

Printer Manager

OPD/SPEC/3

The information in this document is proprietary to ICL, and is supplied to you in confidence on the understanding that you will not disclose it to third parties or reproduce it, and that you will use it solely for the purpose of developing applications software for use with the ICL product or products described in this document.

0 Document Control

0.1 Contents

- 0 Document Control
- 1 General
- 2 Printer Manager Event Handling
- 3 Interfaces to Applications
 - 3.1 OPENPRINT
 - 3.2 PRINT
 - 3.3 CLOSEPRINT
 - 3.4 Director handling of CTRL/PRINT key
 - 3.5 ENQPRINT
- 4 Start up requirements
- 5 Printer Configuration
 - 5.1 Basic configuration
 - 5.2 Housekeeping configuration
- 6 Image print keys
- 7 CMOS mapping of printer configuration
- 8 Programming Hints
 - 8.1 Normal use
 - 8.2 Example of how PM interprets data
 - 8.3 Printing via BASIC
 - 8.4 Using BASIC in transparent mode
 - 8.5 How to drive a printer
 - 8.6 Driving printers via BASIC
 - 8.7 Stack considerations
 - 8.8 Explanation of noticeboard messages

PROAAC

0.2 Changes since previous issue

Minor corrections

Control codes removed and replaced with reference to
CONFIGURATOR

Section 8 added

0.3 Changes forecast

PM may be required to perform more initialisation when
opening a printer. Also a "forms design" function may be
added

0.4 Related documents

Printer Manager Component Product Design Document (CPDD)
contains detailed design and implementation details.

1 General

The printer manager is implemented as a separate activity started by Application Manager at power on. It provides common facilities for applications wishing to use the printer, via four interfaces OPENPRINT, PRINT, ENQPRINT and CLOSEPRINT. Note the name "printer manager" is used to avoid confusion with the Kernel interface to the printer which has taken the name "printer driver", not from any desire to implement VME-like manager software!

There is nothing to stop applications using the Kernel printer driver interface directly if they so wish. However applications should be aware that the printer may be in image-print mode. The printer manager removes from applications the need to take account of the current printer configuration, to deal with printer busy errors, the handling of separate sheet stationery and Noticeboard printer messages, and provides new page, new line and other printer control command handling.

The printer control sequences are assumed by default to be those of the standard OPD printer (the ICL THERMAL) however the user will be allowed to configure certain of the control sequences to suit other attached printers - the EPSON RX80 and the Tonto printer. The configurable options and the actions of the printer manager will limit the possible printers that can be used. For example the initialisation sequence on a EPSON is one simple command, on the OKI most printer features must be reset separately - thus the printer manager will have to initialise by using multiple commands.

The printer manager will not take any special action on starting or completing a print operation, with regard to paper positioning at head of form. This is left to the user, possibly prompted by the application. However the printer manager will assume that the paper is at head of the configured paper size at the start of a print operation, issuing a printer command to that effect, and will make its decision on page positioning on that basis. At start of a print operation the printer manager will extract the current configuration from CMOS, and issue the configured printer initialisation commands and a 'home cursor' command.

During normal printing the print manager will count the data characters printed, and 'newline' commands received, in order to determine the current page position. The application is allowed to issue "new page" and "page position" commands also. "Newline" and "new page" commands will cause the appropriate configured printer commands to be output, and "page position" will generate the appropriate number of new lines and spaces. The printer manager will not generate its own new line or new page commands when its

PROAAC

OPD/SPEC/3
4/0
PM-5

counts reach end of line or end of page, merely resetting its line/column pointers, relying on the ability of the printer to do automatic line wraparound and skip over perforation also (if set).

At end of printing, on CLOSEPRINT, the printer manager will flush the buffer of any as yet unprinted characters by issuing CR, LF.

The printer manager will output to the noticeboard the usual printer information messages, see PSD 76.97.2.1 sec. 9, as:

PRINTING STARTED, after a successful OPENPRINT

PRINTING COMPLETE, after a successful CLOSEPRINT

PRINTER NOT FREE, if OPENPRINT cannot get a printer

PRINTER INOPERABLE, in the special case of printer time-out on queued write, see below.

PRINTING ABANDONED, where the application tells printer manager that it is doing a forced close.

The printer manager will output its print data to the Kernel queued putstring interface, and will issue a 10 second timeout on each operation. If the operation times out before the kernel event indicating the successful completion then the printer will be assumed to have entered a busy state. When this occurs the printer manager will carry out an initial check that the microdrives are not in use. If a microdrive is selected then the timer is restarted for a further 30 second wait. At the end of this extra period, or after the initial 10 second wait if the microdrives were not in use, the message PRINTER INOPERABLE is output to the Noticeboard, with an attention tone. The printer manager will then enter a long timeout of 5 minutes, allowing the user to take action if the busy is due to a condition like 'paper out'. If this further wait expires then the PRINTER INOPERABLE message is refreshed, the application informed its operation has terminated, the printer is closed and its status is marked as DE (device error) for future calls other than CLOSE/OPEN. If at any time during these extra waits the kernel event is received then the operation will have completed successfully, and the timers are cancelled.

Separate sheet stationery printing is allowed for in the printer manager, this option is configurable via the "printing options selection on the housekeeping menu. At end of page in the separate sheet case, the printer manager notes end of page reached. Any subsequent output causes the noticeboard message "NEW PAGE? CTRL/PRINT", to be

PROAAC

displayed. PM then waits for an event from DIRECTOR to say that the appropriate keys have been pressed. There is a 5 minute timeout on this event - if not received the printer is forcibly closed. No further PRINT operations will be allowed before the event is received. After the event it is assumed the printer has been reset, and the current PRINT operation or subsequent ones may proceed. (Note that the length of time the message remains displayed is not guaranteed.)

The interfaces to the printer manager are described in detail in section 3 following.

PROAAC

2

The printer manager events

The printer manager activity will process requests from applications asynchronously with the applications own activities, consequently communication in both directions is triggered by local events.

The applications may specify an event number on the OPENPRINT and PRINT interfaces that printer manager will use to inform them that the requested action has been completed, it may have been successful or have failed. Having received an event the application may proceed with its next call, this may return failure if the previous operation failed or success if the new one has been started.

The interface procedures signal new actions to printer manager with events local to the printer manager.

Printer data is written by queued putstring calls, and the printer manager waits for the completion event or a timer event of 20 seconds. If complete the next sequence is constructed and output, or if no more the applications event is signalled.

Additional timer events are used by printer manager to cater for the wait for CTRL/PRINT key, and the wait for next user action.

PROAAC

3 Interfaces to Applications

3.1 OPENPRINT

3.1.1 OPENPRINT - general

Initially an application will use this interface to start a new print operation. The interface procedure will check that the printer is free and open it via the Kernel interface stopping any other access.

The interface will return success or failure codes as 3.1.2, and also the channel identifier which will be used on subsequent calls to printer manager interfaces.

PROAAC

3.1.2 OPENPRINT Interface

Trap Name: T.PRINMN

Action Value:(D0.B) PM.OPEN

Additional Parameters:

D3.W : Event number (or -1)

Return Information:

D1.L : Channel identifier for this print
operation

Normal and Error Returns:

On return D0.L contains a response code. If this is positive or zero the OPENPRINT call has been successful and D1.L is valid. If negative the call has failed and D0 contains an error code as below. Condition codes are set on return to reflect the state of D0.

Error Returns:

OM : out of memory (unable to create CCA)
from Kernel

IU : printer is already in use elsewhere.

NA : printer manager has not completed
initialisation or has been abandoned.

SL : unable to access CMOS (save/load
probably running)

NF or OR : unable to read CMOS (probably corrupt)

PROAAC

3.1.3 OPENPRINT Description

The interface procedure opens the printer channel for the printer manager, and if successful wakes up printer manager with an event, returning response and channel ID to the application.

If the interface procedure finds the printer manager is busy processing an earlier call or the printer is already open then an error response of IU is immediately made. Similarly OM may be returned if the Kernel open fails with this response.

Printer Manager open routine will take action as follows on receipt of the open event from the interface procedure:

- (i) A 'PRINTING STARTED' message is output to the noticeboard.
- (ii) 480 null bytes (as 6, 80 byte strings) are output to ensure the printer is not left in an interrupted image-print mode, when complete a clear buffer sequence is output. If these queued putstrings fail the OPENPRINT is abandoned, the Noticeboard message PRINTER INOPERABLE is output and the printer closed down. If OK then the open proceeds as follows.
- (iii) The printer details, as configured, are obtained from CMOS, and set up for use by this new print operation. This allows re-configuration between operations, especially those options described in 5.2. Note that the baud rate will be extracted from CMOS and used during the original OPENPRINT interface call to the Kernel open.
- (iv) The printer initialisation sequence as configured (see 3.2.3 (c)) is then output as a queued putstring to the printer and the printer manager will wait for termination or another event as described in 2.
- (v) On completion of the initialisation, if an event has been supplied it is signalled to the application.
- (vi) The print manager returns to its normal wait state for further user events. Note that there is a 30 second timeout on this wait, at which point if there has been no further application events the print manager will check the status of the current application activity. If this no longer exists a forced entry to CLOSEPRINT is made, else a further 30 second wait is started.

PROAAC

3.2 PRINT

3.2.1 PRINT - general

Normally an application having successfully opened the printer will call this interface to carry out its actual printing. The interface procedure will check that the printer is open for the given channel identifier, and that the printer manager is not busy or in a failed state, returning success or error codes correspondingly.

If successful, printer manager is signalled to output the buffer contents and when that is completed it will signal the given event back to the application.

Until the application sees the event it should not change the contents of the buffer, else the print will be unpredictable.

Normally the application having called PRINT will wait for the specified event before using the print manager further, but if further calls are made before print manager has finished (or has decided the printer is inoperable) then PRINT will return the response IU, in use.

The printer manager will process the supplied buffer and output data strings or translated control sequences to the printer, unless the print operation is in "IMAGE PRINT" mode when the buffer is given immediately to queued putstring.

3.2.2 PRINT Interface

Trap Name : T.PRINMN

Action Value (D0.B) : PM.PRINT

Additional Parameters:

D1.L : Channel Identifier
D2.W : String Length in bytes (max value as
for Kernel queued putstring or 80
bytes if this is an image print mode
call)
D3.W : Event number, or -1
D4.B : Image print mode (1=Yes, 0=No)
A1 : Address of buffer

Return Information

None

Normal and Error Returns:

On return D0.L contains a response code. If this is positive or zero the PRINT request has been successful (i.e. accepted as valid for passing to printer manager but not yet carried out!). If negative then the call has failed and D0 contains an error code as below. Condition codes are set to reflect the value in D0 on return.

Error Returns:

IU : print manager is still busy processing
on earlier request on this print
operation
NO : the channel identifier is invalid, or
has been closed.
DE : the printer has gone inoperable
following an earlier operation. The
printer has been closed and will need
to be re-opened, after user action
presumably.
NC : the printer manager is waiting for
user action on separate sheet printing
and the CTRL/PRINT key press.
NA : printer manager has not completed
initialisation or has been abandoned.

PROAAC

3.2.3 PRINT DESCRIPTION

The interface procedure checks that the printer channel is the current one open, that the printer manager is not still busy, that the print manager is not waiting for new sheet action, that the printer has not gone inoperable, and if successful wakes up the printer manager with an event and returns success to the caller.

If the interface procedure finds that the printer channel is not valid then it returns the response NO immediately; if the printer manager is still processing an earlier call the response is IU; if the printer manager is waiting for the user to load a new sheet the response is NC; if the printer has become inoperable the response is DE.

The actions of the printer manager when it receives the PRINT event from the interface procedure are:-

(i) If this print is in image print mode then the buffer is passed complete & unprocessed to queued putstring and the appropriate (10 second) timer is started. The printer manager will then wait either for the putstring event or the timer event.

If the putstring event occurs first, the application is signalled with its event and this PRINT call is complete, the printer manager returning to its event wait.

If the timer times out first, then the printer status is marked as DE, the noticeboard updated "PRINTER INOPERABLE" and any associated event signalled to the caller.

Note that in image print mode there is no check on page positioning.

(ii) If not in image print mode the input buffer is processed - consecutive data characters are output for printing via queued putstring, with the appropriate time out as before, embedded control characters found are actioned as in (iv) below. Any non-printable, or non-recognised control, characters will be ignored. In order to keep the maximum timeout period at 10 seconds the largest string passed to putstring is limited to 80 bytes.

When the buffer has been successfully output or timeout has occurred, the PRINT process terminates as in (i).

(iii) The printer manager will keep column and line position counters, initialised at PRINTOPEN. It will add to

PROAAC

the column counter for every data character printed (including the configurable characters "hash" and "block"), and to the line counter for every application issued 'newline' or before the 'linelength+1' data character is printed although see (a) below. The counters will only be used for processing page position commands (see (n) below), and in deciding page end has been reached for separate sheet stationery (actioned as for application newpage see (b) below).

Note no printer commands are issued by printer manager at line or page end.

(iv) The printer manager actions on the various recognised control codes are:-

(a) Newline. Code \$1C.

Output configured new line string to printer. Reset column count. On all the supported printers the next data byte at line full will cause the line to be printed and a newline executed, also if at line full the next character is a newline control character then only 1 line is fed. The printer manager will thus allow 'newline' through at line full but will only count one extra line to itself.

(b) Newpage. Code \$0C.

Output configured newpage string to printer. Reset line & column counts. If in separate stationery mode then the newpage flag is set. Any subsequent output causes the noticeboard prompt NEW PAGE? CTRL/PRINT to be displayed, a five minute timeout started and enter wait for event. The user will insert a new sheet and press the key, or may just press the key. If the CTRL/PRINT key is pressed this event will free printer manager from the wait, and it will resume the processing of the current buffer. If the wait for the key times out then the printer will be closed and any currently PRINT signalled complete, the action taken is as for printer inoperable. Note one can only hope the message remains in the Noticeboard long enough to be seen. To the printer manager the new page code immediately after page end has been reached in 'separate stationery' mode, is a special case and will be ignored.

(c) Initialise Printer. Code \$07.

This causes the same sequence to be output to the printer as that from OPENPRINT. The sequence is: output 480 nulls, set 1/6 line spacing, homehead,

PROAAC

underlining OFF, subscript OFF, superscript OFF, enlarged print OFF, reset top-of-form, if single-sheet stationery then disable paper end detect else enable it and set correspondence vs draft print quality as configured. The line & column counts are reset.

- (d) Letter quality print on. Code \$04.

Output the configured string (but note this may only produce emphasized print). If the user has configured letter quality on, this string will be issued by OPENPRINT.

- (e) Letter quality print off. Code \$03.

Output the configured string. Note that if draft quality print is configured this string is output by OPENPRINT

- (f) Underline on. Code \$06.

Output the configured string.

- (g) Underline off. Code \$05.

Output the configured string.

- (h) Subscript print on. Code \$16.

Output the configured string.

- (i) Subscript print off. Code \$15.

Output the configured string.

- (j) Superscript print on. Code \$18.

Output the configured string.

- (k) Superscript print off. Code \$17.

Output the configured string.

- (l) Tab n spaces. Code \$09.

This control code is followed by a byte giving the number of column to be tabbed to. Tab 0 is treated as tab (linelength-1). The tab position is checked for being legal ie. to right of current column and less than line length. If it is not legal it is ignored if to the left or equal to the current position. If not legal and greater than the line length OR if tab 0 then the tab command is assumed to be (line length-1) and

the appropriate number of spaces output. The number of spaces depends on whether enlarged print is on or not.

- (m) Print "hash" character. Code \$CA.

Output the configured string.

- (n) Page position. Code \$1F.

This control character is followed by 2 bytes, the first giving the new column position and the second the new line number required. Check that the values are legal ie. beyond current position and within defined page, if not ignore else issue the appropriate number of newlines and spaces. The positioning will also depend on whether enlarged print is on. Note that page position zero is not treated as tab 0 (see (l) above) but as a genuine zero which means it is always ignored!

- (o) Enlarged print. Code \$10.

This is followed by one data byte that is 1 to switch on enlarged, or 0 to switch it off. Note that with this on, each data character adds 2 to column count, and newline may thus occur at linelength-1.

- (p) Carriage return. Code \$0D.

Output the configured homehead string. PM initialises its column counter.

- (q) Perforation skip on. Code \$1A.

Output the configured string. Note that this provides a fixed 1 inch skip always due to the way PM initialises the printer. If continuous stationery is being used perforation skip is set on by OPENPRINT.

- (r) Perforation skip off. Code \$19.

Output the configured string. If single sheet stationery is in use the perforation skip is set off by OPENPRINT.

- (s) Clear buffer/Cancel print. Code \$14

Output the configured string. This code provides a means of flushing the printer buffer without printing contents.

- (t) Escape code. Code \$00.

This control character will cause the printer manager to pass the following byte unprocessed to the printer, it will assume that it is not a data byte. There must be a following byte in the buffer.

- (u) Image print feed line. Code \$01

Output the configured string for image print paper advance.

- (v) Image print single density data. Code \$02.

Output the configured string for single density image data print.

- (w) Image print double density data. Code \$1E

Output the configured string for double density image data print.

- (x) Linefeed. Code \$0A.

This code is passed through unchanged, it is not a configurable option. PM increments its line counter.

- (y) ASCII block character. Code \$7F.

Causes PM to output a little black square.

- (z) Others.

Certain control codes are internal to PM and may not be used across the PRINT interface. The configured sequences corresponding to these are included in the table in section 5.1. They are :-
set line spacing
set top-of-form
paper end detect enable and disable

3.3 CLOSEPRINT

3.3.1 CLOSEPRINT - general

This interface should be used to free the printer from this operation, and make it available for others' use.

The interface procedure will check that the printer is still open and has not become inoperable before signalling printer manager to take action, and returning a response. Note if the printer is inoperable it will already have been closed by the printer manager. The interface allows the application to specify that this is a forced close, in this case the normal close action of the printer manager to write "PRINTING COMPLETED" to the noticeboard is changed to output "PRINTING ABANDONED".

PROAAC

3.3.2 CLOSEPRINT Interface

Trap Name : T. PRINMN

Action Value (DO.B): PM.CLOSE

Additional Parameters:

D1.L : Channel Identifier
D2.B : Print abandoned (=1 Yes, =0 No)
D3.W : Event number (or -1)

Return Information:

None

Normal and Error returns:

On return DO.L contains a response code. If this is positive or zero the CLOSE request has been successful (i.e. accepted and passed to the printer manager for action). If negative then the printer is not open, is busy, or has become inoperable. In the busy case the request will only be passed to printer manager in the print abandoned case. In the inoperable case the request will not be passed to the printer manager, although the Noticeboard message PRINTER INOPERABLE is renewed. Condition codes are set to reflect the value in DO on return.

Error returns:

NO : the channel identifier is invalid, or
has been closed

IU : the printer is still busy processing
an earlier request, printer manager
is only called if abandoning.

NC : the printer manager is waiting for
user action on new page, printer
manager is only called if abandoning.

NA : the printer manager has not been
initialised, or has been abandoned.

PROAAC

3.3.3 CLOSEPRINT Description.

The interface procedure checks that the printer is not in the inoperable state, when doing a normal close. If it is then response IU is returned to the application. The printer manager is not called in this case and any flush buffer is left to a later OPENPRINT.

The interface procedure checks that the channel identifier is the current one open, if not and a normal close an immediate response of NO is made, with no call to printer manager.

The interface procedure checks that the printer manager is not still busy on an earlier action, or that it is not waiting for user action for NEW PAGE, if it is and this call is not 'Print abandoned' then immediate return is made with IU or NC.

In all other cases the interface procedure signals the printer manager to close down the current print operation.

The actions of the printer manager for CLOSEPRINT are:

(i) The printer manager outputs the flush buffer sequence to queued putstring, this is the configured newline string (the homehead that may be necessary on an EPSON being done as part of initialisation on the next open) for normal close, or clear buffer for abandon. If this putstring timesout then the printer is closed, the message PRINTER INOPERABLE output and the printer manager is left in the failed state for a later open to clear.

(ii) Otherwise the CLOSEPRINT has completed successfully, and the Noticeboard message PRINTING ABANDONED or PRINTING COMPLETE output (depending on 'print abandoned' flag). The printer is closed via the kernel and the printer manager goes back to sleep.

PROAAC

3.4 CTRLPRINT

3.4.1 CTRLPRINT - general

This interface is private for DIRECTORs use only - although not policed.

DIRECTOR will use this interface when it sees the CTRL/PRINT key combination. Print manager is signalled that the user has pressed this restart print key, and if in the correct (separate sheet stationery) state resume printing. If PM is not in the correct state the key press is ignored and PM is not invoked.

PROAAC

3.4.2 CTRL/PRINT interface

Trap Name : T. PRINMN

Action Value (D0.B): PM.CTRL P

Additional Parameters:

None

Return Information:

None

Normal return:

On return D0.L will always be zero, and the condition code set correspondingly.

PROAAC

3.4.3 CTRLPRINT Description.

Unless the printer manager is in the wait for NEW PAGE state, this call is ignored.

Otherwise the printer manager will assume the user has inserted a new page and wishes printing to continue (or perhaps has decided to let the print continue anyway). It will then allow the current PRINT operation to continue or if it has now completed signal the completed event to the application (see 3.2.3 (b)).

PROAAC

3.5. ENQPRINT.

3.5.1 ENQPRINT - general

This interface procedure may be called at any time within a print operation.

The procedure will return information about the current page positioning as seen by the printer manager at the time of the call, and the printer status.

3.5.2 ENQPRINT Interface

Trap name : T.PRINTMN

Action Value (D0.B) : PM.ENQ

Additional Parameters :

D1.L : Channel Identifier

Return Information :

D1.L :

W	H	X	Y
---	---	---	---

where W = configured width of page in characters
H = configured height of page in lines
X = current print column position, offset from
start of line
Y = current print line position, offset from
head of page.

Normal and Error returns :

On return D0.L contains a response code. If positive or zero the ENQPRINT request has been successful and the value in D1 is meaningful, otherwise D1 should be ignored, although see IU below. Condition codes are set to reflect state of D0.

Error returns :

- NO : the channel identifier is invalid, or has been closed.
- DE : the printer has gone inoperable following an earlier timeout.
- IU : the printer manager is still processing an earlier call. The values in D1 are those current at the call, but position may now have changed.
- NA : the printer manager has not been initialised, or has been abandoned.

PROAAC

3.5.3 ENQPRINT Description

This interface procedure will return immediately with information request or an error code. If the status IU is returned then the page position returned is not necessarily the final position after the current print operation completes.

PROAAC

4 Start up requirements

Printer Manager is started by the Application Handler at cold start, at entry point PM_PRINT with a stack of 972 bytes.

The print managers' priority is initially 64.

After initialisation of data areas the printer manager will enter an indefinite wait for an event from its interface procedures.

The initialisation process is such that an interface procedure can tell whether it has been called before the printer manager has completed initialisation, in such a case the interface procedure will return the response NA.

The printer manager may be deleted via the store report or may suicide after an abandon event from DIRECTOR. In both these cases a subsequent call will resurrect the printer manager. If deleted/abandoned during a print operation PM loses its data and will return NO to subsequent calls. The next OPENPRINT call will proceed normally.

5. Printer Configuration

There are two different types of printer configuration that the user may carry out. The configuration data from both types is held in CMOS memory with the format defined in section 7.

The first type of configuration is similar in approach and design to that of the main system configuration, and is written as a BASIC program.

This configuration process allows the user to define the operating characteristics and control sequences for the printer to be connected. Additionally as part of this process the user is allowed to choose which of the possible image print options is to map onto the two image print keys available.

The second type of configuration is intended for those printing choices that are made more frequently, and is accessible off the Housekeeping menu.

In both cases when the user has completed a configuration operation, the updated information will be used for the next print operation started, ie. on the next OPENPRINT call, or the next image print key pressed.

5.1 BASIC Configuration.

The initial configuration screen will allow the user to choose to install one of the three standard printers, or to set up in detail some alternate printer.

If an unknown printer is to be set up, or some detail of a standard printer looked at or changed, then the next screens show the printer configuration in detail.

These details are in two classes, the control code sequences for the various print operations, and the printer operational characteristics. A summary is given below of the data in both categories. For the standard printers the data is preset in the program and is displayed for the user to modify if desired, for an unknown printer the ICL THERMAL data is displayed as a guide.

The attached table shows the configured sequences currently supported for the three standard printers.

PROAAC

Table of control codes

This is a summary of the control codes actioned by PM across the user interface. PM uses certain other control codes, for example "set line width" but these cannot be embedded in the data stream to PM.

Code	Meaning
00	escape character
01	image print feed line
02	image print single density data
03	letter quality print off
04	letter quality print on
05	underlining off
06	underlining on
07	top of form
09	tab "n" spaces
0A	linefeed
0C	newpage - executes a formfeed
0D	carriage return
10	enlarged print select/deselect
14	clear buffer/cancel print
15	subscript off
16	subscript on
17	superscript off
18	superscript on
19	perforation skip off
1A	perforation skip on
1C	newline - ie a carriage-return, linefeed
1E	image print double density data
1F	page position
7F	graphics block character
CA	print "hash" character

Printer operational characteristics:

Baud rate
Lines per page
Characters per line

A separate screen allows the user to choose which of the three image print styles is to map onto each of the key combinations, SHIFT/PRINT and ALT/PRINT.

5.2 HOUSEKEEPING CONFIGURATION

The housekeeping menu has a selection :-

PROAAC

9 Printing options

On the Printing Options screen the user may toggle the value of 'Type of Stationery' between 'Continuous' and 'single sheet', and the value of 'Print Quality' between 'Correspondence' and 'Draft'.

Note this part of the user interface is too late to be added to the Handbook, and will probably be described in the Configuration Manual.

6 Image print keys

Only two PRINT keys are available to the image print software; SHIFT/PRINT and ALT/PRINT, however the user is able via the BASIC printer configurator to select which of the three present print styles map onto these two keys.

The director routes only these two keys to the Image print, and the third (CTRL/PRINT) to the Printer manager.

Image printer uses the print manager to open the printer, print each screen chunk and to close the printer. Apart from hopefully removing some code, the printer manager will enable the image printer to close down the data image print successfully in the abandon case.

PROAAC

7 CMOS Mapping of Printer Configuration

Three CMOS records are used.:-

1. Image print configuration. CMOS record \$0D.

A 2 byte record.

First byte contains the colour conversions to black and white for image printing.

The second byte is set up as :-

Top nibble, ALT/PRINT key meaning, values 0/1 or 2

Bottom nibble, SHIFT/PRINT key meaning, values 0/1 or 2

Value 0 is full size (current SHIFT/PRINT), value 1 is full width/half height (current ALT/PRINT) and value 2 is $\frac{1}{2}$ width/ $\frac{1}{2}$ height (current CTRL/PRINT).

2. Printing Options. CMOS record \$11.

A record of 1 byte with format:-

Bit 0 set if continuous stationery

Bit 0 clear if separate stationery

Bit 1 set if letter quality print on

Bit 1 clear if draft print quality

3. Printer Operational Characteristics. CMOS record \$12.

A record of 3 bytes with format:-

1 byte band rate, encoded as per kernel's requirements
see PSD 76.97.3.1 section 14.5.1.

1 byte number of lines per page, binary.

1 byte number of characters per line, binary.

4. Configurable control sequences. CMOS record \$13.

A record of variable length, maximum 147 bytes.

For each defined sequence the format is:

1 byte : number of bytes in the sequence, binary

n bytes: sequence of ASCII characters to a maximum of 6 bytes.

The currently defined configurable sequences are shown in the table in section 5.1

8 Programming hints

8.1 Normal use

Interfacing to PM may be done directly using the interfaces described in section 3. In this case the normal sequence of events is :-

Use the OPENPRINT interface to start off a job.

Use the channel identity returned by the above call and execute one or more PRINT interface calls to actually output the data to the printer. Control codes may be embedded in the data stream. Control codes are usually a single byte and enable paper control sequences (eg formfeed) and formatting sequences (eg superscript print). Control codes are described in section 3.2.3 and summarised in section 5.1.

The ENQPRINT interface may be used at any time after a successful OPENPRINT call.

After all the data has been sent to PM use the CLOSEPRINT interface to terminate the print job and free the printer.

8.2 Example of how PM interprets data

After using the OPENPRINT interface you may send a block of data to PM looking like this :-

0C,1C,09,10,06,48,45,41,44,49,4E,47,1C,1C,05,54,65,78,74

This will cause the printer to throw a page, feed a newline and print the following :-

HEADING

Text

Taking each character in turn,

0C is interpreted by PM as a formfeed

1C is interpreted as a newline (carriage return + linefeed)

09,10 is interpreted as TAB to column 16

06 is interpreted as switch underlining on

48,45,41,44,49,4E,47 are not interpreted by PM and are passed on unchanged to the printer which prints them as ASCII text (underlined).

PROAAC

1C,1C is interpreted as two newlines
05 is interpreted as switch underlining off.
the remaining characters are passed through unchanged.

Characters interpreted by PM are translated to the actual string of characters required to effect the desired result. For example 06 might be translated into <ESC><-><1>. PM uses the configured strings set up by the CONFIGURATOR tool to do this translation. See section 8.5 below.

8.3 Printing via BASIC

It is important to note that BASIC uses the PM interfaces. For example the BASIC statement OPEN #3,pr causes the BASIC interpreter to issue an OPENPRINT call to PM.

Characters that are not recognised by BASIC are translated to a space character before being sent to PM. Thus the statement PRINT #3,chr\$(0); will actually cause a space to be output on the printer.

A PRINT statement not terminated by a semicolon will cause a newline. For example PRINT #3,"Hello" causes a newline after the "o" character.

8.4 Using BASIC in transparent mode

You may wish to force BASIC to pass all characters through to PM without translating some to spaces. To do this the device identity used in the OPEN statement should be PRT. For example the statements

```
OPEN #3,prt  
PRINT #3,chr$(0);
```

will cause BASIC to send a binary zero byte to PM rather than a space character.

Note that when in transparent mode BASIC will append a carriage return to the end of a string if the PRINT statement is not terminated with a semicolon.

8.5 How to drive a printer

You may well wish to drive a printer that is not on the standard OPD repertoire. Or you may wish to make a printer do things not covered in the set of standard PM control codes. Or you may wish to change the way the standard control codes work. We will cover each in turn.

To install a different printer use the CONFIGURATOR tool

CONPRIN and set up the configured control sequences to suit your printer hardware. Use CONPRIN to install your new printer and continue.

To change the way the standard control codes work use CONPRIN. For example when PM initialises the printer it uses the "text line feed" control sequence to set the printers line width. This is defaulted to 1/6". Using CONPRIN this sequence could be changed, for example, to 1/8". All future print jobs would then default to 1/8" line spacing.

To make the printer do things not covered in the standard set of printer control codes you must do it yourself. This may or may not be easy. Suppose you want to send a control sequence of ESC 8 to the printer. You can just set the bytes into the data stream and PM will output them to the printer. No problem. Now suppose you want to send ESC 6 to the printer. If you set the bytes in the data stream PM will pick up the code 06 and translate it to the "turn underlining on" control sequence. You cannot use CONPRIN to turn this translation off. Even if you nullify the configured string for turning underlining on, PM will translate 06 to a null and send it to the printer.

So we need to send characters to PM and tell it not to translate them. We do this by use of the PM escape code (not to be confused with the ASCII ESC character). The PM escape code is a binary zero byte. If we send a zero byte to PM it interprets it as meaning "send the next byte unaltered to the printer". In the example above, to send ESC 6 to the printer we need to send ESC,0,6 to PM which then sends ESC,6 to the printer. (Note that to send a null to the printer you need to send two nulls to PM!).

So by a combination of CONPRIN, the PRINT interface and judicious use of the PM escape character it should be possible to do anything you want with printers.

Readers are referred to the advanced operations manual for a description of how to use CONPRIN.

8.6

Driving printers from BASIC

The preceeding section explained how to drive printers via PM and assumed direct use of the PM interfaces. However you may wish to use BASIC to drive a printer. This introduces a further level of complication. To understand this section you must read all the previous parts of section 8.

The first thing to note is that we must use BASIC in transparent mode as described in section 8.4 above. This is to stop BASIC converting characters it doesn't recognise into spaces.

Suppose we want to send the string ESC,8 to the printer. We can do this by writing

```
PRINT #3,chr$(27);:PRINT #3,chr$(8);
```

or more simply by concatenating the characters :-

```
PRINT #3,chr$(27)&chr$(8);
```

This is OK. However once again we run into trouble if we want to send ESC 6 to the printer as PM would intercept the 06 code. To send ESC 6 to the printer we would write

```
OPEN #3,prt  
PRINT #3,chr$(27)&chr$(0)&chr$(6);
```

Readers are referred to the BASIC manual for further information on BASIC.

8.7 Stack considerations

If using the PM interfaces directly programmers are reminded that the interface procedures run on the callers stack. When any call to PM is made you must ensure that there is sufficient space on your stack to accommodate PM. At least 128 bytes are required. If using BASIC no special considerations apply.

8.8 Explanation of noticeboard messages.

PM monitors the progress of a print job by displaying messages in the noticeboard. Sometimes the message "PRINTER INOPERABLE" is displayed along with a warning tone. However the printer continues printing, after a short while the inoperable message is replaced with "PRINTING STARTED", and everything continues normally. The explanation for this is as follows.

PM sends data to the printer and sets a timeout on the printer replying. Usually the printer will buffer data and PM can send quite large amounts. Once the printer's buffer becomes full it waits until it is empty before replying to an outstanding request from PM. Consequently the timeout may mature. The timeout value is currently twenty seconds though it may be extended if the microdrives are active as these can slow up the printer.

If the printer is unusually slow or if it has an unusually large buffer then the PM timeout may mature and this produces the "inoperable" noticeboard message. However this is only a warning, PM will in fact wait for a further five minutes before taking any other action. This five minute wait is designed to allow a user to take corrective action in the event of a genuine failure eg running out of paper.

If the printer responds to PM within the five minutes then PM retracts the "inoperable" message and carries on. If the five minute timer matures PM fails the print job and returns an error to its user application. The twenty second timer is not configurable unfortunately.

Similarly the "PRINTING COMPLETED" message may appear quite some time before the printer finishes printing. This is due to PM almost filling the printers buffer and completing a job. The printer then chugs along for a while to empty its buffer. PM allows for this fact should another print job be started immediately.

PROAAC