

Printer Applications

Joint PWG / Open Printing Summit - May 2021
Michael R Sweet (Lakeside Robotics)



Open Printing

making printing just work

Before We Begin...

- PWG Antitrust Policy:
 - https://www.pwg.org/chair/membership_docs/pwg-antitrust-policy.pdf
 - The IEEE-ISTO Printer Working Group ("PWG") will not become involved in the business decisions of its Members. The PWG strictly complies with applicable antitrust laws. Every PWG meeting attendee shall comply with this policy. The PWG Officers and PWG Workgroup Officers are responsible to ensure that this policy is adhered to in all PWG activities.
- PWG Intellectual Property Policy:
 - https://www.pwg.org/chair/membership_docs/pwg-ip-policy.pdf
 - TL;DR: Anything you say in a PWG meeting or email to a PWG address can be used in a PWG standard
 - (but please do read the IP policy above if you haven't done so)



Topics

- Printer Applications Are...
- PAPPL: Printer Application Framework
- LPrint
- Gutenprint
- PostScript Printer Application (ps-printer-app)



Printer Applications Are...

- 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
 - Basic IPP Everywhere support only requires PWG Raster, plus JPEG for color printers
 - CUPS library and sample code provide an easy-to-use framework for implementations
- Compatible with CUPS 1.4 and later
 - Can be compatible with iOS® 5 and later with a few small additions (DNS-SD subtype and "image/urf" document format)
 - macOS 10.8® and later support IPP Everywhere™ via the command-line, can be used from the GUI with the same changes needed for iOS support



Open Printing

making printing just work

PAPPL: Printer Application Framework

- Web site and Github repository:
 - <https://www.msweet.org/pappl>
 - <https://github.com/michaelsweet/pappl>
- A simple CUPS-based C framework/library for developing Printer Applications or firmware for any kind of printer or driver on desktops, servers, and in embedded environments
- Supports JPEG, PNG, PWG Raster, Apple Raster, and "raw" printing to printers connected via USB and network (AppSocket/JetDirect) connections
- Works with AirPrint™ (iOS/macOS), IPP Everywhere™ (ChromeOS, Linux), and Mopria® (Android/Windows 10) clients out-of-the-box
- *Will soon be in shipping printers*
- Licensed under the Apache License Version 2.0 with an exception to allow linking against GPL2/LGPL2 software



Open Printing

making printing just work

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: Releases

- December 11, 2020: v1.0
 - First production release
- January 12, 2021: v1.0.1
 - Bug fix release, vendor attribute improvements
- February 20, 2021: v1.0.2
 - Bug fix release, attribute validation improvements, allow updating the version information
- April 21, 2021: v1.0.3
 - Final 1.0.x bug fix release



Open Printing

making printing just work

PAPPL: v1.1 Feature Release

- Current "master" branch
- Hoping to release at the end of May 2021
- <https://github.com/michaelsweet/pappl/projects/1>
 - IPP-USB support (done)
 - Option to disable TLS (done)
 - papplMainLoop persistent state files (done)
 - Start/stop printer sub-commands (to-do)
 - OAuth 2.0/OpenID support (to-do)
 - Proxy (IPP Shared Infrastructure Extensions) support (to-do)



PAPPL: v1.2 Feature Release

- November 2021
- <https://github.com/michaelsweet/pappl/projects/2>
 - Full localization support
 - macOS/GNOME system icons
 - SNMP enhancements (1284 device ID and printer MIB queries)
 - Image interpolation



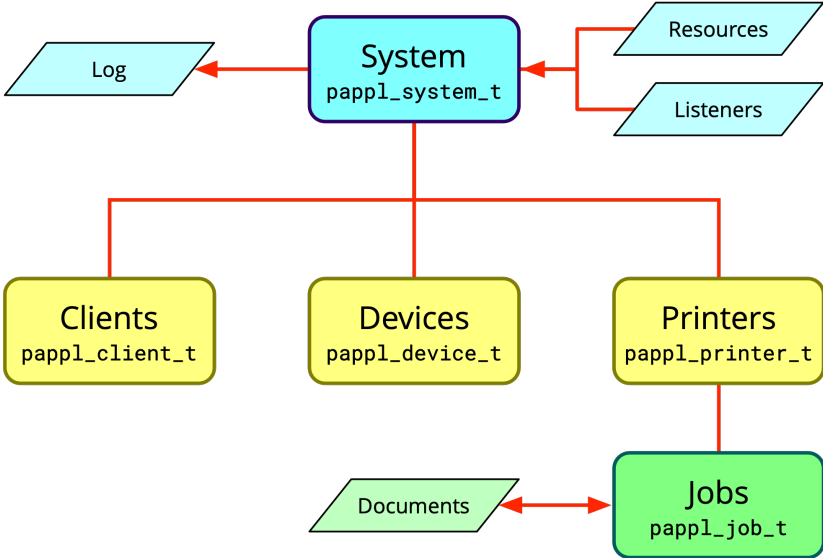
PAPPL: API Overview

- PAPPL Programmers Manual and example printer application project:
 - <https://www.msweet.org/pappl/pappl.html>
 - <https://github.com/michaelsweet/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 Block Diagram



Open Printing

making printing just work

PAPPL: System Object (pappl_system_t) 1/2

- 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 Object (pappl_system_t) 2/2

- 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 Object (`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 Object (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
 - "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



Open Printing

making printing just work

PAPPL: Printer Object (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 Object (pappl_job_t)

- Corresponds to IPP Job object
- Created in response to IPP requests
- One document per job (currently)
- 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:

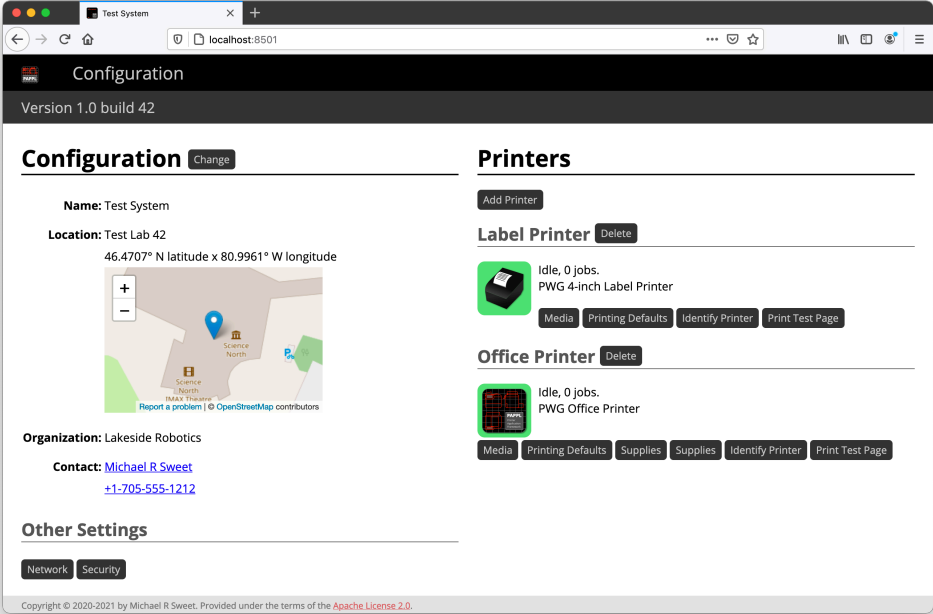
```
printer-app -d myprinter -o option=value filename
```



Open Printing

making printing just work

PAPPL: Web Interface Experience



Open Printing
making printing just work

LPrint



- Web site and Github repository:
 - <https://www.msweet.org/lprint>
 - <https://github.com/michaelsweet/lprint>
- Supports a variety of common label and receipt printers connected via network or USB
 - Currently all Dymo and Zebra label printers, looking at adding others...
- v1.0 was released on February 17, 2020 and formed the basis of PAPPL
- v1.1 was released on ????? ??, 2021 and uses PAPPL



Gutenprint Printer Application

- Web site and SourceForge repository:
 - <http://gutenprint.sf.net/>
 - <https://sourceforge.net/projects/gimp-print/>
- Sambhav Dusad (GSoC 2020 student) started the effort late in 2020 but so far nothing has been pushed to the repository



PostScript Printer Application (ps-printer-app)

- Extra slides from Till:
 - <https://ftp.pwg.org/pub/pwg/liaison/openprinting/presentations/ps-printer-app-2021.pdf>



Q&A



Open Printing
making printing just work

Resources (1/2)

- LPrint
 - <https://www.msweet.org/lprint>
 - <https://github.com/michaelsweet/lprint>
- PAPPL
 - <https://www.msweet.org/pappl>
 - <https://github.com/michaelsweet/pappl>
 - <https://github.com/michaelsweet/hp-printer-app>
- PostScript Printer Application (ps-printer-app)
 - <https://github.com/openprinting/ps-printer-app>



Resources (2/2)

- AirPrint™ information:
 - <https://support.apple.com/en-us/HT201311>
- IPP Everywhere™ v1.1 specifications:
 - <https://ftp.pwg.org/pub/pwg/candidates/cs-ippeve11-20200515-5100.14.pdf>
 - <https://ftp.pwg.org/pub/pwg/candidates/cs-ippeveselfcert11-20200515-5100.20.pdf>
- IPP Everywhere™ printer self-certification tools:
 - <https://istopwg.github.io/ippeveselfcert>
- Mopria® information:
 - <https://mopria.org>



Open Printing

making printing just work