northgate
INFORMATION SOLUTIONS

REALITY

Release 7.2

Printer Definition Maintenance

Document No. UM30002172A
August 1993

Northgate Information Solutions UK Limited
Peoplebuilding 2
Peoplebuilding Estate
Maylands Avenue
Hemel Hempstead
Herts
HP2 4NW

Tel: +44 (0)1442 232424
Fax: +44 (0)1442 256454

www.northgate-is.com

Comment Sheet

Please give page number and description for any errors found:

Page	Error

Please use the box below to: describe any material you think is missing; describe any material which is hard to understand; enter any suggestions for improvement; provide any specific examples of how you use your system which you think might be useful to readers of this manual

Continue on a separate sheet if necessary.

If you would like someone to contact you about documentation, please tick this box. □

Important: *Please enter your name, address and telephone number on the back of this form before returning.*

Manual: REALITY Printer Definition Maintenance,
UM30002172A, August 1993.

Name of Sender: ______________________________

Company Name / Address: ______________________

____

____

Telephone: ___________________________________

Date: ____

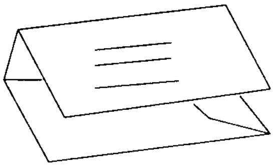
FOLD B

BUSINESS REPLY SERVICE
Licence No. HH698

2

FOLD A

Technical Publications Department
Northgate Information Solutions UK Limited
Boundary Way
Hemel Hempstead
Hertfordshire HP2 7HU



Fold along fold A, then B and C. Tuck in so name and address are on outside.

FOLD C

Chapter 1 About This Manual

Chapter 2 Printer Independent Functions

Chapter 3 Printer Dependent Control Sequences (PDCS)

Chapter 4 PDM Command

Figures

Tables

Chapter 1
About this Manual

Purpose

Overview

This manual decribes how to invoke printer independent functions via the DATA/BASIC PTR function for those who develop prologues and epilogues for form queues and despoolers or develop special printing programs in DATA/BASIC.

The manual also explains how to generate printer-dependent control sequences that perform the requested functions on a given printer.

Contents

The chapters have the following contents:

Chapter 1 lists related documentation about the operating system and presents conventions used in the rest of the manual.

Chapter 2 introduces printer independence and the printer independence functions that can be used with the DATA/BASIC PTR function.

Chapter 3 explains the Printer Dependent Control Sequences (PDCS), the strings of characters that are sent to the printers as function commands, and the symbolic forms that can be used to specify them.

Chapter 4 explains the PDM command and its menus for inputting PDCSs.

Appendix A is a table for converting decimal and hexadecimal numbers and ASCII codes.

Appendix B displays the interrelationship of the PDM program and the files that it updates.

Appendix C describes the Printer Independent Function Sequence (PIFS), which is created when a PTR function is compiled.

Index includes entries for the general topics covered.

Related Documentation

The following documents are either referenced or can be used in conjunction with this manual.

DATA/BASIC Reference Manual

Document Directory and Index

EDITOR Reference Manual

Screen Editor Manual

Glossary of Terms

Using the Spooler

Conventions

This manual uses the following conventions:

`Text` Bold text shown in this typeface is used to indicate input that must be typed at the terminal.

`Text` Text shown in this typeface is used to show text that is output to the screen.

Bold text Bold text in syntax descriptions represents characters typed exactly as shown. For example,

WHO

text Characters or words in italics indicate parameters that must be supplied by the user. For example,

LIST *file-name*

the parameter *file-name* is italicized to indicate that you must supply the name of the actual file defined on your system.

Italic text is also used for titles of documents referred to by this document.

{Braces } Braces enclose options and optional parameters. For example in

BLIST {**DICT**} *file-name item-id* {(*options*}

The word **DICT** can optionally be typed to specify the dictionary of the file.

The values *file-name* and *item-id* must be supplied.

One or more single-letter options can be included, as defined for the command; these must be preceded by an open parenthesis, can be given in any order, and are not separated by spaces. Any number of options can be used except where specified in text.

[Brackets] Parameters shown in syntax descriptions within square brackets indicate that at least one of these parameters must be included.

... In syntax descriptions, indicates that the parameters preceding can be repeated as many times as necessary.

SMALL CAPITALS Represents a named key on the keyboard. For example, BACKSPACE indicates the "Backspace" key

CTRL+KEY Two (or more) key names joined by a plus sign (+) indicate a combination of keys, where the first key(s) must be held down while the second (or last) is pressed. For example, CTRL+X indicates that the CTRL key must be held down while the X key is pressed.

Enter Means to type text and then press RETURN. For example, "Enter the WHO command" means type WHO and then press RETURN.

In general, the RETURN key (shown as ENTER or a symbol on some keyboards) must be used to complete all terminal input unless otherwise specified.

Press Means press a single key or combination of keys but do not press RETURN afterwards.

X'nn' Represents a hexadecimal number.

Chapter 2
Printer Independent Functions

Introduction to Printer Independence

Because each printer manufacturer chooses its own method for specifying printer control strings for performing special functions such as ejecting a sheet, changing fonts, and so on, it is difficult for an operating system producer to provide tables for controlling all printer types.

The REALITY System provides a method by which a despooler can take specially coded strings from the job print data and convert them to control data that is meaningful to a given printer.

Using the PTR function of DATA/BASIC, you can specify a printer function, such as setting line spacing to six lines per inch, that is independent of the printer used. The DATA/BASIC compiler encodes the parameters of the PTR function into a string of characters called a Printer Independent Function Sequence (PIFS). When your print data is despooled, the system translates the PIFS into a command sequence for a particular printer. The sequence is called a Printer Dependent Command Sequence (PDCS).

In the PTR function you specify a function class and a function subclass. The eject-page function, for example, is in class 1, subclass 10. A DATA/BASIC statement that specifies page eject would be:

```
PRINT PTR(1,10)
```

The system compiles this into a coded string of characters that is printer independent (PIFS). When the despooler encounters the PIFS, it converts the PIFS into the printer dependent commands (PDCS) required for its printer to perform the requested function.

If the printer function requires parameters, you also specify them in the PTR function. The Set n CPI function (class 7, subclass 5) sets the character pitch to a specified value. The statement to set the character pitch to 12 is:

```
PRINT PTR(7,5,12)
```

PIFS strings are intended mainly for prologues and epilogues, which you would create using option 3 of the SPM command (explained in the manual *Using the Spooler*). However, you can embed a PIFS in print data created by a DATA/BASIC program.

See the *DATA/BASIC Reference Manual* for the classes and subclasses of printer independent functions and their parameters that you can use with the PTR function. Chapters 3 and 4 of this manual explain how to specify PDCS's for printers that are not defined on the system.

Printer Independent Function Classes

The printer independent functions for use in the PTR function are divided into thirteen classes. Each class has several subclasses. The *DATA/BASIC Reference Manual* presents a table for each class with its subclasses.

The classes and their functions are as follows:

Class	Function
1	Device Control
2	Page Format
3	Line Format
4	Tabulations
5	Horizontal Control
6	Line Spacing
7	Character Pitch
8	Form Layout
9	Special Effects
10	Typeface
11	Character Set
12	Font Selection
13	Color

Chapter 3
Printer Dependent Control Sequences (PDCS)

Introduction to Printer Dependent Control Sequences (PDCS)

Overview

A Printer Dependent Control Sequence is a string of values sent to a printer to command it to perform some function. The sequence is printer dependent because each printer is designed to respond to a given set of values to perform each of its functions. You must check your printer's reference manual for the values that must be sent to your printer to perform each action.

Symbolic Input

The PDM command, which is explained in Chapter 4, accepts a symbolic form of characters and values to create each PDCS. The elements of this symbolic form are explained in this chapter.

An example of a PDCS is the following sequence:

ESC "(s" 8 "H"

The ESC is the ASCII character representaed by the value X'1B'. The number 8 is the ASCII character represented by the value X'38'.

This PDCS is for a laser printer, setting the character pitch to eight characters per inch. It would be invoked by the printer independent function:

PTR(7,1)

Printer Control Block (PtrCB)

The Printer Control Block (PtrCB) is a data structure that you can reference in PDCS strings. The PtrCB has a work byte called the hold byte and byte-size fields for storing various printer values, such as margins, line spacing, font set, and the like. This data structure is required by some printers because the values for certain functions must be combined (ORed) together with another value. An example would be ORing together the values for a subfont and font.

Parameter Maniuplator Function (PRM)

A special function is provided in the symbolic input to the PDM command to allow you to move data into and out of the PtrCB and to combine the data. This is the Parameter Manipulator Function, PRM. It has several symbolic operators and operands to reference the fields of the PtrCB.

Printer Dependent Control Sequence Input Elements

The elements that you can use as input elements for a PDCS are listed in Table 3-1. These elements are the input to the class n subclass menus described in Chapter 4.

Table 3-1. PDM Menu Input Elements.

Element	Explanation
number	Specifies a decimal number.
'*string*' or "*string*"	Specifies a literal string. A literal string is any text delimited by single or double quotation marks.
ASCII name	Specifies any ASCII control character (characters 0 - 31 and 127) by its ASCII name. See Appendix A for the names of ASCII control characters. For example: CR NUL ACK
CHAR(*n*)	Specifies a single character function with a binary value equal to the numeric *n*. For example, CHAR(32) is a blank, and CHAR(65) is the uppercase A.
CTRL(*char*)	Specifies a single-character function with a binary value equal to CTRL+*char*. Values for *char* are limited to the following: @ A - Z [\\ ] ^ _ (underscore)
HEX(*x*)	Generates a string of ASCII characters equivalent to the hexadecimal string defined by *x*. The hexadecimal string must be an even number of digits. For example: HEX(1B20) generates an ESC character followed by a blank space. HEX(30314142) generates the string 01AB.

(Continued)

Table 3-1 . PDM Menu Input Elements (Continued).

Element	Explanation

STR(*e,n*)	Generates a repetition of the element *e* *n* times
	The element *e* can be any combination of the following:
	Literal
	ASCII name
	CHAR(n)
	CTRL(char)
	HEX(x)
	Blanks are not allowed within the function unless they are part of a literal string or as part of a multi-element string in which each element must be preceded by a blank. For example:
	`STR( ESC 'G' ,10)`
	`STR( ESC CTRL(A) , 3)`
	If *e* is only a single element, it does not require leading and trailing blanks. For example:
	`STR(CHAR(6),8)`
	and
	`STR( CHAR(6) ,8)`
	are equivalent expressions.
	Examples:
	`STR('a',5)` generates the string aaaaa.
	`STR(CHAR(64),4)` generates the string @@@@.
	`STR( ESC CTRL(A) , 3)` generates ESC CTRL(A) ESC CTRL(A) ESC CTRL(A).

PRM(*arg*)	Specifies the PRM function described in this chapter.

Examples of PDCSs

The following is the ASCII CAN character:

```
CAN
```

The next string is the ASCII ESC and EM characters followed by the letter R:

```
ESC EM "R"
```

The following is the ASCII ESC character, the ampersand, the letter s, the number 1, and the letter C:

ESC "&s" 1 "C"

The above string could have been expressed as the following:

HEX(1B26733143)

The Printer Control Block (PtrCB)

The Printer Control Block (PtrCB) is a data structure that you can reference in PDCS strings by means of the Parameter Manipulator (PRM) function (explained in the next section). The PtrCB has a work byte called the hold byte and byte-size fields for storing various printer values, such as margins, line spacing, font set, and the like.

Hold Byte

The hold byte is a workspace that can be referenced by several operators of the PRM function. The operators can load the hold byte, perform arithmetic and logical operations on it, and store its contents in a storage byte.

PtrCB Fields Initialization

When a despooler is first started, it moves the values from the class 14 subclass fields to the PtrCB fields.

PtrCB Fields

The PtrCB fields are bytes that are intended for saving various printer values. The mnemonic names, descriptions, and Class 14 subclass initialization fields are presented in Table 3-2.

Table 3-2. Printer Control Block (PtrCB) Fields.

Mnemonic	Description	Class 14, Subclass
PI.TM	Top margin	1
PI.BM	Bottom margin	2
PI.TL	Text length	3
PI.LM	Left margin	4
PI.RM	Right margin	5
PI.LS	Line spacing	6
PI.CP	Character pitch	7
PI.FL	Form length	8
PI.TF	Typeface	9
PI.CS	Character set	10
PI.SP	Separator	11
PI.FS	Font set	12
PI.SF	Subfont	13
PI.LQ	Letter quality	14
PI.HMI	Horizontal Motion Index	15
PI.VMI	Vertical Motion Index	16
PI.EVFU	VFU channel offset	17

The Parameter Manipulator Function (PRM)

The PRM function is a PDCS element that enables you to move parameters from a PIFS string and move them to the printer or the PtrCB. The PRM arguments are the PRM operators and their operands. The PRM operators are described in full in Table 3-3. Operators perform the following tasks:

- Fetch PIFS parameters with or without additional processing and send data to the printer.
- Load data into the hold byte.
- Store data in a field of the PtrCB.
- Perform arithmetic, logical, and shifting operations on the hold byte.
- Write data to printer.
- Disable printer independence processing.

Fetching and Processing PIFS Parameters

The following operators fetch PIFS parameters, process them, and send the result to the printer:

ALLASC sends a series of parameters without processing to the printer.

ONEASC sends one parameter without processing to the printer.

ALLBIN converts a series of parameters to binary and sends each binary result to the printer.

ONEBIN converts one parameter to binary and sends the result to the printer.

ASCSEP sends a series of parameters to the printer; follows each parameter with the contents of PI.SP.

BINSEP converts a series of parameters to binary; sends the binary result of each parameter to the printer followed by the contents of PI.SP.

BIN2B processes one parameter as a signed decimal number; divides by 256; sends quotient byte and remainder byte to printer.

BSOST converts a series of parameters to binary; adds the contents of a specified PtrCB field to each result; sends resulting bytes to printer.

Loading the Hold Byte

The following operators load data into the hold byte:

LP	loads the next PIFS parameter into the hold byte.
LS	loads the contents of the specified PtrCB field into the hold byte.
LN	loads a literal (the next PRM operand) into the hold byte.

Storing Data in a PtrCB Field

These operators store data into a specified PtrCB field:

STH	stores the contents of the hold byte into the field.
STP	stores the next PIFS parameter into the field.

Hold Byte Arithmetic, Logical, and Shifting

The following operators perform the indicated operations:

DH	decrements hold byte by one.
IH	increments hold byte by one.
AH	ANDs the hold byte with a PtrCB field.
OH	ORs the hold byte with a PtrCB field.
XH	XORs the hold byte with a PtrCB field.
SH	shifts the hold byte left one bit, which multiplies the contents by 2.

Writing Data to Printer

The following operators write bytes from the work space:

WH	writes the hold byte to the printer.
WP	writes the specified PtrCB field to the printer.
DATE	writes the system date to the printer in the format: DD MMM YYYY.
TIME	writes the system time to the printer in the format: HH:MM:SS.

Disabling Printer Independence Processing

The following operator does as indicated:

PIOFF	disables printer independence processing for this job.

Table 3-3. PRM Operators.

Operator	Operand	Function
AH	PtrCB field	AND the contents of the hold byte with the contents of the PtrCB field defined by next PRM argument.
ALLASC		Fetch each character of the next series of PIFS parameters and send it as is to the printer. Fetching and sending continue until the sequence delimiter is encountered.
ALLBIN		Fetch the next series of PIFS parameters, convert each parameter to a byte of binary data and send the byte to the printer. Fetching, converting, and sending continue until the sequence delimiter is encountered.
ASCSEP		Fetch each character of the next series of PIFS parameters and send it as is to the printer. After the last character of each parameter, send the contents of PI.SP (separator). Fetching and sending continue until the sequence delimiter is encountered.
BIN2B		Fetch the next PIFS parameter and process it as a signed decimal number in the following steps: 1. Save the sign, if any. 2. Convert the number to binary. 3. If there was no sign or if it was a plus (+), use the binary number as is; if the sign was negative (-), use the result of subtracting the binary number from 65,536. 4. Divide the result of Step 3 by 256 to form two bytes: a quotient byte and a remainder byte. 5. Send the quotient byte to the printer. 6. Send the remainder byte to the printer.
BINSEP		Fetch the next series of PIFS parameters; convert each parameter to a byte of binary data and send the byte to the printer; after each byte of binary data, send the contents of PI.SP (separator) to the printer. Fetching and sending continue until the sequence delimiter is encountered.
BSOST	PtrCB field	Fetch the next series of PIFS parameters; convert each parameter to a byte of binary data; to the byte of binary data add the contents of the PtrCB field defined by the next PRM argument; send the resulting byte to the printer. Fetching, converting, adding, and sending continue until the sequence delimiter is encountered.
DATE		Write the system date to the printer in the format DD MMM YYYY.
DH		Decrement the hold byte by one.
IH		Increment the hold byte by one.
LN	Literal	Load the next PRM argument into the hold byte.
LP		Load the next PIFS parameter into the hold byte.
LS	PtrCB field	Load the contents of the PtrCB field defined by next PRM argument into the hold byte.
OH	PtrCB field	OR the contents of the hold byte with the contents of the PtrCB field defined by the next PRM argument.
ONEASC		Fetch each character of the next PIFS parameter and send it as is to the printer.
ONEBIN		Fetch the next PIFS parameter, convert it to a byte of binary data, and send the byte to the printer.

(Continued)

Table 3-3. PRM Operators (Continued).

Operator	Operand	Function
PIOFF		Disable printer independence processing for this printer for the remainder of the current job.
SH		Shift the contents of the hold byte left one bit (multiply by 2).
STH	PtrCB field	Store the contents of the hold area in the PtrCB field defined by next PRM argument.
STP	PtrCB field	Store the next PIFS parameter in the PtrCB field defined by next PRM argument.
TIME		Write the system time to the printer in the format HH:MM:SS.
WH		Write the contents of the hold area to the printer.
WP	PtrCB field	Write the contents of the PtrCB field defined by next PRM argument to the printer.
XH	PtrCB field	XOR the contents of the hold byte with the contents of the PtrCB field defined by next PRM argument.

Examples of PDCS's with PRM Functions

The difference in PDCSs for different printers is shown by the function PTR(2,3,55) to set the right margin at position 55. For one printer the PDCS is:

```
ESC "Q" PRM(ALLBIN)
```

This PDCS sends the characters ESC and Q to the printer, followed by the parameter (55) converted to binary. The ESC and the Q signal the printer that a binary number is coming next.

For the same function, PTR(2,3,55), the PDCS for another printer is:

```
PRM(STP,PI.RM) ESC "X" PRM(WP,PI.LM,WP,PI.RM)
```

The STP operator stores the 55 in PtrCB field PI.RM. The characters ESC and X are sent to the printer. The WP operator sends the contents of PI.LM and then PI.RM to the printer. This printer requires that both left and right margin numbers follow the ESC and X characters. Hence, the values must be stored in the PtrCB. This printer's PDCS for Set Left Margin, PTR(2,4,*p*), stores the parameter in PI.LM:

```
PRM(STP,PI.LM) ESC "X" PRM(WP,PI.LM,WP,PI.RM)
```

Chapter 4
PDM Command

PDM command

The PDM command allows end users to configure the interface between their own foreign printers and the REALITY Operating System. PDM is a cataloged DATA/BASIC program that you can execute in the SYSPROG or SYSMAN accounts. The PDM program is invoked by the PDM command.

Using PDM, you can define as many printers as your system requires. Your input to PDM is a symbolic form of a Printer Dependent Control Sequence (PDCS). The symbolic elements are presented in Chapter 3.

After you have changed any PDCS strings, you tell PDM to convert the PDCS strings to an object form that is more efficient for a despooler to use.

When you invoke PDM, it displays the PDM main menu, which lists existing printer definitions in alphabetical order. The menu prompts you to select an existing definition for editing or to create a new definition. See Figure 4-1 for a display of the program's main control flow.

If you wish to modify an existing printer definition, PDM displays the class menu and prompts you to enter the class number of any class item you wish to change.

If you wish to create a new printer definition, PDM requires you to start with a copy of an existing definition. PDM then displays the class menu and prompts you to enter the class number of any class item you wish to change.

The class n subclass menu displays the subclasses of class n (a number) and prompts you to change them.

Any changes you enter in reply to the prompts of a class n subclass menu are the PDCS strings. They are stored in source definition items.

When you enter FI at the main menu, the program inputs the symbols and literals of the changed source definition items, converts them to binary data, and stores them in object definition items.

If you enter EX at the main menu, the program exits and does not change the object definition items.

After you have created a printer definition, you must enter its name in the despooler definition for the printer. You do this with the SPM command, option 2, DESPOOLER.CONTROL file maintenance. Enter the printer definition name at option 3, Printer Ind. Id.

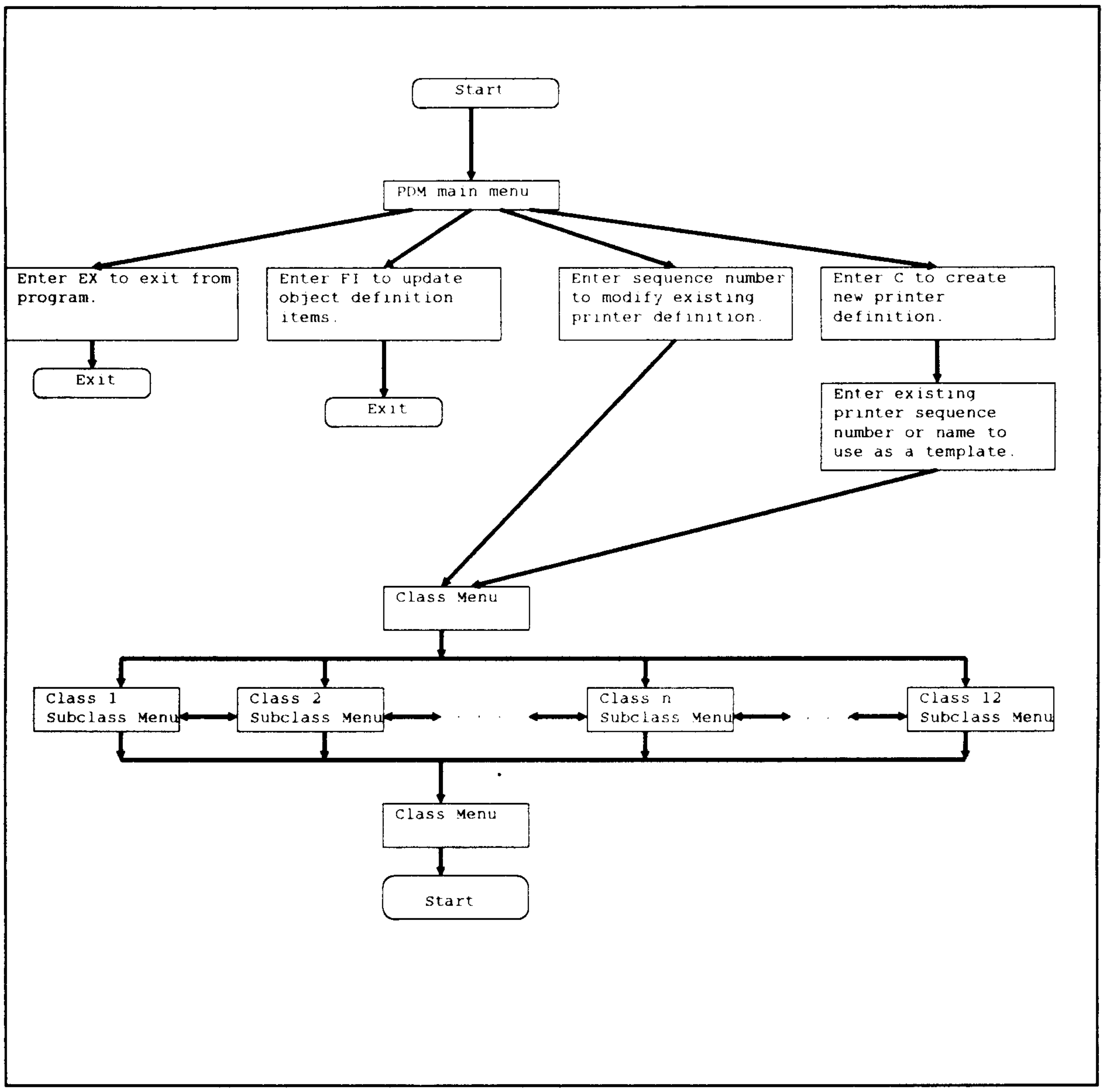

Start
PDM main menu
Enter EX to exit from program.
Enter FI to update object definition items.
Enter sequence number to modify existing printer definition.
Enter C to create new printer definition.
Exit
Exit
Enter existing printer sequence number or name to use as a template.
Class Menu
Class 1 Subclass Menu
Class 2 Subclass Menu
Class n Subclass Menu
Class 12 Subclass Menu
Class Menu
Start

Figure 4-1. PDM Program Main Control Flow.

Invoking the PDM Command

Purpose
You can use the PDM command to add, modify, or delete printer definition items for printers that are foreign to the operating system.

Command Class
Cataloged DATA/BASIC program.

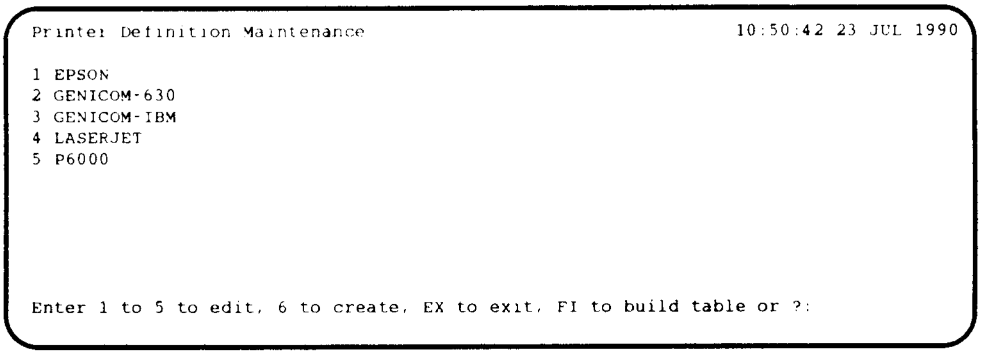
Syntax
PDM

Program Action
The command clears the screen and displays the PDM main menu, which is described in the following section.

Restriction
The PDM command is restricted to the SYSPROG and SYSMAN accounts.

The PDM Main Menu

The PDM command clears the screen and displays the PDM main menu, which includes a list of existing printer definition names in alphanumeric order. Each name is preceded by a sequence number. The main menu looks similar to the following:



```
Printer Definition Maintenance                                  10:50:42 23 JUL 1990

1 EPSON
2 GENICOM-630
3 GENICOM-IBM
4 LASERJET
5 P6000






Enter 1 to 5 to edit, 6 to create, EX to exit, FI to build table or ?:
```

Main Menu Prompt

The main menu prompt is the line:

```
Enter 1 to 5, number to create, EX to exit, FI to build table or ?:
```

Building Printer Definition Table

To convert source definitions to object definitions, enter FI at the main menu prompt.

If you have entered any changes, the command rebuilds the changed object definition items and exits to TCL.

If you have not entered changes, the command displays the prompt:

```
No Changes? Type REBUILD to rebuild all or <cr> to exit:
```

To rebuild all the object definitions enter the word REBUILD. To exit without rebuilding, press **RETURN** only.

Exiting without Building a Table

To exit from the command without building a new table, enter EX at the main menu prompt.

Displaying a Help Message

To display a help message that explains the PDM command, enter a question mark (?) at the main menu prompt.

Creating a New Printer Definition

To create a new printer definition, enter the next number in the list at the main menu prompt. The command verifies that you wish to create a new printer definition with the prompt:

```
Create a new Printer Definition (Y/n):
```

If you do not wish to create a new definition, enter N.

If you wish to enter a new definition, enter Y. The command asks for the name of the new printer definition with the prompt

```
What is the printer name for this new printer definition:
```

Respond with the name of the printer definition.

Rules for a Printer Definition Name

Any name is valid as long as it

- Is unique,
- Contains no blanks,
- Contains 22 or fewer characters,

For the printer definition to be effective, the name must be included in the despooler definition. See the description of the SPM command in *Using the Spooler.*

Selecting a Printer Definition to Use As a Template

After you enter the name of the new printer definition, the command prompts you to enter the name or number of an existing printer definition to use as a template for the new printer definition. The prompt is

```
Input a current printer name or number to use as a template:
```

Enter the name or sequence number of an existing printer definition. The command clears the screen and displays the class menu, which lists all the classes of printer commands.

Modifying or Deleting a Printer Definition

To modify or delete an existing printer definition, enter its sequence number in response to the main menu prompt. The command displays the prompt

```
Modify or Delete printer printer-name (M/d):
```

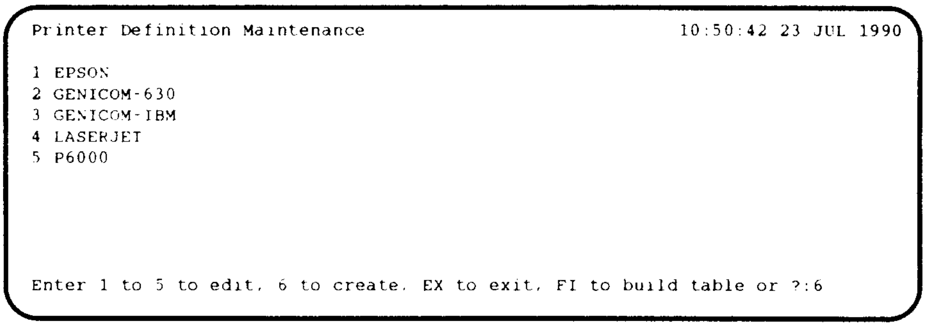
where *printer-name* is the name of the printer.

To delete the definition, enter a D.

To modify the printer definition, enter an M. The command clears the screen and displays the class menu, which lists all the classes of printer commands.

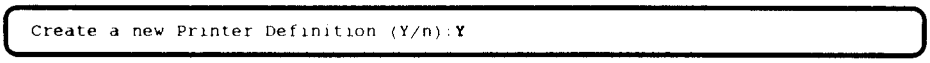
Example of Creating a New Printer Definition

To create a new printer definition enter the next sequence number in response to the PDM main menu prompt as follows:



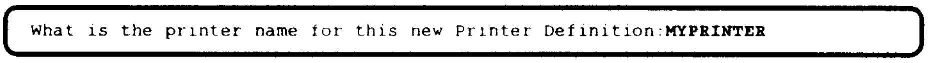
```
Printer Definition Maintenance                                  10:50:42 23 JUL 1990

1 EPSON
2 GENICOM-630
3 GENICOM-IBM
4 LASERJET
5 P6000



Enter 1 to 5 to edit, 6 to create, EX to exit, FI to build table or ?:6
```

The command asks you to verify that you wish to create a new printer definition. Enter Y



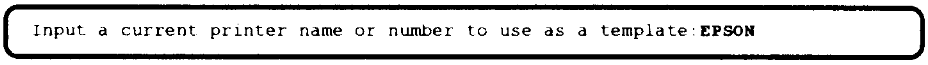
```
Create a new Printer Definition (Y/n):Y
```

PDM then asks you for the name of the new printer definition:



```
What is the printer name for this new Printer Definition:MYPRINTER
```

PDM prompts you to select an existing printer definition to be used as a template for creating your new printer definition:



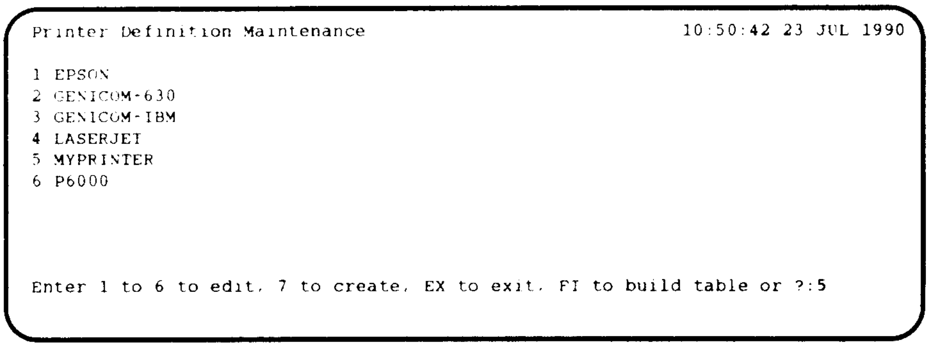
```
Input a current printer name or number to use as a template:EPSON
```

The command clears the screen and displays the class menu. You could have entered 1 instead of the name in reply to this last prompt.

If you press **RETURN** only, PDM stops the creation process and returns to the main menu prompt.

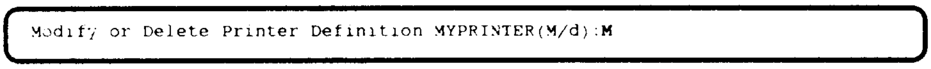
Example of Modifying an Existing Printer Definition

To modify an existing printer definition, enter its sequence number in response to the PDM main menu. The following shows printer definition 5 being entered at the main menu:



```
Printer Definition Maintenance                                    10:50:42 23 JUL 1990

1 EPSON
2 GENICOM-630
3 GENICOM-IBM
4 LASERJET
5 MYPRINTER
6 P6000



Enter 1 to 6 to edit, 7 to create, EX to exit, FI to build table or ?:5
```

PDM then asks if you wish to modify or delete the printer definition. Enter M to modify the item, as follows:

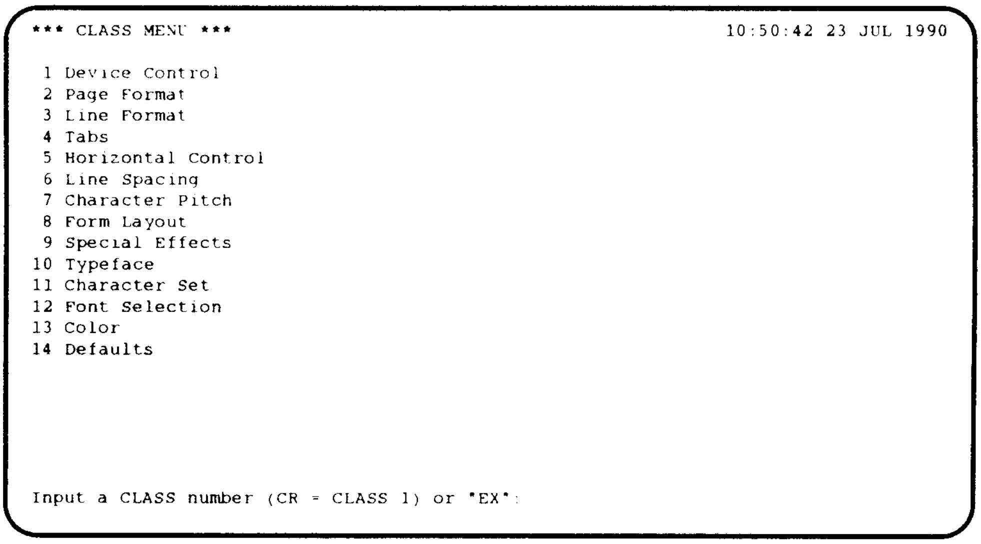


```
Modify or Delete Printer Definition MYPRINTER(M/d):M
```

The command clears the screen and displays the class menu.

The Class Menu

PDM clears the screen and displays the class menu whenever a printer definition is to be modified. The class menu looks similar to the following:



```
*** CLASS MENU ***                                      10:50:42 23 JUL 1990

 1 Device Control
 2 Page Format
 3 Line Format
 4 Tabs
 5 Horizontal Control
 6 Line Spacing
 7 Character Pitch
 8 Form Layout
 9 Special Effects
10 Typeface
11 Character Set
12 Font Selection
13 Color
14 Defaults




Input a CLASS number (CR = CLASS 1) or "EX":
```

Class Menu Prompt

The class menu prompt is the following line:

```
Input a CLASS number (CR = CLASS 1) or "EX":
```

Selecting a Class to Modify or Display

To display the subclasses of a class, enter the class number. If you press **RETURN** only, the program displays class 1.

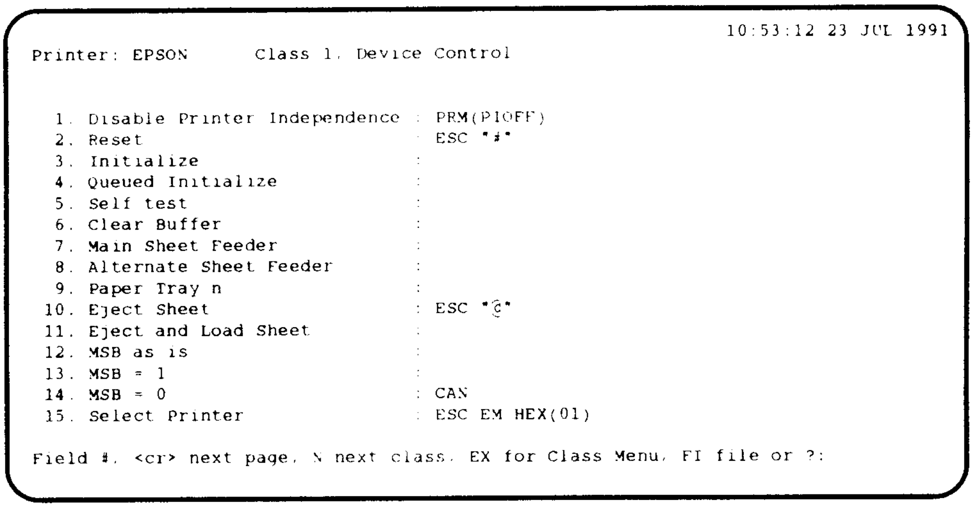
Exiting to Main Menu

To exit from the class menu and return to the main menu, enter EX.

The Class n Subclass Menus

PDM displays a class n subclass menu when you respond to the class menu prompt by entering a number. The command displays the menu with the existing subclass characteristics.

A class n subclass menu looks similar to the following:



```
                                                           10:53:12 23 JUL 1991
Printer: EPSON       Class 1, Device Control


  1. Disable Printer Independence : PRM(PIOFF)
  2. Reset                        : ESC "#"
  3. Initialize                   :
  4. Queued Initialize            :
  5. Self test                    :
  6. Clear Buffer                 :
  7. Main Sheet Feeder            :
  8. Alternate Sheet Feeder       :
  9. Paper Tray n                 :
 10. Eject Sheet                  : ESC "@"
 11. Eject and Load Sheet         :
 12. MSB as is                    :
 13. MSB = 1                      :
 14. MSB = 0                      : CAN
 15. Select Printer               : ESC EM HEX(01)

Field #, <cr> next page, N next class, EX for Class Menu, FI file or ?:
```

Subclass Menu Prompt

The subclass menu prompt has one of the following formats:

```
Field #, N next class, EX for Class Menu, FI file or ?:

Field #, <cr> next page, N next class, EX for Class Menu, FI file or ?:

Field #, <cr> first page, N next class, EX for Class Menu, FI file or ?:
```

Selecting a Field to Modify

When you enter a field number at the subclass menu prompt, the command displays the field's current contents. The field number that you wish to enter does not have to be on the screen.

Entering Data

Enter the new data. For the type of data to enter see the instructions for each field as given in the sections that follow.

Clearing a Field

To clear a field, enter a space.

Displaying a Help Message

To display a help message that explains the requirements of the field, enter a question mark (?).

Exiting Without Changing

To exit from the field without changing the data, press RETURN only.

Updating the Printer Definition Item

To update the printer definition item in file PRINTER, enter FI at the subclass menu prompt. The command updates the item and exits to the class menu.

Exiting without Updating the Printer Definition Item

To exit from a class n subclass menu without updating the printer definition item, enter EX at the subclass menu prompt.

If you have made any changes to a printer definition item, the command prompts:

```
Changes have been made. Exit anyway(Y/N):
```

If you wish to abandon the changes, enter Y.

If you do not wish to exit, enter N.

Displaying a Help Message

To display a help message that explains the class n subclass menu, enter a question mark (?) at the subclass menu prompt.

Class n Subclass Menu Descriptions

The following sections describe the menus for the classes of functions. The description for each subclass includes the menu display, a brief description of its meaning and an example of a PDCS.

Many of the examples are taken from actual PDCS's, but you must check your printer's reference manual for the actual characters or sequences that should be entered.

Class 1 Subclass Menu: Device Control

The class 1 subclass menu has fields that perform device contol operations, such as, reseting the printer, selecting feeder trays, setting bit configurations, and the like.

1 Disable Printer Independence

Disable printer independence. When this element is in effect, the despooler does not interpret the PIFS strings; it simply outputs them to the printer as they are.

Example: `PRM(PIOFF)`

2 Reset

Restore the printer to an initial setting. This usually puts the default settings into force.

Example: `ESC "E"`

3 Initialize

Bring the printer to an initial state. Not all printers clear buffers with this command.

Example: `ESC SUB "I"`

4 Queued Initialize

Reset the printer when the current print job is complete.

Example: `ESC "@"`

5 Self Test

Execute the self test routine in the printer.

Example: `ESC HEX(02) 5`

6 Clear Buffer

Either print or delete all data in the output buffer.

Example: `CAN`

7 Main Sheet Feeder

Select the primary, or default, paper tray.

Example: `ESC "&l" 1 "H"`

8 Alternate Sheet Feeder

Select the alternate, or secondary, paper tray.

Example: `ESC EM HEX(02)`

9 Paper Tray n

Select the paper tray designated by a parameter in the PIFS.

Example: `ESC EM PRM(ALLBIN)`

10 Eject Sheet

Eject a sheet of paper.

Example: `ESC EM "R"`

11 Eject & Load Sheet

Eject the current sheet and reload from the active tray.

Example: `ESC "&10H1H"`

12 MSB as is

Accept the most significant bit (high-order or eighth bit) of each data byte without change.

Example: `ESC "#"`

13 MSB = 1

Force the most significant bit (high-order or eighth bit) of each data byte to one.

Example: `ESC ">"`

14 MSB = 0

Force the most significant bit (high-order or eighth bit) of each data byte to zero.

Example: `ESC "="`

15 Select Printer

Enable on-line control at the printer.

Example: `CHAR(17)`

16 Deselect Printer

Disable on-line control at the printer.

Example: `CHAR(19)`

17 Print Control Sequences On

Enable the printer to print the control sequences it receives. This is a diagnostic function.

Example: `ESC "Y"`

18 Print Control Sequences Off

Disable the printer from printing the control sequences it receives. This is a diagnostic function.

Example: `ESC "Z"`

19 Time

Send the current system time in external format to the printer.

Example: `PRM(TIME)`

20 Date

Send the current system date in external format to the printer.

Example: `PRM(DATE)`

Class 2 Subclass Menu: Page Format

The class 2 subclass menu has fields that determine page format.

1 Set Top Margin

Set the top margin to a specified position.

Example: `ESC "&l" PRM(ALLASC) "E"`

2 Set Bottom Margin

Set the bottom margin to a specified position.

Example: `ESC "&l" PRM(ALLBIN) "B"`

3 Set Right Margin

Set the right margin to a specified position.

Example: `PRM(STP,PI.RM) ESC "X" PRM(WP,PI.LM,WP,PI.RM)`

4 Set Left Margin

Set the left margin to a specified position.

Example: `ESC "&a" PRM(ALLASC) "L"`

5 Set Text Length

Define the number of lines from the top margin to the bottom.

Example: `ESC "&l" PRM(ALLASC) "F"`

6 Clear Vertical Margins

Change the vertical margins to their default settings.

Example: `ESC "C"`

7 Clear Horizontal Margins

Change the horizontal margins to their default settings.

Example: `ESC 9`

8 Clear All Margins

Change both the horizontal and vertical margins to their default settings.

Example: `ESC "C" ESC 9`

9 Set Top of Form

Define the current printer position as the top of form.

Example: `ESC 4`

10 Set Skip Over Perforation

Set the number of line feeds to allow for perforations.

Example: `ESC "N" PRM(ALLBIN)`

11 Clear Skip Over Perforation

Reset the skip-over-perforation parameter to zero.

Example: `ESC "O"`

Class 3 Subclass Menu: Line Format

The class 3 subclass menu has fields that determine line format.

`1 Auto-Justify Off`

Disable the automatic left- and right-justification of print lines.

Example: `ESC "X"`

`2 Left Justify On`

Enable the left-justification of print lines.

Example: `ESC "a" HEX(00)`

`3 Right Justify On`

Enable the right-justification of print lines.

Example: `ESC "a" HEX(02)`

`4 Auto-Center On`

Enable the automatic centering of print lines.

Example: `ESC "a" HEX(01)`

`5 Full Auto-Justify On`

Enable the automatic left- and right-justification of print lines.

Example: `ESC "M"`

`6 Auto-Line Feed On`

Enable automatic advancement of paper after a carriage return.

Example: `ESC 5 1`

`7 Auto-Line Feed Off`

Disable automatic advancement of paper after a carriage return.

Example: `ESC "."`

`8 Auto CR On`

Enable automatic return of print head after a line feed.

Example: `ESC "&s" 1 "C"`

`9 Auto CR Off`

Disable automatic return of print head after a line feed.

Example: `ESC "!"`

Class 4 Subclass Menu: Tabs

The class 4 subclass menu has fields that set and clear tabulation stops.

`1 Set Horizontal Tab Position(s)`

Set horizontal tabulation stops at the specified positions.

Example: `ESC "D" PRM(ALLBIN) NUL`

`2 Clear All Horizontal Tabs`

Clear all horizontal tabulation stops. On some printers this may reassert default tabulation stops.

Example: `ESC "D" NUL`

`3 Set Vertical Tab Position(s)`

Set vertical tabulation stops at the specified positions.

Example: `ESC "B" PRM(ALLBIN) NUL`

`4 Set Vertical Tab Position(s), Channel`

Set vertical tabulation stops at the specified positions on a specified channel. On some printers this may reassert default tabulation stops.

Example: `ESC "b" PRM(ONEBIN) PRM(ALLBIN) NUL`

`5 Select Vertical Channel (VFU)`

Select a specified channel for subsequent use.

Example: `ESC "\" PRM(ONEBIN)`

`6 Clear All Vertical Tabs`

Clear all vertical tabulation stops.

Example: `ESC "B" NUL`

`7 Clear Vertical Channel's Tabs (VFU)`

Clear all tabulation stops on the specified VFU channel.

Example: `ESC "b" PRM(ONEBIN) NUL`

`8 EVFU Load`

Send an EVFU loading sequence to the printer. Each parameter is a channel number, and the numbers must be in line number sequence.

Example: `HEX(1E) PRM(BSOST,PI.EVFU) HEX(1F)`

Class 5 Subclass Menu: Horizontal Control

The class 5 subclass menu has fields that provide control of the horizontal movement of the print head.

1 **Bidirectional Printing On**

Enable printing in both directions.

Example: `ESC "U" HEX(00)`

2 **Bidirectional Printing Off**

Disable printing in both directions, allowing unidirectional printing only.

Example: `ESC "U" HEX(01)`

3 **Move Horizontal, Relative, in Columns**

Change the position of the cursor/head on the current line to the specified column displacement relative to the current position.

Example: `ESC "&a" PRM(ALLASC) "C"`

4 **Move Horizontal, to Absolute Column**

Change the position of the cursor/head on the current line to the specified column displacement from the left margin.

Example: `ESC "&a" PRM(ALLASC) "R"`

5 **Move Horizontal, Relative, in Dots**

Change the position of the cursor/head on the current line to the specified dot displacement relative to the current position.

Example: `ESC "*p" PRM(ALLASC) "X"`

6 **Move Horizontal, Absolute, in Dots**

Change the position of the cursor/head on the current line to the specified dot displacement from the left margin.

Example: `ESC "$" PRM(BIN2B)`

7 **Move Horizontal, Relative, in Decipoints**

Change the position of the cursor/head on the current line to the specified decipoint displacement relative to the current position.

Example: `ESC "&a" PRM(ALLASC) "H"`

8 Move Horizontal, Absolute, in Decipoints

Change the position of the cursor/head on the current line to the specified decipoint displacement from the left margin.

Example: `ESC "&r" PRM(ALLASC) "H"`

9 Horizontal Motion Index

Set the Horizontal Motion Index (HMI), which is the distance between characters or columns.

Example: `ESC "&k" PRM(ALLASC) "L"`

10 Backspace

Move the cursor/head over the most recently printed character.

Example: `BS`

11 Carriage Return

Move the cursor/head to the current left margin.

Example: `CR`

12 Horizontal Tab

Move the cursor/head to the next horizontal tabulation stop.

Example: `HT`

13 (reserved)

This field is reserved.

Class 6 Subclass Menu: Line Spacing

The class 6 subclass menu has fields that set line spacing values, set character heights, and move the cursor/head vertically.

1 6 LPI

Set the paper feed value to six lines per inch (8/48" per line).

Example: `ESC "&l" 6 "D"`

2 8 LPI

Set the paper feed value to eight lines per inch (6/48" per line).

Example: `ESC CHAR(2)`

3 10 LPI

Set the paper feed value to ten lines per inch (7/72" per line).

Example: `ESC 1`

4 Store a Line Spacing Value

Store a paper feed value, in fractional inch units, for later use.

Example: `ESC "A" PRM(ALLBIN)`

5 Set Stored Line Spacing Value

Set the paper feed value to the previously stored paper feed value.

Example: `ESC 2`

6 Set n/360" Line Spacing

Change the paper feed calibration to n/360" per line.

Example: `ESC "+" PRM(ALLBIN)`

7 Set n/180" Line Spacing

Change the paper feed calibration to n/180" per line.

Example: `ESC CHAR(3) PRM(ALLBIN)`

8 Set n/60" Line Spacing

Change the paper feed calibration to n/60" per line.

Example: `ESC "A" PRM(ALLBIN)`

9 Set n/48" Line Spacing

Change the paper feed calibration to n/48" per line.

Example: `ESC "&l" PRM(ALLASC) "C"`

10 Vertical Motion Index

Set the Vertical Motion Index (VMI), which is the distance between lines or rows.

Example: `ESC "J" PRM(ALLBIN)`

11 8 Point Character Height

Set character height to 8 points.

Example: `ESC "(s8V"`

12 10 Point Character Height

Set character height to 10 points.

Example: `ESC "(s10V"`

13 12 Point Character Height

Set character height to 12 points.

Example: `ESC "(s12V"`

14 18 Point Character Height

Set character height to 18 points.

Example: `ESC "(s18V"`

15 Form Feed

Advance the paper until the top of page is at the cursor/head.

Example: `FF`

16 Line Feed

Advance the paper until the cursor/head is at the same character position one line down.

Example: `LF`

17 Half Line Feed

Advance the paper until the cursor/head is at the same character position one-half line down.

Example: `ESC "="`

18 Reverse Line Feed

Reverse move the paper until the cursor/head is at the same character position one line up.

Example: `ESC "^" BS`

19 Vertical Tab

Advance the paper until the cursor/head is at the next vertical tabulation stop.

Example: `VT`

Class 7 Subclass Menu: Character Pitch

The class 7 subclass menu has fields that set character pitch and turn proportional spacing on or off.

`1 Set 8 cpi`

Set character pitch to fixed at eight characters per inch.

Example: `ESC "(s" 8 "H"`

`2 Set 10 cpi (pica)`

Set character pitch to fixed at ten characters per inch.

Example: `ESC "P"`

`3 Set 12 cpi (elite)`

Set character pitch to fixed at twelve characters per inch.

Example: `ESC " " PRM(LN,10,STH,PI.CP,OH,PI.FS,OH,PI.SF,WH)`

`4 Set 15 cpi (micron)`

Set character pitch to fixed at fifteen characters per inch.

Example: `ESC "g"`

`5 Set n CPI`

Set the number of characters per inch to the parameter value.

Example: `ESC "(s" PRM(ALLASC) "H"`

`6 Character Pitch Offset`

Define an additional increment to be added to a character.

Example: `ESC "p" PRM(ALLBIN) "&"`

`7 (reserved)`

This field is reserved.

`8 Proportional Printing On`

Enable proportional spacing during printing.

Example: `ESC "p" 1`

`9 Proportional Printing Off`

Disable proportional spacing during printing.

Example: `ESC "(s" 0 "P"`

10 Set Condensed Printing

Enable condensed, or compressed, printing, which reduces the character pitch.

Example: `SI`

11 Cancel Condensed Printing

Disable condensed, or compressed, printing.

Example: `HEX(12)`

Class 8 Subclass Menu: Form Layout

The class 8 subclass menu has fields that specify the form dimensions and orientation.

1 Set Form Length = n Lines

Define paper length in number of lines.

Example: `PRM(STP,PI.FL) ESC "C" PRM(WP,PI.FL)`

2 Set Form Length, in Inches

Define paper length in inches.

Example: `ESC "C" HEX(00) PRM(ALLBIN)`

3 Set Form for LETTER Size

Define the paper as LETTER size (8.5" X 11").

Example: `PRM(LN,11,STH,PI.FL) ESC "C" NUL HEX(11)`

4 Set Form for LEGAL Size

Define the paper as LEGAL size (8.5" X 14").

Example: `ESC "C" CHAR(14)`

5 Set Form for A4 Size

Define the paper as A4 size (210mm X 297mm).

Example: `ESC "&l" 26 "A"`

6 Set Form for Specified Size

Define paper size as specified by parameter.

Example: `ESC "&l" PRM(ALLASC) "A"`

7 Select PORTRAIT Page Orientation

Set the printer to print across the page width.

Example: `ESC "&l" 0 "O"`

8 Select LANDSCAPE Page Orientation

Set the printer to print across the page length.

Example: `ESC "&l" 1 "O"`

Class 9 Subclass Menu: Special Effects

The class 9 subclass menu has fields that cause special effects, such as double wide characters, shadowing, emphasized characters, outline characters, and the like.

1 **`Print Double Wide Characters`**

Enable printing characters twice their specified width.

Example: `ESC "W" HEX(01)`

2 **`Cancel Double Wide Characters`**

Disable printing double wide characters.

Example: `ESC "W" NUL`

3 **`Print Double High Characters`**

Set printing characters twice their specified height.

Example: `ESC "w" SOH`

4 **`Cancel Double High Characters`**

Disable printing double high characters.

Example: `ESC "w" HEX(00)`

5 **`Print Emphasized Characters`**

Set printing each character twice, the second time a little offset from the first.

Example: `ESC "E"`

6 **`Cancel Emphasized Characters`**

Disable printing emphasized characters.

Example: `ESC "F"`

7 **`Shadow On`**

Enable printing double-strike with offset.

Example: `ESC "q" STX`

8 **`Shadow Off`**

Disable printing double-strike with offset.

Example: `ESC "q" NUL`

9 Outline On

Enable printing outline characters.

Example: `ESC "q" HEX(01)`

10 Outline Off

Disable printing outline characters.

Example: `ESC "q" NUL`

11 Underline On

Enable printing underline.

Example: `ESC "&d" 0 "D"`

12 Underline Off

Disable printing underline.

Example: `ESC "&d@"`

13 Bold On

Enable printing bold characters.

Example: `ESC "(s" 3 "B"`

14 Bold Off

Disable printing bold characters.

Example: `ESC "(s" 0 "B"`

15 Strikethrough On

Enable printing the strikethrough character.

Example: `ESC "(-" HEX(030001) HEX(0201)`

16 Strikethrough Off

Disable printing the strikethrough character.

Example: `ESC "(-" HEX(030001) HEX(0200)`

17 Overscore On

Enable printing overscore.

Example: `ESC "(-" HEX(030001) HEX(0301)`

18 Overscore Off

Disable printing overscore.

Example: `ESC "(-" HEX(030001) HEX(0300)`

19 Superscript On

Enable printing characters a half line above the current line.

Example: `ESC "S" 0`

20 Subscript On

Enable printing characters a half line below the current line.

Example: `ESC "S" 1`

21 Super/Subscript Off

Disable superscript and subscript printing.

Example: `ESC "T"`

22 Set Upright Style

Disable printing italic style.

Example: `ESC "(s" 0 "S"`

23 Set Italic Style

Enable printing italic style.

Example: `ESC "(s" 1 "S"`

24 Select Draft Quality

Enable printing in rough draft quality, which is usually faster than higher quality printing.

Example: `ESC "x" HEX(00)`

25 Select Near Letter Quality

Enable printing in the printer's highest quality.

Example: `ESC "x" HEX(01)`

Class 10 Subclass Menu: Typeface

The class 10 subclass menu has fields that specify typefaces.

1 Select Default Typeface

Restore the default typeface.

Example: `ESC "(s0T"`

2 Select ROMAN Typeface

Select Roman typeface.

Example: `ESC "k" HEX(00)`

3 Select GOTHIC Typeface

Select Gothic typeface.

Example: `ESC " " PRM(LN,00,STH,PI.FS,LN,20,STH,PI.SF,OH,PI.CP,WH)`

4 Select COURIER Typeface

Select Courier typeface.

Example: `ESC "(s3T"`

5 Select PRESTIGE Typeface

Select Prestige typeface.

Example: `ESC "k" HEX(03)`

6 Select HELVETICA Typeface

Select Helvetica typeface.

Example: `ESC "(s0T"`

7 Select SCRIPT Typeface

Select Script typeface.

Example: `ESC "k" HEX(04)`

8 Select Typeface n

Select typeface by parameter.

Example: `ESC "k" PRM(ALLBIN)`

Class 11 Subclass Menu: Character Set

The class 11 subclass menu has fields that specify character sets and control codes.

1 `Character Set 1`

Select the primary, or default, character set.

Example: `ESC 7`

2 `Character Set 2`

Select the secondary, or alternate, character set.

Example: `ESC "t" HEX(01)`

3 `Graphics Set 1`

Select the primary, or default, graphics set.

Example: `ESC "t" HEX(02)`

4 `Graphics Set 2`

Select the secondary, or alternate, graphics set.

Example: `ESC "t" HEX(03)`

5 `Select INTERNATIONAL Set`

Select an INTERNATIONAL character set.

Example: `ESC "R" PRM(ALLBIN)`

6 `Select Character Set n`

Select the specified character set.

Example: `ESC "(" PRM(ALLASC)`

7 `Select Extended Characters`

Select the current set characters represented by numbers in the range 128 to 255.

Example: `SO`

8 `Deselect Extended Characters`

Disable use of extended characters.

Example: `ESC 5`

9 `Select User Set`

Select the character set downloaded, or defined, by the user.

Example: `ESC "%" HEX(01)`

10 Deselect User Set

Deselect the user-loaded character set and re-enable the normal, or default, set.

Example: `ESC "t" HEX(00)`

11 Remap User's Set

Remap user-loaded characters from positions 0-127 to positions 128-255.

Example: `ESC "t" HEX(02)`

12 Enable Control Codes

Enable treating character codes 128-159 as control codes.

Example: `ESC HEX(07)`

13 Print Control Codes

Enable treating character codes 128-159 as printable characters.

Example: `ESC HEX(06)`

Class 12 Subclass Menu: Font Selection

The class 12 subclass menu has fields that specify which font to use.

1 Select Primary Font Set

Select the primary, or default, font set.

Example: `SI`

2 Select Font Set 2

Select the secondary, or alternate, font set.

Example: `SO`

3 Select Font Set 3

Select the third font set.

Example: `ESC " " PRM(LN,02,STH,PI.FS,OH,PI.SF,OH,PI.CP,WH)`

4 Select Font Set 4

Select the fourth font set.

Example: `ESC " " PRM(LN,03,STH,PI.FS,OH,PI.SF,OH,PI.CP,WH)`

5 Select Font Set 5

Select the fifth font set.

Example: `ESC " " PRM(LN,04,STH,PI.FS,OH,PI.SF,OH,PI.CP,WH)`

6 Select Font Set 6

Select the sixth font set.

Example: `ESC "f" HEX(05)`

7 Select Font Set 7

Select the seventh font set.

Example: `ESC "$7"`

8 Select Font Set 8

Select the eighth font set.

Example: `ESC "F" 8`

9 Select Subfont 1

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,00,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

10 Select Subfont 2

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,20,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

11 Select Subfont 3

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,40,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

12 Select Subfont 4

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,60,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

13 Select Subfont 5

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,80,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

14 Select Subfont 6

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,A0,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

15 Select Subfont 7

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,C0,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

16 Select Subfont 8

Select a font from a set of fonts.

Example: `ESC " " PRM(LN,E0,STH,PI.SF,OH,PI.FS,OH,PI.CP,WH)`

Class 13 Subclass Menu: Color

The class 13 subclass menu has fields that select color.

1 Black

Select black printing color.

Example: `ESC "%c" 0 NUL`

2 Magenta

Select magenta printing color.

Example: `ESC "&M"`

3 Cyan

Select cyan printing color.

Example: `ESC HEX(1B)`

4 Violet

Select violet printing color.

Example: `ESC STX "cv" ETX`

5 Yellow

Select yellow printing color.

Example: `ESC CHAR(47)`

6 Red

Select red printing color.

Example: `ESC "&c" 3 HEX(00)`

7 Green

Select green printing color.

Example: `ESC 3 NUL`

Class 14 Subclass Menu: Defaults

The class 14 subclass menu has fields that contain default initialization settings for various options. When the despooler first starts, it moves these values into the indicated PtrCB fields.

1 Top Margin Default

The top margin initialization value into PI.TM.

Example: `NUL`

2 Bottom Margin Default

The bottom margin initialization value into PI.BM.

Example: `NUL`

3 Text Length Default

The text length initialization value into PI.TL.

Example: `HEX(00)`

4 Left Margin Default

The left margin initialization value into PI.LM.

Example: `NUL`

5 Right Margin Default

The right margin initialization value into PI.RM.

Example: `CHAR(79)`

6 Line Spacing Default

The line spacing initialization value into PI.LS.

Example: `CHAR(6)`

7 Character Pitch Default

The character pitch initialization value into PI.CP.

Example: `CHAR(08)`

8 Form Length Default

The form length initialization value into PI.FL.

Example: `CHAR(66)`

9 Typeface Default

The typeface initialization value into PI.TF.

Example: `NUL`

10 Character Set Default

The character set initialization value into PI.CS.

Example: `CHAR(01)`

11 Separator

A character, such as a comma, to be used as a separator character in control sequences into PI.SP.

Example: `"!"`

12 Default Font Set

The font set initialization value into PI.FS.

Example: `NUL`

13 Default Subfont

The subfont set initialization value into PI.SF.

Example: `HEX(00)`

14 Default Print Quality

The letter quality initialization value into PI.LQ.

Example: `1 "q%"`

15 Default HMI

The HMI initialization value into PI.HMI.

Example: `CHAR(08)`

16 Default VMI

The VMI initialization value into PI.VMI.

Example: `CHAR(6)`

17 Default EVFU

The EVFU initialization value into PI.EVFU.

Example: `0`

Appendix A
Decimal, Hexadecimal, and ASCII Table

Decimal	Hexadecimal	ASCII	Decimal	Hexadecimal	ASCII
000	00	NUL	048	30	0
001	01	SOH	049	31	1
002	02	STX	050	32	2
003	03	ETX	051	33	3
004	04	EOT	052	34	4
005	05	ENQ	053	35	5
006	06	ACK	054	36	6
007	07	BEL	055	37	7
008	08	BS	056	38	8
009	09	HT	057	39	9
010	0A	LF	058	3A	:
011	0B	VT	059	3B	;
012	0C	FF	060	3C	<
013	0D	CR	061	3D	=
014	0E	SO	062	3E	>
015	0F	SI	063	3F	?
016	10	DLE	064	40	@
017	11	DC1	065	41	A
018	12	DC2	066	42	B
019	13	DC3	067	43	C
020	14	DC4	068	44	D
021	15	NAK	069	45	E
022	16	SYN	070	46	F
023	17	ETB	071	47	G
024	18	CAN	072	48	H
025	19	EM	073	49	I
026	1A	SUB	074	4A	J
027	1B	ESC	075	4B	K
028	1C	FS	076	4C	L
029	1D	GS	077	4D	M
030	1E	RS	078	4E	N
031	1F	US	079	4F	O
032	20	SPACE	080	50	P
033	21	!	081	51	Q
034	22	"	082	52	R
035	23	#	083	53	S
036	24	$	084	54	T
037	25	%	085	55	U
038	26	&	086	56	V
039	27	'	087	57	W
040	28	(	088	58	X
041	29	)	089	59	Y
042	2A	*	090	5A	Z
043	2B	+	091	5B	[
044	2C	,	092	5C	\
045	2D	-	093	5D	]
046	2E	.	094	5E	^
047	2F	/	095	5F	_

Decimal	Hexadecimal	ASCII
096	60	`
097	61	a
098	62	b
099	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL
128	80	
129	81	
130	82	
131	83	
132	84	
133	85	
134	86	
135	87	
136	88	
137	89	
138	8A	
139	8B	
140	8C	
141	8D	
142	8E	
143	8F	
144	90	
145	91	
146	92	
147	93	
148	94	
149	95	
150	96	
151	97	
152	98	
153	99	
154	9A	
155	9B	
156	9C	
157	9D	
158	9E	
159	9F	
160	A0	
161	A1	
162	A2	
163	A3	
164	A4	
165	A5	
166	A6	
167	A7	
168	A8	
169	A9	
170	AA	
171	AB	
172	AC	
173	AD	
174	AE	
175	AF	
176	B0	
177	B1	
178	B2	
179	B3	
180	B4	
181	B5	
182	B6	
183	B7	
184	B8	
185	B9	
186	BA	
187	BB	
188	BC	
189	BD	
190	BE	
191	BF	

Decimal	Hexadecimal	ASCII	Decimal	Hexadecimal	ASCII
192	C0		224	E0	
193	C1		225	E1	
194	C2		226	E2	
195	C3		227	E3	
196	C4		228	E4	
197	C5		229	E5	
198	C6		230	E6	
199	C7		231	E7	
200	C8		232	E8	
201	C9		233	E9	
202	CA		234	EA	
203	CB		235	EB	
204	CC		236	EC	
205	CD		237	ED	
206	CE		238	EE	
207	CF		239	EF	
208	D0		240	F0	
209	D1		241	F1	
210	D2		242	F2	
211	D3		243	F3	
212	D4		244	F4	
213	D5		245	F5	
214	D6		246	F6	
215	D7		247	F7	
216	D8		248	F8	
217	D9		249	F9	
218	DA		250	FA	
219	DB		251	FB	(SB)
220	DC		252	FC	\(SVM)
221	DD		253	FD	](VM)
222	DE		254	FE	^(AM)
223	DF		255	FF	_(SM)

Figure B-1 displays the relationship of the PDM program and the files that it uses.

Source definition items are stored in the PRINTER file in the SYSFILES account. The item-id format is printer-name:class. For example, EPSON:8 is the source definition item for class 8 of the EPSON printer. The attributes of the item hold the subclass data.

When you enter FI at the PDM main menu, the program inputs the control sequences of any changed printer definitions in file PRINTER, converts the data to binary, and stores the output in object definition items in file PRINTER.TABLES. The format of the item-ids is printer-name:class. For example, EPSON:8 is the object definition item for class 8 of the EPSON printer.

An object definition is invoked by a Printer Independence Function Sequence (PIFS) embedded in a job print data.

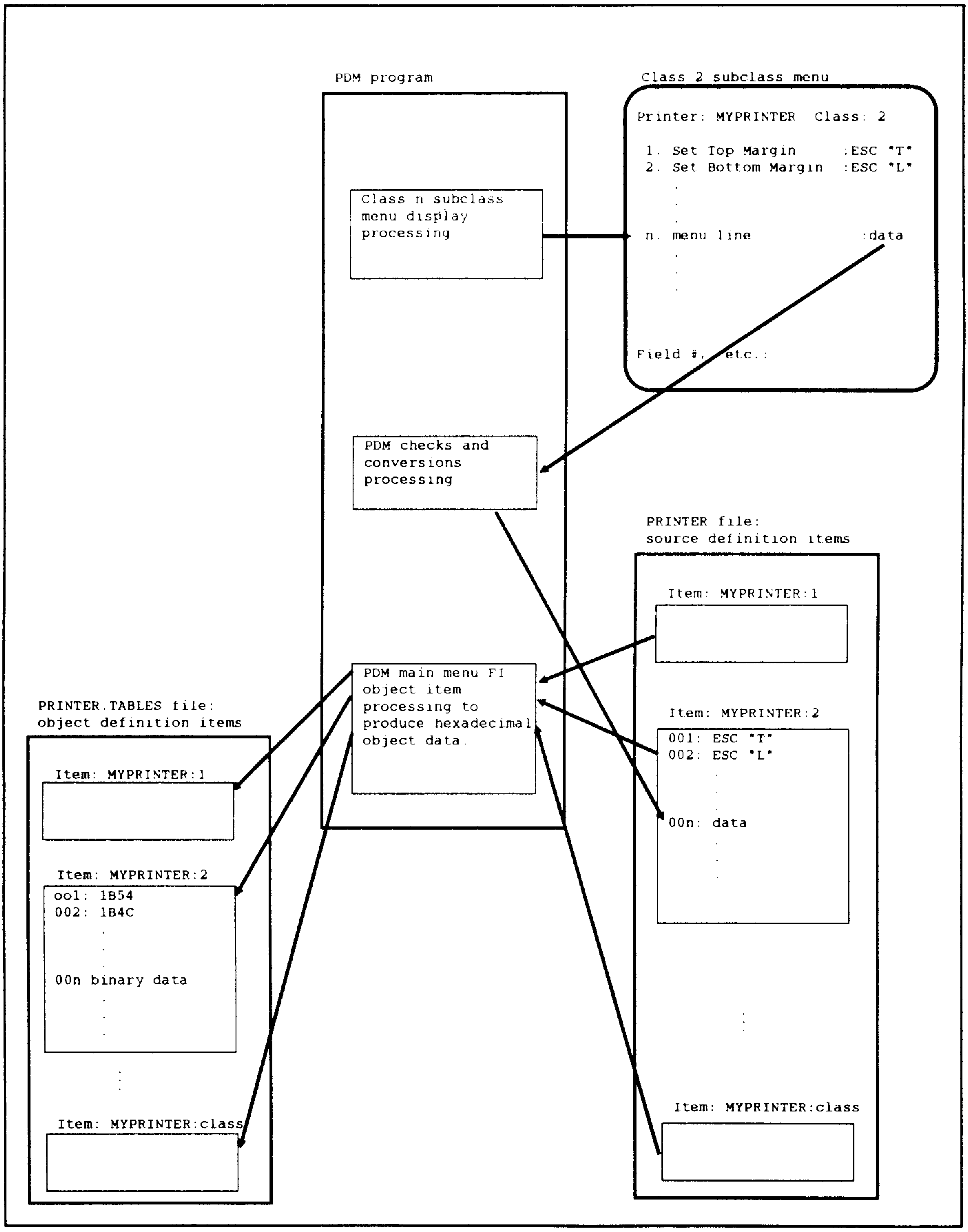

PDM program
Class n subclass menu display processing
PDM checks and conversions processing
PDM main menu F1 object item processing to produce hexadecimal object data.
Class 2 subclass menu
Printer: MYPRINTER Class: 2
1. Set Top Margin :ESC "T"
2. Set Bottom Margin :ESC "L"
n. menu line :data
Field #, etc.:
PRINTER file: source definition items
Item: MYPRINTER:1
Item: MYPRINTER:2
001: ESC "T"
002: ESC "L"
00n: data
Item: MYPRINTER:class
PRINTER.TABLES file: object definition items
Item: MYPRINTER:1
Item: MYPRINTER:2
ool: 1B54
002: 1B4C
00n binary data
Item: MYPRINTER:class

Figure B-1. PDM Program Processing and Files.

Overview

A Printer Independent Function Sequence (PIFS) is a sequence of special characters that invoke a Printer Dependent Control Sequence (PDCS). When the DATA/BASIC compiler processes a PTR function, it generates a PIFS. The PIFS function code references a class and subclass function.

As a despooler is processing a job, it checks for the presence of PIFS strings. If it finds one, it reads the item containing the referenced class and subclass, processes the PDCS, and sends the control data to the printer.

PIFS Format

A PIFS comprises the following characters in order:

- Value mark (], X'FD'), which is the introducer character.
- Tilde (~, X'7E'), which is the confirmation character.
- Function code, which consists of

  Class number.

  Colon (:, X'3A').

  Subclass number.
- Optional parameters: Each parameter consists of

  Subvalue mark (\, X'FC'), which is the parameter introducer.

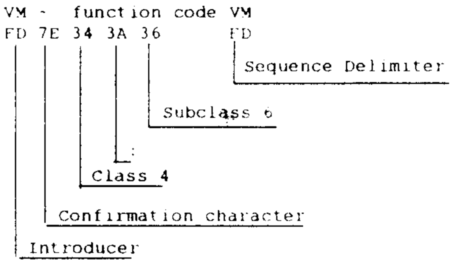
  ASCII characters.
- Value mark, which is the sequence delimiter. The system also treats the attribute mark (X'FE') or segment mark (X'FF') as a delimiter.

In summary the format is as follows:

]~*class*:*subclass*{\parameter}...]

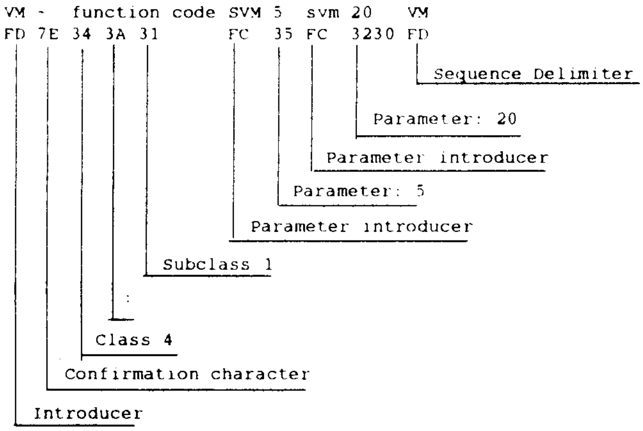
Example 1

This example invokes the function Clear All Vertical Tabs. The PIFS function code references class 4 and subclass 6



Example 2

This example invokes the function Set Horizontal Tabs, which is class 4, subclass1. The example includes two parameters, 5 and 20.


VM - function code SVM 5 svm 20 VM
FD 7E 34 3A 31 FC 35 FC 3230 FD
Sequence Delimiter
Parameter: 20
Parameter introducer
Parameter: 5
Parameter introducer
Subclass 1
:
Class 4
Confirmation character
Introducer