APS Linux Driver

Installation Manual

1. Table of contents

1.Table of contents..........2
2.Revision history..........3
3.Installing the driver..........4
3.1.Installing a printer..........4
4.How to use..........6
4.1.APS CUPS driver..........6
4.1.1.Postscript printing..........6
4.1.2.Plain text printing..........6
4.1.3.Plain text printing using text processor..........7
4.1.4.Driver options..........8
4.1.5.Ticket and page modes..........10
4.2.APS library..........11

http://www.aps-printers.com/

A.P.S. reserves the right to make changes without notice to the product to improve reliability, function or design.

A.P.S. does not assume any liability arising out of the application or use of the product or circuit described herein.

2. Revision history

Rev.	Date	Page	Revision item
A	09/03/2006	-	First issue
B	21/04/2006	-	Applies to driver rev. 0.2 and above
C	16/06/2006	-	Applies to driver rev. 0.3 and above
D	12/10/2006	-	Applies to driver rev. 0.4 and above
E	22/02/2008	-	Applies to driver rev. 0.6 and above Added checkneop CUPS option
F	09/06/2008	-	Added inter-character spacing option Added line spacing option

3. Installing the driver

The APS Linux Driver is distributed in source form. Each release comes as a single archive, with *x.y* being the release number of the package.

```
aps_linux_driver-x.y.tar.gz
```

Installation of the APS Linux Driver requires compilation of its various components. The following procedure unpacks the archive into /tmp directory and then compiles and installs from it. Final installation must be done as superuser.

```
$ tar xvzf aps_linux_driver-x.y.tar.gz /tmp
$ cd /tmp/aps_linux_driver-x.y/src
$ make
$ su
[enter superuser password]
# make install
```

The CUPS subsystem must then be restarted to take into account the new printer models installed. This step depends on the Linux distribution, but it is generally a matter of typing a command like the following (as superuser).

```
# /etc/init.d/cups restart
```

3.1. Installing a printer

During start and restart, the CUPS subsystem uses the APS Linux Driver to automatically search for connected printers. This step is done only once when CUPS starts, so any printers connected later will not be detected unless the CUPS subsystem is restarted.

To add a printer, one may use the CUPS web interface at http://localhost:631 and follow the instructions. A printer can also be added from the command line by directly specifying printer model and URI.

```
$ lpadmin -p CP324MRS -E \
         -v aps:/dev/ttyS0?type=serial+baudrate=9600+handshake=rtscts \
         -m cp324mrs.ppd
```

This creates a printer named `CP324MRS` using model definition file `cp324mrs.ppd`. This printer is a serial printer connected to device `/dev/ttyS0`, which default configuration is 9600 bauds, hardware RTS/CTS handshaking.

```
$ lpadmin -p CP290MRS -E -v aps:/dev/parport0?type=parallel -m cp290mrs.ppd
```

This creates a printer named `CP290MRS` using model definition file `cp290mrs.ppd`. This printer is a parallel printer connected to device `/dev/parport0`.

```
$ lpadmin -p CP324HRS -E -v aps:/dev/usb/lp0?type=usb -m cp324hrs.ppd
```

This creates a printer named `CP324HRS` using model definition file `cp324hrs.ppd`. This printer is a USB printer connected to device `/dev/usb/lp0`.

Other options exist for each port type. Refer to document `driver_libaps.pdf` for more information about APS printers URI.

4. How to use

The APS Linux Driver is made of two components: a low-level communication library (the APS library), and a printer driver for the CUPS subsystem. The APS CUPS driver uses the APS library to communicate with APS printers. The APS library may also be used as a stand-alone component in user applications to get more control when communicating with printers.

4.1. APS CUPS driver

The APS CUPS driver enables user to print from any application to supported APS printers.

The APS CUPS driver provides filters for the CUPS internal raster format (RIP) and for plain text. The APS CUPS driver may be used in three different ways.

4.1.1. Postscript printing

When printing, most applications generate a file in Postscript format which in turn is rendered by CUPS in its internal raster format. In this context, the APS CUPS driver for RIP is automatically selected by CUPS and the file is printed using graphics printing commands. For example, printing from OpenOffice will generate a Postscript file sent to CUPS and printed by the APS CUPS driver using graphics commands.

This generates a large volume of data to be sent to the printer and might limit the printing speed when using RS232 or parallel ports.

The amount of data can be automatically reduced using the *Optimize printing* CUPS option (turned on by default). This option compresses blank dotlines as well as leading and trailing blank spaces so that only non-blank data is transmitted. This option generally increases printing speed in Postscript printing mode; however it can lead to jerky paper movement in some extreme cases (for example, succession of full black and full white dotlines).

4.1.2. Plain text printing

The APS CUPS driver is also able to print text directly using the internal fonts of the printer. When CUPS detects that the file to be printed is a text-only file, it automatically selects the APS CUPS driver for plain text and skips the intermediate rendering in internal raster format. The following command will print the file plain_text.txt using internal fonts of the printer.

```
$ lp -d APS_printer plain_text.txt
```

Printing in this fashion will give the highest printing speed. However the complexity of tickets is limited because text formatting options cannot be used and graphics cannot be printed.

4.1.3. Plain text printing using text processor

The APS CUPS driver provides a way to circumvent the limitations of plain text printing. When option *Process embedded control codes* is checked, the APS CUPS driver analyses plain text data and substitutes tokens surrounded by < > by their ASCII code equivalents.

The following example shows how to embed text formatting options in a plain text ticket. This example uses the APS command set (MRS, HRS and KCP printer series).

```
Normal printing.<LF><LF><LF>
<TAB>Printed with <ESC>%<1>font number 1.
<ESC>!<0x10>Double-height printing.
```

Following data will be sent to the printer after processing.

```
0000  4e 6f 72 6d 61 6c 20 70  72 69 6e 74 69 6e 67 2e  Normal p rinting.
0010  0a 0a 0a 0a 09 50 72 69  6e 74 65 64 20 77 69 74  .....Pri nted wit
0020  68 20 1b 25 01 66 6f 6e  74 20 6e 75 6d 62 65 72  h .%.fon t number
0030  20 31 2e 0a 1b 21 10 44  6f 75 62 6c 65 2d 68 65   1...!.D ouble-he
0040  69 67 68 74 20 70 72 69  6e 74 69 6e 67 2e 0a     ight pri nting..
```

Tokens can either be a string, a decimal number, an hexadecimal number or an octal number. Invalid tokens are printed without processing.

- `<ESC>` is translated as ASCII code 27;
- `<27>` is translated as ASCII code 27;
- `<0x1b>` is translated as ASCII code 27;
- `<033>` is translated as ASCII code 27.

Token strings include ASCII control code names.

TOKEN STRING	ASCII CODE	TOKEN STRING	ASCII CODE
<NUL>	0	<SOH>	1
<STX>	2	<ETX>	3
<EOT>	4	<ENQ>	5
<ACK>	6	<BEL>	7
<BS>	8	<TAB>	9
<LF>	10	<VT>	11
<FF>	12	<CR>	13
<SO>	14	<SI>	15
<DLE>	16	<DC1>	17
<DC2>	18	<DC3>	19
<DC4>	20	<NAK>	21
<SYN>	22	<ETB>	23
<CAN>	24	<EM>	25
<SUB>	26	<ESC>	27
<FS>	28	<GS>	29
<RS>	30	<US>	31

4.1.4. Driver options

The APS CUPS driver provides several options summarized in table below.

Option field	Description
Serial printing baudrate	Set serial baudrate used during printing. (*serial printers only*)
Serial printing handshaking	Set serial handshaking used during printing. (*serial printers only*)
Parallel printing write mode	Set parallel printing write mode (polling or IRQ based). (*parallel printers only*)
Printing timeout	Printing is automatically cancelled after this timeout elapsed. This value can be set to None to have the driver wait indefinitely.
Dynamic division	Set maximum number of dots that can be fired per strobe. Large dynamic division setting may impact printing speed. This option also affects maximum current consumed by the printer.
Maximum print speed	Set maximum printing speed.
Print intensity	Set printing intensity.
Optimize printing	Optimize graphics printing in Postscript printing mode.
Internal font	Set internal font used while printing plain text.
Process embedded control codes	When checked, process plain text data to interpret embedded control codes.
Final cut mode	Set type of cut to perform after printing ticket.
Enable page mode	When checked, enable page mode.
Forward feed after ticket	Set distance to feed in forward direction after printing ticket.
Backward feed after ticket	Set distance to feed in backward direction after printing ticket.
Maximum ticket length	Set maximum ticket length. Tickets longer than this value are cut in pieces of maximum length using integrated cutter.
Check NEOP status	Enable or disable near end-of-paper (NEOP) status checking.
Inter-character spacing	Set number of pixels between successive text characters
Line spacing	Set number of dotlines between successive text lines

Driver options can be tuned using the CUPS web interface. These can also be modified manually using following command:

```
$ lpoptions -d APS_Printer -o option=value -o option=value ...
```

Table below shows various option names and allowed values.

Option field	Option	Value
Serial printing baudrate	`prbaudrate`	0 = 1200 bauds 1 = 2400 bauds 2 = 4800 bauds 3 = 9600 bauds 4 = 19200 bauds 5 = 38400 bauds 6 = 57600 bauds 7 = 115200 bauds
Serial printing handshaking	`prhandshake`	1 = Software flow control (XON/XOFF) 2 = Hardware flow control (RTS/CTS)
Parallel printing write mode	`parmode`	0 = Polling 1 = IRQ
Printing timeout	`prtimeout`	Timeout in milliseconds. Zero disables timeout.
Dynamic division	`dynadiv`	Maximum number of black bytes per dotline. Zero sets maximum (unlimited) current consumption.
Maximum print speed	`maxspeed`	Print speed in mm/s.
Print intensity	`intensity`	Print intensity in percent.

Optimize printing	`optprint`	False = Optimization disabled
		True = Optimization enabled

Internal font	`font`	0 = APS font 0 1 = APS font 1 2 = APS font 2

Process embedded control codes	`process`	False = Processing disabled
		True = Processing enabled

Final cut mode	`finalcut`	0 = No cut at end of ticket 1 = Partial cut at end of ticket 2 = Full cut at end of ticket

Enable page mode	`pagemode`	False = Ticket mode enabled
		True = Page mode enabled

Forward feed after ticket	`fwdfeed`	Distance to feed in dotlines.
Backward feed after ticket	`backfeed`	Distance to feed in dotlines.

Option field	Option	Value
Maximum ticket length	`maxlength`	Maximum ticket length in dotlines.

Check NEOP status	`checkneop`	False = NEOP status checking disabled
		True = NEOP status checking enabled

Inter-character spacing	`charspacing`	Character spacing in pixels
Line spacing	`linespacing`	Line spacing in dotlines

For example, following command sets maximum printing speed to 50mm/s with a print intensity of +30% on printer CP324HRS.

```
$ lpoptions -d CP324HRS -o maxspeed=50 -o intensity=30
```

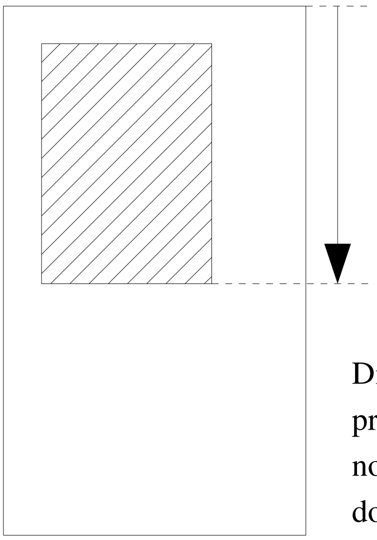
4.1.5. Ticket and page modes

Some applications, including web browsers, generate Postscript printing data entire page by entire page, even if only a portion of the page is non-blank.

In ticket mode, the APS CUPS driver stops printing at the last non-blank dotline, which is generally the intended behaviour of application developpers.

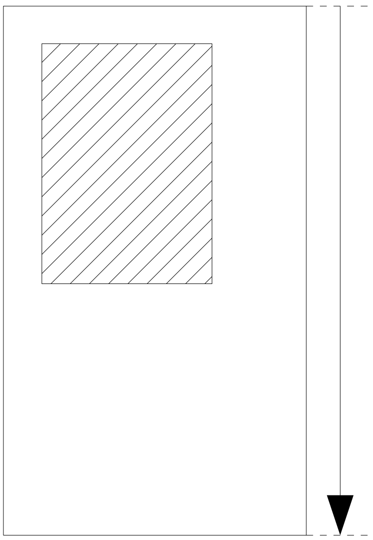
In page mode, the APS CUPS driver prints all data output by the application, including all trailing blank dotlines.

Following drawings illustrate the two behaviours.



Driver stops
printing at last
non-blank
dotline

Ticket mode



Driver prints
full Postscript
page

Page mode

In ticket mode, extra blank dotlines can be printed after the ticket using the *Forward feed after ticket* CUPS driver option.

4.2. APS library

The APS library can be used directly in user applications to get more control when communicating with printers.

The APS library is a C library that provides an easy interface to communicate with APS printers. Using the APS library requires programming.

Refer to document `driver_libaps.pdf` for information about the APS library.