OpenPrinting®

PostScript Printer Application

Till Kamppeter – OpenPrinting
5 May 2021

Introduction

- In the **Snap Store**: https://snapcraft.io/ps-printer-app
- **PAPPL**-based
- First working model and production-ready Printer Application for **standard A4/Letter office printers**
- First **retro-fitting** Printer Application
- First Printer Application for **high-level/vector graphics output format**
- First project **really exercising PAPPL** leading to many bug reports and feature requests where most got fixed/implemented
- Supports **1000s of PostScript printers** by ~4000 included PPDs, a generic PPD, and possibility to upload original PPD which comes with printer

PPD file handling

- Everything based on **libppd**
- **Lists PPDs** with duplicate elimination, user-friendly sorting
- **Auto-assigns PPD** to make/model from device ID, generic PPD to CMD: of device ID
- Lists **standard**, **vendor-specific**, and **installable accessory** options
- Associates **standard options** with **IPP attributes** and **ready media**
- Modifies **Printing Defaults** list depending on **installed accessories**
- Adds **PPD's PostScript code** to job data streams

PPD file handling

- Printers can be queried for option defaults and installed accessories if PPD supplies appropriate PostScript code
- Allows user to **add his own PPDs**, checks file format, rejects and warns, prefers user PPDs on auto-assigning to printer, allows removal

Further Properties

- Uses **filter functions** to print **PDF** and **PostScript** input data, to have no inbetween raster step, maintaining maximum print quality
- **PostScript** jobs are **not passed through unfiltered/raw**, but always with PPD code applied
- Uses **PAPPL's raster printing functionality** to print **raster input**, allowing completely **streaming** and "infinite" jobs, conversion to grayscale, and to dithered bi-level (for fast draft printing)
- **A4/Letter auto-selection** by locale environment variables
- The **~4000 PPDs** (from **foomatic-db** and **HPLIP**) in the Snap are compressed with **pyppd**

Next Steps

- We are **nearly feature-complete**, missing parts are due to pending features in PAPPL:
  - Human-readable strings for vendor options
  - Internationalization/Localization
  - Ink level check, like CUPS does via SNMP in backends
- **Base for future retro-fitting Printer Applications**, we will spin out most retro-fit-relevant functionality into a library

Questions / Comments