Linux Foundation Collaboration Summit - April 2011 - San Francisco

Attendees

```
Christopher Arnaiz (Kyocera)
Danny Brennan (IBM)
Raghavendra Chitpadi (HP)
Zhenjian Dai (Kyocera)
Daniel Dressler (Independent, GSoC Foomatic Intern - call-in)
Hal Engel (KDE, OpenICC - call-in)
Jochen Faas (EFI)
Till Kamppeter (Linux Foundation/Canonical, OP Manager)
John Layt (KDE printing)
George Liu (Ricoh)
Kevin Luo (Kyocera)
Tim McCann (Konica Minolta)
Ira McDonald (High North/Samsung, OP Chair, PWG IPP WG Co-Chair)
Andrew Mitchell (HP, PWG Cloud Imaging WG Co-Chair)
Bruce Nordman (Lawrence Berkeley National Lab, IETF EMAN Co-Chair)
Jeremy Pennini (Ricoh)
Glen Petrie (Epson - call-in)
Hitoshi Sekine (Ricoh)
Craig Shifman (Konica Minolta)
Mike Sweet (Apple, PWG Chair)
Jerry Thrasher (Lexmark)
Charles Torreyos (Lexmark)
Randy Turner (Almalfi)
Paul Tykodi (TCS, PWG IPP WG Co-Chair - call-in)
Michael Vrhel (Artifex, Ghostscript)
Bill Wagner (TIC, PWG WIMS WG Chair)
Tim Waugh (Red Hat printing - call-in)
Rick Yardumian (Canon - call-in)
```

Preface

- Summary prepared by Ira McDonald (High North, OP Chair) based on:

```
* Conference agenda (see below)
* Presentation slides (see below)
* Discussions during sessions
* Ira's participation in person in all sessions
```

- Summary is archived at:

```
ftp://ftp.pwg.org/pub/pwg/fsg/April2011_OPSummit/
 Open-Printing-Summit-Summary-20110504.htm
```

- Agenda of OPS (with links to presentation slides) is archived at:

```
https://www.linuxfoundation.org/collaborate/workgroups/openprinting/
  openprinting-summit-san-francisco-2011
```

- Recordings of OPS sessions are archived at:

```
http://www.openprinting.org/download/meetingnotes/op-summit-2011/
  OP-Summit-2011-day1-1-20110406.mp3
  OP-Summit-2011-day2-1-20110407.mp3
  OP-Summit-2011-day2-2-20110407.mp3
  OP-Summit-2011-day3-1-20110408.mp3
  OP-Summit-2011-day3-2-20110408.mp3
```

OP/PWG Joint Working Session - OP JTAPI (Glen Petrie, Epson)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  JTAPI.GSOC_.201104.06.pdf
```

Open Printing Job Ticket API/1.0 standard is archived at:

```
ftp://ftp.pwg.org/pub/pwg/fsg/Released-Specifications
  jtapi-version1.0.0.zip
```

OP JTAPI working directories are at:

```
ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/
```

Excerpts from slides and discussion:

- What is JTAPI ?

```
- JTAPI stands for:
  Job Ticket Application Programming Interface
- Pronounced "jay-tappy", "Job Ticket API", or "jay tee API"
```

- Objectives of JTAPI

```
- To create and consume job tickets but not define a new job ticket
- To be job ticket syntax neutral
- To isolate the application from the content of the job ticket
- To be programming language neutral
- To import and export multiple job ticket formats
```

- Status of JTAPI

```
- Completed w/ reference C header files in January 2005
- Approved by Free Standards Group in March 2005
- Published by Free Standards Group in September 2005
```

- JTAPI WG Members (w/ their current affiliations in 2011)

```
- Claudia Alimpich (InfoPrint Solutions - JTAPI WG Chair)
- Jody Goldberg (Gnome)
- Tom Hastings (retired from Xerox)
- Till Kamppeter (Linux Foundation/Canonical, OP Manager)
- Ira McDonald (High North, OP Chair)
```

- Glen Petrie (Epson)

¿ Existing Job Ticket Formats

- PWG Job Ticket
  - Defined in PWG Semantic Model 1.0 (PWG 5105.1-2004)
  - Based entirely on IPP/1.1 (RFC 2911)
  - Open, extensible, XML-based job ticket standard

- CIP4 Job Definition Format
  - Defined by CIP4, an international printing standards body
  - Current version is CIP4 JDF/1.4a (2009)
  - Open, extensible, XML-based job ticket standard
  - Subset is defined in Integrated Digital Printing (IDP) ICS

- Adobe Job Ticket
  - Defined by Adobe and used in Adobe PostScript

- PWG Micro Job Ticket (MJT)
  - Work-in-progress

- Vendor Job Tickets
  - Defined by printer vendors and ISVs

¿ JTAPI Object Model

- Follows model in ISO Document Printing Application (DPA) (ISO 10175)
- Complete abstract model w/ UML relationship diagrams

¿ JTAPI C Header Files

- Each object defined in a separate file
- Common extensible methods for attributes
- Data/object model that is object oriented
- Defines objects that are familiar to the printing industry
  - Job, Document, Insert Sheet, Media, Stitching, Hole Making, etc.
- Defines relationships between objects
- Defines operations to be performed on objects
- Defines attributes of objects
- Defines well-known enumerated values of all attributes

¿ Google Summer of Code 2011 - JTAPI Project - Approach

- Review OP JTAPI model and API in detail
- Review PWG Job Ticket specification
- Create Test Job Ticket
  - Manually create a minimum of 3 representative Job Tickets (text files) to be used for testing and evaluation
- Define the command-line Test Application to exercise the JTAPIs
  - Include an initial set of commands
- Create Thin-Thread implementation of the individual JTAPIs and the Test Application
  - This will be the first demonstrational implementation and the start code for detailed development
  - This will include minimum documentation on how to use the Test Application
- Enhance individual JTAPIs and the Test Application to provide full

```
  functionality
  - Provided update documentation as required
- Project Demonstration
```

Driver Packaging Tutorial (Till Kamppeter, OP Manager)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  PrinterDriverTutorial_0.pdf
```

Excerpts from slides and discussion:

- Why auto-download of distro-independent driver packages?

```
- Distributions do not ship all available printer drivers
- Free drivers from upstream need to be compiled by users
  - Driver installation too complicated for unexperienced users
- Manufacturers make packages only for a few major distributions
- Driver packages often difficult to find on manufacturer's web sites
- Testing/packaging effort for manufacturers and driver developers too
  high to ship binary driver packages for all distributions
```

- Existing Infrastructure we make use of

```
- OpenPrinting database (former linuxprinting.org), central database for
  printer/driver info
- LSB provides tools and infrastructure to create
  distribution-independent binary packages
```

- Solution - Distribution-independent printer driver packages

```
- Based on LSB 4.1 for binary format
- Using CUPS, Ghostscript (with IJS, CUPS Raster and OpenPrinting Vector
  interfaces), Perl, and foomatic-rip which is on every distribution
- LSB DDK (Driver Development Kit) helps packaging the drivers correctly
- Make packages part of OpenPrinting database (or link them at least
  from there), so that they can be easily found
- Infrastructure for automatic package lookup, download, installation,
  and auto update through the internet by printer setup tools and
  package managers
- system-config-printer (Fedora/Red Hat, Ubuntu, Mandriva) already
  supports automatic download of driver packages (with Jockey)
```

- Advantages of Solution

```
- Distribution-independent
  - One package for Linux, instead of one for Red Hat, one for SuSE, one
    for Ubuntu, ...

- Binary packages
  - User does not need to compile, system is also suitable for
    closed-source drivers
```

- Same installation method for all driver packages
  - A printer setup tool can easily install them automatically

- One query location at the OpenPrinting web site
  - Easy to find for both humans and printer setup tools
  - Granting redistribution permissions of non-free drivers is much easier

- Driver query API for printer setup tools
  - All needed info available: License, supplier, support contact, print quality indices
    - So the setup tool and the user can easily find the driver suiting best for him.

- Distributions look up drivers at OpenPrinting
  - Distributions do not need to support all printer models
  - So drivers newer than the distro are available, for updates and for new printer models
  - Distribution CDs do not get overloaded with printer drivers and PPD

¿ Still needed - Manufacturers take full responsibility for their drivers

- Distributions are supposed to download these non-distro packages by default
- Users would make distros responsible if something goes wrong
- Manufacturers should sign a legal agreement to take responsibility

¿ Still needed - Cryptographic code in drivers and export restrictions

- Use only standardized cryptographic technologies which come already with the OS
- Host the driver packages on the manufacturer's site and link only from OpenPrinting
  - Repository on manufacturer's site must be indexed for RPM and DEB (for automatic updates)
  - Repository linked from OpenPrinting web site to allow same look-up and download mechanism as for directly hosted drivers
  - Links on OpenPrinting web site have to be kept up-to-date

¿ Still needed - Signing

- Packages uploaded by manufacturers must be electronically signed

¿ Still needed - Repositories handled like at distros

- main: Drivers of trusted sources (usually manufacturers) who have signed responsibility agreement go here, only from this repository distributions automatically download and install by default (like "main" in the distros)

- contrib: Upload to here does not require signing the agreement, but to automatically download from here the user has to activate this repository (like "contrib", "universe", .) in the distros

¿ Still needed - Maintainer scripts: Only pre-defined procedures

- Pre/post-install/uninstall scripts

- To avoid arbitrary system changes by printer driver packages

- Procedures pre-defined as macros in the LSB DDK
  - Add /opt/<supplier>/... to $PATH
  - Symlink CUPS backends, filters, filter rules, and PPDs installed in /opt to appropriate system directories
  - Update PPDs of existing queues for this driver
  - Set up, start, and restart driver-specific daemons
  - Restart CUPS
  - Clean up all of the above when uninstalling

2011 Major Issues in Linux Printing (Till Kamppeter, OP Manager)

Slides are archived at:

https://www.linuxfoundation.org/sites/main/files/2011MajorIssues.pdf

Excerpts from slides and discussion:

¿ Starting in OP Summit 2006 - Printing Dialog

- GNOME printing dialog was still asking for the print command, no printer list, no options, no CUPS support

- After my feature request on the GNOME list and a flamewar started by Linus Torvalds GTK comes out with a CUPS-supporting dialog in GTK 2.10.x (by the way, 2.10.x finally made it into the LSB in 2011).

- At the OpenPrinting Summit hosted by Lanier (heute Ricoh) in Atlanta OpenUsability started to design the Common Printing Dialog
  - Dialog with Usability in mind
  - Same dialog for all applications and all desktops (KDE, GNOME, Firefox, OpenOffice.org, etc.)
  - Feature completeness to support everything which CUPS supports
  - Also supports application-specific options

- Problems of today:
  - MANPOWER!
  - No volunteers, we need to pay developers
  - No sponsors to pay developers and usability research/design

¿ Starting in OP Summit 2006 - PDF Printing Workflow

- Replacing PostScript as print job format by PDF
  - One can always tell pages apart
  - Transparency and other new graphical objects
  - More compact files

- Filters written by Koji Otani, Tobias Hoffman, Till Kamppeter

- Implemented for the first time in Ubuntu Jaunty Jackalope (8.10, Oct 2008) and at the same time in Debian

```
- Lots of bug fixes (and PDF interpreter improvements) afterwards

- Not yet adopted by Red Hat and SUSE (Red Hat will probably adopt in
  Fedora 16)

- Problems of today:
  - PDF interpreter performance for certain files
  - Filters are contributed by many persons who (and whose employers)
    are copyright owners
    - This requires contributor agreements with Apple and/or hosting
      of CUPS extensions for Linux on OpenPrinting
```

- New issues at OP Summit in 2011 - Color Management

```
- To get same color output quality as commercial OS
```

- New issues at OP Summit in 2011 - Performance

```
- Filters, renderers, and drivers are often too slow, especially on
  complex input files
```

- New issues at OP Summit in 2011 - QA

```
- New versions of programs, especiall of applications have often
  regressions concerning printing, and the printing functionality does
  not get enough tested, for example f-spot crashes when clicking on
  "Print"
```

- New issues at OP Summit in 2011 - MANPOWER!

```
- Difficult to find volunteers, even GSoC students. Important coding
  tasks do not get done:  JTAPI, CPD, SANE in LSB, ...
```

- New issues at OP Summit in 2011 - Device ID Matching

```
- New OP database of all actual IEEE 1284 Device IDs contributed by all
  printer/MFD manufacturers (Tim Waugh, Red Hat)
```

- New issues at OP Summit in 2011 - Support for MFD Functions

```
- New standard framework/approach for using other MFD Functions (Scan,
  Fax, Email, etc.) (Tim Waugh, Red Hat)
```

Status of CUPS (Mike Sweet, Apple, PWG Chair)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  cups-openprinting-april-11.pdf
```

Excerpts from slides and discussion:

¿ Introduction

- CUPS is the standards-based, open source printing system developed by Apple for Mac OS X and other UNIX-like operating systems
- CUPS 1.4.x is the current stable branch
- Final 1.4.7 release coming out in a few months
- CUPS 1.5.x is the current development branch
  - Beta testing starting soon

¿ CUPS Legal Stuff

- Still GPL2/LGPL2
- Still no plans to change to GPL3/LGPL3
- Name of the software and project is now officially just "CUPS"
  - Old logo and long name are going away
- New agreement for significant contributions:
  - http://www.cups.org/AppleContributorAgreement_2011-03-10.pdf
  - Summary: effectively joint copyright on contributions

¿ CUPS 1.5 Changes - Security

- Job/printer/subscription access control
- SSL certificate validation/revocation
- Kerberos changes/simplification
- Web interface configuration option

¿ CUPS 1.5 Changes - Command-line programs

- Help
- Extended information
- Additional feature parity between System V and Berkeley commands

¿ CUPS 1.5 Changes - Bonjour support

- Goal is to add full support for Avahi
- Have patches but not all contributors have signed new agreement

¿ CUPS 1.5 Changes - IPP support

- ipptool
- IPP Everywhere

¿ CUPS 1.5 Changes - PWG Raster support

- PWG Raster == subset of CUPS Raster v2 (compressed)
- Simple changes for existing raster producers:
  - cupsRasterOpen(fd, CUPS_RASTER_WRITE_PWG)
  - Send full page image (no margins)
  - Look at FINAL_CONTENT_TYPE to determine whether to send CUPS Raster or PWG Raster

```
  - Add line to .convs file for "image/pwg-raster", e.g.:
    application/vnd.cups-postscript image/pwg-raster pstoraster
  - New rastertopwg filter for existing CUPS Raster producers
  - imagetoraster filter will be updated with native PWG Raster support
  - Will be sending patches to Artifex for Ghostscript PWG Raster
    support in gdevcups
```

¿ Not for CUPS 1.5

```
- PDF filters
  - Not all contributors have signed the new agreement
  - Still need to do a thorough code/design review
- Remote access to driver resources (ICC profiles, icons, etc.)
  - Need to define a bundling format and address security issues
- ICC support in imageto* filters
  - Out of time
```

¿ CUPS API Changes

```
- ipp_t reference-counted starting with CUPS 1.4.4
  - Resolves a long-standing issue with collections
  - ippDelete only frees memory when the reference count goes to 0
  - Documentation has been updated
- PPD header (<cups/ppd.h>) no longer included from main CUPS header
  (cups/cups.h) starting with CUPS 1.5
  - Existing programs should include both headers, even for prior
    releases of CUPS
```

¿ IPP Everywhere

```
- New standards work being done in the Printer Working Group
  - http://www.pwg.org/ipp
- The future of CUPS
- Printers discovered using Bonjour, LDAP, or SLP, queried and printed
  to using IPP and PDF and/or bitmap files (JPEG or PWG/CUPS Raster)
- Standard IPP job tickets - no PPDs
- Existing network printers and direct-connect printers will continue to
  be supported using CUPS (PPD-based) drivers, with CUPS exposing these
  printers as "IPP Everywhere" printers
- Long-term goal is to eliminate the need for printer drivers, PPD
  files, and complicated printing/driver UI
```

¿ CUPS Future - Overview

```
- Printing has changed a lot since 1999
- People are printing different things and printing less
- Mobile/wireless devices are prevalent
- Applications are a lot smarter
  - and so are printers!
- Need to address changing requirements, capabilities, and use cases
```

¿ CUPS Future - Major changes

```
- Tighter coupling between scheduler, filters, and printer
- Focus on a few key file formats (JPEG, PDF, PWG Raster)
- Focus on "smart" printers/services (i.e., IPP Everywhere, Cloud
  Imaging)
```

```
- List of available printers is dynamic (not a static list)
- Drop support for legacy technologies, formats, protocols, and features
- Greater use of threading and launch-on-demand
```

- 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 1.5 is out
```

Energy Management, Energy Star, EMAN (Bruce Nordman, LBNL)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  printers-and-energy.pdf
```

Excerpts from slides and discussion:

- Paper

```
- Paper in printers/copiers may be larger use of energy than electricity
  - approximately 16 Wh/sheet (data circa 1995)
- Duplexing and n-up imaging important for energy
```

- Power Control Elements

```
- slide of many different power buttons, with no standardization
```

- IEEE 1621 Power Control - Key Elements

```
- 3 basic power states: On, Sleep, Off
- Standard colors for power states
- Sleep metaphor
  - "Wake-up"
- Hibernate
  - form of Off
```

- Energy Star - Test Procedure for Printers/Copiers

```
- Set of active cycles followed by sleep
- Average across test cases
```

```
- Calculation method - see slides
```

¿ Energy Star - Network Connectivity Proxy

```
- Proxy operation
  - PC awake; becomes idle
  - PC transfers network presence to proxy on going to sleep
  - Proxy responds to routine network traffic for sleeping PC
  - Proxy wakes up PC as needed

- Proxy locations
  - Device internal (NIC)
  - Immediately adjacent switch
  - "Third-party" device elsewhere on network

- Proxy protocols
  - ARP, DHCP, TCP, ICMP, SNMP, SIP, ...

- Proxy purpose
  - Reduce power required for idle or sleeping printers (and PCs, etc.)
  - Standard is ECMA-393
  - Includes SNMP, Wake on TCP SYN, ...
```

¿ IETF EMAN

```
- Goal - define basic mechanism to report energy and power data
- Scope - all products, primarily monitoring, include complications not
  applicable to printers
- Power States - 3? 6? 12? 100?
  - Current thinking - IANA registry of sets of states
  - Initially - IEEE 1621, ACPI, DMTF, PWG Power Model/MIB
```

Status of IPP (Ira McDonald and Mike Sweet, PWG IPP WG officers)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  ipp-openprinting-april-11.pdf
```

Excerpts from slides and discussion:

¿ Activities since OPS in April 2010

```
- Published the second edition of the IPP/2.0 specification, which adds
  the IPP/2.2 conformance level for high-end/enterprise printing
- Published the IPP:  Job and Printer Extensions - Set 2 specification
  which includes proof print, saved print, and other related operations
  and extensions (required for IPP/2.2)
- Adopted a charter for the IPP Everywhere project within the working
  group to define standards to support "driverless" and mobile printing,
  scanning, facsimile, and other multifunction services
- Began work on IPP Everywhere with 4 key specifications in development
- and more on the way
```

¿ IPP Version 2.0 Second Edition (PWG 5100.12)

- IPP/2.0 is basically a reboot of IPP that brings together all of the approved IPP standards and extensions under new versions of IPP, loosely grouped as follows:
  - 2.0 for small desktop/SOHO printers
  - 2.1 for medium workgroup printers
  - 2.2 for large workgroup/enterprise printers/copiers

- IPP/2.0 reinforces several key conformance items from IPP/1.1 that were not always followed:
  - HTTP chunking for streamed print jobs
  - HTTP Upgrade for encrypted print jobs
  - HTTP Expect for early request termination (for authentication)
  - Handling of unsupported attribute values, specifically IPP collections and the "noValue" tag

¿ IPP Everywhere

- IPP Everywhere takes IPP/2.0 and adds requirements necessary to support "driverless" and mobile printing, scanning, facsimile, etc.
- Taking a two-phase approach
  - First phase/edition is for printing only
  - Second phase/edition is for multifunction (print, scan, fax, etc.)
- First phase to be completed by Q1 2012
- Four key specifications in the first phase:
  - IPP Everywhere First Edition: umbrella spec
  - IPP Job and Printer Extensions - Set 3: supply levels, geolocation, identification, Kerberos, media selection, ICC profiles, icons, etc.
  - PWG Raster Format: low-level raster format for all printers
  - IPP over HTTPS Transport Binding and "ipps" URI Scheme: new RFC to register the "ipps" URI scheme for secure printing

¿ IPP Everywhere Summary

- Discovery:
  - Bonjour/DNS-SD for local printers
  - LDAP/SLP for printers within an organization

- Transport:
  - IPP/2.0 (of course)

- Document Formats:
  - JPEG for photos
  - PDF for documents on "office" printers
  - PWG Raster for documents on "consumer" printers ("office" and "consumer" are generalizations)

- Job Tickets:
  - copies, finishings, ipp-attribute-fidelity, job-accounting-user-id, job-billing-info, job-mandatory-attributes, job-name, job-password, job-password-encryption, media/media-col, multiple-document-handling, orientation-requested, output-bin, overrides, page-ranges, print-color-mode, print-quality, print-rendering-intent, printer-resolution, sides

- Printer Attributes:
  - media/media-col-ready, media-col-database, printer-geolocation,

```
printer-icc-profiles, printer-icons,
printer-mandatory-job-attributes, printer-organization-name,
printer-organizational-unit, printer-supply,
printer-supply-description, printer-supply-info-uri, printer-uuid
```

Mobile Printing - Google Cloud Print demo (Hitoshi Sekine, Ricoh)

Slides are archived at:

```
<slides not available - to be requested from Ricoh>
```

Excerpts from slides and discussion:

- Live demo - not presented

```
- Wrong printer accidentally shipped without Java application installed
```

- Development process and demo screenshots - slides

```
- Used Google Cloud Print API
  - Very limited print options
- Java application installed on printer
  - Based on Google SDK
- How to extend print options and UI?
  - Not clear
```

Mobile Printing - CPD Mobile (Glen Petrie, Epson)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  CPD.Mobile.201104.06.pdf
```

Excerpts from slides and discussion:

- Objective - Support CPD in the Mobile World

```
- A device may have limited system memory resources
- A device may have limited system processing capabilities (cpu speed)
- A device may have limited display area and display resolution
```

- Objective - Support the CPD in the Embedded World (Optional)

```
- A device may have limited system memory resources
- A device may have limited system processing capabilities (cpu power)
- A device may have limited user interfacing capabilities
- A device has NO display
```

- Print Dialog Hierarchy - Can the Mobile Device (MD) support

```
- a dialog box? (how big?)
- pull down menus?
- hierarchical menus? (same as display on a printer)
- one or more iconic?
- keystroke commands?
- gesturing (double tap to print)?
- a physical button action?
- combinations of the above?
```

- End-User Printing Interface/Intent

```
- Level 1 - "Just Print"
  - The Application determines the print parameters
- Level 2 - "Just Print Predefines"
  - Print As Text
  - Print As Web Page
  - Print As Graphics
  - Print As Photo
- Level 3 - "Print My Way"
  - "Full" Print Options
```

- End-User Printer Setup - iPod Touch Example

```
- Associate WiFi ?? Printer
  - Just as the iPod Touch can "auto join" a WiFi network, it also "auto
    associates" a specific printer within that network.
- Printing can be "Turned On/Off"
- KISS Principle - "No Auto Discovery"
  - Represents a First Stage Capability
  - End-User will have to Enter IP Address
```

CPD - Being Common (Glen Petrie, Epson)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  CPD.MakeCommon.201104.06.pdf
```

Excerpts from slides and discussion:

- Objective - Make CPD Common but Adaptable

```
- Common Applies To:
  - The Application Programming Interface to the CPD
  - The Print Solution or Platform Interface to the CPD
  - The Print Job Ticket objects, elements, attributes and values.
  - The User Print Dialog terms and meaning that don't change.
  - The managed Application, Platform and Print-Vendor extensions
- Adaptable Applies To:
  - The User Interface is based on the Human Interface Guide (HIG) for
    the target Solution, Platform and/or Application
  - The User Interface is scalable based on target Solution, Platform,
    Application and/or User preferences.
  - The User Interface is extensible by the target Solution, Platform,
    Application and/or Print-Vendor.
```

¿ CPD - Current State of Affairs

- CPD is 5 years old
  - Since CPD was identified as a project and need by OpenPrinting:
    - Prototype level new print dialog UI has be proposed and documented
    - Prototype code as been started and is on-going

- CPD in the next 5 years
  - If the need still exist for the CPD to "be common", then,
    - a common approach to the UI is necessary.
  - If the need still exist for the CPD to "have a default UI"; then,
    - a generic CPD UI is necessary.
  - If the need still exist for the CPD to "manage extensions", then,
    - an the establishment of extension registry is necessary
  - If the need still exist for a CPD "approach to print dialog", then,
    - a set of OpenPrinting Guidelines is necessary.

- CPD CANNOT TAKE 5 MORE YEARS

¿ CPD - Approach to being Common

- Identify Applications, Solutions, Platforms and Distro's that will represent the basis of CPD.
- Locate HIG for each Application, Solution, Platform and Distro identified above.
- Document the common and unique HIG factors for and between all above Applications, Solutions, Platforms and Distro's.
- Develop a CPD specification that provides coherence and exceptions of the above Applications, Solutions, Platforms and Distro's.
- Identify new parts only where absolutely necessary.

¿ CPD - OpenPrinting Guidelines

- While the "look-and-feel" of a specific CPD might change, an "OpenPrinting Guidelines" will provide
  - Definition of objects, elements and attributes
  - Definition ranges or the set of extensible and non-extensible values
  - Groupings of related objects, elements and attributes
  - Interrelationships between objects, elements, attributes and their values
  - Constraints between objects, elements, attributes and their values

¿ CPD - Managing Extensions

- Extensions; Application, Print Vendors, Solution/Platform always exist BUT they are unmanaged.
- Who is the only group that is confused? - the Users
- Beyond the OpenPrinting Guidelines there exist the need for a registry of terms, acronyms and values such that anyone wanting to add an extension to CPD MUST use values in the registry.

¿ Next steps for Common Printing Dialog

- Is Linux Foundation in Europe to accept CPD funds from German BSI?

- Till/LF to contact German BSI directly

```
- Finish CPD DBUS libraries ASAP

- Finish CPD UI for *one* target application/platform/printer

- Scope the proposed OP Behavior Guidelines spec

- Telecons/email on the proposed OP Behavior Guidelines spec

- Take CPD to Joint Desktop summit in summer 2011

- Take CPD to mobile conferences to stimulate interest

- MOVE FAST - no more 5 years to half-way stage
```

CPD - Skins (Glen Petrie, Epson)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  CPD.Skins_.201104.06.pdf
```

Excerpts from slides and discussion:

- Objective - Make CPD Common but Adaptable

```
- See "CPD - Being Common" above
```

- Print Dialog Hierarchy

```
- Does being Adaptable mean Chaos !
  - No, it means managed choices
- Which Dialog to Use?
  - If there are two or more instantiations of the dialog which is used?
    - The Applications or The Solution/Platform
  - The User is typically (always) using an application; therefore the
    application has first level priority on the UI appearance.  The
    Solution/Platform dialog is used when the application does not want
    to create (or manage) a print dialog.
    - This does not mean the application can not add extensions to the
      Solution/Platform print dialog in same manor a Print Vendor can.
- What if the Solution/Platform has no print dialog!
  - OpenPrinting will define and create a generic, HIG independent print
    dialog that Applications, Solutions or Platforms can use.  See
    separate slide presentation.
```

- CPD Skins - Will Skins confuse the User?

```
- Don't know? However,
  - terminology will be common!
  - skins can be (more) common on a single solution/platform!
  - skins can be (more) common for application on different
    solutions/platforms
```

- CPD Skins - OpenPrinting Skin Guidelines

```
- While the skin's "look-and-feel" might change, an "OpenPrinting
```

```
Skin Guidelines" will provide
- Definition of objects, elements and attributes
- Definition ranges or the set of extensible and non-extensible values
- Groupings of related objects, elements and attributes
- Interrelationships between objects, elements, attributes and their
  values
- Constraints between objects, elements, attributes and their values
```

Status of Ghostscript (Michael Vhrel, Artifex)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  ghostscript-openprinting-2011.pdf
```

Excerpts from slides and discussion:

- Ghostscript Overview - The Basics

```
- Ghostscript is a document conversion and rendering engine.
- Written in C ANSI 1989 standard (ANS X3.159-1989)
- Essential component of the Linux printing pipeline.
- Dual GPL/Proprietary licensed.  Artifex owns the copyright.
- Source and documentation available at www.ghostscript.com
```

- Ghostscript Overview - Devices

```
- Understanding devices is a major key to understanding ghostscript.
- Devices can have high-level functionality, e.g., pdfwritecan handle
  text, images patterns, shading, fills, strokes and transparency
  directly.
- Devices may be set up to handle only certain high-level operations.
- Graphics library has "default" operations, e.g., text turns into
  bitmaps, images decomposed into rectangles.
- In embedded environments, calls into hardware can be made.
- Raster devices require the graphics library to do all the rendering.
```

- Changes to GS since 2010 Open Printing Summit

```
- New ICC color management added (9.0)
- Free type font rendering as default and new font engine API (9.0)
- Fixes for several issues with CUPs color spaces (9.01)
- High speed halftoningusing SSE2 commands. (9.02)
```

- Upcoming Changes to GS (release 9.03*)

```
- Support for anti-aliasing when source contains transparency (in trunk,
  testing)
- Support for littleCMS2.1 (in trunk, testing)
- Object based color rendering (development started)
- Support for output rendering intent (development started)
- Support for proofing profiles, device link profiles and profile
  override (hopefully)
- Ghostscript is moving to git...
```

- Ghostscript Color Architecture

- Easy to interface different CMM with Ghostscript.
- ALL color spaces defined in terms of ICC profiles.
- Linked transformations and internally generated profiles cached.
- Easily accessed manager for ICC profiles.
- Devices communicate their ICC profiles and have their ICC profile set.
- Operates efficiently in a multithreaded environment.
- Handles named colors with ICC named color profile or proprietary format.
- Color management of Device-N colors.
- Includes object type (e.g. image, graphic, text) and rendering intent into the computation of the linked transform (upcoming)
- Proofing, profile override and device link profiles (upcoming)

¿ Conversion of PS and PDF Color Spaces

- PS and PDF CIE color spaces are converted to ICC forms that the CMM can handle.
- PS mappings are all 1-way: Device to CIEXYZ or CIEXYZ to Device
- Procedural mappings are sampled.
- Because of the multiple matrix operations and procedural mappings, some PS color spaces that do not include MLUTs will give rise to ICC profiles that do include MLUTs.

¿ Profile Cache

- Ghostscript creates ICC profiles from PDF and PS CIE colorspacedefinitions (e.g., CalRGB, CIEABC, CIEDEFG)
- To avoid repeated creations, these profiles are cached based upon a hashcode that is related to the resource ID.
- Cache is designed such that MRU item is at the top of the list.
- Profiles are only released if we are at maximum number (or memory), new request is made and a reference count is one.

¿ Device N color spaces (PDF and PS)

- For Device N output, very simple to provide capability for N-color ICC profile.
- Many desire to have CM with CMYK and to pass additional spot colors unmolested.
- For DeviceNinput color, XPS requires ICC profile. PDF and PS use an alternate tint transform.
- Architecture provides capability to define N-color ICC profile for DeviceN input colors to replace the alternate tint transform if desired.

¿ Halftoning

- Recent inclusion of high speed halftoning with an 8 bit threshold array.
- Makes use of SSE2 128bit registers to operate on 16 pixels at a time.
- Current support in trunk is for monochrome output devices only.
- For release 9.03 we should have in place support for high speed halftoning for CMYK planar devices.

Status of system-config-printer (Tim Waugh,

Red Hat)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  system-config-printer-status.pdf
```

Excerpts from slides and discussion:

- GNOME 3 - Big changes coming

```
- Major UI changes - see slides
```

- GNOME 3 - What about system-config-printer?

```
- New System Settings Printers module meant be simple, not to replace
  system-config-printer
- Lessons learned in system-config-printer can be applied to new
  configuration tools, e.g., choosing the best driver
- Other desktops
```

- GNOME 3 - Printing integration

```
- No need for system-config-printer-applet any more
- Notifications for printer added/removed
- Notifications for job status
- Automatic install of driver packages when printer connected, using
  PackageKit
- System Settings: simple printer and job operations
```

- system-config-printer 1.3

```
- Better driver selection
  - XML rules for constructing preference list
  - Better model name comparison logic (Till Kamppeter)
  - CMD-based PPD elimination (George Liu)
- XML rules for constructing preference list
  - Foomatic's XML database can only speak about Foomatic drivers
  - Aim is to have a database that can speak about any arbitrary driver,
    even those not yet shipped/written:
    1. For the given make and model, build a preference list
       of types of driver
    2. Classify available drivers
    3. Sort them into preferred order
- Better model name comparison logic
  - Normalized "spelled-out" form when comparing names
- CMD-based PPD elimination - the problem:  optional PostScript module
  - When to use PostScript PPD?
  - More generally:
    - How do we know if a PPD requires an optional PDL module to be
      installed?
- CMD-based PPD elimination - the solution:  use CMD field in
  1284DeviceID
  - Device's IEEE 1284 Device ID tells us which command
    sets are supported
  - So match this against the PPD
```

- Roadmap

```
- Expect more developments in GNOME 3
- Manufacturers: lists of correct Device IDs would help!
  - Fuzzy matching is unreliable
  - Drivers need to declare correct Device IDs
```

Joint session with LSB Workgroup (Jeff Licquia)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  lsb-printing-2011.pdf
```

Excerpts from slides and discussion:

- LSB 4.1 Printing Requirements

```
- CUPS 1.2.x full API with http/ipp/ppd functions
- Printing API of Qt 4
- Printing API of GTK 2.10.x (especially CUPS-based printing dialog)
- Renderer/Driver interfaces
  - IJS
  - CUPS Raster
  - OpenPrinting Vector
- Foomatic-rip
- Search path for PPD files
```

- What would be great for next LSB?

```
- SANE - for multi-function devices
  - Not possible for LSB 4.1, due to test suite manpower issues (Jeff)
- D-Bus - for inter-process communication between filters, backends and
  GUI
  - Probable in LSB 4.1, others have already asked (Jeff)
- Udev - for device detection and permission setting
  - Possible in LSB 4.1, but Kernel folks will object (Jeff)
```

Testing Linux printing workflow components (George Liu, Ricoh)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  PrintingQualityControl_Ricoh_0.pdf
```

Excerpts from slides and discussion:

- Testing Ricoh Driver Against New Linux Distributions

```
- Fixed testing/release cycle (to plan for engineering resources).
```

- Release cycle is not in sync with Linux Distributions release cycle.
  (Red Hat: Every 6 months; Open SUSE: Every 8 months;
  Ubuntu: Every 6 months)
- Testing limited to certain Linux Distributions.
  (Fedora, Open SUSE, Ubuntu, RHEL, SLE, etc)
- Only General Available distributions are tested
  (No pre-release distributions)

¿ Testing New Ricoh Printer Drivers

- Focusing on Printer Driver Functionality.
- Print jobs submitted from command line to actual printers.
- Defects specific to Ricoh devices have the highest priority.
  (Usually a must fix)
- Submit bug report to appropriate components in Linux Printing
  Subsystem.
- Work around general problems in Linux Printing Subsystem

¿ Defects in Linux Printing

- Majority of the problems reported during Ricoh's driver testing are
  not specific to Ricoh.
- Some problems still exist today (see slides for examples)

¿ Many New Features in Linux Printing - Are they well tested?

- New PDF Workflow filters
- New versions of PDF rendering and writing libraries.
  - Poppler/Cairo.
- New Ghostscript capabilities
- New foomatic-rip 4.0
- New printer discovery capabilities in CUPS
- Openprinting Distribution Independent driver packages
- Fedora Driver Packages with 1284Device ID tagging

¿ Whom is Testing Which Module? - How is Testing Done?

- Ricoh tests printer driver final format conversion module.
- Not enough to guarantee Linux user a smooth printing experience.
- Other module also need testing
- Need to understand test coverage provided by Linux distributions.

¿ Solution - Every Layer Should Do Printing Test

- Application
- Printing Workflow Filter Chain
- Vendor Specific Printer Driver

¿ Solution - Reference Linux Printing Test Suite?

- Application vendor and Linux Distributions use a Postscript printer to
  validate Printing.
- Compile a collection of documents of application format (Open Office,
  HTML, PDF, Image, Txt, etc) and provide it to Application Vendors to
  test on the Postscript printer.
- Compile a collection of print data files (PDF, Postscript, generated

```
  by application) to Linux Distribution to test on the Postscript
  printer.
- Give the collection of print data files (PDF, Postscript, generated by
  application) to Printer vendor to test their devices.
```

- Solution - Could There be an Effective Spool Print Test?

```
- This idea has been brought up many times before.
- Create printer queue for each driver.
  (print to dev/null or print to file)
- Submit random files with random picked options.
- Verify log or size of ripped file.
```

- Open Issues

```
- Discovery
- Driver Matching
- Installation
```

- New ideas at OP Summit in 2011 - Testing Printing Workflow

```
- Ghostscript sample test files
  - Access to these for printer vendor testing?

- OP repository of sample test files from vendors/users
  - XML metadata to describe edge conditions tested?

- Need AUTOMATED testing in software build/commit processes
  - OP to create guidelines to encourage printer vendors

- Use CUPS ipptool to test actual printers
  - OP to encourage printer vendors to test "last mile" in workflow

- IEEE 1284 Device ID
  - All should use PWG IEEE 1284 Command Set (PWG 5107.2-2010)
    (standard tokens for well-known PDLs)

- IPP Everywhere "printer-uuid" attribute
  - Use to eliminate duplicates (from multiple discovery/networks)
  - Works even for multi-homed printers (w/ multiple IP addresses)
```

colord: Color Daemon (Richard Hughes, Red Hat)

Slides are archived at:

```
https://www.linuxfoundation.org/sites/main/files/
  colord.pdf
```

Excerpts from slides and discussion:

- Slides presented by Tim Waugh (Red Hat)
- Basic principles - Gamut

- Human eye can only capture a certain range of colors
- Devices can only capture or produce a certain range of colors
- Mapping from one color-space to another (RGB to CMYK)
- sRGB vs AdobeRGB vs ProPhotoRGB
- Basic problem
  - Camera (14bit RAW RGB),
  - Display (8bit PNG sRGBish)
  - Printer (CMYK)

¿ Basic principles - ICC Profiles

- Set of data that characterizes a device or color space
- Generic prolles are bad...
- Solution - End-to-end color managed workflow

¿ What is colord: High Level Architecture

- Really, only dealing with device to prolle mapping.
- Provides a DBus API for system frameworks to query
- Provides a persistant database backed store that is preserved across reboots
- Provides the session for a way to set system settings, for instance setting the display prolle for all users and all sessions

¿ Key Concepts

- Qualifiers
  - Already delned by Apple for use in CUPS
- Hard and soft relationships
  - Hard = user set mapping, and default
  - Soft = autogenerated mapping, and used as fallback
- DeviceId is unique to the device, e.g., xrandr-LVDS1
- Object path is an actual remote DBus object for the device
- So we can use any language with a DBus binding to interact with colord devices and profiles

¿ How It Works: Overview

- System daemon
- System activated when required
- PolicyKit to control access to privileged operations
- One SQLite database for the persistent device to profile mappings
- One SQLite database for the virtual devices

¿ Licensing Basics

- daemon is GPLv2+
- libcolord (requires GObject and GIO) is LGPLv2+
- DBus interface has no `linking', so possible for use in proprietary software

¿ What does GNOME Color Manager do?

- Call CreateDevice for each connected XRandr screen.
- Create an ICC prolle lle for each Xrandr device using the EDID (optional).

```
- Call CreateProfile for each prolle found in the home directory.
- For each ::profile-added event check if the EDID md5 metadata matches.
- For each ::device-added event check the device modified property
  (optional).
- For each ::device-added event from a Xrandr device, send the gamma
  ramp to X.
```

Wrap-up (Ira McDonald and Till Kamppeter, OP officers)

No slides

Excerpts from discussion:

- 2011 Major Issues in Linux Printing

```
- Pursue ideas from session (see above)
```

- Common Printing Dialog

```
- Pursue ideas from session (see above)
```

- Testing Printing Workflow

```
- Pursue ideas from session (see above)
```

- Back-to-back OPS and IEEE-ISTO PWG meetings in 2012

```
- Mike Sweet and Ira McDonald will pursue with PWG Steering Committee
```