CUPS Plenary

Michael Sweet, Apple Inc.
May 14, 2013

Introduction

- CUPS is the standards-based, open source printing system developed by Apple Inc. for OS X and other UNIX®-like operating systems.
- CUPS 1.6.x is the current stable branch
  - Final 1.6.3 release coming out soon
  - Maybe one more 1.6.x release after that
- CUPS 1.7.x is the current development branch
  - Beta testing started last month (April 2013)
  - Probable 1.7.0 release this summer

cups.org Web Site

- Old Easy Software Products server failed April 22, 2013
- Currently running a limited version of the site on a temporary Apple server
  - Provides access to released source code tarballs
  - Bug reports/feature requests via bugreport.apple.com
  - Mailing lists, third-party links, and PPD files currently offline
- Plan is to restore our backups onto a new Apple server as soon as possible

CUPS 1.6 Release History

- CUPS 1.6.0 released July 25, 2012
- CUPS 1.6.1 released July 27, 2012
  - Urgent packaging and localization fixes
- CUPS 1.6.2 released March 18, 2013
  - Lots of Avahi, colord, and libusb changes/fixes
  - Localizations
  - Security changes to move file/directory/user configuration to cups-files.conf
- CUPS 1.6.3 coming soon...

CUPS 1.7 Features

- Infrastructure support for paid printing, external quota systems, etc.
  - Requires OS user interface and printer/service support (part of IPP Paid Printing Extensions)
- Automatic data compression support when printing via IPP
- New APIs for IPP Everywhere printing to support printer-specific UI
  - Ready (loaded) media support
  - Enumeration of supported attributes and values
- New APIs for server applications

CUPS 1.7 Developer Changes

- Many existing enumerated types have been renamed for consistency and to reduce programming errors, for example:
  - ipp_status_t enum names now use the IPP_STATUS prefix; IPP_OK is now IPP_STATUS_OK, IPP_BAD_REQUEST is now IPP_STATUS_ERROR_BAD_REQUEST, etc.
- Compatibility defines/typedefs are provided for backwards compatibility (no code changes required)
- Final 1.7 documentation will include a list of enum changes

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 1.7.x
- #define _PPD_DEPRECATED “”
  - Turns off PPD warnings

New Destination APIs

- *cupsFindDestDefault* - find the default option values for a destination
- *cupsFindDestReady* - find the ready (loaded) media, finishings, etc. for a destination
- *cupsFindDestSupported* - find the supported values for each option
- *cupsGetDestMediaByIndex*, *cupsGetDestMediaCount* - get supported/ready media details (dimensions, margins, etc.)
- *cupsGetDestMediaDefault* - get default media details

New User-Agent APIs

- *cupsSetUserAgent* - set the default User-Agent string for HTTP and IPP requests
- *cupsUserAgent* - get the default User-Agent string for HTTP and IPP requests
- User-Agent defaults to:

CUPS/major.minor (OS and version) IPP/2.0

New HTTP APIs

- *httpAddrCopyList* - copy an address list from httpAddrGetList
- *httpAddrPort* - get the port number associated with an address
- *httpAssembleUUID* - make a RFC 4122 urn:uuid: style UUID string
- *httpConnect2* - connect with timeout and cancel support
- *httpSetDefaultField* - set the default value for fields, e.g. Accept-Encoding, Server, User-Agent

New HTTP Server APIs

- *httpAcceptConnection* - accept a new HTTP connection from a listen socket
- *httpAddrListen* - listen for connections on the given address
- *httpGetContentEncoding* - get a common content encoding between client and server
- *httpGetExpect* - get the value of the Expect request header
- *httpPeek* - peek at bytes from the client

New HTTP Server APIs (con't)

- *httpReadRequest* - read a HTTP request from the connection
- *httpWriteResponse* - write a HTTP response to the connection

New IPP APIs

- *ippContainsInteger*, *ippContainsString* - does the IPP attribute contain the given integer or string value?
- *ippCreateRequestedArray* - creates a CUPS array containing all of the attributes that should be returned ("requested-attributes" support for servers)
- *ippGetOctetString*, *ippSetOctetString* - get and set octetString (binary blob/string) values
- *ippAddStringf*, *ippAddStringfv*, *ippSetStringf*, *ippSetStringfv* - add or set string attributes using printf-style format strings

New IPP APIs (con't)

- *ippNewResponse* - create a new ipp_t response message from a given request
- *ippValidateAttribute*, *ippValidateAttributes* - validate one or multiple IPP attributes and their values

New PWG APIs

- *pwgFormatSizeName* - generate a PWG media size name
- *pwgInitSize* - get size and margin information from an IPP request
- *pwgMediaForLegacy*, *pwgMediaForPPD*, *pwgMediaForPWG* - look up dimensions for the given media size name
- *pwgMediaForSize* - look up the PWG media size name for the given dimensions


CUPS
Future

CUPS Future

- Continue march to ubiquitous printing via IPP Everywhere
  - Focus on PDF, JPEG, and PWG Raster
  - 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?
  - Use new server APIs in libcups

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

- Additional discovery/directory service support
  - Bring back LDAP and SLP support, this time using the standard schema
  - DNS-SD/mDNS enhancements being discussed in the IETF
    - http://tools.ietf.org/html/draft-lynn-mdnsext-requirements-01
- Potential Cloud-based printing support
  - Based on IPP Shared Infrastructure Extensions (IPPSIX) in the PWG

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 1.7 is out

Resources

- CUPS web site:
  - http://www.cups.org/
- CUPS roadmap (currently offline)
  - http://www.cups.org/roadmap.php

Q&A