CUPS Plenary

Michael Sweet, Apple Inc.

April 24, 2012

Introduction

- CUPS is the standards-based, open source printing system developed by Apple Inc. for OS X and other UNIX®-like operating systems.
- CUPS 1.5.x is the current stable branch
  - Final 1.5.3 release coming out soon
  - Probably one more 1.5.x release
- CUPS 1.6.x is the current development branch
  - Beta testing starting soon
  - Probable 1.6.0 release this summer

Legal Stuff

- Still GPL2/LGPL2
- Still no plans to change to GPL3/LGPL3
- Name of the software and project is officially just “CUPS”
- Legal agreement for significant contributions:
  - http://www.cups.org/AppleContributorAgreement_2011-03-10.pdf
  - Summary: effectively joint copyright on contributions

Major Changes in CUPS 1.6

- Creation of a new CUPS Filters project in Open Printing
- Introduction of new dynamic APIs inspired by IPP Everywhere
- Support for Avahi and colord
- Deprecation of PPD-based APIs and drivers
- Removal of old printer sharing mechanisms: CUPS browsing and the custom LDAP/SLP schema

New CUPS Filters Project

- Filters, backends, and support software that is not used on OS X but is critical for Open Printing is now part of a separate CUPS Filters project within Open Printing
- Other common software remains part of the core CUPS software hosted at cups.org
- Users shouldn’t notice a difference
- The goal is to provide the most up-to-date software required by Open Printing while retaining a stable core for all to use

New Dynamic APIs

- The computing world has changed significantly since 1999 and CUPS 1.0
  - Infrastructure now shared, wireless networks, device capabilities and resources significantly greater
- New dynamic destination APIs that allow programs to:
  - Get a list of available printers based on network visibility as well as traditional local queues
  - Get a list of ready and/or supported media (and other options) for a selected printer
  - Still use cups_dest_t and cups_option_t (easy migration)

New Dynamic APIs

- cupsConnectDest - connects to a destination returning a http_t *; supports callbacks, interruption, and timeouts
- cupsCopyDestInfo - gets supported options, ready media, localizations, and status
- cupsEnumDests - enumerates local and network printers to a callback
  - uses Bonjour for discovery but future versions will also support/use LDAP, SLP, and potentially Cloud
  - cupsEnumDests also supports filtering by capabilities
  - cupsGetDests only gets local printers

New Dynamic APIs

- cupsCheckDestSupported and cupsCopyDestConflict
  - Replace ppdConflicts, ppdInstallableConflict, and cupsResolveConflicts
- cupsGetDestMediaByName and cupsGetDestMediaBySize
  - Replace ppdPageSize and friends
- cupsLocalizeDestOption and cupsLocalizeDestValue
  - Replace ppdLocalize and friends
  - Localizations loaded on demand

New Dynamic APIs

- cupsCancelDestJob, cupsCloseDestJob, cupsCreateDestJob, cupsFinishDestDocument, cupsStartDestDocument
  - Replace corresponding APIs without “Dest” in the name
  - Existing non-dest APIs are not currently deprecated (but probably will be in a future release)

Avahi Support

- Avahi is now supported using its “native” C API
- Based on patches from Red Hat (Tim Waugh) and Till Kamppeter
  - Final implementation much different than original patches, addresses known bugs in Avahi...
- As part of the Avahi integration, also improved the mDNSResponder integration in cupsd
- Supports multiple DNS subtypes, e.g., _cups for CUPS and _print for IPP Everywhere

colord Support

- Based on patches from Red Hat (Richard Hughes)
  - This will show up in the cups.org Subversion repository very soon

Deprecation of PPD-Based APIs and Drivers

- Don’t Panic!
- Deprecation just means that we will no longer enhance the PPD APIs and drivers in CUPS because the functionality offered by PPDs has been superseded by native IPP (Everywhere)
- Existing printers and software will continue to work as before
    - Drivers need #define to remove deprecation warning
- Clients of a CUPS server can use PPD options or standard IPP attributes and values mapped to/from the PPD (since CUPS 1.0, actually)

Removal of CUPS Browsing

- CUPS browsing used UDP broadcasts to announce the availability of a printer
  - Not good for wireless networks since it causes a degradation of performance
- cupsd had to wake up periodically
  - Bad for power management
- Had to broadcast numeric addresses to deal with name resolution issues
- IPv4 only
- Often blocked by corporate networks

Removal of Custom LDAP/SLP

- cupsd used a custom schema for registering/sharing printers via LDAP and SLP
  - Never registered with schema and poorly defined
- The implementations had serious performance and power issues (particularly for SLP)
- Never-ending bug reports (“doesn’t work with my network”) due to configuration problems and site-specific issues
- We’ll return LDAP and SLP support in a future release, this time using the update to RFC 3712 and optimizing for power and speed

Other Changes in CUPS 1.6

- Time-based job history/preservation/cancel
  - Works with “MaxJobs” to provide the desired amount of history
  - By default, users can reprint jobs up to 1 day later
  - History and job file preservation times are separately configurable
  - Held and long-running jobs can be canceled
- PPD files can now be installed as tar files
  - Support gzip in 1.6 (7x smaller than gzip’d PPD files)
  - Tracking future changes to support bzip2 (15x smaller) and xz/lzma (540x smaller!)

Other Changes in CUPS 1.6

- Authentication:
  - CUPS now supports long usernames and passwords (up to 127 characters)
  - cupsd supports “auto” AuthType setting to automatically enable Kerberos
- IPP:
  - Default request version is now 2.0 (was 1.1)
  - IPP APIs moving to opaque structures like HTTP APIs with accessor APIs
    - Structures are hidden by default but can be “revealed” for existing code with a #define

CUPS
Future

CUPS Future

- Continue march to ubiquitous printing via IPP Everywhere
    - Focus on PDF, JPEG, and PWG Raster
    - Ready media, standardization of options and values
    - Better status/state reporting
- Multithreaded cupsd to address performance and functional issues
    - DNS resolution for Internet/Cloud deployments
    - Potentially long-running operations like Get-Jobs
    - Live status/ready media for printers?

CUPS Future

- Launch-on-demand via systemd and others
  - like we already do on OS X via launchd
- Better power support on Linux and others
  - like we already do on OS X
  - idle sleep, ‘dark wake’, forced sleep
- Better network awareness on Linux and others
  - like we already do on OS X
  - detect network changes, default-printer-per-network

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 make this scale from consumer electronics to high-end servers?
  - Can we do this quickly?
  - How do we coordinate with OSS that is not part of CUPS?

CUPS Future

- Timeframe/Schedule
  - No schedule yet
  - Will be planning after CUPS 1.6 is out

Other News

- cups.org is moving!
- Currently running on the old Easy Software Products server
  - Easy Software Products is now closed and all hosted projects are getting moved off this server
- cups.org is moving to an Apple server
  - Existing references to {ftp|svn}.easysw.com will change to {ftp|svn}.cups.org
- Will post announcement on cups.org before the switchover happens

Resources

- CUPS web site:
  - http://www.cups.org/
- CUPS roadmap
  - http://www.cups.org/roadmap.php

Q&A