

Designing and Packaging Printer and Scanner Drivers

OpenPrinting MC on Linux Plumbers 2020
August 28, 2020



Open Printing

making printing just work

Till Kammeter, OpenPrinting

What we had

- **Printer drivers**

- PPD files
- Filters, perhaps also backends
- All has to be in CUPS-specific directories

- **Scanner drivers**

- Shared libraries with SANE ABI in SANE-specific directories

- **Packaging**

- Binaries were built specific to destination distro and packaged in DEB or RPM packages
- For each distro drivers need to be built, packaged, and tested separately
- As files need to be in specific directories drivers cannot be installed with CUPS in a Snap or with scanning user applications in Snaps



Open Printing

making printing just work

What we want

- **Sandboxed packaging – Snaps**

- Distribution-independent: Install from Snap Store on any distro running snapd
- More security: Every package with all its libraries and files in its own sandbox, fine-grained control for communication between packages
- All-Snap distributions

- **But**

- You cannot drop driver files into directories of a snapped CUPS or snapped user applications, Snaps do not see the system's files
- Snaps only communicate via IP or D-Bus, not by files

- **Also**

- CUPS is deprecating support for PPD files, working by itself only in driverless IPP mode.



Open Printing

making printing just work

The New Architecture

- **Printer/Scanner Applications emulating an IPP device**
 - Easily snappable: Communicates only via IP
 - Multi-function device support, Printing, Scanning, and Fax Out can be done in one Snap/Application
 - Web admin interface for vendor/device-specific GUI
 - Behaves like a network printer/scanner/multi-function device
- **CUPS for printing (and fax out)**
 - CUPS discovers and uses all driverless IPP printers it finds, physical ones and Printer Applications and auto-creates temporary print queues
 - CUPS spools jobs, does page management, converts jobs to the needed format
- **IPP Scan for scanning**
 - Scanning user applications scan on IPP scanners via IPP Scan
 - In the beginning, retro-fit with sane-airscan SANE backend in the Snap
 - In the future direct IPP scan
 - SANE only hidden in Scanner Applications for retro-fit of classic drivers



- **PAPPL**

- libpappl: Library providing everything what Printer/Scanner Applications have in common
 - Daemon
 - Web admin interface
 - IPP server emulation
 - Job handling
 - Answering all IPP requests, especially get-printer-attributes
 - Printer discovery and setup, also automatically on daemon start
- Only what is specific to the supported devices needs to be implemented



- **cups-filters 2.x**

- libcupsfilters

- Filter functions

- To convert data formats during print/scan job execution
 - Re-using the code of the CUPS filters: pdftopdf, pdftops, pstops, rastertops, rastertopdf, ...
 - Chaining filter function when conversion cannot be done with a single filter
 - All filter functions have the same interface, taking input/output streams, job attributes/options, printer capabilities, log function, and filter-specific parameters

- Auxiliary functions, for IPP attribute handling, calling filter functions from PAPPL filter, ... will get added as needed

- libppd

- All PPD handling functions of libcups and some more: IPP attributes ↔ PPD Options, find PPDs
 - PPDs are deprecated in CUPS and everything PPD supporting will be removed soon
 - For retro-fitting existing classic printer drivers without need of rewriting

- Customized build options for the individual Snap

- No libppd, no libqpdf, Raster-only, no Ghostscript/Poppler, ...



- **snapcraft**

- Printer/Scanner Application to be packaged as a Snap → Distribution-independent
- Upload to Snap Store → Easily available for everyone
- Snapcraft building is similar to RPM/DEB building: You have an instruction file (snapcraft.yaml) and running the snapcraft tool builds the package according to this
- In contrary to RPM/DEB all dependencies (libraries, ...) are included in the Snap, this makes it distribution-independent
- Advanced Security: Snaps are isolated from each other and from the host system, communication only through defined interfaces: network, usb-raw, avahi-control, ...
- Client (CUPS, SANE frontend) communicates only via IPP, Snap communicates also with device
- Under development: snapcraft plugins and extensions to simplify snapping Printer/Driver Applications, to avoid re-including common instructions in snapcraft.yaml and for constant quality
- Planned: Finding Snaps in the Snap store by hardware signature → Driver auto-installation



Design Guidelines

- **1 Printer/Scanner Application = 1 Snap**
- **Printer/Scanner/Fax support can be in a single application, to easily support multi-function devices**
- **Recommended: 1 Printer/Scanner Application per project or manufacturer/product line: Gutenprint, HPLIP, SANE, foo2zjs, Epson, Canon, Ricoh, ...**
- **NOT 1 Printer/Scanner Application per device → A lot of clutter and code duplication**
- **1 Printer/Scanner Application = 1 Port**
- **For more than 1 device on 1 Application use URI: `ipp://localhost:<PORT>/ipp/print/<NAME>`**
- **DNS-SD service names must be always the same, independent of order Application start at boot or of device discovery**
- **Web admin interface should allow suppressing auto-setup for selected devices, manual setup of additional devices/instances, configuration of options not accessible via IPP**
- **sane-airscan in SANE Application must be built without IPP Scan to avoid recursive discovery infinite loop (“Scanner bomb”)**



Further Activity

- **Despite of the incredible work of our whole team we did not finish on the tools and resources yet**
- **Google Season of Docs 2020**
 - Piyush Goyal will write “Tutorial and Design Guidelines for Printer/Scanner drivers in Printer Applications”
 - September 14, 2020 – December 5, 2020
- **Continued development of the tools**
 - Finalization of PAPPL to get released as 1.x and moved to OpenPrinting
 - Finalization of cups-filters, esp. all filters as filter functions to get 2.x, spin out cups-browsed in own project (and Printer Application Snap)
 - Gaining experience with the development of Gutenprint and PostScript-PPD Printer Applications
 - chroot jail to retro-fit closed-source classic printer/scanner drivers
 - Plug-in and extension for snapcraft for quickly and easily snapping Printer Applications
 - Finalizing the CUPS Snap (mainly waiting for work of snapd team)



Thank You

Thank You !!

Twitter: [Open_Printing](#)

Telegram: [Join Us](#)

Email: till@linux.com



Open Printing

making printing just work