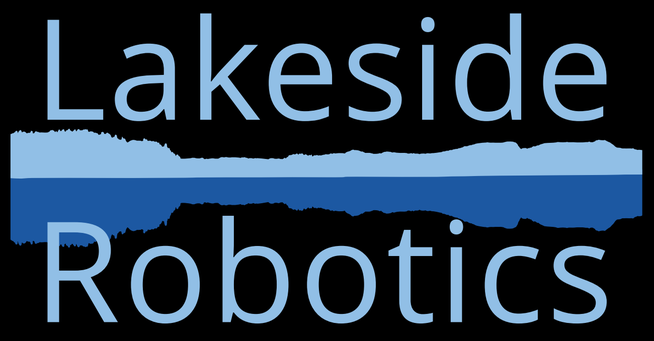


Printer Applications and PAPPL

August 2024

Michael R Sweet (Lakeside Robotics)

Topics

- Printer Applications Are…
- But…
- PAPPL

Printer Applications Are…

- Standalone programs that can be packaged, distributed, installed, and run in many ways (native packages, Snap, Flatpak, AppImage, Docker, etc.)

- A replacement for CUPS printer drivers
  - Options are replaced by IPP attributes
  - Driver-specific UI is provided by the Printer Application

- An implementation of an IPP Everywhere™ Printer
  - Makes non-IPP printers look and act like IPP printers

- Compatible with all current desktop, server, and mobile operating systems/Linux distributions
  - Takes advantage of AirPrint™, IPP Everywhere™, and Mopria® support

But...

- Writing an IPP service from scratch is challenging
- AirPrint™ and Mopria® each have some unique attributes/values to be supported in addition to the common IPP, mDNS/DNS-SD, and file format "stuff"
- Most developers don't have the resources to certify a custom implementation

PAPPL - Overview

- "Printer Application Framework"
  - A simple CUPS-based C framework/library for developing Printer Applications
  - Implements all of the required IPP, mDNS/DNS-SD, and file format support for you
  - JPEG, PNG, PWG Raster, Apple Raster, and "raw" printing support is standard
    - PDF and other formats can be supported easily
- Web site w/documentation and Github repository:
  - https://www.msweet.org/pappl
  - https://github.com/michaelrsweet/pappl
- Licensed under the Apache License Version 2.0 with an exception to allow linking against GPL2/LGPL2 software

PAPPL - Platform Support

- Runs on desktops, servers, and in embedded environments
  - *BSD, Linux, macOS®, QNX®, Windows® 10+
  - Used in shipping printers
- Works out-of-the-box with AirPrint™ (iOS®/macOS®), IPP Everywhere™ (ChromeOS, Linux), and Mopria® (Android™/Windows® 10+) clients
- IPP-USB "gadget" support on Linux (requires a suitable USB controller)

PAPPL - Key Contributors

- Michael Sweet (Lakeside Robotics): lead developer
- Jai Luthra (GSoC 2020): auto-setup, hp-printer-app, networking improvements, DNS-SD and SNMP discovery, papplMainloop API, snapcraft prototyping
- Sambhav Dusad (GSoC 2020): job persistence, live log viewer, test pages, web interface enhancements
- Didier Raboud (Debian Project): documentation and packaging

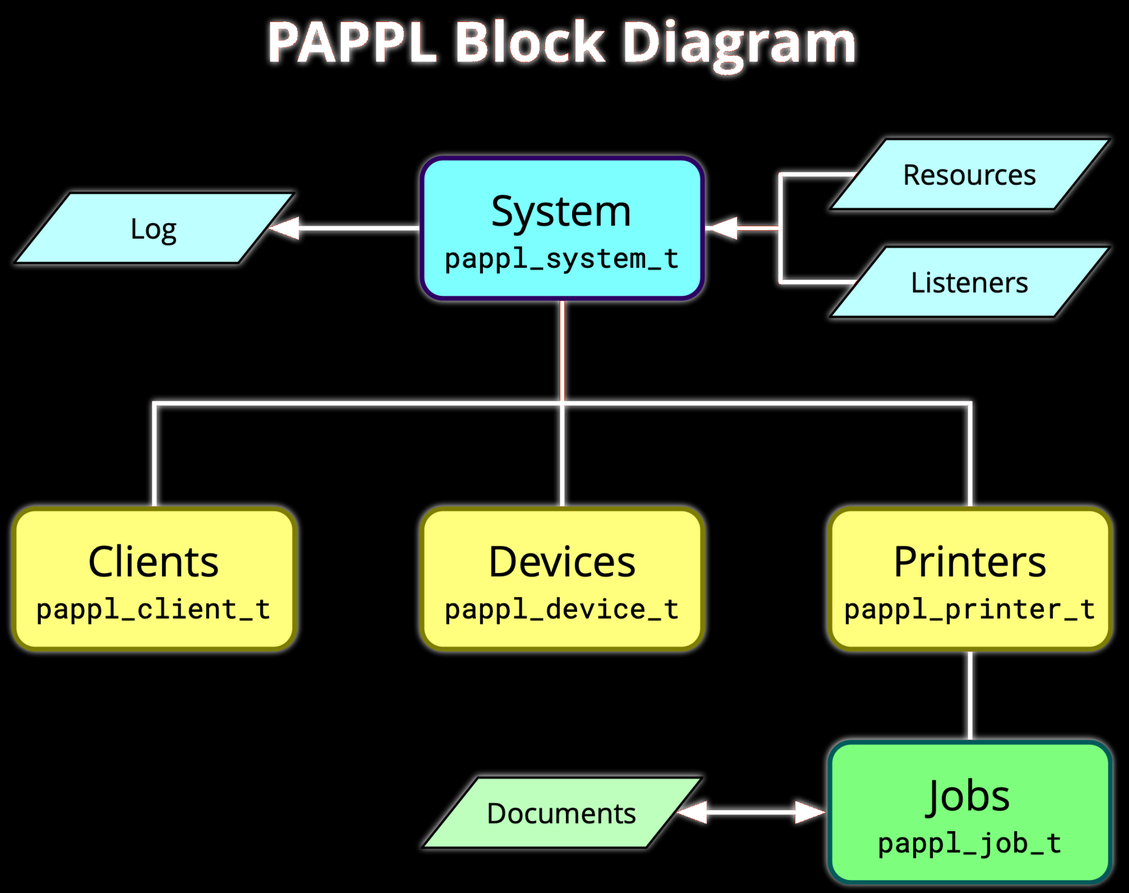
PAPPL - v1.4.x

- Current stable branch ("v1.4.x")
- Release History:
  - v1.4.7: August ??, 2024:
    - PAM-based authentication did not work on Linux due to a glibc incompatibility
    - Fixed the web interface for setting the admin and print groups
    - Fixed the web interface for adding network printers on non-standard port numbers
    - Fixed some USB gadget error conditions
    - Fixed the Wi-Fi configuration web page
    - Fixed a logging deadlock issue
    - Fixed some threading issues
    - Fixed how PAPPL responds to an unsupported request character set

PAPPL: API Overview

- PAPPL Programmers Manual and example printer application project:
  - https://www.msweet.org/pappl/pappl.html
  - https://github.com/michaelrsweet/hp-printer-app
- PAPPL provides five main objects:
  - The System (`pappl_system_t`): The main object that manages the whole printer application
  - Clients (`pappl_client_t`): The objects that manage client connections
  - Devices (`pappl_device_t`): The objects that manage printer connections
  - Printers (`pappl_printer_t`): The objects that manage printers
  - Jobs (`pappl_job_t`): The objects that manage print jobs
- Simple "main loop" API takes care of standard sub-commands and behaviors

PAPPL - API Overview



PAPPL - System - pappl_system_t

- Manages all client and device connections, listeners, the log, printers, and resources
- Implements a subset of the IPP System Service (PWG 5100.22)
- Provides an embedded web interface
- Supports USB gadgets on Linux:
  - Classic USB printer class
  - IPP-USB
  - USB serial (for debug console/logs)
  - Mass storage
- Can be configured for single printer mode or multiple/variable printer mode

PAPPL - System - pappl_system_t

- Listeners support both TCP/IP and local (domain) listening sockets
  - Default is to listen for IPv4 and IPv6 connections from any address
- Logging can be to a file, stderr, or syslog/systemd
  - Log monitoring to a file allows web interface to provide remote monitoring
  - Five levels of logging (fatal, error, warn, info, debug)
- Resources can be dynamic (callback) or static (files or memory), System or Printer specific
  - Not currently exposed as PWG 5100.22 Resource objects

PAPPL - Client - pappl_client_t

- Provides access to client connections
- Client connections and the life cycle of the `pappl_client_t` objects are managed automatically by the System object
- Provides support methods for authentication, HTML forms, and the default page header and footer

PAPPL - Device - pappl_device_t

- Provides access to output device connections and to list available output devices.
- Output devices are accessed using Uniform Resource Identifier (URI) strings such as "file:///path/to/file-or-directory", "socket://11.22.33.44", and "usb://make/model?serial=number".
- Included URI schemes:
  - "dnssd": Network (AppSocket) printers discovered via DNS-SD/mDNS (Bonjour)
  - "file": Local files and directories
  - "ipp" and "ipps": Internet Printing Protocol printers (no discovery)
  - "snmp": Network (AppSocket) printers discovered via SNMPv1
  - "socket": Network (AppSocket) printers using a numeric IP address or hostname and optional port number
  - "usb": Local USB printer

PAPPL - Printer - pappl_printer_t

- Implements IPP Everywhere™ (PWG 5100.14) and some extensions to provide compatibility with the full range of mobile and desktop client devices
- Each printer is connected to a Device and uses a driver to process document data and produce output
  - Drivers have a required raster interface to support Apple and PWG Raster
  - Optional "raw" file printing interface
  - Filters to support other formats - JPEG and PNG included, PDF and others can be easily added

PAPPL - Job - pappl_job_t

- Corresponds to IPP Job object
- Created in response to IPP requests
- One or more documents per Job
- Apple and PWG raster are streamed while other formats are spooled
  - Printer can be configured to spool multiple Jobs, but only one streamed Job can be active at any time
- Job history can be configured (0 to infinity), completed/canceled/aborted Jobs stay in history for at least 60 seconds

PAPPL - Main Loop API

- A single `papplMainLoop` function provides the standard web interface and command-line to a Printer Application
- Command-line supports common "sub-commands" to do different things:

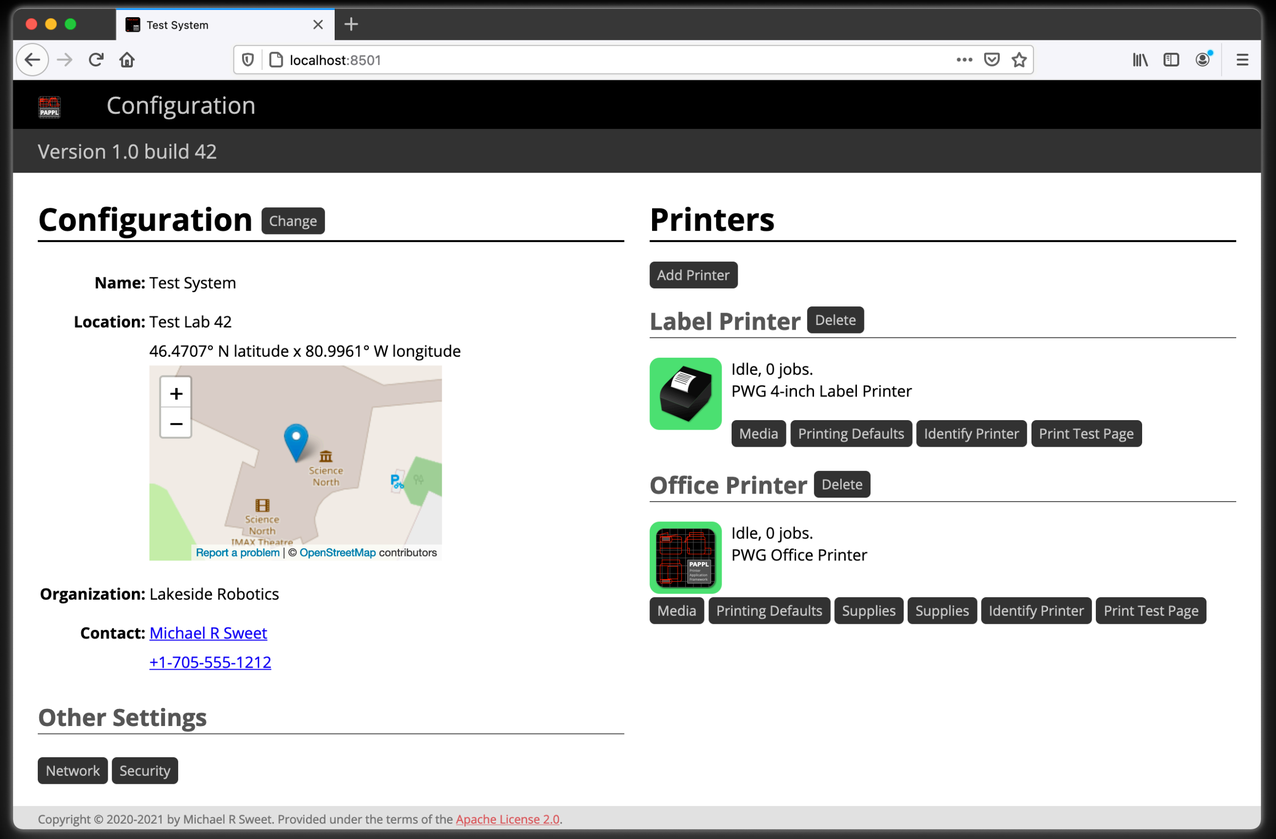
add	autoadd	cancel	default
delete	devices	drivers	jobs
modify	options	printers	server
shutdown	status	submit	

- Callbacks can be used to customize everything

PAPPL - Command-Line Experience

- List drivers:
  `printer-app drivers`
- List available printers/devices:
  `printer-app devices`
- Add a printer:
  `printer-app add myprinter -m auto -v usb://Example/Printer`
- List available options:
  `printer-app options -d myprinter`
- Print something:
  `printer-app -d myprinter -o option=value filename`

PAPPL - Web Interface Experience



PAPPL - Current Applications

- braille-printer-app - https://github.com/OpenPrinting/braille-printer-app
- ghostscript-printer-app - https://github.com/OpenPrinting/ghostscript-printer-app
- gutenprint-printer-app - https://github.com/OpenPrinting/gutenprint-printer-app
- HP Printer App - https://github.com/michaelrsweet/hp-printer-app
- hplip-printer-app - https://github.com/OpenPrinting/hplip-printer-app
- LPrint - https://github.com/michaelrsweet/lprint
- pappl-retrofit - https://github.com/OpenPrinting/pappl-retrofit
- ps-printer-app - https://github.com/OpenPrinting/ps-printer-app

PAPPL - v2.0.x

- Current feature branch ("master")
- Release schedule:
  - v2.0.0: September/October 2024
- Features:
  - https://github.com/michaelrsweet/pappl/milestone/8
  - IPP System Service v1.1 support
  - IPP Shared Infrastructure Extensions v1.1 support
  - IPP OAuth Extensions v1.0 support
  - IPP Document Object v1.2 support with multiple document jobs
  - "job-password/-encryption" and "job-release-action" support
  - "job-sheets/-col" support
  - "proof-copies" support
  - ipptransform support (from libcups v3)

PAPPL - v2.0.x

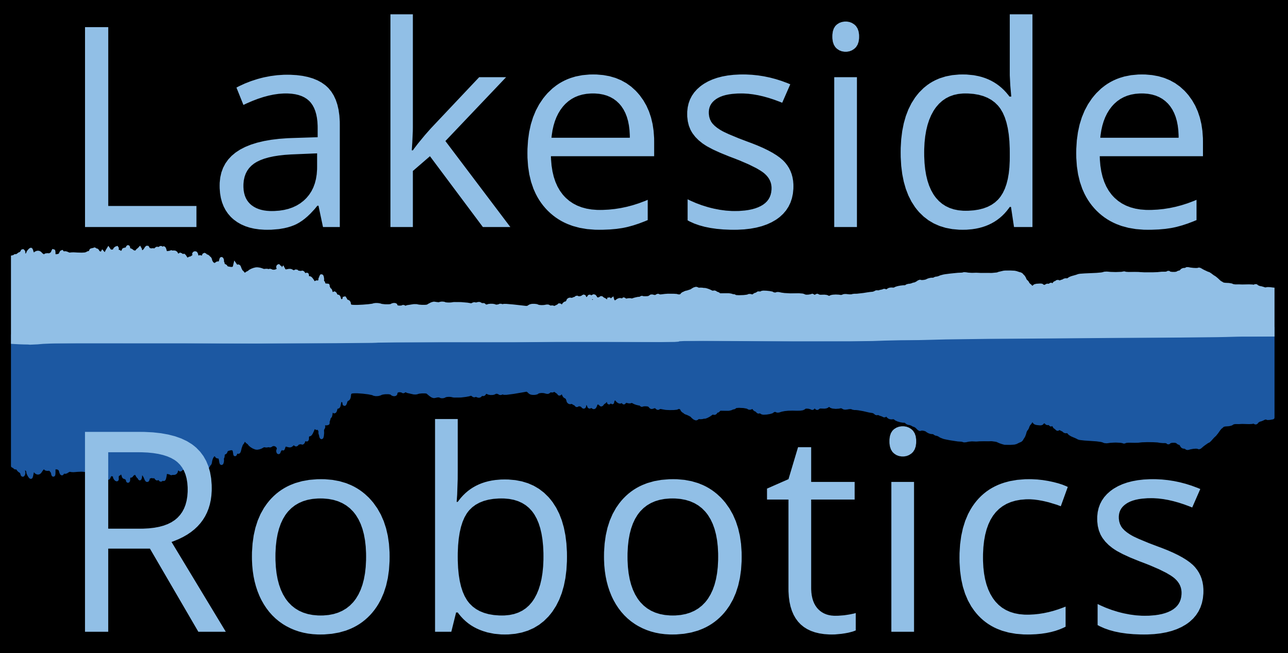
- **PAPPL 2.0.x is not binary-compatible with PAPPL 1.x**
- **PAPPL 2.0.x requires CUPS 2.5 or later**
- API changes may require some source changes to v1.x-based printer applications - full details are part of the PAPPL 2.0 documentation
  - https://www.msweet.org/pappl/pappl2.html
  - https://www.msweet.org/pappl/pappl2.epub
  - Biggest changes are using `size_t` for array sizes instead of `int` and adding the document number argument to several functions
  - Also a handful of renamed constants/functions/types for consistency

PAPPL - Infrastructure Support

- The new `PAPPL_SOPTIONS_INFRA_SERVER` and `PAPPL_SOPTIONS_INFRA_PROXY` server options enable Infrastructure Printer (server) and Proxy functionality
- Infrastructure Printers can be created using the new `papplPrinterCreateInfra` function
  - Infrastructure Printers can support multiple output devices, which are added manually via `papplPrinterAddInfraDevice` or dynamically via IPP Register-Output-Device requests
  - Registration callbacks can be set with `papplSystemSetRegisterCallbacks` to enable dynamic output device registration
- Proxies can be added/enabled using the `papplPrinterAddInfraProxy` function
  - This will relay/proxy jobs from an Infrastructure Printer to a local Printer

PAPPL - Future Stuff

- Load/Save Callbacks for Printers
  - To load/save printer-specific data without having to write your own full system load/save callbacks
- Presets
  - Standard API and UI for providing IPP "job-presets-supported" values
- Scanning support for multifunction and standalone scanners
  - eSCL (what AirPrint and Mopria use) and maybe IPP Scan if there is interest
- SMB support
  - For printing through old Windows servers


Lakeside
Robotics