CUPS Plenary

Joint PWG/Open Printing Meeting
Lexington, KY
April 16, 2019

Michael Sweet, Apple Inc.

CUPS Plenary

Topics

- Introduction
- CUPS 2.2
- CUPS 2.3
- CUPS Deprecations
- CUPS Future
- Q&A
- Resources

CUPS Plenary

Introduction

- CUPS is the standards-based, open source printing system developed by Apple Inc. for macOS® and other UNIX®-like operating systems.
- The CUPS web site, source code, and bug database are hosted on Github
    - https://www.cups.org/
    - https://github.com/apple/cups/

CUPS Plenary

CUPS 2.2

- CUPS 2.2 is the current stable branch with patch releases every 2-3 months:
  - CUPS 2.2.0 was released September 13, 2016
  - CUPS 2.2.1 was released October 3, 2016
  - CUPS 2.2.2 was released January 17, 2017

    …
  - CUPS 2.2.10 was released December 7, 2018
  - CUPS 2.2.11 was released March 22, 2019
- A "final" CUPS 2.2.12 release is planned before CUPS 2.3.0

CUPS 2.2

License

- CUPS 2.2.x and earlier continue to use the GNU GPL2/LGPL2 with exceptions for software on Apple operating systems (macOS, iOS, tvOS, etc.)
- Future CUPS security fixes will also be explicitly provided under the same license
    - File Github issues for other important bug fixes (larger than a few lines of code) that you need from a newer Apache-licensed CUPS release

CUPS 2.2

Developer Resources

- "New" CUPS Programming Manual
    - https://www.cups.org/doc/cupspm.html
    - https://www.cups.org/doc/cupspm.epub
- Documents CUPS APIs as well as best practices
- Includes lots of examples
- Please provide feedback via Github and/or cups-devel mailing list

CUPS 2.2

Security Changes

- CUPS 2.2.10: CVE-2018-4300 - Linux session cookies used a predictable random number seed

CUPS 2.2

Known Issues in CUPS 2.2.11

- Scheduler crashes after adding a printer and sending a print job (Issue #5554)
  - Fix pending, but this issue has been happening sporadically for several years and we only now have a way to reproduce it reliably
- Problems with "job-password" (Issue #5557 and #5558)
  - Fixed in CUPS master and branch-2.2
- Changes in CUPS 2.2.10 broken some Samsung printer drivers (Issue #5562)
  - Fixed in CUPS master and branch-2.2

CUPS Plenary

CUPS 2.3

- CUPS 2.3 is the next feature release:
  - CUPS 2.3.0 tentatively scheduled for June/July 2019
  - Additional 2.3.x updates planned through the end of 2021
- Primary Focus of CUPS 2.3:
  - License Change
  - IPP Everywhere
  - Print Accounting
  - Scheduler

CUPS 2.3

License Change

- CUPS 2.3 and later will be distributed under the terms of the Apache License Version 2.0
  - Eliminates compatibility issues with projects that use GPL3, LGPL3, AGPL3, and the Apache License Version 2.0
- Will include an exception for GPL2/LGPL2-only software (same as LLVM and a few other projects)

CUPS 2.3

IPP Everywhere

- Localization of attributes and values, including printer-specific values from a printer's .strings files
- IPP Job Presets support
- IPP "finishing-template" support
- Closing of any remaining CUPS API "holes" preventing applications from using IPP Everywhere instead of printer drivers
- Bug fixes

CUPS 2.3

Print Accounting

- The scheduler now tracks the total number of media sheets and only logs the count once a job completes
  - The previous mix of progress (from filters) and total (from printer) values could yield incorrect accounting results
  - Also simplifies accounting software that uses the page_log file - now just a single line for each job that is printed

CUPS 2.3

Scheduler

- Now generate a per-printer .strings file for client-side localization
- Bonjour (sharing) host name can now be set
- Now support the "printer-id (integer(1:65535))" Printer Status attribute
- Scripted CGI programs are supported differently:
  - Now rely on execute bit and #! header
  - No more hardcoded script interpreters

CUPS 2.3

CUPS API

- Improved media selection support, including a new function:

```
int
cupsAddDestMediaOptions(http_t *http,
    cups_dest_t *dest, cups_dinfo_t *dinfo,
    unsigned flags, cups_size_t *size,
    int num_options, cups_option_t **options);
```

- New option encoding function:

```
ipp_attribute_t *
cupsEncodeOption(ipp_t *ipp,
    ipp_tag_t group_tag, const char *name,
    const char *value);
```

CUPS 2.3

CUPS API, con't

- `cupsCopyDestConflicts` now supports collection attribute ("media-col", "finishings-col", etc.) constraints
- HTTP header values can now be longer than the old static limit (`HTTP_MAX_VALUE`)
- The '-D_IPP_PRIVATE_STRUCTURES=1' cheat no longer works when including the `<cups/ipp.h>` header
    - The `ipp_t` and `ipp_attribute_t` structures are now fully private (moved to private header)
    - Use the accessor functions (`ippGetXxx`/`ippSetXxx`) which were added in CUPS 1.6 *nine* years ago

CUPS 2.3

CUPS API, con't

- The '-D_PPD_DEPRECATED=""' cheat for the `<cups/ppd.h>` header file no longer works
  - There is no longer a way to disable compile-time warnings when using the PPD functions
  - The CUPS destination APIs are the replacement for all PPD functionality and have been since CUPS 1.4 which was released *eleven* years ago

CUPS 2.3

CUPS API, con't

- `cupsRaster` functions now included in both libcups and libcupsimage
  - libcups contains the actual implementation, with stubs in libcupsimage pointing to the libcups functions
  - Goal is to eliminate libcupsimage in a future version of CUPS
  - "cups-config --image --libs" returns the same thing as "cups-config --libs" now

CUPS Plenary

CUPS Deprecations

- We periodically deprecate functionality that either is no longer necessary or will prevent us from improving CUPS
- When we deprecate something:
  - We announce the deprecation as far in advance as possible
  - We display a warning that the functionality is going away in a future release of CUPS
  - We help developers and users migrate to any replacement functionality, if applicable
- Deprecation is a necessary step prior to removal from CUPS
- *Deprecated items are still functional until removed*

CUPS Plenary

CUPS Deprecations, con't

- After a transition period, deprecated items are removed from CUPS
  - Deprecated CUPS APIs are never fully removed from shared libraries (non-functional stubs remain) to preserve binary compatibility
- We've had some hard exceptions over the years:
  - Security issues forced us to do a hard transition of some cupsd.conf directives to cups-files.conf
  - Security issues forced us to drop interface script support
  - Performance and architectural issues forced us to drop CUPS browsing before Avahi was fully supported/deployed

CUPS Deprecations

CUPS 2.2.7: Deprecate Raw Print Queues

- *Raw queues will continue to work in CUPS 2.2.x/2.3.x*
- Why deprecate them?
  - Raw queues pointing to shared printers cause problems for sandboxed applications on macOS and applications using AppArmor/SELinux on Linux (no direct network access)
  - Raw queues pointing to label printers, etc. require applications to provide printer-specific UI and print data, the opposite of what CUPS is about
  - Raw queues do not work with file: device queues, which people still occasionally use with special-purpose printers and software

CUPS Deprecations

CUPS 2.3: Deprecate Printer Drivers

- *Printer drivers and the PPD APIs will continue to work in CUPS 2.3.x*
- PPD files were deprecated in CUPS 1.4 (*eleven* years ago) but we didn't have a replacement strategy for printer drivers at that time
- IPP Everywhere (and related standards) provide the replacement for most printer drivers
  - Strategy for other printers and drivers is to use Printer Applications
- We hope to remove printer driver support in the CUPS feature release following 2.3.x

CUPS Deprecations

CUPS 2.3: Deprecate Printer Drivers, con't

- Why deprecate printer drivers?
  - At least 98% of all printers sold since 2010 support IPP, Apple/PWG Raster, and JPEG; many (about half) support PDF
    - Holdouts are industrial label printers and certain vertical market printers
  - PPDs and drivers have been holding us back from offering a better user experience (ready media, localization, full range of printer options/values), improved document processing, and improved accounting
  - PPDs and drivers are a security and distribution nightmare

CUPS Future

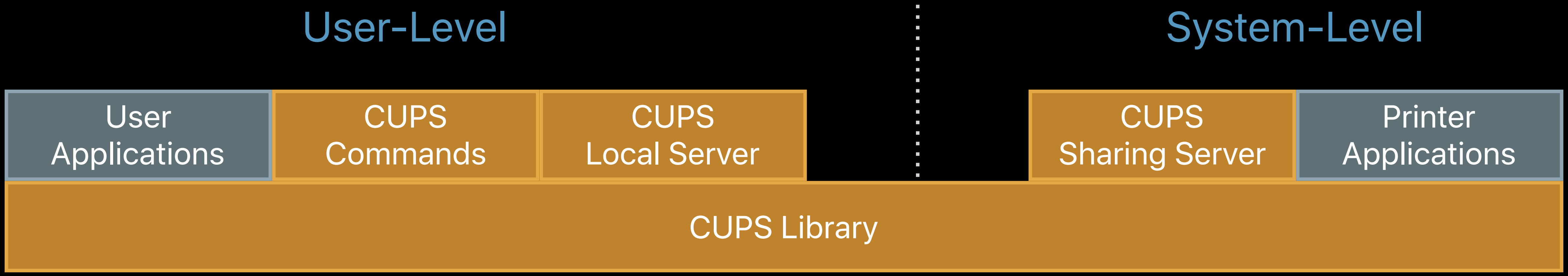
CUPS Future

Topics

- Modular Printing System
- OAuth 2.0
- Printer Applications

CUPS Future

Modular Printing System



CUPS Future

Modular Printing System

- CUPS Commands: lp, lpr, lpstat, cancel, etc.
- CUPS Local Server handles local print requests
  - Only temporary IPP Everywhere print queues
  - Runs as user
- CUPS Sharing Server handles network print requests
  - Full print accounting/ACLs/pre-processing of documents
  - OAuth 2 and PAM-based authentication
  - IPP Shared Infrastructure Extensions/System support
- CUPS Library: libcups, as exists today

CUPS Future

OAuth 2.0

- Replacement for Kerberos SSO
- Doesn't require root access so CUPS Sharing Server can run in a restricted system account
- Also used for common Cloud printing and web-based solutions
- Many open source solutions available, including my own:
  - https://www.msweet.org/moauth
- SAML and Webauthn authentication backends are commonly available, too

CUPS Future

OAuth 2.0, con't

- CUPS implementation in libcups will support OpenID/RFC 8414 compliant OAuth 2.0 authorization servers
  - Authorization server is reported via IPP "oauth-authorization-server-uri (uri)" Printer Description attribute from printer/sharing server
  - Bearer and refresh tokens will be cached per-user/auth-server
  - Authorization UI will be presented using embedded web view - only available when printing from system console
    - Command-line tool for registering bearer token, too

CUPS Future

OAuth 2.0, con't

- CUPS sharing server will use token introspection (RFC 7662) and support OAuth scopes for ACLs
    - Also support multiple authentication methods (e.g. Basic and Bearer) to support username + password and OAuth against a common authentication service

CUPS Future

Printer Applications

- Replacement for printer drivers that looks like an IPP Everywhere printer on the network (or just the local system)
- Simplest applications just needs to accept PWG Raster data, convert it to the printer language (PCL, PostScript ESC/P, etc.) and send it to the device, e.g., USB and AppSocket
- More complex applications can do PDF and produce vector output, e.g., PostScript
- Can be distributed via the Mac App Store, Homebrew project, Linux distribution packages, snapcraft, AppImage, Docker, etc.

CUPS Future

Printer Applications - ippeveprinter

- Replacement for the old ippserver sample code in CUPS
- ippeveprinter has three main ways to setup the printer:
  - Using the old generic printer offered by ippserver (with some minor improvements)
  - Using the newer ippserver attribute files (with bugs fixed)
  - Using a PPD file (limited to PostScript printers)
- Print commands handle document transforms and device communication

CUPS Future

Printer Applications - ippeveprinter, con‘t

- Two print commands are bundled with ippeveprinter:
  - ippevepcl for generic PCL printers
    - Like the HP PCL Laser Printer driver bundled with CUPS
    - Use with either an attribute file or the default attributes
  - ippeveps for PostScript printers
    - Use with PPD file
    - Runs CUPS PDF to PS filter (cgpdftops or pdftops) to convert PDF content
- Other print commands can be created and used

CUPS Future

Printer Applications - ippeveprinter, con't

- Output from commands can be sent to:
  - A network printer via AppSocket ("port 9100")
  - The spool directory or an alternate output directory
- Currently the following features are not exposed (but are planned):
  - Authentication
  - Output through non-root CUPS backends like smb and usb
  - PIN/password printing
  - Web controls, e.g., disable GET supply update requests

CUPS Future

Printer Applications - ippeveprinter, con't

- Why are we providing this application?
  - Great for testing clients and printing issues - you can use ipptool to get a snapshot of a printer's capabilities and then simulate the same printer on your development system
  - Gives driver developers a simple framework for supporting non-IPP printers without legacy PPD baggage
  - Will give everyone more experience with printing using just IPP

CUPS Future

CUPS ippeveprinter vs. PWG ippserver

- ippeveprinter implements a single IPP Everywhere Printer
- ippserver implements the IPP System Service with multiple IPP Printers
    - IPP Everywhere, IPP 3D Printing Extensions, IPP Shared Infrastructure Extensions, etc.
- Print commands written for ippserver will work with ippeveprinter, but the opposite may not be true since ippserver lacks the CUPS backends and filters

CUPS Plenary

Q&A

Questions?

CUPS Plenary

Resources

- CUPS Web Site
  - https://www.cups.org/
- CUPS Repository
  - https://github.com/apple/cups
- CUPS Programming Manual
  - https://www.cups.org/doc/cupspm.html
  - https://www.cups.org/doc/cupspm.epub