Gutenprint: Introduction

- Gutenprint, formerly named Gimp-Print, is a GPL-licensed community-maintained suite of high quality printer drivers.
  - Project started 7 years ago when I bought an Epson Stylus Photo EX from a friend to print photographs and found no Linux driver! I started from Mike Sweet's GIMP Print plugin 2.0 and went from there.
  - Project team is entirely volunteer members of the community.
  - Gimp-Print 4.2, released over 4 years ago, has been very successful. Current release, 4.2.7, remains in wide use.
  - Gutenprint 5.0 is a major rearchitecture, supporting more printers with capabilities beyond the limits of the Gimp-Print 4.2 architecture. We support over 700 printers in 6 families from 30 vendors.
  - Gimp-Print and Gutenprint are portable to other POSIX-compliant operating systems with an ANSI C compiler, including Mac OS X (10.2 up), Solaris, HP-UX, BSD, etc.

Gutenprint: Mission

- Gutenprint's mission is to **provide the best possible print quality consistent with user needs on a wide range of devices to the FOSS community.**
  - This doesn't mean we only care about maximum quality photo/fine art printing – we support the entire range of printing activities.
- Provides raster printer driver API used by CUPS and Ghostscript (IJS) drivers and other printing-related applications.
  - Print plugin for the GIMP (owned/maintained by Gutenprint)
  - Print plugin for Cinepaint (owned/maintained by Cinepaint)
  - PhotoPrint (independent FOSS application for printing photos)
  - QuadTone RIP (independent partially closed source general purpose RIP)

Gutenprint: Community Involvement

- **CUPS**: Gutenprint supplies a CUPS raster driver and PPD files for all supported printers. The PPD files are generated directly from driver data.
  - Mike Sweet has commit privilege to our repository.
  - Gimp-Print 4.2 and Gutenprint 5.0 can coexist on one system.
- **Foomatic**: Gutenprint supplies Foomatic data for all supported printers. As with CUPS, the Foomatic data is generated from information provided by the driver package.
  - Till Kamppeter has commit privilege to our repository.
- Next goal in this area is support for CUPS 1.2 functionality such as enhanced data types and on-the-fly generation of PPD files.

Gutenprint: Community Involvement

- Drive technical improvements in these packages to better meet our needs. Examples:
  - 16-bit data from CUPS. 16-bit images are now widely available (e. g. TIFF); need broader support in FOSS applications.
  - Improved handling of CMYK (and *n*-channel) data in CUPS.
  - On-the-fly PPD file generation: *cups-genppd* utility can serve as a model for this.
    - *cups-genppdupdate* utility can serve as a model for on the fly update of PPD files.
  - Foomatic data auto-generation.

Gutenprint: Technical Challenges

- Limited information availability about printers from vendors.
  - Vendors have different policies about releasing information to developers, but nobody is perfect in this regard. Epson has been helpful!
  - We need information about using all printer capabilities, including margin expansion/full bleed, high speed modes, special paper feeds and handling requirements, etc.
    - “How to put dots on the page”
  - We *don't* need information about color transforms, screening algorithms, and similar technologies.
    - “What dots to put on the page”
    - The community has resources for this, e. g. Raph Levien's EvenTone and EvenBetter screening algorithms.
  - It would be useful to know about ink and drop size characteristics to save time.

Gutenprint: Technical Challenges

- Limitations of PPD files
  - PPD files offer very restricted data types and little control over the organization and presentation of options.
    - CUPS 1.2 offers improvements for enabled applications, but lack of curves and arrays inhibits us from offering full general-purpose RIP capabilities.
    - Offering options such as full bleed that change the imageable area is very unwieldly, as it requires duplicating all available paper sizes.
    - PPD files offer very limited dynamic option availability and no option hiding.
  - No (standard) way to reset PPD file to original defaults.
    - “I'm scared to experiment because I can't just go back to the original settings if I mess up.”
    - Latest version of *cups-genppdupdate* offers this for Gutenprint PPD files.
  - We are using multiple sets of PPD files for basic and advanced functionality, but that's clumsy.

Gutenprint: Technical Challenges

- PPD proliferation
  - No way to use one PPD file for multiple printers without requiring user to look up which PPD file to use.
    - 700 PPD files, but we really need maybe 150.
  - 17 languages to date
    - Duplicate PPD files cause user confusion; some tools don't display the PPD file language making it extra hard to find the right file.
    - CUPS 1.2 should help here.
  - Simplified vs. full featured
  - Foomatic vs. native CUPS
  - Total is close to **50,000** PPD files!
    - *2½ orders of magnitude* more than we need.

Gutenprint: Technical Challenges

- Presentation of options (user interface)
  - Gutenprint offers a wide variety of options that most users don't need, and that even power users will generally calibrate infrequently.
    - Not just “set and forget” – e. g. as printers age, linearization curves change.
  - Gutenprint 5 API offers tools for managing presentation (what kind of control, what data type, what options are available, grouping, how advanced a particular control is).
  - We have no user interface experts on the team.
- **Hard rule**: it must be possible for advanced users to have full control over the output (we're not going to entirely get rid of useful options just because they're hard for a non-expert).
  - Individual options are open for negotiation based on technical merit.

Gutenprint: Technical Challenges

- Color management
  - Multiple solutions exist (LittleCMS, Argyll). Which to use?
  - Profiles for other (e. g. OEM) drivers won't give good results.
    - Color transforms and linearization are different.
  - Is the driver the right place for this?
    - Yes! We could generate *n*-channel output transparently from RGB/CMY(K).
    - No! If CUPS provides this (and 16-bit, *n*-channel output), we don't need to duplicate the code and it may be less efficient.
  - Other projects (Cinepaint, PhotoPrint) already provide color managed printing using the Gutenprint engine, but does this scale?

Gutenprint: Organizational Challenges

- Developer burnout
  - I've been leading this project for almost 7 years on a time-available volunteer basis. I'm getting burned out!
  - Other developers have dropped out, mostly after the 4.2 release.
  - We've had a few very good new people join the project, but not enough to generate a lot of momentum.
  - Key technologies (color management, EvenBetter Screening, full general-purpose RIP capability) have not been implemented due to lack of developer cycles/expertise.
- No UI expertise on the team

Gutenprint: Wishlist

- Move beyond PPD files to a more flexible configuration mechanism – will need support from spoolers and applications.
- Vendors provide sufficient information to permit the community to write drivers for printers.
  - We believe that open specifications and community involvement are of more value than OEM closed source drivers as they permit improvements to be leveraged across all printer makes and all system architectures.
  - Vendor source code/developer contributions would be welcome.
  - Access to printer samples is important.
  - Vendors should compete on printer functionality and quality (improvements in inksets, papers, mechanisms, performance, and so forth).