

PyKota's internals

Table of Contents

<u>Chapter 1. PyKota's internals</u>	1
<u>Notes</u>	1

Chapter 1. PyKota's internals

Last modified on \$Date: 2005-03-06 17:52:43 +0100 (dim, 06 mar 2005) \$

To account for pages or eventually ink usage [\[1\]](#) , you must plug your accounting system somewhere into the printing system you use. One way to do this without having to modify the printing system itself, is by using a *filter*.

A filter is a computer program which takes data in one format as its input, and outputs the same data but transformed into another format. CUPS already contains many filters. For example there's one filter named **pstops** which accepts PostScript data as its input, and, as its name implies, outputs PostScript data too, but after having eventually rearranged the pages to fit several pages on a single sheet of paper, or other manipulations like that.

The **pstops** filter described above is also in charge of doing basic page accounting, but PyKota currently doesn't use this facility since it may prove to be unreliable depending on the drivers used or if a paper jam occurs for example.

So to do its own accounting, PyKota has its own filters, for CUPS it's named **cupspykota**, which is in fact a CUPS backends wrapper. The procedure to plug the correct filter into your printing system is described in the chapter.

Currently with CUPS' internal accounting mechanism, the **pstops** filter can be bypassed. That's why PyKota uses its own backend. The **cupspykota** backend wrapper ensures that jobs can't bypass it, so you can use any printer with any driver and any command line option, and you can be sure that your print job will be correctly accounted for.

When using the `hardware` accounting method, PyKota launches the script you specified to ask the printer for its internal page counter at the start and at the end of the print job, and computes the values' difference to know the job's size.

When using the `software` accounting method, the command you specified is launched with the job's data on its standard input. Your command must print the job's size in number of pages on a single line on its standard output. This number is then read by PyKota and used to update the current user's quota information.

If a problem occurs, it is logged either to the filter's standard error or to the system logger, depending on your preferences in PyKota's configuration files. Also if a print quota is reached you may choose if the administrator, the user, both or no-one will receive an email message explaining the situation and proposing a solution. You can even configure another action instead of sending email messages if you want.

Notes

- [\[1\]](#) PyKota doesn't currently account for ink usage, it only accounts pages. To account for ink usage, you should use PrintBill instead for now.