CUPS Development for 2006-2007

• Current roadmap available on cups.org

• Tentative release schedule:
  - December 2006 - CUPS 1.3, CUPS DDK 1.1
  - June 2007 - CUPS 1.4
  - December 2007 - CUPS 1.5

• Monthly PPD releases from the cups.org printer driver database (CUPS-specific)

• Transition of ESP Ghostscript code/patches to main Ghostscript tree

• Print dialog UI extensions
CUPS 1.3 Highlights

• Kerberos authentication
  - *Existing cupsDoAuthentication() function provides application support transparently (no code changes in apps/toolkits!)*

• Side-channel API
  - *Provides out-of-band communication with backends to get the interface state, get the current 1284 device ID, and do a "soft reset" of the interface*

• New USB backend
  - *libusb and direct Linux kernel interface support*
CUPS 1.3 Highlights

• Scheduler
  – *Printer PPD/attribute cache (faster startup)*
  – *CUPS-Get-PPDs supports ppd-make-and-model and ppd-device-id lookups*
  – *Use poll(), epoll(), or /dev/kqueue instead of select()*
  – *LDAP browsing improvements*

• Web interface
  – *Change the name of auto-detected printers*
  – *Auto-configure printers using SNMP*
  – *Other SNMP improvements (configure, partial probes, etc.)*
  – *Improved on-line help searching*
CUPS DDK 1.1 Highlights

• Support for multi-language PPDs
• New "ppdmerge" utility
  - Allows vendors to produce multi-language PPDs easily from existing single-language PPDs
• New CUPS driver interface
  - Allows vendors to ship PPD compiler source files (*.drv, *.po) instead of/with pre-generated PPD files
• Updated Epson and HP raster drivers
  - Support for Epson impact and newer inkjet printers
  - Additional PJL option support for PCL printers/plotters
CUPS 1.4 Highlights

• DNS-SD a.k.a. Bonjour
  - *Already standard on Mac OS X*
  - *Used for printer discovery, printing, and printer sharing*
  - *CUPS 1.4 implementation will address scaling issues (no more 1 file descriptor per shared printer)*

• Standard IPC between filters and applications
  - *Will allow drivers to prompt the owner of a job with a message and 1-3 action buttons*
  - *Implementation TBD*
  - *Need to be careful about security - we don't want to repeat Microsoft's "Windows messaging" mistakes*
CUPS 1.4 Highlights

• Image printing
  - *Scaling mode that ignores margins*
  - *Multi-page TIFF support*
  - *Improved BMP support*

• Web interface
  - *cupsd.conf templates*
  - *Custom option support*
  - *Other usability improvements*

• CUPS API
  - *New async request interface*
CUPS 1.4 Highlights

• PDF support
  - Use poppler instead of embedded Xpdf code
  - Add support for N-up, page selection, and other printing options
  - pdftoraster (ESP) for direct printing of PDF files to non-PS printers
  - pdftopdf (Japan) for direct printing of PDF files to PDF-capable printers
    - New cupsJCLToPDFInterpreter attribute in PPD file
  - imagetopdf (Japan) for direct printing of image files to PDF-capable printers
  - texttopdf (ESP) for direct printing of text files to PDF-capable printers
CUPS 1.5 and Beyond

• Networking features
  - Listen on an interface
  - Selective sharing on specific interfaces

• Various job processing features
  - RIP to disk
  - Preprocess job files

• Better plain text printing
  - More charsets, XHTML-Print, etc.

• Server plug-ins
  - Custom IPP policies, job accounting, job handling, quotas, etc.
ESP Ghostscript Transition

- Current ESP Ghostscript is based on old 8.15 Ghostscript release
- Current Ghostscript (8.54) is now provided under GNU GPL
- Ideally want to merge ESP Ghostscript changes back into GPL Ghostscript
  - Maintain one version of Ghostscript, not N versions
  - Non-Artifex drivers can be placed in a separate subdirectory for convenience ("addons" and "pstoraster" for ESP Ghostscript)
  - Get current ESP Ghostscript developers involved with GPL Ghostscript development?
Print Dialog UI Extensions

• Driver and application developers want to add their own options/controls/views

• Can't rely on binary plug-ins installed on the local computer since the printer might be shared from another system with a different OS or CPU

• Can't rely on the user or application using the extensions

• Localization is important
  – *So is conforming to the HIG of the local desktop environment*
Print Dialog UI Extensions

Requirements

• Support systems with little memory, CPU, or display

• Support non-graphical printing
  – *Drivers can't depend on the user selecting or entering information*

• Cross-platform
  – *Different operating systems, CPUs, and run-time environments*

• Cross-desktop
  – *KDE and GNOME (at least) on Linux, Aqua on OS X, etc.*

• Support localization/I18N

• Controls/capabilities are TBD
Print Dialog UI Extensions
Embedding in PPD Files

• PPD attributes can have arbitrarily long values, allowing for XML and other types of data to be embedded
• Offers a backwards-compatible solution
• Solves the shared printer problem
• Identifying the UI requirements and the language to use for the UI is the key issue!
Print Dialog UI Extensions

UI Description Languages

• Mozilla's XML User-Interface Language (XUL)
  – Pros: well-established, developer tools available, several implementations
  – Cons: XULrunner is big (25MB on Linux i686, compiled), uses multiple files (may be hard to embed)

• GNOME's Glade
  – Pros: Already available to GTK+ apps via libglade, developer tools available, libglade is LGPL and small
  – Cons: no scripting support, controls have to be hooked up to handlers via signals
Print Dialog UI Extensions

UI Description Languages

• OASIS's UIML
  – **Pros**: Open standard, scripting support, platform/vendor/desktop neutral
  – **Cons**: May be too simple, not widely implemented/accepted, layout of controls not specifically under developer control