

printer

COLLABORATORS

	<i>TITLE :</i> printer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	printer	1
1.1	printer.doc	1
1.2	printer.device/CMD_FLUSH	1
1.3	printer.device/CMD_INVALID	1
1.4	printer.device/CMD_RESET	2
1.5	printer.device/CMD_START	2
1.6	printer.device/CMD_STOP	2
1.7	printer.device/CMD_WRITE	2
1.8	printer.device/PRD_DUMPRPORT	5
1.9	printer.device/PRD_PRTCOMMAND	17
1.10	printer.device/PRD_QUERY	17
1.11	printer.device/PRD_RAWWRITE	18
1.12	printer.device/PWrite	19

Chapter 1

printer

1.1 printer.doc

CMD_FLUSH	CMD_STOP	PRD_QUERY
CMD_INVALID	CMD_WRITE	PRD_RAWWRITE
CMD_RESET	PRD_DUMPRPORT	PWrite()
CMD_START	PRD_PRTCOMMAND	

1.2 printer.device/CMD_FLUSH

NAME

CMD_FLUSH - abort all I/O requests (immediate)

FUNCTION

CMD_FLUSH aborts all stopped I/O at the unit.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Command	CMD_FLUSH
io_Flags	IOB_QUICK set if quick I/O is possible

1.3 printer.device/CMD_INVALID

NAME

CMD_INVALID - invalid command

FUNCTION

CMD_INVALID is always an invalid command, and sets the device error appropriately.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Command	CMD_INVALID
io_Flags	IOB_QUICK set if quick I/O is possible

1.4 printer.device/CMD_RESET

NAME

CMD_RESET - reset the printer

FUNCTION

CMD_RESET resets the printer device without destroying handles to the open device.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Command	CMD_RESET
io_Flags	IOB_QUICK set if quick I/O is possible

1.5 printer.device/CMD_START

NAME

CMD_START - restart after stop (immediate)

FUNCTION

CMD_START restarts the unit after a stop command.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Command	CMD_START
io_Flags	IOB_QUICK set if quick I/O is possible

1.6 printer.device/CMD_STOP

NAME

CMD_STOP - pause current and queued I/O requests (immediate)

FUNCTION

CMD_STOP pauses all queued requests for the unit, and tries to pause the current I/O request. The only commands that will be subsequently allowed to be performed are immediate I/O requests, which include those to start, flush, and finish the I/O after the stop command.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible
io_Device	preset by the call to OpenDevice
io_Command	CMD_STOP
io_Flags	IOB_QUICK set if quick I/O is possible

1.7 printer.device/CMD_WRITE

NAME

CMD_WRITE -- send output to the printer

FUNCTION

This function causes a buffer of characters to be written to the current printer port (usually parallel or serial). The number of characters is specified in `io_Length`, unless `-1` is used, in which case output is sent until a `0x00` is encountered.

The Printer device, like the Console device, maps ANSI X3.64 style 7-bit printer control codes to the control code set of the current printer. The ANSI codes supported can be found below.

NOTES

Not all printers will support all functions. In particular you may not assume that the MARGINS or TABS can be set. Close to half the supported printers don't fully implement one or the other. If you want the features of margins or tabs you will need to fake it internally by sending out spaces.

Note that the printer device may have already sent out a "set margins" command to the printer. If you are faking your own margins, be sure to cancel the old ones first. (use the "aCAM" command)

Defaults are set up so that if a normal AmigaDOS text file is sent to PRT:, it has the greatest chance of working.

(AmigaDOS text files are defined as follows:)

tabs	- every 8
CR (0x0D)	- moves to start of current line
LF (0x0A)	- moves to start of next line

IO REQUEST

<code>io_Message</code>	<code>mn_ReplyPort</code> set
<code>io_Device</code>	preset by <code>OpenDevice</code>
<code>io_Unit</code>	preset by <code>OpenDevice</code>
<code>io_Command</code>	<code>CMD_WRITE</code>
<code>io_Length</code>	number of characters to process, or if <code>-1</code> , process until <code>0x00</code> encountered
<code>io_Data</code>	pointer to block of data to process

RESULTS

`io_Error` : if `CMD_WRITE` succeeded, then `io_Error` will be zero. Otherwise `io_Error` will be non-zero.

SEE ALSO

`printer.h`, `parallel.device`, `serial.device`, `Preferences`

ANSI X3.64 style COMMANDS

aRIS	ESCc	hard reset	
aRIN	ESC#1	initialize to defaults	
aIND	ESCD	true linefeed (lf)	
aNEL	ESCE	return,lf	
aRI	ESCM	reverse lf	*

aSGR0	ESC[0m	normal character set	
aSGR3	ESC[3m	italics on	
aSGR23	ESC[23m	italics off	
aSGR4	ESC[4m	underline on	
aSGR24	ESC[24m	underline off	
aSGR1	ESC[1m	boldface on	
aSGR22	ESC[22m	boldface off	
aSFC	SGR30-39	set foreground color	
aSBC	SGR40-49	set background color	
aSHORP0	ESC[0w	normal pitch	
aSHORP2	ESC[2w	elite on	
aSHORP1	ESC[1w	elite off	
aSHORP4	ESC[4w	condensed on	
aSHORP3	ESC[3w	condensed off	
aSHORP6	ESC[6w	enlarged on	
aSHORP5	ESC[5w	enlarged off	
aDEN6	ESC[6"z	shadow print on	
aDEN5	ESC[5"z	shadow print off	
aDEN4	ESC[4"z	doublestrike on	
aDEN3	ESC[3"z	doublestrike off	
aDEN2	ESC[2"z	Near Letter Quality (NLQ) on	
aDEN1	ESC[1"z	NLQ off	
aSUS2	ESC[2v	superscript on	
aSUS1	ESC[1v	superscript off	
aSUS4	ESC[4v	subscript on	
aSUS3	ESC[3v	subscript off	
aSUS0	ESC[0v	normalize the line	*
aPLU	ESCL	partial line up	*
aPLD	ESCK	partial line down	*
aFNT0	ESC(B	US char set (default)	or Font 0
aFNT1	ESC(R	French char set	or Font 1
aFNT2	ESC(K	German char set	or Font 2
aFNT3	ESC(A	UK char set	or Font 3
aFNT4	ESC(E	Danish I char set	or Font 4
aFNT5	ESC(H	Sweden char set	or Font 5
aFNT6	ESC(Y	Italian char set	or Font 6
aFNT7	ESC(Z	Spanish char set	or Font 7
aFNT8	ESC(J	Japanese char set	or Font 8
aFNT9	ESC(6	Norwegian char set	or Font 9
aFNT10	ESC(C	Danish II char set	or Font 10
aPROP2	ESC[2p	proportional on	*
aPROP1	ESC[1p	proportional off	*
aPROP0	ESC[0p	proportional clear	*
aTSS	ESC[n E	set proportional offset	*
aJFY5	ESC[5 F	auto left justify	*
aJFY7	ESC[7 F	auto right justify	*
aJFY6	ESC[6 F	auto full justify	*
aJFY0	ESC[0 F	auto justify off	*
aJFY3	ESC[3 F	letter space (justify)	*
aJFY1	ESC[1 F	word fill(auto center)	*

aVERP0	ESC[0z	1/8" line spacing	
aVERP1	ESC[1z	1/6" line spacing	
aSLPP	ESC[nt	set form length n	
aPERF	ESC[nq	set perforation skip to n lines (n>0)	
aPERF0	ESC[0q	perforation skip off	
aLMS	ESC#9	Left margin set	*
aRMS	ESC#0	Right margin set	*
aTMS	ESC#8	Top margin set	*
aBMS	ESC#2	Bottom margin set	*
aSTBM	ESC[Pn1;Pn2r	set T&B margins	*
aSLRM	ESC[Pn1;Pn2s	set L&R margin	*
aCAM	ESC#3	Clear margins	
aHTS	ESCH	Set horiz tab	*
aVTS	ESCJ	Set vertical tabs	*
aTBC0	ESC[0g	Clr horiz tab	*
aTBC3	ESC[3g	Clear all h tab	*
aTBC1	ESC[1g	Clr vertical tabs	*
aTBC4	ESC[4g	Clr all v tabs	*
aTBCALL	ESC#4	Clr all h & v tabs	*
aTBSALL	ESC#5	Set default tabs (every 8)	
aEXTEND	ESC[Pn"x	Extended commands	
		This is a mechanism for printer drivers to support extra commands which can be called by ANSI control sequences	
aRAW	ESC[Pn"r	Next 'Pn' chars are raw (ie. they are not parsed by the printer device, instead they are sent directly to the printer.	

(*) indicates that sending this command may cause unexpected results on a large number of printers.

1.8 printer.device/PRD_DUMPRPORT

NAME

PRD_DUMPRPORT - dump the specified RastPort to a graphics printer.

FUNCTION

Print a rendition of the supplied RastPort, using the supplied ColorMap, position and scaling information, as specified in the printer preferences.

IO REQUEST

io_Message	mn_ReplyPort set if quick I/O is not possible.
io_Command	PRD_DUMPRPORT.
io_Flags	IOB_QUICK set if quick I/O is possible.
io_RastPort	ptr to a RastPort.
io_ColorMap	ptr to a ColorMap.
io_Modes	the 'modes' flag from a ViewPort structure, (the upper word is reserved and should be zero).
io_SrcX	x offset into the RastPort to start printing from.
io_SrcY	y offset into the RastPort to start printing from.

io_SrcWidth width of the RastPort to print (from io_SrcX).
io_SrcHeight height of the RastPort to print (from io_SrcY).
io_DestCols width of the printout in printer pixels.
io_DestRows height of the printout in printer pixels.
io_Special flag bits
 (some of which pertain to DestCols and DestRows).
-if SPECIAL_MIL is set, then the associated
 parameter is specified in thousandths of
 an inch on the printer. ie. if DestCols = 8000,
 DestRows = 10500 and SPECIAL_MILROWS and
 SPECIAL_MILCOLS is set then the printout would be
 8.000 x 10.500 inches.
-if SPECIAL_FULL is set, then the specific dimension
 is set to the maximum possible as determined
 by the printer limits or the configuration
 limits; whichever is less.
-if SPECIAL_FRAC is set, the parameter is
 taken to be a longword binary fraction
 of the maximum for that dimension.
-if all bits for a dimension are clear,
 (ie. SPECIAL_MIL/FULL/FRAC and ASPECT are NOT set)
 then the parameter is specified in printer pixels.
-if SPECIAL_CENTER is set then the image will be
 put between the left and right edge of the paper.
-if SPECIAL_ASPECT is set, one of the dimensions
 may be reduced/expanded to preserve the aspect
 ratio of the print.
-SPECIAL_DENSITY(1-7) this allows for a maximum of 7
 different print densities. DENSITY1 is the lowest
 density and the default.
-SPECIAL_NOFORMFEED - this allows for the mixing of
 text and graphics or multiple graphic dumps on page
 oriented printers (usually laser jet printers).
 When this flag is set the page will not be ejected
 after a graphic dump. If you perform another
 graphic dump without this flag set OR close the
 printer after printing text after a graphic dump,
 the page will be ejected.
-if SPECIAL_TRUSTME is set then the printer specific
 driver is instructed to not issue a reset command
 before and after the dump. If this flag is NOT
 checked by the printer specific driver then setting
 this flag has no effect. Since we now recommend
 that printer driver writers no longer issue a reset
 command it is probably a safe idea to always set
 this flag when calling for a dump.
-if SPECIAL_NOPRINT is set then the following is done:
 Compute print size, set 'io_DestCols' and
 'io_DestRows' in the calling program's 'IODRPreq'
 structure and exit, DON'T PRINT. This allows the
 calling program to see what the final print size
 would be in printer pixels. Note that it modifies
 the 'io_DestCols' and 'io_DestRows' fields of your
 'IODRPreq' structure. It also sets the print
 density and updates the 'MaxXDots', 'MaxYDots',
 'XDotsInch', and 'YDotsInch' fields of the
 'PrinterExtendedData' structure.

There following rules for the interpretation of `io_DestRows` and `io_DestCols` that may produce unexpected results when they are not greater than zero and `io_Special` is zero. They have been retained for compatability. The user will not trigger these other rules with well formed usage of `io_Special`.

When `io_Special` is equal to 0, the following rules (from the V1.1 `printer.device`, and retained for compatibility reasons) take effect. Remember, these special rules are for `io_DestRows` and `io_DestCols` and only take effect if `io_Special` is 0).

- a) `DestCols>0 & DestRows>0` - use as absolute values.
ie. `DestCols=320 & DestRows=200` means that the picture will appear on the printer as 320x200 dots.
- b) `DestCols=0 & DestRows>0` - use the printers maximum number of columns and print `DestRows` lines. ie. if `DestCols=0` and `DestRows=200` than the picture will appear on the printer as wide as it can be and 200 dots high.
- c) `DestCols=0 & DestRows=0` - same as above except the driver determines the proper number of lines to print based on the aspect ratio of the printer. ie. This results in the largest picture possible that is not distorted or inverted.
Note: As of this writing, this is the call made by such program as `DeluxePaint`, `GraphicCraft`, and `AegisImages`.
- d) `DestCols>0 & DestRows=0` - use the specified width and the driver determines the proper number of lines to print based on the aspect ratio of the printer. ie. if you desire a picture that is 500 pixels wide and aspect ratio correct, use `DestCols=500` and `DestRows=0`.
- e) `DestCols<0` or `DestRows>0` - the final picture is either a reduction or expansion based on the fraction $|DestCols| / DestRows$ in the proper aspect ratio.
Some examples:
 - 1) if `DestCols=-2 & DestRows=1` then the printed picture will be 2x the AMIGA picture and in the proper aspect ratio.
(2x is derived from $|-2| / 1$ which gives 2.0)
 - 2) if `DestCols=-1 & DestRows=2` then the printed picture will be 1/2x the AMIGA picture in the proper aspect ratio.
(1/2x is derived from $|-1| / 2$ which gives 0.5)

NOTES

The printer selected in preferences must have graphics capability to use this command. The error '`PDERR_NOTGRAPHICS`' is returned if the printer can not print graphics.

Color printers may not be able to print black and white or greyscale pictures -- specifically, the Okimate 20 cannot print these with a color ribbon: you must use a black ribbon instead. If the printer has an input buffer option, use it. If the printer can be uni or bi directional, select uni-directional; this produces a much cleaner picture. Most printer drivers will attempt to set unidirectional printing if it is possible under software control.

Please note that the width and height of the printable area on

the printer is in terms of pixels and bounded by the following:

- a) $WIDTH = (RIGHT_MARGIN - LEFT_MARGIN + 1) / CHARACTERS_PER_INCH$
- b) $HEIGHT = LENGTH / LINES_PER_INCH$

Margins are set by preferences.

For BGR printer support, the YMC values in the printer specific render.c functions equate to BGR respectively, ie. yellow is blue, magenta is green, and cyan is red.

Data Structures

The printer specific and non-specific data structures can be read ONCE you have opened the printer device. Here is a code fragment to illustrate how to do just that.

```
#include <exec/types.h>
#include <devices/printer.h>
#include <devices/prtbase.h>
#include <devices/prtgfx.h>

struct IODRPREq PReq;
struct PrinterData *PD;
struct PrinterExtendedData *PED;

open the printer device / if it opened...
if (OpenDevice("printer.device", 0, &PReq, 0) == NULL) {
    get pointer to printer data
    PD = (struct PrinterData *)PReq.io_Device;
    get pointer to printer extended data
    PED = &PD->pd_SegmentData->ps_PED;
    let's see what's there
    printf("PrinterName = '%s', Version=%u, Revision=%u\n",
        PED->ped_PrinterName, PD->pd_SegmentData->ps_Version,
        PD->pd_SegmentData->ps_Revision,);
    printf("PrinterClass=%u, ColorClass=%u\n",
        PED->ped_PrinterClass, PED->ped_ColorClass);
    printf("MaxColumns=%u, NumCharSets=%u, NumRows=%u\n",
        PED->ped_MaxColumns, PED->ped_NumCharSets, PED->ped_NumRows);
    printf("MaxXDots=%lu, MaxYDots=%lu, XDotsInch=%u, YDotsInch=%u\n",
        PED->ped_MaxXDots, PED->ped_MaxYDots,
        PED->ped_XDotsInch, PED->ped_YDotsInch);
    CloseDevice(&PReq);
}
```

Preferences

If you want the user to be able to access the printer preferences items without having to run preferences (like DPAINT II's printer requestor), here is what you do. You can look at the printer's copy of preferences by referring to 'PD->pd_Preferences' (the printer device MUST already be opened at this point). After you have this you could put up a requestor and allow the user to change whatever parameters they wanted. BEAR IN MIND THAT YOU ARE RESPONSIBLE FOR RANGE CHECKING THESE SELECTIONS! Listed below are the printer preferences items and their valid values.

```
PrintPitch      - PICA, ELITE, FINE.
PrintQuality    - DRAFT, LETTER.
```

```

PrintSpacing      - SIX_LPI, EIGHT_LPI.
PrintLeftMargin   - 1 to PrintRightMargin.
PrintRightMargin  - PrintLeftMargin to 999.
PaperLength       - 1 to 999.
PrintImage        - IMAGE_POSITIVE, IMAGE_NEGATIVE.
PrintAspect       - ASPECT_HORIZ, ASPECT_VERT.
PrintShade        - SHADE_BW, SHADE_GREYSCALE, SHADE_COLOR.
PrintThreshold    - 1 to 15.
PrintFlags        - CORRECT_RED, CORRECT_GREEN, CORRECT_BLUE, CENTER_IMAGE,
                   IGNORE_DIMENSIONS, BOUNDED_DIMENSIONS,
                   ABSOLUTE_DIMENSIONS, PIXEL_DIMENSIONS,
                   MULTIPLY_DIMENSIONS, INTEGER_SCALING,
                   ORDERED_DITHERING, HALFTONE_DITHERING,
                   FLOYD_DITHERING, ANTI_ALIAS, GREY_SCALE2
PrintMaxWidth     - 0 to 65535.
PrintMaxHeight    - 0 to 65535.
PrintDensity      - 1 to 7.
PrintXOffset      - 0 to 255.

```

Asynchronous I/O

The recommended way to do asynchronous i/o is...

a) To send requests for i/o.

```

struct IORequest *ioreq;
struct MsgPort *port;
UBYTE signal;

port = ioreq->io_Message.mn_ReplyPort;
signal = port->mp_SigBit;

SendIO(ioreq);  send request
Wait(signal);  wait for completion (go to sleep)
while ((Msg = GetMsg(port)) != NULL) {  get ALL messages
}

```

b) To abort a previous request for i/o.

```

struct IORequest *ioreq;

AbortIO(ioreq);  abort request
WaitIO(ioreq);  wait for reply

at this point you can re-use 'ioreq'.

```

Note that in the above examples 'ioreq' could be any one of...

- a) struct IOStdReq a standard i/o request
- b) struct IODRPRReq a dumphrport i/o request
- c) struct IOPrtCmdReq a printer command i/o request

It is recommend that you do asynchronous i/o in your programs and give the user a way of aborting all requests.

In general densities which use more than one pass should only be used for B&W shade dumps. They can be used for Grey-Scale or Color Shade dumps BUT the output may tend to look muddy or dark. Also multiple pass Color dumps tend to dirty or smear the ribbon (ie. yellow will get contaminated with the other colors on the ribbon; you've been warned).

Alphacom_AlphaPro_101

-
1. Daisywheel printer (text only).

Brother_HR-15XL

-
1. Daisywheel printer (text only).

CalComp_ColorMaster

-
1. Thermal transfer b&w/color printer (text and graphics).
 2. Use Black ribbon for non-color dumps; Color ribbon for color dumps.
 3. Linefeeds # of vertical dots printed.
 4. Densitie(s) supported are 203x200(1) dpi.
 5. This is a dual printer driver. Select a PaperSize of 'Narrow Tractor' for use with the ColorMaster; 'Wide Tractor' for use with the ColorView-5912 (which uses 11 x 17 inch paper).

CalComp_ColorMaster2

-
1. Thermal transfer b&w/color printer (text and graphics).
 2. Use Black ribbon for non-color dumps; Color ribbon for color dumps.
 3. Linefeeds # of vertical dots printed.
 4. Densitie(s) supported are 203x200(1) dpi.
 5. This is a dual printer driver. Select a PaperSize of 'Narrow Tractor' for use with the ColorMaster; 'Wide Tractor' for use with the ColorView-5912 (which uses 11 x 17 inch paper).
 6. This driver is the same as the Calcomp_ColorMaster driver EXCEPT it is approximately 2 times faster (during color dumps) and requires LOTS of memory (up to 1,272,003 bytes for a full 8 x 10 inch (1600 x 2000 dot) color dump). Typically full-size (color) dumps are 1600 x 1149 dots and require 730,767 bytes. Memory requirements for the ColorView-5912 are up to 2,572,803 bytes for a full 10 x 16 inch (2048 x 3200 dot) color dump. Typically full-size (color) dumps are 2048 x 2155 dots and require 1,732,623 bytes. The memory requirements are 1/3 when doing a non-color printout (on both the ColorMaster and ColorView).

Canon_PJ-1080A

-
1. Ink jet b&w/color printer (text and graphics).
 2. Linefeeds # of vertical dots printed.
 3. Densitie(s) supported are 83x84(1) dpi.

CBM_MPS1000

-
1. Dot matrix b&w printer (text and graphics).
 2. Linefeeds # of vertical dots printed (-1/3 dot if PaperType = Single). *2
 3. Density XDPI YDPI XYDPI Comments

1	120	72	8640	
2	120	144	17280	two pass

3	240	72	17280		*1
4	120	216	25920	three pass	
5	240	144	34560	two pass	*1
6	240	216	51840	three pass	*1
7	same as 6				

Diablo_630

1. Daisywheel printer (text only).

Diablo_Advantage_D25

1. Daisywheel printer (text only).

Diablo_C-150

1. Ink jet b&w/color printer (text and graphics).
2. Always linefeeds 4 dots (limitation of printer).
3. A PaperSize of 'Wide Tractor' selects a maximum print width of 8.5 inches (for wide roll paper).
5. Densitie(s) supported are 120x120(1) dpi.

EpsonQ (24-pin Epson compatible)

1. Dot matrix b&w/color printer (text and graphics).
2. Drives all EpsonQ (LQ1500, LQ2500, etc.) compatible printers.
3. Linefeeds # of vertical dots printed.
4. Density

	XDPI	YDPI	XYDPI	Comments
1	90	180	16200	
2	120	180	21600	
3	180	180	32400	
4	360	180	64800	*1
5,6,7	same as 4			
5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.6 inches (for wide carriage printers).
6. A PaperType of 'Single' uses only 16 of the 24 pins, whereas a PaperType of 'Fanfold' uses all 24 pins. The 'Single' option is useful for those printers which have a weak power supply and cannot drive all 24 pins continuously. If during a single pass of the print head you notice that the top two thirds of the graphics are darker than the bottom one third then you'll probably need to drop down to 16 pins.

EpsonX[CBM_MPS-1250] (8/9-pin Epson compatible)

1. Dot matrix b&w/color printer (text and graphics).
2. Drives all EpsonX (EX/FX/JX/LX/MX/RX, etc.) compatible printers.
3. Linefeeds # of vertical dots printed (-1/3 dot if PaperType = Single). *2
4. Density

	XDPI	YDPI	XYDPI	Comments
1	120	72	8640	
2	120	144	17280	two pass
3	240	72	17280	*1
4	120	216	25920	three pass
5	240	144	34560	two pass
6	240	216	51840	three pass
7	same as 6			

5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.6 inches (for wide carriage printers).
6. Use this driver if you own a CBM_MPS-1250 (as it is EpsonX compatible).

EpsonXOld (8/9-pin Epson compatible)

1. Dot matrix b&w printer (text and graphics).
2. Drives all very old EpsonX (EX/FX/JX/LX/MX/RX, etc.) compatible printers.
3. Linefeeds # of vertical dots printed.
4. Density XDPI YDPI XYDPI Comments

1	60	72	4320	
2	120	72	8640	(double speed) *1
3	120	72	8640	
4	240	72	17280	*1
5	120	72	8640	(for use on old Star printers)
6	240	72	17280	(for use on old Star printers) *1
7	240	72	17280	(same as density 4) *1

5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.6 inches (for wide carriage printers).
6. Use this driver if the EpsonX driver doesn't work properly in graphics or text mode on your EpsonX compatible printer.

generic

1. Text only printer.

Howtek_Pixelmaster

1. Plastic ink jet b&w/color printer (text and graphics).
2. Linefeeds # of vertical dots printed.
3. Density XDPI YDPI XYDPI Comments

1	80	80	6400	
2	120	120	14400	
3	160	160	25600	
4	240	240	57600	
5,6,7	same as 4			

4. Maximum print area is 8.0 x 10.0 inches.

HP_DeskJet

1. Ink jet non-color printer (text and graphics).
2. Linefeeds # of vertical dots printed.
3. Density XDPI YDPI XYDPI Comments

1	75	75	5625	
2	100	100	10000	
3	150	150	22500	
4	300	300	90000	
5,6,7	same as 4			

4. Maximum print area is 8.0 x 10.0 inches.

HP_LaserJet (LaserJet+/LaserJetIII compatible)

1. Laser engine non-color printer (text and graphics).
2. Linefeeds # of vertical dots printed.

3. Density	XDPI	YDPI	XYDPI	Comments
1	75	75	5625	
2	100	100	10000	
3	150	150	22500	
4	300	300	90000	
5,6,7	same as 4			

4. Maximum print area is 8.0 x 10.0 inches.

HP_PaintJet

1. Ink jet b&w/color printer (text and graphics).
2. Linefeeds # of vertical dots printed.
3. Densitie(s) supported are 180x180(1) dpi.

HP_ThinkJet

1. Ink jet non-color printer (text and graphics).
2. Linefeeds # of vertical dots printed.
3. Density

XDPI	YDPI	XYDPI	Comments
1	96	96	9216
2	192	96	18432
3,4,5,6,7	same as 4		

Imagewriter II (Imagewriter compatible)

1. Dot matrix b&w/color printer (text and graphics).
2. Linefeeds # of vertical dots printed.
3. Density

XDPI	YDPI	XYDPI	Comments
1	80	72	5760
2	120	72	8640
3	144	72	10368
4	160	72	11520
5	120	144	17280 two pass
6	144	144	20736 two pass
7	160	144	23040 two pass

Nec_Pinwriter (24-wire Pinwriter compatible (P5/P6/P7/P9/P2200))

1. Dot matrix b&w/color printer (text and graphics).
2. Drives all Nec 24-wire Pinwriter compatible printers.
3. Linefeeds # of vertical dots printed.
4. Density

XDPI	YDPI	XYDPI	Comments
1	90	180	16200
2	120	180	21600
3	180	180	32400
4	120	360	43200 two pass
5	180	360	64800 two pass
6	360	180	64800
7	360	360	129600 two pass

5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.6 inches (for wide carriage printers).

Okidata_92

1. Dot matrix non-color printer (text and graphics).

2. Always linefeeds 7/72 inch (limitation of printer in graphics mode).
3. Densitie(s) supported are 72x72 dpi.

Okidata_293I

-
1. Dot matrix b&w/color printer (text and graphics).
 2. Drives 292 or 293 using the IBM interface module.
 3. Linefeeds # of vertical dots printed (-1/2 dot if PaperType = Single) *3
 4. Density XDPI YDPI XYDPI Comments

1	120	144	17280	
2	240	144	34560	
3	120	288	34560	two pass
4	240	288	69120	two pass
5,6,7	same as 4			
 5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.6 inches (for wide carriage printers).

Okimate-20

-
1. Thermal transfer b&w/color printer (text and graphics).
 2. Use Black ribbon for non-color dumps; Color ribbon for color dumps.
 3. Linefeeds an even # of dots printed. (ie. if 3 printed, 4 advanced).
 4. Densitie(s) supported are 120x144(1) dpi.

Quadram_QuadJet

-
1. Ink jet b&w/color printer (text and graphics).
 2. Linefeeds # of vertical dots printed.
 3. Densitie(s) supported are 83x84(1) dpi.

Qume_LetterPro_20

-
1. Daisywheel printer (text only).

Seiko_5300

-
1. Thermal transfer b&w/color printer (graphics only).
 2. Use Black ribbon for non-color dumps; Color ribbon for color dumps.
 3. Density XDPI YDPI XYDPI Comments

1	152	152	23104	drives CH-5301 printer
2	203	203	41209	drives CH-5312 printer
3	240	240	57600	drives CH-5303 printer
4, 5,6,7	same as 3			

You must select the proper density to drive the specific printer that you have.

4. This driver is not on the V1.3 Workbench or Extras disk. It is available on BIX and directly from Seiko.

Seiko_5300a

-
1. Thermal transfer b&w/color printer (graphics only).
 2. Use Black ribbon for non-color dumps; Color ribbon for color dumps.
 3. Density XDPI YDPI XYDPI Comments

1	152	152	23104	drives CH-5301 printer
2	203	203	41209	drives CH-5312 printer
-

3	240	240	57600	drives CH-5303 printer
4, 5, 6, 7		same as 3		

You must select the proper density to drive the specific printer that you have.

4. This driver is the same as the Seiko_5300 driver EXCEPT it is approximately 2 times faster (during color dumps) and requires LOTS of memory (up to 1,564,569 bytes for a full 8 x 10 inch (1927 x 2173 dot) color dump). Typically full-size (color) dumps are 1927 x 1248 dots and require 898,569 bytes. The memory requirements are 1/3 when doing a non-color printout.
5. This driver is not on the V1.3 Workbench or Extras disk. It is available on BIX and directly from Seiko.

Tektronix_4693D

1. Thermal transfer b&w/color printer (graphics only).
2. Densitie(s) supported are 300x300(1) dpi
3. Due to the way the printer images a picture none of the printer preferences options affect the printout with the following exceptions:
 - a) Aspect - Horizontal, Vertical
 - b) Shade - B&W, Grey_Scale, Color
 ...as a result of this only full size pictures can be printed.
4. Keypad menu option 3b COLOR ADJUSTMENT may be set from the keypad. For normal prints this option should be set to "do not adjust".
5. Keypad menu option 3d VIDEO COLOR CORRECTION may be set from the keypad. For normal prints this option should be set to "do not adjust".
6. Keypad menu option 5 BACKGROUND COLOR EXCHANGE may be set from the keypad. For normal prints this option should be set to "print colors as recieved".
7. Once a picture has been printed additional copies may be printed whithout resending by using the printers keypad.
8. This driver is not on the V1.3 Workbench or Extras disk. It is available on BIX and directly from Tektronix.

Tektronix_4696

1. Ink jet b&w/color printer (text and graphics).
 2. Always linefeeds 4 dots (limitation of printer).
 3. Densities supported are 121x120(1), 242x120(black)(2) and 242x120(color)(3).

Selecting a density of 2 or higher really doesn't give you true 242 dpi resolution since the printer only has 121 x dots per inch. Instead this mode tells the printer to go into it's double pass mode. Here, it outputs a line of dots at 121 dpi; and outputs the line again (shifted to the right by 1/242 of an inch). This produces much more vibrate colors and gives the illusion of more resolution. One drawback is that large areas of solid colors (red, green, and blue specifically) tend to over-saturate the paper with ink. Density1 outputs all colors in one pass. Density 2 does a double pass on black. Density 3 does a double pass on all colors. Density 1 to 3 correspond to the printer's graphics printing modes 1 to 3 (respectively).
 4. This driver is not on the V1.3 Workbench or Extras disk. It is available on BIX and directly from Tektronix.
 5. A PaperSize of 'Wide Tractor' selects a maximum print width of 9.0 inches (for wide roll paper).
-

Toshiba_P351C (24-pin Toshiba compatible)

1. Dot matrix b&w/color printer (text and graphics).
2. Drives all Toshiba_P351C compatible printers.
3. Linefeeds # of vertical dots printed.
4. Density XDPI YDPI XYDPI Comments

1	180	180	32400	
2	360	180	64800	
3,4,5,6,7		same as 2		
5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.5 inches (for wide carriage printers).

Toshiba_P351SX (24-pin Toshiba compatible)

1. Dot matrix b&w/color printer (text and graphics).
2. Drives all Toshiba_P351SX (321SL, 321SLC, 341SL) compatible printers.
3. Linefeeds # of vertical dots printed.
4. Density XDPI YDPI XYDPI Comments

1	180	180	32400	
2	360	180	64800	
3	180	360	64800	two pass
4	360	360	129600	two pass
5,6,7		same as 4		
5. A PaperSize of 'Wide Tractor' selects a maximum print width of 13.5 inches (for wide carriage printers).

Xerox_4020

1. Ink jet b&w/color printer (text and graphics).
2. Always linefeeds 4 dots (limitation of printer).
3. This driver is IDENTICAL to the Diablo_C-150 driver EXCEPT it outputs all black dots TWICE. This is a special feature of this printer and produces much more solid, darker black shades. Please note that some printing time overhead results from this feature; if you don't want it use the Diablo_C-150 driver.
4. Densities supported are 121x120(1) and 242x240(2) dpi.
 Selecting a density of 2 or higher really doesn't give you true 240 dpi resolution since the Xerox_4020 only has 121 x dots per inch.
 Instead this mode tells the printer to go into it's pseudo 240 dpi mode. Here, it outputs a line of dots at 121 dpi; moves the paper up 1/240 of an inch and outputs the line again (shifted to the right by 1/240 of an inch). This produces much more vibrant colors and gives the illusion of more resolution. One drawback is that large areas of solid colors (red, green, and blue specifically) tend to over-saturate the paper with ink.
5. A PaperSize of 'Wide Tractor' selects a maximum print width of 9.0 inches (for wide roll paper).

Notes

- *0 - on most printers friction fed paper tends to produce better looking (ie. less horizontal banding) graphic dumps than tractor fed paper.
- *1 - in this mode the printer cannot print two consecutive dots in a row.

It is recommended that you only use this density for B&W Shade dumps.

- *2 - only when 72 YDPI is selected. This option is useful if you notice tiny white horizontal strips in your printout.
- *3 - only when 144 YDPI is selected. This option is useful if you notice tiny white horizontal strips in your printout.

1.9 printer.device/PRD_PRTCOMMAND

NAME

PCPRD_PRTCOMMAND -- send a command to the printer

FUNCTION

This function sends a command to either the parallel or serial device. The printer device maps this command to the control code set of the current printer. The commands supported can be found with the printer.device/Write command. All printers may not support all functions.

IO REQUEST IOPrtCmdReq

io_Message	mn_ReplyPort set
io_Device	preset by OpenDevice
io_Unit	preset by OpenDevice
io_Command	PRD_PRTCOMMAND
io_PrtCommand	the actual command number
io_Parm0	parameter for the command
io_Parm1	parameter for the command
io_Parm2	parameter for the command
io_Parm3	parameter for the command

RESULTS

Errors: if the PRD_PRTCOMMAND succeeded, then io_Error will be zero. Otherwise io_Error will be non-zero. An error of -1 indicates that the command is not supported by the current printer driver. This could be used to check if the connected printer supports a particular command (*italics for example*).

SEE ALSO

printer.device/Write printer.h, parallel.device, Preferences

1.10 printer.device/PRD_QUERY

NAME

PRD_QUERY - query printer port/line status

FUNCTION

This command returns the status of the printer port's lines and registers. Since the printer port uses either the serial or parallel port for i/o, the actual status returned is either the serial or parallel port's status.

IO REQUEST

io_Message mn_ReplyPort set if quick I/O is not possible
 io_Device preset by the call to OpenDevice
 io_Command PRD_QUERY
 io_Data ptr to 2 UBYTES where result will be stored.

RESULTS

io_Data	BIT	ACTIVE	FUNCTION (SERIAL DEVICE)
LSB	0	low	reserved
	1	low	reserved
	2	low	reserved
	3	low	Data Set Ready
	4	low	Clear To Send
	5	low	Carrier Detect
	6	low	Ready To Send
MSB	7	low	Data Terminal Ready
	8	high	read buffer overflow
	9	high	break sent (most recent output)
	10	high	break received (as latest input)
	11	high	transmit x-OFFed
	12	high	receive x-OFFed
	13-15		reserved

io_Data	BIT	ACTIVE	FUNCTION (PARALLEL DEVICE)
	0	hi	printer busy (offline)
	1	hi	paper out
	2	hi	printer selected
			(WARNING: the bit 2 line is also connected to the serial port's ring indicator pin on the A500 and A2000)
	3-7		reserved

io_Actual 1-parallel, 2-serial

1.11 printer.device/PRD_RAWWRITE

NAME

PRD_RAWWRITE - transparent write command

FUNCTION

This is a non standard write command that performs no processing on the data passed to it.

IO REQUEST

io_Message mn_ReplyPort set if quick I/O is not possible
 io_Command PRD_RAWWRITE
 io_Flags IOB_QUICK set if quick I/O is possible
 io_Length the number of bytes in io_Data
 io_Data the raw bytes to write to the printer

1.12 printer.device/PWrite

NAME

PWrite - internal write to printer port

SYNOPSIS

```
error = (*PrinterData->pd_PWrite)(buffer, length);  
D0                                A0      D0
```

FUNCTION

PWrite writes 'length' bytes directly to the printer. This function is generally called by printer drivers to send their buffer(s) to the printer.

This function is accessed by referencing off the PrinterData (PD) structure. Below is a code fragment to show how to get access to a pointer to the PrinterData (PD) structure.

```
#include <exec/types.h>  
#include <devices/printer.h>  
#include <devices/prtbase.h>  
  
struct IODRPREq PReq;  
struct PrinterData *PD;  
struct PrinterExtendedData *PED;  
  
/* open the printer device (any version); if it opened... */  
if (OpenDevice("printer.device", 0, &PReq, 0) == NULL) {  
  
    /* get pointer to printer data strcuture */  
    PD = (struct PrinterData *)PReq.io_Device;  
  
    /* write something directly to the printer */  
    (*PD->pd_PWrite) ("Hello world\n", 12);  
  
    CloseDevice(&PReq); /* close the printer device */  
}
```