OpenPrinting®

Snap, OCI, cups-filters, CPDB, desktop integration of CUPS 3.x

Till Kamppeter – OpenPrinting

6 May 2024

cups-filters: Introduction

- **cups-filters takes up everything from CUPS which Mac OS X does not need** (CUPS 1.6.x)
  - Started end of 2011 by OpenPrinting, **overtaking most of CUPS' filters**
  - Switched filters over from PostScript-centric to **PDF-centric workflow**
  - **cups-browsed** introduced end of 2012, to introduce **browsing of DNS-SD-advertised remote CUPS queues**, as CUPS dropped its own broadcasting/browsing
  - **13** years of **further development** added things like driverless printing support, clustering, support for Printer Applications, IPP standards, PPD-less...

cups-filters Development: Project has been split

- **cups-filters split into 5 parts now**
  - **libcupsfilters**
    - Filter functions and more for Printer Applications/drivers
  - **libppd**
    - All PPD support of CUPS 2.x and more
  - **cups-filters**
    - Filter/backend executables for CUPS 2.x
  - **braille-printer-app**
    - Printer Application for Braille embossers
  - **cups-browsed**
    - Daemon for printer clusters and legacy printer sharing

cups-filters Development: Project has been split

- Splitting done to **separate PPD file support**, to easily fade it out, discontinue it in the future
  - **libcupsfilters**
    - Completely free of PPD file support
    - Filter functions take
      - Printer IPP attributes instead of PPDs
      - Job IPP attributes for options
  - **libppd**
    - PPD and PostScript output support for legacy
    - All functionality from CUPS: libcups, ppdc, cups-driverd, cupstestppd
    - For filter functions: Wrapper, PPD → IPP converter

cups-filters Development: Project has been split

- **cups-filters**
  - CUPS filter/backend executables for CUPS 2.x (legacy)
  - Uses filter functions with PPD support from libppd
  - **`foomatic-rip`** supports PPDs on its own
- **braille-printer-app**
  - Once **converted to Printer Application**, no PPD file support any more
- **cups-browsed**
  - Currently still doing classic queues with PPDs
  - To be **converted to Printer Application**, then free of PPD file support

cups-filters Development: libcupsfilters

- **Filter functions**
  - Converted **all CUPS filters** into **filter functions**
  - Added some **new filter functions**: **`cfFilterPWGToRaster()`**, **`cfFilterUniversal()`**, **`cfFilterExternal()`**
  - Filter functions work **without PPDs**, using **printer and job IPP attributes**)
  - To use filter functions with PPDs, **libppd** provides **data structure extension**, **PPD → IPP attributes converter** and **wrappers**
  - Use **parameters** instead of environment variables
  - All logging into **log function**, no leaks into stderr
  -

cups-filters Development: libcupsfilters

- **Raster data handling**
  - Conversion of image/raster format, color spaces/depth, generate Raster headers, select color space/depth for job
- **IPP Attribute handling**
  - **`get-printer-attributes`**, resolve DNS-SD URIs, select setting from printer/job attributes, select resolutions and page sizes/margins
- **Make/Model/Device ID string handling**
  - Sanitizing, comparing, readability
- **Human-readable strings/translations**
  - Message catalogs from IPP services and from CUPS

cups-filters Development: libppd

- **All PPD file support functionality taken from CUPS 2.x**
  - libcups: All **`ppd...()`** API functions (**`ppd.h`**), made some internal functions public
  - **`ppdc`**: Utilities also some library functions added to API
  - **`cups-driverd`**: Functionality as API functions
  - **`cupstestppd`**: **`ppdTest()`** function, **`testppdfile`** utility
- **Added functionality**
  - PPD support for **filter functions**
    - Convert PPD options/attrib. into printer IPP attributes
    - Wrapper filter functions for special needs
  - Wrapper to turn filter functions into CUPS filters

cups-filters Development: libppd

- **Improvements and fixes**
  - PPD support for **filter functions** (**`ppd-filter.h`**)
    - **`ppdFilterLoadPPD()`**: PPD file → Printer IPP attributes converter (calls also **`ppdLoadAttributes()`**)
    - Wrapper filter functions only for special cases: **`ppdFilterExternalCUPS()`**, **`ppdFilter...ToPDF()`**, **`ppdFilterUniversal()`**
    - Named (“**`libppd`**”) extension for filter function data structure, to hold PPD file name and data, for wrappers
    - **`ppdFilterCUPSWrapper()`**: Wrapper to make PPD-supporting CUPS filter executables from filter functions
  - **`ppdFilterExternalCUPS()`**: Call external, also proprietary, CUPS filters and backends

cups-filters Development: libppd

- **Improvements and fixes**
  - Also **PostScript output is obsolete** and PostScript printers always come with PPD files
    => Move PostScript output filter functions to libppd
    - **`ppdFilterPSToPS()`**, **`ppdFilterPDFToPS()`**, **`ppdFilterRasterToPS()`**, **`ppdFilterImageToPS()`**
  - **`ppdFilterEmitJCL()`**: **JCL/PJL support** in filter functions **moved to libppd**, commands usually supplied by PPD files
  - Support for **page size variants** (**`A4`**, **`A4.Borderless`**, **`A4.Duplex`**, ...), for different margins, sometimes also different print size (overspray).
  - Support for **duplicate sizes** with different names

cups-filters Development: libppd

- **We want to do away with PPDs. Why libppd?**
  - **Legacy printer driver support**
    - No re-writing of driver code for which one has not the hardware for testing
    - Proprietary drivers
    - See also pappl-retrofit
  - **Put together all PPD file support functionality in just one library**
    - Easy fading out of PPD file support
    - Easy discontinuing maintenance
    - Without loss of all the other functionality

cups-filters Development: libppd

- **BUT …**
  - **DO NOT create new PPD files and classic CUPS drivers only because we have this library!**
  - **This library is ONLY for retro-fitting drivers and PPD files which already exist, to avoid re-writing unmaintained, untestable, or proprietary code for devices you cannot buy any more!**
- **If you want to write printer drivers …**

  … create Printer Applications!

cups-filters Development: cups-filters

- **CUPS backend** and **filter** executables for **CUPS 2.x**
- **Filter executables** of general conversion filters **replaced** by small code stub, calling the corresponding **filter function** via **`ppdFilterCUPSWrapper()`**
- Only **`foomatic-rip`** and printer drivers (**`rastertoescpx`**, **`rastertopclx`**, **`commandtoescpx`**, **`commandtopclx`**) are actually implemented in cups-filters
- Backends **`parallel`**, **`serial`**, **`beh`**
- **`driverless`** utility to retro-fit driverless printing support into classic printer setup tools
- **Sample PPD files** for non-IPP PDF printers and PCL-XL printers

cups-filters Development: cups-filters

- **Further improvements and fixes**
  - Streaming mode via **`filter-streaming-mode`** option to run filters optimized or streaming, currently affects **`pdftopdf`**, **`gsto...`**, **`foomatic-rip`** filters.
  - **`imagetops`** implemented via **`ppdFilterImageToPS()`**
  - **`sys5ippprinter`** removed: CUPS does not support System V interface scripts any more, and this first approach of PPD-less printing got never adopted ...
  - **`urftopdf`** removed: CUPS supports URF/Apple Raster by itself

cups-filters Development: cups-browsed

- **Auto-create local CUPS queues** for network IPP printers and remote, shared CUPS queues
- Create **permanent queues** if CUPS would print with a **temporary queue**, for out-of-date print dialogs
- Create **clusters of printers**, also of different models
  - Automatic for equally-named printers of different servers
  - Manual by config file
- **Legacy CUPS broadcasting/browsing** to connect with CUPS <= 1.5 servers and clients
- Legacy **LDAP** support
- **Highly configurable**

cups-filters Development: 2.0.0 Release

- **License: Apache 2.0 + (L)GPL2 exception**, same as CUPS
- Cleaned up **naming style to match CUPS:**
  - API functions: "**`cfCamelCase()`**", "**`ppdCamelCase()`**"
  - Library-internal (private) functions: "**`_cfCamelCase()`**", "**`_ppdCamelCase()`**"
  - File-local (static) functions: "**`underscore_separated()`**"
- Cleaned up all the code to coding style of CUPS
- Bumped **soname** to **2**

cups-filters Development: Finalizing 2.0.0 Release

- **Improvements and fixes approaching final 2.0.0**
  - **Only bug fixes** **(News Sep 2023)**
  - **First security vulnerability** fix for **CVE-2023-24805**
    - Arbitrary command execution by the **`beh`** (Backend Error Handler) backend
    - **`beh`** is rarely used, but it was a great **dress rehearsal for security vulnerabilities**
    - Wrote **security bug handling tutorial** in **May 2023 News**
  - **Synced up libppd with CUPS**, 3 years of fixes and improvements
  - **cups-browsed** busy loop fix from multi-threading support

cups-filters Development: braille-printer-app

- **Not yet released, needs to get converted to Printer Application first**
- Work was started in GSoC 2022 project, GSoC 2023 proposal did not get a contributor slot from Google
- **Arun Patwa** will do the conversion in **GSoC 2024**

cups-filters Development: Next steps

- **libcups 3.x support**
  - **DONE** for **2.1.0**, in **libcupsfilters** and **libppd**
    - Thanks to **Gayatri Kapse** and **Biswadeep Purkayastha** for the conversion
  - **cups-browsed**: Needs to be checked, probably together with conversion to Printer Application
  - **Also in 2.1.0: Fixed API functions for DNS-SD URI resolution** to easily pass to its support by libcups3.
  - Otherwise **transition was easy**, we prepared well by the **design of libppd**

cups-filters Development: Next steps

- **Further plans**
  - Due to our strategy of developing consumers (Printer Applications, ...) and then **developing cups-filters 2.x features as needed by the consumers**, we have reached a **good feature-completeness**.
  - **Future releases** (2.1.0, ...) will happen as new features are needed.
  - **2.1.0** will be mainly for **libcups 3.x support**
  - **2.1.0 or later:** Convert `cfFilterPDFToPDF()` filter function to **standard C** using **PDFio** instead of QPDF and generally **get rid of C++ and QPDF**
    - GSoC project by **Uddhav Phatak**

cups-filters Development: Next steps

- **Further feature ideas**
  - **cups-browsed** in its own Snap **separate** from the CUPS Snap
  - Turn **cups-browsed** into a **Printer Application**
  - Options for the `./configure` script for partial builds: No libqpdf, raster-only printing/scanning, … to **allow Snaps build only the part of cups-filters which they actually need.**
  - **Documentation** of libraries with Mike Sweet's codedoc utility
    - Look for contributors through Canonical's **Documentation Academy** program

OpenPrinting Development: Ghostscript

- **No Artifex presentation this year, after switch of PDF interpreter to C only bug fixes**
- No complaints about performance or output quality of new PDF interpreter
- No changes on Apple/PWG/CUPS Raster output device by me
- Artifex "low on development resource"
- Linux distro contact is **Chris Liddell**

OpenPrinting Development: Security vulnerability handling

- OpenPrinting uses **GitHub** to maintain all its code
- By default **no way to report private (security) bug**
- Discovered that Michael Sweet's PAPPL repo **accepts security bugs**
- **Way to activate it not intuitive**, needs to get **activated in Settings**, under "Code security and analysis" **AND** bug report **template** needs to get created.
- Talked with **Mark Esler from Canonicals Security Team** on how to handle security bugs and **Mark reported to GitHub** about GitHub's security bug support activation
- **GitHub improved by not needing bug report template any more**

OpenPrinting Development: Security vulnerability handling

- **Enabled Security bug reporting for all our repositories**
- Created “**Security Team**” with access to the security bugs: Till Kamppeter, Michael Sweet, Zdenek Dohnal, Thorsten Alteholz, Johanne Meixner, Aveek Basu
- **Received 2 security bug reports** on cups-filters so far:
  - CVE-2023-24805 on cups-filters: Arbitrary command execution by the beh (Backend Error Handler) backend
  - CVE-2023-4504 on libppd: Postscript Parsing Heap Overflow
- Based on this experience created a **tutorial about security bug handling with GitHub** in **May 2023 News**

OpenPrinting Development: libcups3/CUPS 3.x/PAPPL 2.x support

- **libcups3 support added** (thanks Gayatri Kapse & Biswadeep Purkayastha) **(News Oct 2023)**
  - **libcupsfilters**, **libppd**, **pappl-retrofit**
  - Followed how Michael Sweet did it in **PAPPL 1.4.x**
  - **Actual code is using libcups3 API**
  - For building with libcups2 an additional header file is included which **“translates” libcups3 API to libcups2 API with macros** (`libcups2-private.h`)
  - In a few cases still some C code is needed for libcups2 support, have put it into `libcups2.c`
  - Rarely used “`#ifdef HAVE_LIBCUPS2`” in actual code
- To Do: cups-browsed, needs to be turned into Printer App
- To Do: PAPPL 2.x support in pappl-retrofit

OpenPrinting Development: The CUPS 3.x Snaps

- **Sharing server**
  - Current CUPS Snap will be the base
- **Local server**
  - User daemon
  - Triggered by D-Bus
  - Optionally run permanently from login to logout?
- **Snapping (snapcraft.yaml, scripts, ...) in cups-sharing and cups-local repos**
  - Move snapping already into CUPS 2.5.x repo?

Desktop Integration: Common Print Dialog Backends 2.x

- In the course of **adding CPDB support to the GTK print dialog** GSoC contributor Gaurav Guleria has **added many features to CPDB itself**, leading to the **2.x generation**
- Acquire printer details **asynchronously** (non-blocking)
- **Synchronous** printer data fetching on **backend activation**
- **Backends signal** frontends on **printer updates**
- **Option groups**
- Interfaces for **human-readable/translated** group, option, and setting names
- Retrieve media dimensions from a given “**`media`**” setting
- Support for **margin variants** for the same media size (borderless, ...)
- Support for configurable user and system-wide **default printers**

Desktop Integration: Common Print Dialog Backends Snap

- To complete OpenPrinting Snaps we need **Snap of CPDB CUPS backend**
- **General method to snap CPDB backends**
- Backend is **user daemon**, **D-Bus-triggered**
- **Requirements for snappability** **(News March 2024)**
  - Job passed as **stream**, not as files, frontend receives socket path via D-Bus
  - **Remove Job count**, **job list** and **cancel Job** methods
  - Filtering printer list by **backend methods** not by frontend being D-Bus service sending signals
  - Add **printers of newly appearing backends** to list
- Thanks, Biswadeep Purkayastha, for working on this!

Desktop Integration: Print Dialogs

- We need to get **Common Print Dialog Backends** into all dialogs
- **CUPS CPDB backend** then takes care of changes: No use of PPD files, temporary queues, … **Backend itself DONE** (thanks, Gaurav Guleria), also needs libcups3 support
- **Status of dialogs**:
  - **GTK**: Gaurav Guleria 2022, merge request accepted → **DONE**
  - **Qt**: Gaurav Guleria 2022, shortly before getting merged
  - **Mozilla** (Firefox/Thunderbird): Feature request posted, Kushagra Sharma 2024
  - **Chromium**: Kushagra Sharma 2023, shortly before getting merged
  - **LibreOffice**: Posted on dev mailing list, with reference to a first approach back in 2017, Biswadeep Purkayastha 2024

Desktop Integration: Printer Setup Tools

- **Main Window**
  - List **all IPP print services** as reported by DNS-SD
  - List Printer Applications and their queues or printer/fax in a group
  - No duplicates for IPv4/IPv6, IPPS, interfaces
  - Buttons for web interface, add new queue, show jobs …
- **Add Printer Wizard**
  - List of discovered **non-driverless** USB/network printers
  - Button to see list of **Printer Applications** supporting the printer, installed ones and available in Snap Store (look-up service on OpenPrinting)
  - Buttons to setup printer with selected Printer Application and to install Printer Application from Snap Store
- **Do not remove support for permanent CUPS queues and classic drivers**

Desktop Integration: GNOME Control Center

- Extension of the “**Printers**” module for the New Architecture
- **Main view**
  - **List IPP print services**, each is a print destination for CUPS, without need of actual CUPS queue
  - For MF devices with printer and fax or Printer Applications with various queues, **group the services**.
  - Also list **classic CUPS queues**, for universal compatibility with all CUPS versions.
  - On IPP services button to **open web interface in browser**, no “Set Options”, “Remove printer”, ...
- **Add Printer:**
  - Adding support for finding suitable Printer Applications to **cups-pk-helper**
  - UI to assign and handle both Printer Applications and classic drivers
- **Thanks** to Lakshay Bandlish (GSoC 2020), Divyasheel (2021), Mohit Verma (2022, 2023)
- Also support by the **Canonical Desktop and Design Teams**

Desktop Integration: KDE Print Manager

- **KDE Print Manager** is printing part of **KDE Settings**
- Needs to follow the same principal scheme as **GNOME's "Printers" module**
- **KDE Developer Mike Noe** opened feature request for **general improvements on Print Manager**
  - **KDE Developer Nate Graham** commented about **CUPS 3.x needs**
  - **I added detailed comment** on what is needed with **reference to GNOME and Mohit Verma's work**
- I also had **longer private mail thread with Mike Noe**
- **Mike Noe posted feature request for CUPS 3.0 support**
- So it seems that Mike Noe is on it … **(News April 2024)**

Desktop Integration: system-config-printer

- **No further feature development** on system-config-printer at Red Hat, “**maintenance mode**”
- But **system-config-printer is actually used**, with non-GNOME, non-KDE desktops (in several Ubuntu flavors)
- Switch to GNOME or KDE tool just for a printer setup tool **pulls too many dependencies**
- system-config-printer also includes **scp-dbus-service** for device identification and driver assignment, **used also by the KDE and GNOME tools**
- Therefore **we resume development**, to make it **supporting CUPS 3.x**, also following Mohit Verma’s GNOME work
- **GSoC 2024 project by Shivam Jaiswal (News April 2024)**

Questions / Comments