



# Printing on the Desktop

Till Kämpeter, OpenPrinting

Joint Meeting of **OpenPrinting** and **Desktop Architects**  
on the **Linux Foundation Collaboration Summit**

Austin, Texas, April 10, 2008





## The Dialog

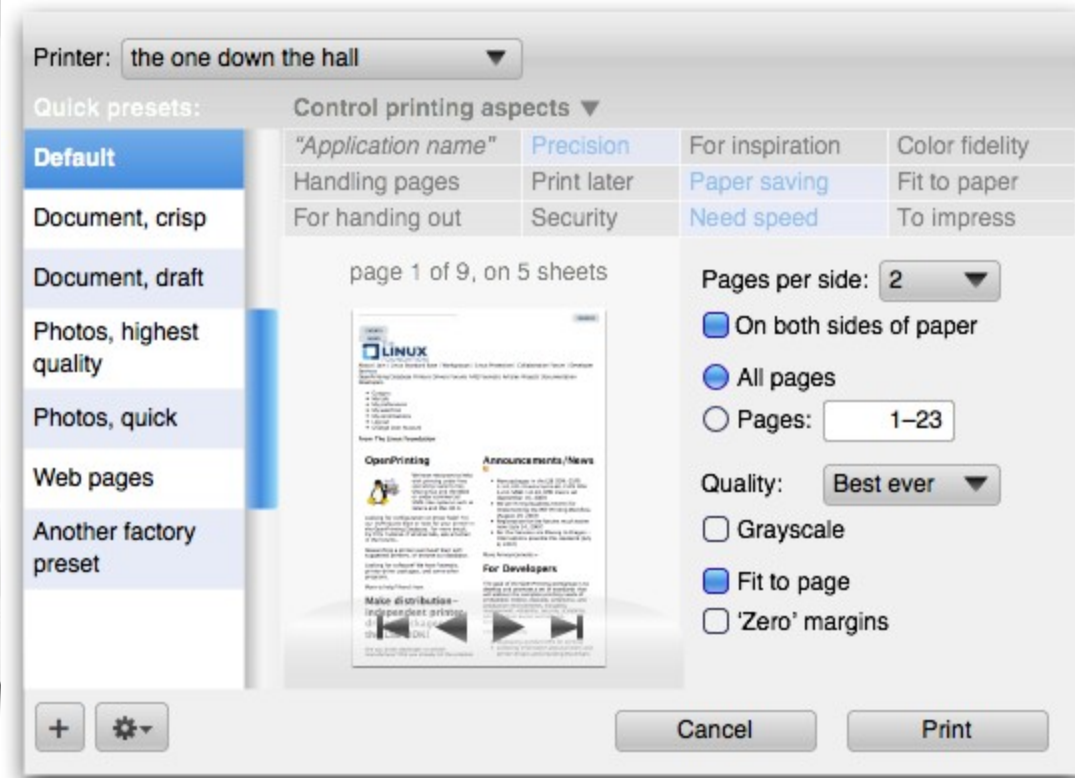
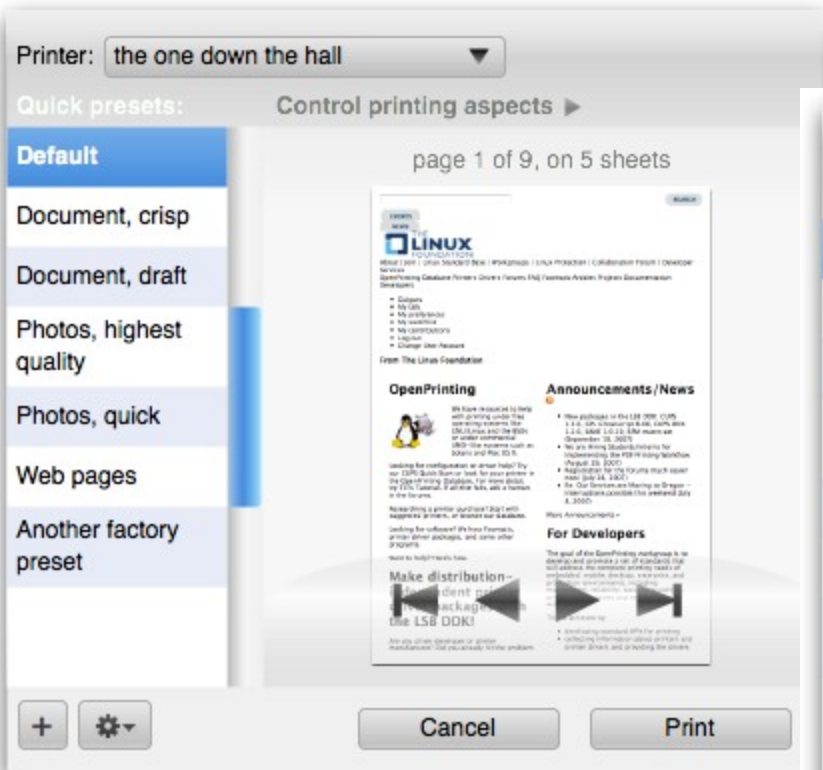
- **Designed by a team of OpenUsability in cooperation with OpenPrinting**
- **One dialog for all desktops and all applications**
  - User knows how to print from app A, then he knows how to print from app B
- **Dialog provides access to full printer functionality but is not cluttered**
  - By default, only basic functionality is shown: Preview, buttons for common printing tasks/qualities
  - On demand detailed options can be displayed by task
- **Dialog will be provided by the desktop**
  - On a KDE desktop always the KDE version appears, also if called from a GNOME application



# Screenshots

## Standard view

## Tag list and some options





- **The design is completed**
- **Now we need to write specifications**
  - For the **desktop GUI developers** to code the dialog
  - For the **desktop infrastructure developers** to write the interface code for the applications
  - For the **driver developers/printer manufacturers** to tag the driver options and to define the common printing tasks and their settings
  - The specs will go **beta** on the **Common Printing Dialog Kick-Off Meeting** in Berlin **end of May**
  - The specs will get **released** before the **OpenPrinting Meeting in Tokyo** on **July 8**
- **This summer the dialog will go live**
  - **Two students** have applied for coding the dialog in the **Google Summer of Code 2008**
  - Coding either by one student or by two, one KDE, the other GNOME
  - Two mentors: Josef Spillner, expert on dialog auto-generation, and Jonathan Riddell, expert on KDE/GNOME dual-UI apps



- **Interface between the Common Printing Dialog and the application**
  - Portland? DAPI? D-Bus?
  - Which data to exchange between app and dialog?
  - Which format for data about app-specific options? PPD-like?
  - How should the dialog return the user settings on the application-specific options? Simple key-value pairs?
  - Interface between app itself and code to call dialog: Let app use API of already existing dialog to avoid modifications in app
  - Who will write this part of the specs?



## Part II: PDF as Standard Print Job Format

- **We agreed on the OSDL Printing Summit in Atlanta April 2006 on switching to PDF as standard print job format**
- **Advantages**
  - PDF is the common platform-independent web format for printable documents
  - Portable
  - Easy post-processing (N-up, booklets, scaling, ...)
  - Easy Color management support
  - Easy High color depth support (> 8bit/channel)
  - Easy Transparency support
  - Smaller files
  - Linux workflow gets closer to Mac OS X



- **Replacing CUPS filters `psto . . .`, `. . . tops`, and `pstops` by `pdfto . . .`, `. . . topdf`, and `pdftopdf`**
  - Most filters implemented by OpenPrinting Japan workgroup
  - `texttopdf/utf8topdf` and `pdftoijs` to be implemented by a Google Summer of Code student
  - Getting all these into CUPS
- **Making `foomatic-rip` (universal wrapper for printer drivers) PDF-aware**
  - Implementation by Linux Foundation intern, nearly completed
- **Modularizing of Ghostscript built-in drivers**
  - Debashis Panda (Kyocera) is working on this together with a Google Summer of Code student



- **Applications must switch to PDF, too**
  - What is the state of the art here?
    - KDE
    - GNOME
    - OpenOffice.org
    - Other desktops/applications
- **Switchover on printing system side planned this year, after Google Summer of Code**
  - Switchover of all applications not required



# Anything Else? Questions?

**Anything else concerning desktop printing?**

**Any questions?**