

How to dissect a CUPS job control file '/var/spool/cups/cNNNNNNN'?

Asked 6 years ago Modified 4 years, 11 months ago Viewed 6k times



4



When printing a job to a CUPS server, you can set up the *cupsd.conf* parameters `PreserveJobHistory` and `PreserveJobFiles` to control how many jobs you want to keep.

CUPS always temporarily stores the actual print job files in directory `/var/spool/cups/`. The spool files as submitted by the print client (before CUPS' conversion chain of filters kicks in) are always named `dNNNNNNN-001` (starting with a 'd' as in 'datafile') where `NNNNNNN` is the job ID assigned by CUPS. If you submit a multi-document print job, the second document's spool file within the same job ID is named `dNNNNNNN-002`, and so on...

Also, the same directory will hold files starting with another character, the *control files* and they will be named `cNNNNNNN` for each job.

I want to dissect these control files files.

When I use the `strings` tool, it only reveals part of what I want to get at:

Example:

```
sudo strings /var/spool/cups/d00089

attributes-charset
utf-8H
attributes-natural-language
en-us
printer-uri
%ipp://localhost:631/printers/hp2B
job-originating-user-name
kurtpfeifleB
job-name
hosts!
copies
finishings
job-cancel-after
job-hold-until
no-hold!

job-priority
job-sheets
noneB
none!
number-up
job-uuid
-urn:uuid:ca854775-f721-34a5-57e0-b38b8fb0f4c8B
job-originating-host-name
localhost!
time-at-creation
time-at-processing
time-at-completed
job-id
job-state
job-state-reasons
processing-to-stop-point!
job-media-sheets-completed
job-printer-uri
(ipp://host13.local:631/printers/hp!
```

Also, that `strings` output doesn't look very nice.

Question: Is there a programmatic (or other) way to dissect these CUPS job control files and get at their complete content with all included info?

binaryfiles

cups

ipp-protocol

Share

edited Dec 8, 2018 at 23:50

Improve this question

Follow

asked Dec 8, 2018 at 23:37



Kurt Pfeifle

89.9k ● 24 ● 266 ● 353

Some data like printer name and print date can be obtained through `lpstat` :

unix.stackexchange.com/a/159955/101920 – mgutt May 25, 2023 at 14:45

2 Answers

Sorted by:

Highest score (default)



Found an answer myself...

12



When you compile CUPS from sources, there is the subdirectory *cups*. It also holds a *Makefile* specific to this subdir. That Makefile contains a build target named *"unittests"* which is **NOT** built by default!





But if you run `make unittests` it not only runs its unit tests, but also creates a few command line utilities which potentially can be put to quite some good use outside of unit tests too!

For the case to solve my problem, it turns out that the `testipp` CLI utility is pure gold. See yourself:

```
sudo ./testipp /var/spool/cups/c00089

operation-attributes-tag:

    attributes-charset (charset): utf-8
    attributes-natural-language
(naturalLanguage): en-us

job-attributes-tag:

    printer-uri (uri):
ipp://localhost:631/printers/hp
    job-originating-user-name
(nameWithoutLanguage): kurtpfeifle
    job-name (nameWithoutLanguage): hosts
    copies (integer): 1
    finishings (enum): none
    job-cancel-after (integer): 10800
    job-hold-until (keyword): no-hold
    job-priority (integer): 50
    job-sheets (1setOf nameWithoutLanguage):
none, none
    number-up (integer): 1
    job-uuid (uri): urn:uuid:ca854775-f721-34a5-
57e0-b38b8fb0f4c8
    job-originating-host-name
(nameWithoutLanguage): localhost
    time-at-creation (integer): 1472022731
    time-at-processing (integer): 1472022731
    time-at-completed (integer): 1472022732
    job-id (integer): 89
    job-state (enum): completed
    job-state-reasons (keyword): processing-to-
```

```
stop-point
  job-media-sheets-completed (integer): 0
  job-printer-uri (uri):
ipp://host13.local:631/printers/hp
  job-k-state (integer): 1
```

Unfortunately, running `make install` will not install this tool into the system, and thusly it never gets exposed to any CUPS administrator! Also, Linux distro packagers may easily overlook it. `testipp` runs under the radar for most CUPS geeks.

There are a few more useful utilities built by `make unittests`:

These are: `testadmin`, `testarray`, `testcache`, `testclient`, `testconflicts`, `testcreds`, `testcups`, `testdest`, `testfile`, `testgetdests`, `testhttp`, `testi18n`, `testlang`, `testoptions`, `testppd`, `testpwg`, `testraster`, `testsnmp`.

Unfortunately, no Linux distro currently builds and ships these useful tools. ***So if you know a distro packager, please point him to this discovery here and ask her to package a nice bundle*** for all end-users of `cups-test-utils.rpm`, or `cups-test-utils.deb` or `cups-test-utils.tgz` or whatever the package name suffixes for his \$distro are! `

Share Improve this answer

Follow

answered Dec 9, 2018 at 1:31



Kurt Pfeifle

89.9k ● 24 ● 266 ● 353



5



I had a similar issue - we needed the duration of jobs and discovered your thread. testlPP worked fine so far, but I didn't want to compile cups just for these little neat tool and also I need it to integrate in other go based applications.

I've started to implement small CLI util in go which also can be used as library <https://github.com/ui-kreinhart/go-cups-control-files>

Share Improve this answer

answered Jan 12, 2020 at 11:14

Follow



[user1363989](#)

61 ● 1 ● 2
