



CUPS Plenary

Michael Sweet, Apple Inc.
April 28, 2015

Introduction

- CUPS is the standards-based, open source printing system developed by Apple Inc. for OS X and other UNIX®-like operating systems.
- CUPS 2.0.x is the current stable branch
 - 2.0.3 coming out soon
 - Two more 2.0.x releases planned
- CUPS 2.1.x is the current development branch
 - Beta testing will start June/July 2015
 - Probable 2.1.0 release October 2015

CUPS 2.0 Release History

- CUPS 2.0.0 released October 1, 2014
- CUPS 2.0.1 released November 4, 2014
 - Security: POODLE changes (disabled SSLv3)
- CUPS 2.0.2 released February 9, 2015
 - Security: Buffer overflow in cupsRasterReadPixels

CUPS 2.1

- IPP Everywhere improvements
- Logging improvements
- ipptool improvements
- Initial 3D printer support

CUPS 2.1 - IPP Everywhere

- lpadm and web interface now support automatic IPP Everywhere printer setup:
 - Specify "everywhere" for the model with lpadm:

`lpadm -p name -E -v printer-uri -m everywhere`
 - In the web interface, select "IPP Everywhere" for the printer driver
 - Automatically creates a PPD based on capabilities reported by the printer

CUPS 2.1 - Logging

- ASL (OS X) and journald (Linux/systemd) support
- Allows for simple log filtering, for example to show all of the messages for job 42:

`journalctl job-id=42` [journald]

`syslog -k job-id 42` [ASL]

- (Not yet in the public Git repository...)

CUPS 2.1 - ipptool

- EXPECT-ALL directive to test all attributes of a given name (like in Get-Jobs response)
- NOT-EXPECT directive to allow test failure based on a match (beyond "EXPECT !name")
- WITH-VALUE-FROM predicate to compare against another attribute's value

CUPS 2.1 - 3D Printer Support

- Based on current IPP 3D Printing Extensions white paper
- Just the basic "plumbing" to support 3D printers
 - MIME media types for common stuff
 - 3D "printer-type" capability bit to filter out 3D printers from 2D print dialogs, and vice-versa
 - Some PPD extensions to support materials and capabilities
- No standard filters or camera support
- (Not yet in the public Git repository...)

CUPS Developer “Cheats”

- `#define _CUPS_NO_DEPRECATED 1`
 - Turns off compatibility defines/typedefs for enums
 - Marks deprecated functions and types as unavailable so you get a compile error instead of a warning
- `#define _IPP_PRIVATE_STRUCTURES 1`
 - Makes `ipp_t` structure public
 - Will be removed after *CUPS 2.0.x*
- `#define _PPD_DEPRECATED “”`
 - Turns off PPD warnings

CUPS Future



CUPS Future

- Continue march to ubiquitous printing via IPP Everywhere
 - Focus on PDF, JPEG, and PWG Raster
 - Better status/state reporting
- Investigate alternate (per-user) spooler implementations that just do IPP Everywhere - no printer drivers
 - CUPS server APIs and `ippserver` sample code make this feasible
 - Run in parallel with traditional `cupsd` for driver-based queues

CUPS Future

- Better power support on Linux and others
 - Like we already do on OS X
 - Idle sleep, forced sleep
- Better network awareness on Linux and others
 - Like we already do on OS X
 - Detect network changes, default-printer-per-network
- Better support for 3D Printing/Printers
 - `cupsd` for hosted drivers
 - `ippserver` for embedded implementations

CUPS Future

- Additional discovery/directory service support
 - Bring back LDAP support, this time using the standard schema
 - DNS-SD/mDNS enhancements being discussed in the IETF
 - <http://tools.ietf.org/wg/dnssd/>
- Potential Cloud/infrastructure and release printing support
 - Based on IPP Shared Infrastructure Extensions (INFRA) in the PWG
 - Solves some of the harder network accessibility issues

CUPS Future

- Challenges
 - Can we make these changes transparent to applications, i.e., will we be able to stay binary compatible?
 - Can we provide a consistent user experience on all platforms, i.e., do we have all of the tools/libraries we need for networking, USB, graphics, etc?
 - Can we do this quickly enough?

Resources

- CUPS Web Site
 - <http://www.cups.org/>
- CUPS Roadmap
 - <http://www.cups.org/roadmap.php>

Q&A