units of Vertical Motion.

Maximum Units
Maximum number of vertical units for a single relative movement supported by the printer.

Move Up
Commands that move the printhead up the page a specified distance. At print-time WP stores in variable VS the number of printer units to move. The HP LaserJet+ could use [27] "*p-" ASCII(VS) "Y".

Move Down
Commands that move the printhead down the page a specified distance. At print-time WP stores in variable VS the number of printer units to move. The HP LaserJet+ could use [27] "*p+" ASCII(VS) "Y".

|** Absolute Vertical Motion**

This screen lists printer control strings for performing vertical motion measured from the top edge of the

page. The position is a multiple of the printer's Units of Vertical Motion.

Move to Vertical Position
Commands that move the printhead to the specified vertical position. At print-time WP stores in variable YPOS the position to which the printer should move. The HP LaserJet+ could use [27] "*p" ASCII(YPOS) "Y".

|** Font Libraries ***
A font library is a collection of fonts that use the same resource type (memory, cartridge, print wheel). Often the fonts are grouped based on the ability to share those fonts on different printers. There are two types of libraries: libraries that may be shared by more than one printer, and those that are local to a printer definition. In .PRS files there are only non-shared libraries. To make setting automatic font changes and substitute fonts easier, in .PRS files all of the fonts in the printer definition are displayed in the same list, so the library selection list is skipped. Automatic font changes and substitute fonts may only be set up in non-shared libraries. Fonts may be copied from a shared library to a non-shared library by marking the fonts and using the copy or move option. Note: A shared library often has more than one printer definition using it, so if you <u>move</u> the font from a shared library to a non-shared library it will not be accessible by the other printer definition.

Font Library Name
When a non-sharable library is created without using a pattern, no library name is entered (the name of the library will be the printer's name in all uppercase).

Mark (*)
All libraries in the .ALL file are displayed, only the libraries which the selected printer uses are marked with the asterisk (*). Copying or doing a text out while editing a printer will perform the operation only on the font library that is highlighted. You can mark sharable font libraries to copy to another file or to do a text out only if you enter the font library screen from the top level (**Ctrl-F8**).

Update Flag
The update flags for sharable libraries in the .ALL files that ship with WordPerfect are clear. The update flags for sharable libraries that users or 3rd party programs create are automatically set.

PTR /UPDATE <FONTSADD.ALL> <NEW.ALL>

can be used to update a new .ALL file (from WordPerfect) with the sharable libraries that the user added to the previous .ALL file.

|** Fonts**

A font is a collection of characters with specific attributes in common (e.g., orientation, character set, pitch, point size, typeface). At least one font must be defined for each printer. Some fonts may be members of font groups (sets of fonts that must be loaded and unloaded as a unit). When editing a group, the fonts which belong to the group are marked (*). Each font may require some of the printer's font resources (e.g., memory for soft fonts, slots for font cartridges).

Font Name
Any leading parenthetical expression, though displayed in WP with the font name in the Cartridges and Fonts list, is <u>not</u> displayed at font selection time. Such expressions may contain cartridge information (e.g., (AD, Port) Courier 10pt 10pitch, (AZ, Land) Courier 10pt 10pitch).

|*** Font: (name)**

Typeface
This category lists attributes which define the appearance of the font characters. Each font must specify one typeface defining this appearance.

Orientations
This category lists orientations (portrait or landscape) in which the font may be printed. Depending on the capabilities of the printer, the font may be printed in more than one orientation.

Character Map
This category lists the font's character map (list of font characters and the control strings required to print them). Each font must specify one character map defining the characters in the font.

Size and Spacing Information
This category lists the font character primary sizes (point size, leading), horizontal and vertical motion resolutions, and which proportional-spacing table (if any) describes the character widths. Mono-spaced fonts need no proportional-spacing table.

Load and Select Strings
This category lists command strings the printer requires for loading, unloading, selecting, and deselecting the current font.

Groups
This category lists the current printer's font groups. A group is a collection of fonts which must be loaded and unloaded as a unit (e.g., fonts on a cartridge, soft fonts on a diskette). If the current font is a member of a font group, the group should be specified here.

Quality
This category lists up to three printing qualities (draft, medium, high) that may apply to the current font. The font may support one or more printing qualities.

Miscellaneous Questions
Miscellaneous information that WordPerfect needs to know in order to handle various situations better.

Automatic Font Changes (non-sharable fonts only)
This category lists automatic font changes (AFCs) for the current font's printing attributes (e.g., bold, italics, large, subscript). An AFC is the name of a font WP should use to represent a printing attribute for the given font. For example, when WP is printing text in the current font, an AFC could direct WP to use another font to print italicized text. Only fonts available in the current printer definition can be used to define each font's AFCs. When defining a printer, define all fonts before creating AFCs for each.

Substitute Fonts (non-sharable fonts only)
This category lists which fonts can be used to print characters not found in the current font. Up to nine substitute fonts can be listed in priority of the size and appearance best matching the current font. When defining a printer, define all fonts before creating substitute font lists for each.

|**** Typefaces ***

This screen displays the names of all typefaces defined in the current file. A typeface is the general appearance and structure of the characters in a font. Within a printer definition, each font must specify one typeface that describes its characters. Under the font definition, that typeface is marked (*).

|***** Typeface: (name) ****

This screen displays definition categories for the typeface named at the top of the screen. A typeface is the general appearance and structure of characters in a font. Each printer font must specify one typeface that describes its characters.

|****** Appearance/Style *****

This screen lists attributes describing the general appearance, style, or use of a typeface. Those attributes that apply to the current typeface are marked (*).

Casual
This typeface is appropriate in casual settings (e.g., Dom Casual).

Connected Letters
The letters are connected to each other like cursive handwriting (e.g., Commercial Script).

Decorative
The typeface is purely decorative and would probably be used sparingly and to draw attention (e.g., typefaces designed with rainbows, stars and stripes, vines and leaves).

Formal
Typeface is appropriate in formal settings (e.g., script or novel typefaces used for wedding invitations and restaurant menus).

Futuristic
Typeface appears futuristic (e.g., like a computer display or LCD).

Old Style
Typeface is old fashioned (e.g., Old English, Raphael, Uncial).

Script or Calligraphic
Typeface is fashioned after characters made with hand brushed strokes. Script typefaces generally have as few strokes as possible and are smooth and flowing (e.g., Commercial Script), whereas calligraphic

typefaces have many strokes, often at sharp angles (e.g., Old English).

|******** Attributes ***

This screen lists printing attributes that apply to a font's typeface characters. Those attributes that define the current typeface are marked (*).

Italic or Oblique
Characters are italicized or slanted.

Outline
Each character shows the profile or outline of the letter it represents with white (open) centers.

Shadow
Characters look three dimensional because of gray or black shadows in one or two directions.

Small Caps
Lowercase letters are capitals printed smaller than uppercase capitals.

|******** Serifs ***

This screen lists attributes that define serifs in a typeface. Serifs are fine lines projecting from the main strokes of a character, usually at the corners. Those attributes that apply to the current typeface are marked (*).

Cupped
Serifs are concave or cupped.

Exaggerated
Most serifs are prominent or long.

Hairline
Serifs are very thin or hairline strokes.

Slab
Serifs are nearly as thick as (or thicker than) the main strokes and are somewhat rectangular.

Slanted
Serifs make an obvious angle (e.g., in characters bdlk).

Transitional
The angles between main strokes and serifs are curved.

Triangular
The angles between main strokes and serifs are sharp.

Ball
Serifs are stubby, round, or spiral.

|****** Shape ***

This screen lists attributes describing the general shape of typeface characters. The shape of a character is determined by its main strokes. Those attributes that apply to the current typeface are marked (*).

Curved or Bowed Lines
Diagonal lines are not straight (e.g., in the letter X).

Nonconnecting Enclosures
Lines that form enclosures do not connect (e.g., in letters bdgpqBP).

Round
General character shape is circular rather than oval (e.g., in letters CDGOQ).

Square
General character shape is square or geometric rather than oval (e.g., in letters ABCDGJKO, OCR fonts).

|****** Stress (Line Thickness) ***

This screen lists attributes that describe the stress, or line thickness, of characters in a typeface. Those attributes that apply to the current typeface are marked (*).

Angular
Slanted lines (and portions of curves) have thicker strokes than horizontal and vertical lines.

Exaggerated
The difference between thick and thin strokes within a character is extreme (e.g., Broadway or Manhattan typefaces).

Uniform
All lines have nearly the same thickness.

|****** Weight ***

This screen lists attributes describing the typeface weight. Weight describes how bold or black a typeface is. The attribute that describes the current typeface is marked (*).

Demi─bold fonts are intended for those who want a base font that is a little darker than normal.

|****** Proportions ***

This screen lists typeface character proportions with respect to the point size.

Capital Height (% of Point Size)
Height of uppercase letters.

Descender Height (% of Point Size)
Distance from the bottom of descenders (e.g., fgjpqy) to the baseline (lowest point on most capital letters).

Lowercase x Height (% of Point Size)
Height of lowercase characters not including ascenders and descenders.

Lowercase t Height (% of Point Size)
Height of lowercase letter t (including ascender, excluding descender).

Maximum Ascender Height (% of Point Size)
Height of tallest ascender (e.g., bdlh). An ascender is that part of a character extending above the x height.

Slant Adjust (± % of Point Size)
Signed amount that the top of each capital letter is moved forward from its upright position in proportion to its point size.

|**** Orientations**

This screen lists printing orientations supported by WP. The class of orientations in which the current font may be printed is marked (*).

Portrait
Font prints only left-to-right across narrow direction of page.

Landscape
Font prints only bottom-to-top across long direction of page.

Reverse Portrait
Font prints only right-to-left (upside down) across narrow direction of page.

Reverse Landscape
Font prints only top-to-bottom across long direction of page.

Portrait & Landscape
Font can print in both portrait and landscape orientations.

Portrait, Landscape, Reverse Portrait, & Reverse Landscape
Font can print all four orientations.

|**** Character Maps ***

This screen displays all character maps defined in the current file. A character map is a table of characters and the printer command string mapped to each. To print a character, WP sends the printer string mapped to

it. Each printer font must specify one character map that defines the characters it contains and the printer codes that print them. Under a font definition that character map name is marked (*).

|***** Character Map: (name) ***

This screen defines the characters of a font or family of fonts. Each character that exists in the font is assigned a control string (the character mapping) for printing the character. No strings are given for characters not supported by the font. Each font must specify one character map defining its characters. Many fonts may use the same character map.

At print-time, if WP needs to print a character not defined in the current font, it searches that font's substitute font list for one that does have the character. WordPerfect will print the characters not found in the maps using the printer's graphics capabilities (if any).

Number
The numeric character ID, comprised of two parts: the character set number and the character number within the set (e.g., 0,126 indicates character set 0, character 126).

The Go To key positions the cursor on a specified character. The character specification may be a letter (ASCII character), single number (character set number), or two numbers separated by a comma (character set and character numbers).

Description
The character name.

Words within line-draw character names represent to which sides of the character cell (top, right, bottom, left) a line extends. Uppercase words denote double lines, lowercase words denote single lines.

Printer Command String
The command string required to print the character.

|**** Size and Spacing Information**

This screen lists information on the size of characters and spaces in the current font. Each font character has the same width unless a proportional-spacing (PS) table has been selected below.

Point Size (1 Point = 1/72 Inch)
Distance between consecutive baselines of text in the current font. The point size includes the point size and the leading.

Font Cell Height (Points)
Height of font cell measured from top of capitals and ascenders (bdhkl) to bottom of descenders (gjpqy), not including white space left above and below characters (leading).

Additional Leading (Points)
Amount of white space to add between lines of text in the current font.

Width Scale Factor (PS and PostScript fonts only)
Value by which the current PS table widths should be scaled to describe this font. Used for PS tables defined for similar fonts of differing sizes (e.g., a font with normal and double widths).

Pitch (Characters per Inch) (non-PS and non-PostScript fonts only)
Width (including white space) of current font's characters (mono-spaced). Pitch is the distance the printhead moves after printing each character.

Font Width (non-PS and PostScript fonts only)
Width (in Horizontal Spacing Units) of font's characters. For PostScript fonts, the width is measured in PostScript units.

Character Cell Adjust (± 1200ths)
Signed distance (in 1200ths of an inch) by which all font characters must be shifted right to left-justify each within its character cell. A value entered here is equivalent to a horizontal adjust factor entered for every character in the font.

For fonts which include much white space in each character cell, this value is useful to adjust all characters left so that even when mixed with other fonts, the page's left margin is straight. (The eye wants to see characters, not character cells, lined up on the left margin.)

Baseline Bias Factor (Points)
Amount by which font characters must be lowered or raised to bottom-justify each on the baseline. The baseline is an imaginary line drawn through the lowest point of non-descending capital letters.

A positive bias indicates that the characters print above the desired baseline and causes WP to print them lower; a negative bias indicates that the characters print below the desired baseline and causes WP to print them higher.

Amount to Slant Font (Degrees) (PostScript fonts only)
Signed number of degrees (from -90 to 90, exclusive) the font is slanted clockwise.

Horizontal Spacing Units (non-PostScript fonts only)
Units of horizontal motion for this font (in units per inch).

Vertical Spacing Units (non-PostScript fonts only)
Units of vertical motion for this font (in units per inch).

Proportional-Spacing Table: (name)
Name of PS table defining this font's character widths ("None" for fixed pitch font). Press [Enter] to select, deselect, or edit PS table.

NOTE
When a PS Table is assigned to the current font, the following information from the table appears at the bottom of the screen. It cannot be changed on this screen, but can in the PS Table definition.

Average Scaled Width (PS Table Units):
Average width of characters in this font. Average width of characters in PS table scaled by Width Scale Factor given above.

Average PS Table Width (PS Table Units):
Average width of characters in PS table.

PS Table Units:
For non-PostScript fonts, units for measuring character widths in the PS table (in units per inch). For PostScript fonts, -1. (Each PostScript character width is a function of the font's point size.)

Point Size PS Table Was Created for:
Point Size of font for which the PS Table was created.

|**** Proportional-Spacing Tables ***

This screen displays all proportional-spacing tables defined in the current file. A proportional-spacing (PS) table is a list of characters in a proportionally spaced font and the width of each. WP uses the PS table information to perform right justification, tabulation, repositioning on a second pass, and other horizontal positioning functions. Within a printer definition, each font may (if its characters are not all the same width) specify one PS table that defines the widths of its characters.

|***** Proportional-Spacing Table: (name) ****

This screen defines the character widths, adjust factors, and kerning values of a proportionally spaced (PS) font or family of fonts. Each character defined in the PS font's character map should be assigned a width, and optionally, an adjust factor and table of values for kerning the character with others. This information is used by WP to calculate proper horizontal positioning and right justification.

The arrow keys move the cursor among Width, Adjust. [Tab] moves the cursor from Width/Adjust/Kern to Units, Point Size and Kern Table.

Number
The numeric character ID, comprised of two parts: the character set number and the character number within the set (e.g., 0,126 indicates character set 0, character 126).

The Go To key positions the cursor on a specified character. The character specification may be a letter (ASCII character), single number (character set number), or two numbers separated by a comma (character set and character numbers).

Description
The character name.

Words within line-draw character names represent to which sides of the character cell (top, right, bottom, left) a line extends. Uppercase words denote double lines, lowercase words denote single lines.

Width
Width of character cell in terms of PS table units, displayed at the bottom of the screen.

Adjust
Signed number of printer motion units needed to center character within its cell width. Useful for characters which print off-center (e.g., on some older daisy wheel fonts). Printer motion units are 1200ths of an inch for PostScript definitions.

Units
Units for the PS table values (in units/inch, e.g., 300 = 300ths of an inch).

Point Size
Point Size of font for which the PS table was created. Other fonts (of different size) may also use this PS table.

Kerning Table
Multiple PS tables may share the same kerning table.

Tab
Rotates cursor from Width/Adjust/Kern to Units to Point Size.

|****** Kerning Tables * ***

Table of kerning values for current character. For characters that should be kerned (horizontally adjusted to correct wide or narrow inter-letter white space) when they follow the cursor character, the kerning values are entered in printer motion units. Printer motion units are 1200ths of an inch for PostScript definitions.

|******* First Character of Kern Pair: (name) ** ****

This screen lists characters that should be kerned (horizontally adjusted to correct wide or narrow inter-letter white space) when they follow the character named at the top of the screen.

The Go To key positions the cursor on a specified character. The character specification may be a letter (ASCII character), single number (character set number), or two numbers separated by a comma (character set and character numbers).

Value
Signed distance (in printer motion units) that the cursor character should be right-adjusted to kern it properly after the first character (named at the top of the screen). A negative number moves the character left.

Second Character of Kern Pair
Name of characters that may be kerned when they follow the first character (named at the top of the screen).

Next kern value
Alt-F3 will move the cursor to the next entry which has a value defined.

Words within line-draw character names represent to which sides of the character cell (top, right, bottom, left) a line extends. Uppercase words denote double lines, lowercase words denote single lines.

|****** Load and Select Strings

This screen lists control strings that govern loading and selection of the current font. The WPDL expressions making up the font strings may refer to font and printing attribute variables to determine which commands should be sent to the printer.

Load Font
Commands that load this font. If no commands are given here, the font should be specified as built-in (always present) or as part of a group.

Unload Font
Commands that unload this font (if it is not built-in or part of a group).

Select Font
Commands that select this font as the active font.

Deselect Font
Commands that counteract any of Select Font string's side effects that are not counteracted by other fonts' selection strings. Many fonts do not need a deselect string.

|****** Groups

This screen displays the names of all font groups defined for the current printer definition. Each group may require some of the printer's font resources (e.g., memory for soft fonts, slots for font cartridges).

|******* Group: (name)

This screen displays definition categories for the font group named at the top of the screen. A group is a collection of fonts which are loaded and unloaded as a unit (e.g., fonts on a cartridge, soft fonts on a diskette).

|******** Fonts in Group

This screen lists all the fonts in the current printer definition. Fonts marked (*) belong to the font group named at the top of the screen. You may press **Alt-F3** to position to the next font that is marked.

|******** Load/Unload Commands

This screen lists control strings for loading and unloading the current font group.

Load Group
Commands that load the font group. May use variable GRPNUM for a unique group ID (set by WP).

Unload Group
Commands that unload the font group.

|****** Miscellaneous Numbers**

Resource Quantity Used
Cartridge fonts use 1 cartridge slot, memory groups may use a certain amount of memory in kilobytes or a certain number of fonts depending on how the resource is set up.

|****** Miscellaneous Questions**

Include in PRS file
The Group and the fonts which belong to the group will automatically be included when WP creates a PRS file from the ALL file. Should not set for fonts that are not built-in and cannot be marked with a plus in WordPerfect (only swappable or built-in fonts).

|**** Automatic Font Changes For (name) (non-sharable libraries only)**

This screen displays automatic font changes (AFCs) for the current font's printing attributes (e.g., bold, italics, large, subscript). An AFC is the name of a font WP should use to represent a printing attribute for the given font. For example, when WP is printing text in the current font, an AFC could direct WP to use another font to print italicized text. Only fonts available in the current printer definition can be used to define each font's AFCs.

Switch
Displays cross-reference list of all fonts and their AFCs.

|***** Attribute, Character Set or Orientation**

Mark the font (*) to switch to when printing a character with the attribute or in the character set or orientation listed in the header.

|**** Automatic Font Change Cross-Reference For (name)**
   (non-sharable libraries only)

This screen displays a cross-reference of automatic font changes (AFCs) for fonts in the current printer definition. In the cross-reference table, all fonts (rows) and all printing attributes (columns) are listed. The table entry specifies whether an AFC exists for the font-attribute combination, and whether that AFC is to the cross-referenced font named at the top of the screen.

Font Names and Printing Attributes
Each row names a font in the current printer definition. Each column symbolizes a printing attribute for which an AFC can be specified. The attribute symbols are defined below.

Symbol	Attribute	Symbol	Attribute
X	Extra Large Print	Ñ	Multinational 1
V	Very Large Print	2	Multinational 2
L	Large Print	╫	Box Draw
s	Small Print	¶	Typographical Sym.
f	Fine Print	☺	Iconic Symbol
▲	Superscript	±	Math
▼	Subscript	⌠	Math Extension
O	Outline	Σ	Greek
c	Small Caps	H	Hebrew
I	Italics	C	Cyrillic
▒	Shadow	¥	Japanese Kana
+	Redline	U	User
=	Double Underline	↑	Portrait
B	Bold	←	Landscape
—	Strikeout	↓	Rev. Portrait
_	Underline	→	Rev. Landscape
A	ASCII		

○

For the column attribute, the row font has an AFC, but not to the cross- referenced font named at the top of the screen.

*

For the column attribute, the row font has an AFC to the cross-referenced font named at the top of the screen.

Enter
Change cross-reference font (named at the top of the screen) to the cursor font. In the cross-reference table, an asterisk (*) indicates an AFC to this font.

Switch
Displays AFCs for current font.

|****** **Substitute Fonts**

This screen displays all fonts of the current printer definition and gives a prioritized list of which (if any) can substitute for the current font. A substitute font is one that WP may use to print any character not defined in the current font. Up to nine substitute fonts can be listed in priority of the size and appearance best matching the current font.

Priority

Valid priorities are 1 through 9, with 1 being the highest. For each font you may type any priority, but the current priority list is recalculated if necessary to keep all priorities consecutive. You may type asterisk (*) for the next unused priority or if priority is unimportant.

|****** Quality

This screen displays the print qualities supported by the current font. When defining a new font you may specify more than one quality for it if you want it used for printing in any of those qualities, otherwise, you may define each quality as a separate font and mark only one quality for each.

At print-time, the WP user selects the print quality desired and WP sets the WPDL built-in variable QUALITY to that quality (0=draft, 1=medium, 2=high). Within the printer definition the QUALITY variable may be used to determine which control strings should be sent to the printer, but such strings should not change the current typeface, pitch, or other font attributes as a side-effect. (If they do, they specify a <u>font change</u> rather than a quality change.)

For example, the Star NB24-15 could be defined to print in Pica for draft quality, emphasized for medium quality, and NLQ for high quality. All three qualities would be marked (*), and the font selection string would test QUALITY to determine which control strings to send to the printer.

High Quality
Current font useable in high-quality printing.

Only high-quality fonts can be selected in WP for printing, but if the user desires to print in another quality, WP prints in the best-matching font that does support that quality. That is, while draft fonts cannot be user-selected for printing, they may be WP-selected to replace high quality fonts for draft printing.

Medium Quality
Current font useable in medium-quality printing.

Draft Quality
Current font useable in draft-quality printing.

The primary reason for defining draft fonts is to allow fast printing, so any slow-printing font (e.g., a PS font that uses graphics movement to microspace for right justification) should not claim to support draft quality.

|****** Miscellaneous Questions

This screen lists miscellaneous font attributes. Those attributes that apply to the current font are marked (*).

Scaleable
Printer can scale this font to any point size (from one point up).

Use Font Only for Automatic Font Changes
Font is not intended to be selected in WP as a base font. The font should not appear in the WP font selection list.

Don't Use Font for Automatic Font Changes
Font will appear in the WP base font list and will have automatic font changes for it, but it will not appear in any other font's automatic font change list.

Diacriticals Positioned (Vertically) for Uppercase Characters
Font diacriticals are designed to be printed with uppercase (rather than lowercase) characters. If marked, WP moves the diacriticals down for printing over lowercase characters. If not marked, WP moves the diacriticals up for printing over uppercase characters.

Diacriticals Are Not Already Adjusted (Horizontally) in Italic Fonts
For italic or slanted fonts, diacriticals do not take into account the font slant (i.e., they appear where they would over non-slanted characters). The slant is defined with the font's typeface information.

Include in PRS file
Font will automatically be included when WP creates a PRS file from the ALL file. The font should be a built-in font, or one that can be loaded during a print job (shown as a plus (+) on the Cartridges and Font screen in WP).

|* Forms**

This screen displays the name and attributes of each form defined for the current printer. A form describes the size and type of sheet of paper on which the printer can print. When editing a form definition, use cursor keys to access each area in the table.

Form Type
General category describing the form (e.g., blank standard, letterhead, envelopes). A user-defined form type is available.

Size (Top & Side)
Physical sizes of top (inserted edge) and side.

Text Adjust (Top & Side)
Amount to adjust the top or left bias. A negative top adjust will move the text down. A negative left adjust will move the text right.

D = Duplexing
  N = No duplexing
  L = Long edge duplexing (side duplexing if portrait)
  S = Short edge duplexing (top duplexing if portrait)

O = Orientation
  P = Portrait
  L = Landscape
  B = Both portrait and landscape
     (only the ALL OTHERS form should be marked as both)

I = Initially Present?

WordPerfect will prompt the user to load the form at the start of the print job if the form is not initially present. WordPerfect will prompt the user to load the initially present form at the end of the job if another form needed to be loaded in its place.

L = Labels
  N = No labels
  Y = There is a label definition (also used for logical pages on a physical page).

Location

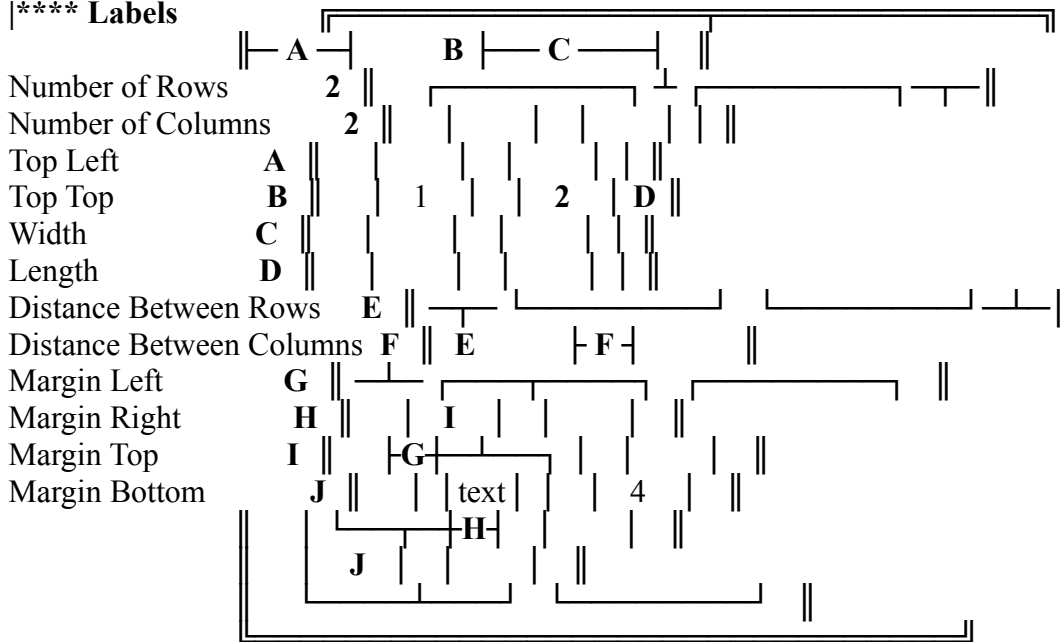Printer location of form type (continuous feeder, manual feeder, form bin).

|**** Form Types

The form type is independent of size. A user-defined type is available but its use may limit a document's ability to be ported to other printers, so this type should be used to describe only forms that cannot fit any standard type. The type describing the current form is marked (*).

[All Others]

Form not described by any standard or user-defined type.

|**** Labels

```
                              ┌──────────────────────────────────────┐
                  ║─ A ─┤      B ├─ C ──────┤  ║
Number of Rows        2  ║    ┌─────────────┐ ┌──────────────┬──┬─║
Number of Columns      2  ║    │       │  │        │ │ ║
Top Left          A  ║    │   │  │      │ │ ║
Top Top           B  ║    │  1  │  │  2   │ D ║
Width             C  ║    │   │  │      │ │ ║
Length            D  ║    │   │  │      │ │ ║
Distance Between Rows    E  ║ ─┬─  └──────────┘  └──────────┘ ─┴─║
Distance Between Columns F  ║  E       ├ F ┤        ║
Margin Left          G  ║ ─┴─  ┌──────┬──────┐  ┌──────────┐  ║
Margin Right         H  ║   │  I  │   │    │   │ ║
Margin Top          I  ║   ├G┤ ─┴─   │    │   │ ║
Margin Bottom         J  ║   │ │ text │  │    │ 4  │  ║
                         ║   │ ─┤H├─  │    │   │ ║
                      J  │ │    │  │    │ ║
                         ║─ └──────┴──────┘  └──────────┘  ║
                         ║──────────────────────────────────║
```

|**** Form Locations

This screen lists standard locations for forms on a printer. The location of the current form is marked (*).

Continuous

Form comes from printer's continuous form feeder.

Manual
Form comes from printer's manual feeder.

Bin:
Form comes from a printer's sheet feeder bin. The bin number is defined on the Sheet Feeder Bins screen.

|* Graphics**

Bitmap Graphics
This category lists methods for printing graphics.

List of Bytes that Need Special Handling
This category is for printers which do not allow certain bytes in graphics data. See special handling string under graphic expressions.

Resolutions
List of graphics resolutions (low, medium, high) supported by the printer. This information is left blank for printers that do not support graphics printing.

Rules and Shaded Boxes
This category lists methods for printing rules (blocks of colors, shades, or patterns).

|** Bitmap Graphics**

This screen lists methods (driver types) for bitmap graphics printing. The default method for the current printer is marked (*).

Monochrome Column Data Format
Printer prints multi-line rasters (a column of dots) at each printhead position. Dot matrix printers usually print graphics in column data format (e.g., IBM Graphics, Epson).

Monochrome Raster Data Format
Printer can print individual rasters (single dots) at each printhead position. Laser printers usually print graphics in rasters data format (e.g., HP LaserJet).

Graphics Not Supported
Printer has no bitmap graphics capabilities (e.g., Diablo 630).

|*** Monochrome Column Data Format**

This screen displays categories that define bitmap graphics printing.

Graphics Expressions
This category lists methods for printing bitmap graphics.

List of Bytes that Need Special Handling
This category is for printers which do not allow certain bytes in graphics data. See special handling string under graphic expressions.

Data Format Flags
This category lists formats defining data bytes sent to the printer.

|**** Graphics Expressions**

This screen lists printer control strings that govern column bitmap graphics printing.

Start Graphics Data Transfer
Commands that turn on graphics in a given density for a given number of data bytes. WP stores the print-time density character (from the Graphics Resolution table) in variable GRAPHDENSITY and the number of data bytes in variable GRAPHCOUNT. Sent at the beginning of each printhead pass of graphics printing.

End Graphics Data Transfer
Commands that are sent at the end of each printhead pass of graphics printing.

End Graphics
Commands that turn off graphics printing. Sent at end of each printhead pass of graphics printing.

Baseline Bias Factor
Printhead pin number (0-origin) specifying which pin prints at the baseline (lowest point of most capital letters).

Special Handling String
Expression that changes byte values (see List of Bytes that Need Special Handling) that are not allowed in graphics data to values that will produce the same or similar results. The byte is contained in the WPDL built-in variable **GRAPHBYTE**.

|**** Data Format Flags**

This screen lists formats defining data bytes sent to the printer.

Most Significant Bit at Bottom of Column
Whether high-order bit of data value is the at the bottom of the raster column (rather than the top).

Set Unused Bits in Data Bytes
Whether unused bits in data byte (those marked in the graphics resolution table with '.') should be set to 1 (rather than 0).

|*** Monochrome Raster Data Format**

This screen lists printer control strings that govern raster bitmap graphics printing.

Set Graphics Resolution
Commands that set the printer to a specified resolution. At print-time WP stores the resolution in variables XRESOLUTION and YRESOLUTION. Sent at the beginning of a figure. The HP LaserJet could use [27] "*t" ASCII(XRESOLUTION) "R".

Start Graphics
Commands that initialize graphics printing (e.g., [27] "*r1A" on the HP LaserJet). Sent at the beginning of a figure.

Start Graphics Data Transfer
Commands specifying number of graphics data bytes. WP stores the print-time number of bytes in variable GRAPHCOUNT. Sent at the beginning of each horizontal raster line. The HP LaserJet could use [27] "*b" ASCII((GRAPHCOUNT*300) / XRESOLUTION) "W".

End Graphics Data Transfer
Commands that are out put at the end of line of data transfer.

End Graphics
Commands that turn off graphics printing (e.g., [27] "*rB" on the HP LaserJet). Sent at the end of a figure.

Data Encoding Type
Value that specifies how graphics should be encoded. 0=raw (no encoding), 1=sixel encoding with 3F hex offset, 2=sixel encoding with 40 hex offset.

Special Handling String
Expression that changes byte values (see List of Bytes that Need Special Handling) that are not allowed in graphics data to values that will produce the same or similar results. The byte is contained in the WPDL built-in variable **GRAPHBYTE**.

|** Enhancement Board**

WordPerfect needs to know if you are using one of the enhancement boards listed.

|** List of Bytes that Need Special Handling**

This is a list of bytes which the "Special Handling String" (under Graphic Expressions) will be sent instead of the byte each time the byte appears in the graphics data.

|** Resolutions**

List of graphics resolutions supported by the current printer. Some printers have no graphics capabilities. Three resolutions are used by WP for low, medium, and high resolution graphics printing. The resolution and motion units are required for every printer that can print graphics. The other columns contain information useful only for column data graphics formats such as are printed by dot matrix printers.

Resolution (Vert and Horz)
The number of (vertical and horizontal) dots per inch that can be printed. At print-time WP stores these values in variables XRESOLUTION and YRESOLUTION.

Any vertical resolution may be defined that is both a multiple of the printhead's pins per inch and a divisor of the vertical motion units.

Motion Units (Vert and Horz)
Number of (vertical and horizontal) units per inch that the printhead can move while in graphics printing mode.

D = Graphics Density Number (column data format only)
ID (usually representing the printing density) which specifies the current graphics mode being defined. At print-time WP places this value in variable GRAPHDENSITY for use in graphics transfer strings.

The density may be an ASCII value that represents the horizontal resolution. For example, on an Epson printer this value could be 75 (K) for regular density graphics or 76 (L) for double density. In either case the graphics transfer string could be [27] GRAPHDENSITY LO(GRAPHCOUNT) HI(GRAPHCOUNT).

S = Pin Spacing Unit (Pins Per Inch) (column data format only)
Distance between consecutive pins on the printhead in pins per inch (e.g., 72 for Epson FX).

P = Number of Printhead Pins Used for Graphics (column data format only)
Number of printhead pins used for graphics printing.

Data Mask (column data format only)
Which bits are significant in each data byte. To print a column of dots, a 24-pin printer may require four data bytes, each of six significant bits and two unused bits. WP gives each unused bit the value 0, unless the Set Unused Bits in Data Bytes flag is marked (under the Column Bitmap Graphics, Data Format Flags screen).

|** Rules and Shaded Boxes**

This screen lists methods (driver types) for printing rules and other shaded boxes. Many laser printers (e.g., HP LaserJet+, Canon LBP) have built-in rule printing features, and most others can print rules but may require complex graphics control strings to do so. The default method for the current printer is marked (*).

Single Intensity Shading (Rules)
All rules are printed with a single shading intensity, color, or pattern.

Multiple Intensity Shading
Rules may be printed with various shading intensities, colors, or patterns (e.g., HP LaserJet+).

Rules Not Supported
Printer has no gray shading or rule printing capabilities.

|*** Single Intensity Shading (Rules)**

This screen lists control strings for printing rules in an intensity (shading) defined by the printer. A rule is a shaded (usually black) rectangle.

Set Rule Width
Commands that set the rule to a specified width. At print-time WP stores the width in variable GRAPHWIDTH.

Set Rule Height
Commands that set the rule to a specified height. At print-time WP stores the height in variable GRAPHHEIGHT.

Draw Rule
Commands that print the defined rule.

|***** **Multiple Intensity Shading**

This screen lists control strings for printing boxes in a variety of intensities (shades).

Set Box Width
Commands that set the box to a specified width. At print-time WP stores the width in variable GRAPHWIDTH. The HP LaserJet+ could use [27] "*c" ASCII(GRAPHWIDTH) "A".

Set Box Height
Commands that set the box to a specified height. At print-time WP stores the height in variable GRAPHHEIGHT. The HP LaserJet+ could use [27] "*c" ASCII(GRAPHHEIGHT) "A".

Set Box Intensity
Commands that set the box to a specified shading intensity. At print-time WP stores the intensity in variable GRAYSCALE. The HP LaserJet+ could use [27] "*c" ASCII(GRAYSCALE) "G".

Draw Box
Commands that print the defined shaded box (e.g., [27] "*c2P" on the HP LaserJet+).

|*** **Resources**

This screen displays the names of all font resources available in the current printer definition. A resource is anything that limits the number of fonts or groups which can be loaded at a given time (e.g., memory for soft fonts, slots for font cartridges).

Resource Type or Name
Type of resource. You can have a different name appear in the resource screen in WordPerfect by also specifying a name. This is necessary when there are more than one resource of the same type.

Quantity
Number of resources available (in terms of Units which follow).

ID = Resource ID Number
Unique numeric identifier. At print-time, WP stores the resource ID in the built-in WPDL variable RESOURCE at every font change. This variable may be referenced within any other WPDL expression to determine which resources are being used.

Type/Order
The swapping or loading method for fonts or font groups using the current resource. The method type is "load" or "swap." The method order may be "any" (any order) "lifo" (last in, first out), or "page" (may be swapped if not used on the current page).

I = Intervention Required?
Yes or **N**o specifying whether loading of fonts or groups into this resource requires the user to take some action (e.g., insert a font cartridge).

D = Display Libraries in WordPerfect?
Yes or **N**o specifying whether font libraries are displayed in the cartridges and fonts selection in WordPerfect.

|** Method of Loading or Swapping Fonts or Groups**

This screen lists swapping and loading methods for fonts or font groups. The loading method used by the current printer is marked (*).

Fixed
The fonts must remain loaded for the entire print job. They cannot be loaded or swapped during a print job.

Swap/Any
The fonts may be swapped in any order.

Load/Any
The fonts may be loaded in any order.

Swap/LIFO
The fonts may be swapped out only in the reverse order in which they were swapped in or loaded. (The last in must be the first out.)

Load/LIFO
The fonts may be unloaded only in the reverse order in which they were loaded. (The last in must be the first out.)

Swap/Page
The fonts must remain loaded during the printing of an entire page. They cannot be swapped out while a page is being printed.

|* Attribute Methods**

This category lists methods for printing attributes when no automatic font change is defined for the font when the attribute is on. Different methods may be used by using the WPDL SETMETHOD() command. For expressions requesting a signed value, use positive numbers to move down or right and negative numbers to move up or left.

|**** Bold

This screen lists methods (driver types) for printing bold text when no bold automatic font change is defined. The default method for the current printer is marked (*).

Auto Bold
Printer has a bold feature that automatically prints bold text without changing the current font.

Double Overstrike on Same Pass (True HMI Only)
Printer uses HMI for horizontal motion. Bold characters should be printed twice with a horizontal offset between each overstrike. Daisy wheel printers usually use one of the HMI driver types.

Triple Overstrike on Same Pass (True HMI Only)
Printer uses HMI for horizontal motion. Bold characters should be printed three times, with a horizontal offset between each overstrike. Daisy wheel printers usually use one of the HMI driver types.

Multiple Pass Overstrike
Bold characters should be printed a given number of times, each on one pass of the printhead. A horizontal offset may be given for each bold pass.

Backspace Overstrike
Printer has backspace capabilities. Bold characters should be printed a given number of times by backspacing for each overstrike. No horizontal offset is given for each overstrike.

Done by AFCs only
Printer has no capability for printing bold text, or all bold printing is governed by automatic font changes.

|***** Auto Bold

This screen lists control strings for printing bold text with the printer's bold feature.

Auto Bold On
Commands that turn on bold feature.

Auto Bold Off
Commands that turn off bold feature.

|***** Double Overstrike on Same Pass (True HMI Only)

This screen lists control strings for printing bold text by using HMI movement to overstrike each character.

Horizontal Offset (1200ths)
Distance between the pair of images (overstrikes).

|*** Triple Overstrike on Same Pass (True HMI Only)**

This screen lists control strings for printing bold text by using HMI movement to print each character three times.

Horizontal Offset (1200ths)
Distance between each pair of images (overstrikes).

|*** Multiple Pass Overstrike**

This screen lists control strings for printing bold text by overstriking bold characters on multiple printhead passes, possibly with a given horizontal offset between each pair of passes.

Number of Passes
Number of images (original + overstrikes) for bold text. The minimum and default value is 2.

Initial Horizontal Offset Bold Passes (1200ths)
Distance between first pair of images (passes 1 and 2).

Additional Horizontal Offset Each Pass (1200ths)
Distance between each pair of overstrike images (passes 2 and 3, 3 and 4, etc.).

|*** Backspace Overstrike**

This screen lists control strings for printing bold text by backspacing and overstriking each bold character a specified number of times.

Number of Images
Number of images (original + overstrikes) for bold text. The minimum and default value is 2.

|** Color**

This screen lists methods (driver types) for color printing. The default method for the current printer is marked (*).

Auto Color
Printer has a color feature that automatically changes the color of text being printed.

Color Not Supported
Printer has no capability for printing color text.

|*** Auto Color**

This screen displays categories that define color capabilities on the current printer.

Printer String
This category lists the strings that control the printer's color capabilities.

RGB (Red, Green, Blue) Table
This category lists the printer's available colors.

|**** Printer String**

This screen displays the printer strings that control the printer's color capabilities.

Set Color
Commands that set the printer text color. WPDL variable COLOR should be used to determine the print-time color to be set. (At print-time, WP sets COLOR with the ID from the RGB table at every color change.)

|**** RGB (Red, Green, Blue) Table**

This screen lists the printer's available colors. Each color is defined by what percent of red, green, and blue make up the color. Cyan, magenta, and yellow percentages are complements of red, green, and blue percentages. Either set of percentages defines each color.

ID Code
ID to represent each color. At print-time, for each color change, WP sets the built-in variable COLOR to the ID given here. COLOR may be tested in the Set Color string to determine the current color and send the appropriate printer codes to change to that color.

Red, Green, Blue
Percentage of each base color found in the printer color being defined. Black is 100% red, 100% green, 100% blue.

Cyan, Magenta, Yellow
Percentage of each base color found in the printer color being defined. Black is 0% cyan, 0% magenta, 0% yellow.

Maximum Value
Maximum value the printer uses to define a base color. At print-time, for each color change, WP sets the built-in variables RED, GREEN, and BLUE to the new color's red, green, and blue percentages multiplied by this maximum value. For example, if the RGB maximum is 255, then for a color made up of 67% red, 0% green, and 33% blue, RED=171, GREEN=0, BLUE=84. These variables may be used in the Set Color string to set the new color.

|** Double Underline**

This screen lists methods (driver types) for double underlining when no double underline automatic font

change is defined. The default method for the current printer is marked (*).

Auto Double Underline on Same Pass
Printer has a double underline feature that automatically underlines without changing the current font. Text and underlining are printed on one printhead pass.

Auto Double Underline on Separate Pass
Printer has a double underline feature that automatically underlines without changing the current font. Text is printed on one printhead pass and double underlining on another. Useful for printers that cannot double underline motion spaces (e.g., intra-word microspacing) on text-printing pass. Also useful for adjusting vertical position of double underline that would be printed on text-printing pass.

WP's Double Underline on Separate Pass
Printer has no double underline feature. WP should use a character (e.g., '=') to double underline on a second printhead pass.

Twin Pass - Auto Underline
Printer has no double underline feature, but has an underline feature. Printer should make two underline passes with a vertical offset between each pass.

Twin Pass - WP's Underline
Printer has no double underline feature and no character (e.g., '=') useful for double underlining. WP should use a character (e.g., '-') to underline on two separate passes, with a vertical offset between each underline pass.

Same as Single Underline
Double underlined characters should be printed with single underlining.

Done by AFCs Only
Printer has no double underlining capabilities, or all double underlining is governed by automatic font changes.

|*** Auto Double Underline on Same Pass**

This screen lists printer control strings for double underlining with the printer's double underline feature on the same pass on which text is printed.

Auto Double Underline On
Commands that turn on double underline feature.

Auto Double Underline Off
Commands that turn off double underline feature.

|*** Auto Double Underline on Separate Pass**

This screen lists printer control strings for double underlining with the printer's double underline feature on a

printhead pass after the text has been printed.

Auto Double Underline On
Commands that turn on double underline feature.

Auto Double Underline Off
Commands that turn off double underline feature.

Character to Use for Movement
For printers that cannot double underline motion spaces (e.g., microspacing), value of WP character to use for horizontal movement (e.g., 32 or ' '). Value = 256 * char_set_num + char_num.

Vertical Movement (1200ths)
For adjusting vertical position of double underline, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/3). A negative value to move up, a positive value to move down.

|*** WP's Double Underline on Separate**

This screen lists printer control strings for double underlining using a character (e.g., '=') on a second printhead pass after text has been printed.

Double Underline Character
Value of WP character to use for double underlining (e.g., 61 or '='). Value = 256 * char_set_num + char_num.

Vertical Movement (1200ths)
For adjusting vertical position of double underline character, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/3).

Adjustment Between Characters (1200ths)
Signed distance double underline character should be reprinted to the right so that consecutive characters touch to form a continuous line. The amount of white space between two printed double underline characters.

Adjustment at Start of Double Underlining (1200ths)
Signed distance double underline character should be reprinted to the right so that first character is fully double underlined. About half the amount of white space between two printed double underline characters.

Adjustment at End of Double Underlining (1200ths)
Signed distance double underline character should be reprinted to the right so that last character is fully double underlined. About half the amount of white space between two printed double underline characters.

Use positive numbers to move right and negative numbers to move left.

|***** Twin Pass - Auto Underline

This screen lists printer control strings for double underlining with the printer's single underline feature on two printhead passes after the text has been printed. A vertical offset specifies the distance between the underline passes.

Auto Underline On
Commands that turn on underline feature.

Auto Underline Off
Commands that turn off underline feature.

Character to Use for Movement
For printers that cannot underline motion spaces (e.g., microspacing), value of WP character to use for horizontal movement (e.g., 32 or ' '). Value = 256 * char_set_num + char_num.

Vertical Movement First Pass (1200ths)
For adjusting vertical position of first underline, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/3).

Vertical Movement Second Pass (1200ths)
For adjusting vertical position of second underline, signed distance the line should be moved down from the first underline. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/4).

Use negative values to move up, positive values to move down.

|***** Twin Pass - WP's Underline

This screen lists printer control strings for double underlining using a character (e.g., '-') on two printhead passes after text has been printed. A vertical offset specifies the distance between the underline passes.

Underline Character
Value of WP character to use for underlining (e.g., 95 or '_', 45 or '-'). Value = 256 * char_set_num + char_num.

Vertical Movement First Pass (1200ths)
For adjusting vertical position of first underline character, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/9).

Vertical Movement Second Pass (1200ths)
For adjusting vertical position of second underline character, signed distance the line should be moved down from the first underline. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/8).

Adjustment Between Characters (1200ths)
Signed distance underline character should be reprinted to the right so that consecutive characters touch to form a continuous line. The amount of white space between two printed underline characters.

Adjustment at Start of Double Underlining (1200ths)
Signed distance underline character should be reprinted to the right so that first character is fully underlined. About half the amount of white space between two printed underline characters.

Adjustment at End of Double Underlining (1200ths)
Signed distance underline character should be reprinted to the right so that last character is fully underlined. About half the amount of white space between two printed underline characters.

|** Italics**

This screen lists methods (driver types) for printing italic text when no italics automatic font change is defined. The default method for the current printer is marked (*).

Auto Simulated Italics (NOT Italics)
Printer has an automatic feature (e.g., oblique or color printing) that represents italics without changing the current font.

Same as Underline
Italics should be printed as underlined characters.

Done by AFCs Only
Printer has no italics capabilities, or all italics printing is governed by automatic font changes.

|*** Auto Simulated Italics (NOT Italics)**

This screen lists control strings for using a printer feature representing italics (e.g., oblique or color printing) on the same pass on which text is printed.

Auto Simulated Italics On
Commands that turn on simulated italics feature.

Auto Simulated Italics Off
Commands that turn off simulated italics feature.

|** Outline**

This screen lists methods (driver types) for outlining when no outline automatic font change is defined.

Auto Outline
Commands to control the printers outline feature (if one).

Done by AFCs Only
Outline will print in the base font unless the font has an automatic font change to an outline font.

|**** Redline

This screen lists methods (driver types) for redlining when no redline automatic font change is defined. Redlining is used to identify proposed additions to a document, usually by gray-shading the added text. The default method for the current printer is marked (*).

Auto Redline on Same Pass
Printer has a redline feature that automatically prints redline text without changing the current font. The text and redline are printed on one printhead pass.

Auto Redline on Separate Pass
Printer has a redline feature that automatically prints redline text without changing the current font. Text is printed on one printhead pass and redline on another. Useful for printers that cannot redline motion spaces (e.g., intra-word microspacing) on text-printing pass. Also useful for adjusting vertical position of redline that would be printed on text-printing pass.

WP's Redline on Separate Pass
Printer has no redline feature. WP should use a character (e.g., ▦) to redline text on a second printhead pass.

Backspace Redlining
Printer has backspace capabilities. Backspace over each character and print a redline character (e.g., ▦).

Done by AFCs Only
Printer has no redline capabilities, or all redlining is governed by automatic font changes.

|***** Auto Redline on Same Pass

This screen lists control strings for printing redline text with the printer's redline feature on the same pass on which text is printed.

Auto Redline On
Commands that turn on redline feature.

Auto Redline Off
Commands that turn off redline feature.

|***** Auto Redline on Separate Pass

This screen lists control strings for printing redline text with the printer's redline feature on a printhead pass after the text has been printed.

Auto Redline On
Commands that turn on redline feature.

Auto Redline Off
Commands that turn off redline feature.

Character to Use for Movement
For printers that cannot redline motion spaces (e.g., microspacing), value of WP character to use for horizontal movement (e.g., 32 or ' '). Value = 256 * char_set_num + char_num.

Vertical Adjustment
For adjusting vertical position of redline, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/7). A negative value to move up, a positive value to move down.

|*** WP's Redline on Separate Pass**

This screen lists control strings for printing redline text using a character (e.g., ▦) on a second printhead pass after text has been printed.

Redline Character
Value of WP character to use for redlining (e.g., 300h or 768 for ▦). Value = 256 * char_set_num + char_num.

Vertical Movement (1200ths)
For adjusting vertical position of redline character, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/9).

Adjustment Between Characters (1200ths)
Signed distance redline character should be reprinted to the right so that consecutive redline characters touch to form a continuous line. The amount of white space between two printed redline characters.

Adjustment at Start of Redlining (1200ths)
Signed distance redline character should be reprinted to the right so that first character is fully redlined. About half the amount of white space between two printed redline characters.

Adjustment at End of Redlining (1200ths)
Signed distance redline character should be reprinted to the right so that last character is fully redlined. About half the amount of white space between two printed redline characters.

|*** Backspace Redlining**

This screen lists control strings for printing redline text by backspacing and printing a character (e.g., ▦).

Redline Character
Value of WP character to use for redlining (e.g., 300h or 768 for ▦). Value = 256 * char_set_num + char_num.

|**** Shadow

This screen lists methods (driver types) for printing shadowed text when no shadow automatic font change is defined. Shadowed characters look three dimensional because of gray or black shadows cast in one or two directions. The default method for the current printer is marked (*).

Auto Shadow
Printer has a shadow feature that automatically prints shadow text without changing the current font.

Multiple Pass Shadowing
Shadowed characters should be printed a given number of times, each on one pass of the printhead. A horizontal and vertical offset may be given for each shadow pass.

Done by AFCs Only
Printer has no capability for printing shadow text, or all shadowing is governed by automatic font changes.

|***** Auto Shadowing

This screen lists control strings for printing shadow text with the printer's shadow feature on the same pass on which text is printed.

Auto Shadow On
Commands that turn on shadow feature.

Auto Shadow Off
Commands that turn off shadow feature.

|***** Multiple Pass Shadowing

This screen lists control strings for printing shadow text by overstriking shadow characters on multiple printhead passes, possibly with given horizontal and vertical offsets between each pair of passes.

Number of Passes
Number of images (original + overstrikes) for shadow text. The minimum and default value is 2.

Initial Horizontal Offset (1200ths)
Signed distance between first pair of images (passes 1 and 2). A negative value to move left, a positive value to move right.

Additional Horizontal Offset Each Pass (1200ths)
Signed distance between each pair of overstrike images (passes 2 and 3, 3 and 4, etc.). A negative value to move left, a positive value to move right.

Initial Vertical Offset (1200ths)
Signed distance between first pair of images (passes 1 and 2). A negative value to move up, a positive value

to move down.

Additional Vertical Offset Each Pass (1200ths)
Signed distance between each pair of overstrike images (passes 2 and 3, 3 and 4, etc.). A negative value to move up, a positive value to move down.

|** Strikeout**

This screen lists methods (driver types) for printing strikeout text when no strikeout automatic font change is defined. Strikeout is used to identify proposed deletions from a document. The default method for the current printer is marked (*).

Auto Strikeout on Same Pass
Printer has a strikeout feature that automatically prints strikeout text without changing the current font. The text and strikeout are printed on one printhead pass.

Auto Strikeout on Separate Pass
Printer has a strikeout feature that automatically prints strikeout text without changing the current font. Text is printed on one printhead pass and strikeout on another. Useful for printers that cannot print strikeout line through motion spaces (e.g., intra-word microspacing) on text-printing pass. Also useful for adjusting vertical position of strikeout line that would be printed on text-printing pass.

WP's Strikeout on Separate Pass
Printer has no strikeout feature. WP should use a character (e.g., '-') to strikeout text on a second printhead pass.

Backspace Strikeout
Printer has backspace capabilities. Backspace over each character and print a strikeout character (e.g., '-').

Done by AFCs Only
Printer has no strikeout capabilities, or all strikeout printing is governed by automatic font changes.

|*** Auto Strikeout on Same Pass**

This screen lists control strings for printing strikeout text with the printer's strikeout feature on the same pass on which text is printed.

Auto Strikeout On
Commands that turn on strikeout feature.

Auto Strikeout Off
Commands that turn off strikeout feature.

|*** Auto Strikeout on Separate Pass**

This screen lists control strings for printing strikeout text with the printer's strikeout feature on a printhead

pass after the text has been printed.

Auto Strikeout On
Commands that turn on strikeout feature.

Auto Strikeout Off
Commands that turn off strikeout feature.

Character to Use for Movement
For printers that cannot print strikeout line through motion spaces (e.g., microspacing), value of WP character to use for horizontal movement (e.g., 32 or ' '). Value = 256 * char_set_num + char_num.

Vertical Adjustment
For adjusting vertical position of strikeout line, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/7). A negative value to move up, a positive value to move down.

|*** WP's Strikeout on Separate Pass**

This screen lists control strings for printing strikeout text using a character (e.g., '-') on a second printhead pass after text has been printed.

Strikeout Character
Value of WP character to use for strikeout (e.g., 45 or '-'). Value = 256 * char_set_num + char_num.

Vertical Movement (1200ths)
For adjusting vertical position of strikeout line character, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/9). A negative value to move up, a positive value to move down.

Adjustment Between Characters (1200ths)
Signed distance strikeout character should be reprinted to the right so that consecutive characters touch to form a continuous line. The amount of white space between two printed strikeout characters.

Adjustment at Start of Strikeout (1200ths)
Signed distance strikeout character should be reprinted to the right so that first character is fully struck out. About half the amount of white space between two printed strikeout characters.

Adjustment at End of Strikeout (1200ths)
Signed distance strikeout character should be reprinted to the right so that last character is fully struck out. About half the amount of white space between two printed strikeout characters.

|*** Backspace Strikeout**

This screen lists control strings for printing strikeout text by backspacing and printing a character (e.g., '-').

Strikeout Character
Value of WP character to use for strikeout (e.g., 45 or '-'). Value = 256 * char_set_num + char_num.

|**** Subscript

This screen lists methods (driver types) for subscripting when no subscript automatic font change is defined.

Auto Subscript
Commands to control subscripting.

Subscript with Font Change/Movement
Subscripting will be done with an automatic font change or by moving the base font down. See miscellaneous questions screen.

|**** Superscript

This screen lists methods (driver types) for superscripting when no superscript automatic font change is defined.

Auto Superscript
Commands to control superscripting.

Superscript with Font Change/Movement
Superscripting will be done with an automatic font change or by moving the base font up. See miscellaneous questions screen.

|**** Underline

This screen lists methods (driver types) for underlining when no underline automatic font change is defined. The default method for the current printer is marked (*).

Auto Underline on Same Pass
Printer has an underline feature that automatically underlines without changing the current font. Text and underlining are printed on one printhead pass.

Auto Underline on Separate Pass
Printer has an underline feature that automatically underlines without changing the current font. Text is printed on one printhead pass and underlining on another. Useful for printers that cannot underline motion spaces (e.g., intra-word microspacing) on text-printing pass. Also useful for adjusting vertical position of underline that would be printed on text-printing pass.

WP's Underline on Separate Pass
Printer has no underline feature. WP should use a character (e.g., '_') to underline on a second printhead pass.

Backspace Underlining

Printer has backspace capabilities. Backspace over each character and print underscore.

Done by AFCs Only
Printer has no underlining capabilities, or all underlining is governed by automatic font changes.

|*** Auto Underline on Same Pass**

This screen lists printer control strings for underlining with the printer's underline feature on the same pass on which text is printed.

Auto Underline On
Commands that turn on underline feature.

Auto Underline Off
Commands that turn off underline feature.

|*** Auto Underline on Separate Pass**

This screen lists printer control strings for underlining with the printer's underline feature on a printhead pass after the text has been printed.

Auto Underline On
Commands that turn on underline feature.

Auto Underline Off
Commands that turn off underline feature.

Character to Use for Movement
For printers that cannot underline motion spaces (e.g., microspacing), value of WP character to use for horizontal movement (e.g., 32 or ' '). Value = 256 * char_set_num + char_num.

Vertical Movement (1200ths)
For adjusting vertical position of underline, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/3).

|*** WP's Underline on Separate Pass**

This screen lists printer control strings for underlining using a character (e.g., '_') on a second printhead pass after text has been printed.

Underline Character
Value of WP character to use for underlining (e.g., 95 or '_', 45 or '-'). Value = 256 * char_set_num + char_num.

Vertical Movement (1200ths)

For adjusting vertical position of underline character, signed distance the line should be moved down. Since fonts of different sizes may need different vertical movements, this value should be a function of the font size (e.g., PTSIZE/3).

Adjustment Between Characters (1200ths)
Signed distance underline character should be reprinted to the right so that consecutive characters touch to form a continuous line. The amount of white space between two printed underline characters.

Adjustment at Start of Underlining (1200ths)
Signed distance underline character should be reprinted to the right so that first character is fully underlined. About half the amount of white space between two printed underline characters.

Adjustment at End of Underlining (1200ths)
Signed distance underline character should be reprinted to the right so that last character is fully underlined. About half the amount of white space between two printed underline characters.

|*** Backspace Underlining**

This screen lists printer control strings for underlining by backspacing and printing a character (e.g., '_').

Underline Character
Value of WP character to use for underlining (e.g., 95 or '_'). Value = 256 * char_set_num + char_num.

Number of Images
Number of images (original + overstrikes) for underline character. Useful to make underlining distinguishable from underline character used within the text.

|* Helps and Hints about Printer**

This screen contains explanatory notes about the current printer. These notes are displayed in WP at printer selection time. They should explain any idiosyncracies the printer has. While editing in this screen, bold (F6) and underline (F8) attributes may be toggled.

|* Miscellaneous Numbers**

This screen lists restrictions the printer places on minimum margins and the number of fonts that can be printed on each page. A minimum margin is the area in which nothing can be printed at the edge of a page.

Minimum Top Margin
Distance (in inches) between top edge of paper and top-most printing position.

Minimum Bottom Margin
Distance (in inches) between bottom edge of paper and bottom-most printing position.

Minimum Left Margin
Distance (in inches) between left edge of paper and left-most printing position.

Minimum Right Margin
Distance (in inches) between right edge of paper and right-most printing position.

Maximum # of Fonts/Page
Maximum number of separate fonts (not font changes) the printer allows on each page (e.g., 16 for the HP LaserJet).

Look Ahead Region For Text While Printing Graphics
For printing graphics on a printer like the HP DeskJet which uses an algorithm to sort the graphics and text that it receives. Need to set question in miscellaneous questions.

|* Miscellaneous Questions**

Calculate Super/Subscript Positions
Allows WP to calculate vertical adjustment for super/subscript characters. (WP takes into consideration the baseline bias of each character in the super/subscript font.) This flag allows WP to print all super/subscript characters so that they overlap the base font lowercase x by 1/8 of the x-height.

This flag should not be marked if the printer has fixed vertical motion units (e.g., must move 1/6 inch on every line feed). If this flag is not marked, although the super/subscript characters may have baseline biases, no additional vertical adjustment is made to print them.

Don't Set Up AFC's for Super/Subscripts
Set if all super/subscripts are done by the auto super/subscripts drivers.

Print Super/Subscripts, Advances, Graphics in Top-to-Bottom Order
Printer cannot back up vertically. This flag directs WP to sort each page by vertical printhead position and print the page from top to bottom. With this flag set, each logical line containing super/subscripts requires multiple horizontal passes to print.

Scan Look Ahead Region for Text While Printing Graphics
For printers like the HP DeskJet. Need to set the look ahead region in miscellaneous numbers.

Don't Print Unmapped Characters Using Graphics (Print Space)
WordPerfect 5.1 will print characters that have no codes to send to the printer in the character maps using the printer's graphics commands if the printer supports graphics and this question is not set (and not printing in draft).

Character Cell Centers around Print Head (Using HMI)
Some printers (Toshiba) will spread the extra space evenly on both sides of the character, resulting in a jagged margin using the algorithm which assumes that all of the white space is placed after the character.

Portrait and Landscape Available in Each Font
Every font on this printer can be printed in portrait and landscape orientations.

Four Orientations Available in Each Font
Every font on this printer can be printed in portrait, landscape, reverse portrait, and reverse landscape orientations.

Rotate Graphics in Landscape
WordPerfect needs to rotate graphics data (the printer does not automatically do it)

Manual Feed and Continuous Forms Load in Same Place
So WordPerfect will prompt the user to load manual feed or continuous. Both manual feed and continuous cannot be loaded at the same time.

Long Edge Duplexing Supported / Short Edge Duplexing Supported
These questions will automatically be set by PTR when the duplexing questions in the forms are changed.

Printer Supports Auto Multiple Copies
This should be set if the initialize printer strings are set up for the printer to handle multiple copies based on the built-in WPDL variable COPIES.

Use Space Width from PS Table (else 3M)
Causes the optimal space width (usually 1/3 of the point size) to be the width of the space character in the PS table.

Use Device Halftoning if Supported
WordPerfect will send the raw gray scale data to the printer if the current graphic image placement is supported by the printer's halftoning capabilities.

|* Soft Font Format Type**

This screen lists general printer classifications identifying various downloadable font formats. The format describing the current printer is marked (*).

HP Series II compatible printers can print more characters and can print each character larger than the HP Plus printers.

Other
Printer does not support soft fonts, or the format is not described by any other type listed here.

|* Sheet Feeders

This screen displays names for all sheet feeders defined in the current file. A sheet feeder is a collection of bins (form feeding devices). The WP user may define which sheet feeders are used and on which printers.

| Sheet Feeder: (name)**

|* Bins**

This screen lists all bins defined for the current sheet feeder. A bin is a device that feeds forms into a printer.

Bin No.
Unique, consecutive numeric ID for each bin defined. Begins with 1.

Bin Description
Name of current sheet feeder bin.

|* Printer Strings**

This screen lists printer strings for sheet feeder bin selection and page control.

Select Bin
Commands that select forms from a given bin for printing. At print-time, WP stores the bin number in variable BIN.

Insert Form
Commands that insert a form from the selected bin.

Eject Form
Commands that eject a form from the printer.

|* Helps and Hints about Sheet Feeder**

This screen contains explanatory notes about the current sheet feeder. These notes should explain any idiosyncracies of the sheet feeder. While editing in this screen, bold (F6) and underline (F8) attributes may be toggled.

Printer Program Function Keys

Cancel Key (F1)
While editing WPDL expressions, cancels edit changes and restores former expression.

Setup Key (Shift-F1)
Allows you to toggle the name sorting feature. PTR automatically sorts all name lists alphabetically before displaying them on the screen, but since sorting names in a large .ALL file can cause noticeable pauses, you can disable name sorting with this key.

Forward Search Key (F2)
While viewing any list, searches forward through the list for the specified string. You are prompted for the search string. Press **Search** again to perform the search.

Backward Search Key (Shift-F2)
While viewing any list, searches backward through the list for the specified string. You are prompted for the search string. Press **Search** again to perform the search.

Consistency Check Key (Ctrl-F2)
Displays possible inconsistencies in the cursor definition that may cause **WordPerfect** not to print correctly.
The errors are written to **PRS.ERR**.
Switch Key (Shift-F3)
Toggles between Automatic Font Change list and Automatic Font Change Cross Reference list.

Switches from local variables to global variables.

Go to Next Marked (Alt-F3)
Moves the cursor to the next entry that is marked in the list (groups) or the next entry which has a defined
value (maps, ps tables, kern tables).

Character Maps Key (F4)
At the top level, switches to the screen of character mapping tables defined for all printers in the current file.

Copy Key (Ctrl-F4)
Copies cursor definition (or all marked definitions) to a specified file. You are prompted for the destination
filename. A full path is allowed. File definitions with identical names in the destination file are replaced
with the incoming definitions.

Definition Out or Expression In/Out Key (Ctrl-F5)
While editing a WPDL expression or a "Helps and Hints" screen, retrieves from or saves to a WordPerfect
document. Retrieved text is inserted into the expression or help screen at the cursor position. Saved text
includes the entire expression or help screen.

While viewing certain definition names (printers, sharable libraries, typefaces, character maps, proportional-
spacing tables, kerning tables, or sheet feeders), dumps a text version of the cursor definition (or all marked
definitions if at the top level) to a WordPerfect document. If the currently loaded file has only one printer
definition, performing a Definition Out of the printer definition dumps the entire file (including group, font,
map, ps table, typeface, and sheet feeder definitions). Performing a Definition Out on a sharable library will
give detailed font information, which the Definition Out of a Printer in a file with more than one definition
does not, without giving the lengthy map, ps table and typeface information that a Definition Out of a one
printer in the file produces. You may bring the file into WordPerfect to print the file or to extract
information. Definition Out with document compare is a good way to see changes that were made to the
ALL file.

Proportional-Spacing Tables Key (F6)
At the top level, switches to the screen of proportional-spacing tables defined for all printers in the current
file.

Kerning Tables Key (Alt-F6)
At the top level, switches to the screen of kerning tables defined for all printers in the current file.

Typefaces Key (Shift-F6)
At the top level, switches to the screen of typefaces defined for all printers in the current file.

Exit/Restart Key (F7)
On a sublevel of the program, returns you to the next higher level.

On the highest level of the program, returns you to the operating system or clears the currently loaded file. You are prompted for whether the current file should be saved and whether you want to exit the program. You must provide a name for any newly created file.

Quit Key (Alt-F7)
On any level of the program, acts like **Exit** on the highest level (i.e., returns you to the operating system or clears currently loaded file). You are prompted whether the current file should be saved and whether you want to exit the program.

Printer Definitions Key (F8)
At the top level, switches to the screen of printers defined in the current file.

Sharable Font Libraries Key (Ctrl-F8)
At the top level, switches to the screen of sharable font libraries defined in the current file.

Sheet Feeder Definitions Key (Shift-F8)
At the top level, switches to the screen of sheet feeders defined in the current file.

Global Variables Key (Alt-F8)
Toggles between global string, integer and function variables.

String Variables Key (F9)
If you are editing a printer definition or map string it takes you to local string variables otherwise (font, group, sheet feeder strings) it takes you to global string variables.

Integer Variables Key (Alt-F9)
If you are editing a printer definition or map string it takes you to local integer variables otherwise (font, group, sheet feeder strings) it takes you to global integer variables.

Function Variables Key (Shift-F9)
If you are editing a printer definition or map string it takes you to local function variables otherwise (font, group, sheet feeder strings) it takes you to global function variables.

Built-in Variables Key (Ctrl-F9)
Places you in the screen of built-in WPDL variables set by WP while printing. Operational when you are editing a printer definition or map string.

Save Key (F10)
Saves the current file to disk. You are prompted for the destination filename. A full path may be specified.

Retrieve Key (Shift-F10)
When a file is not already loaded, loads a specified file from disk. You are prompted for the filename. A full path may be entered.

Tab Key
When viewing a multi-field screen, rotates you through the fields. On some screens the arrow keys can similarly be used.

When performing a name search on a list of names, fills in the missing portion of the current word with characters from the cursor name, up to the next space. Used to save typing of long names for which the first several characters are identical in many listed names. For example, to locate HP LaserJet Series II when HP LaserJet 2000 is also listed, instead of typing the complete name you may type "HP L**[Tab]**S". The tab key fills in the remaining portion of the current word.

New Page Key (Ctrl-Enter)
When editing a WPDL expression, moves the expression into a window for editing. Useful for seeing a two-dimensional view of the current expression. Ctrl-Page Up takes you to editing in the full box.

Escape Key
When editing a WPDL expression, places the decimal representation of the escape key, [27], in the expression.

"Go To" Key (Ctrl-Home)
In all but character name lists, locates the nth entry in the list. You are prompted for the value of n.

In character name lists, locates the specified character. You may specify a letter (ASCII character set, given character), a number (given character set, first character), or two numbers separated with a comma (given character set and character).

* (Set Mark Key)
When viewing a name list on one of the program's highest levels (e.g., printers, font libraries, sheet feeders, typefaces, character maps, proportional-spacing tables, kerning tables), selects cursor definition for next **Copy** or **Text Out** operation.

When viewing a sublevel name list (e.g., fonts under a group definition, character maps under a font definition), selects cursor definition to be included in higher level being edited. Some sublevel lists allow many items to be selected (e.g., many fonts for a font group), and others allow only one item to be selected (e.g., one character map for a font).

Backspace (Clear Mark Key)
When viewing a name lists, clears any mark (*) on the cursor name.

When editing a WPDL expression, deletes character to left of cursor.

Insert Key
When viewing a name list, adds a definition to the list. You are prompted for a pattern (existing definition) and new definition name.

When editing a WPDL expression, toggles the insert/typeover editing mode.

Delete Key
When viewing a name list, deletes the cursor definition from the list. You are prompted to confirm the deletion.

When editing a WPDL expression, deletes character at the cursor.

Previous Font (F11, Ctrl-P)
On many screens below font definitions, shows the current screen for the previous font in the list.

Next Font (F12, Ctrl-N)
On many screens below font definitions, shows the current screen for the next font in the list.