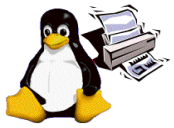
# OpenPrinting Vector Printer Driver API

Printing Summit Lexington 2006

Osamu MIHARA <mihara.osamu@fxpsc.co.jp>
OpenPrinting WG
Fuji Xerox Printing Systems Co., Ltd.



# Printer Driver & Objectives

## Printer Driver API is:

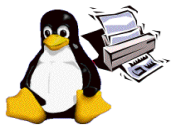
- A printer driver interface for requesting driver/printer information and accepting/printing print jobs.
  - Vector API's supporting PDL based printers
  - Raster API's supporting Raster based printers

## Printer Driver API contains:

- Commands to query/set capabilities
- Commands to create and control print jobs
- Vector / Raster transfer commands

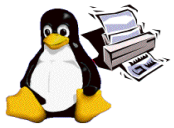
## Objectives

- To be a common interface for printing to printers
- To isolate the application from the details of individual printers
- To isolate the application from the details of individual PDLs
- To have printer drivers support a set of common job properties
- Performance Optimization
  - Achieve full speed printing
  - Utilizes graphical acceleration feature supported by printer controllers



# Printer Driver API

- Job Control
  - Open/Close driver
  - Set Job/Document/Page attributes
- Graphics State Operation
  - Set attributes for each graphics objects
- Drawing Operations
  - Path
  - Text
  - Bitmap Image
  - Scanline
  - Raster Image
- Stream Data (embedded PDL)



# Printer Driver API – Details (1)

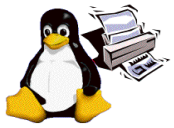
- Printer Context
  - OpenPrinter()
    - Create printer context
    - Register API entry pointers
    - Specify file descriptor for data stream
  - ClosePrinter()
    - Closes printer context
    - Driver releases all resources

- Job Control
  - A print job consist of documents.
  - A document consist of pages.
  - StartJob(), EndJob()
  - StartDoc(), EndDoc()
  - StartPage(), EndPage()
  - Job, doc and page attributes are specified by each StartXxx() function.

- Query Device Capabilities & Information
  - QueryDeviceCapability()
    - Query if the device can do number-up, duplex, etc.
    - Information such as media size, media source and etc. which are supported by the device can be retrieved.
  - QueryDeviceInfo()
    - Query current settings of the device.

- Graphics State Object Operations
  - Graphics State is managed as GS object
    - Operation to GS – InitGS, SaveGS, RestoreGS
  - Controls to each items in GS
    - CTM (Coordinate Translate Matrix)
    - Color Space
    - Raster Operation – ROP3
    - Fill Mode – even/odd or winding
    - Alpha Constant
    - Line Style – width, dash/solid, cap, join
    - Paint Mode – opaque or transparent
    - Stroke and fill color – brush control
    - Foreground and background color – solid brush

- Path Operations
  - A path is a virtual track object
    - Will be visible by stroke or fill operations
    - Will be used to define clip region
  - Lines, rectangles, polygons, arc/pie and bezier are all treated as “path.”
  - Operations:
    - NewPath() – Declare start of a path
    - EndPath() – Declare end of a path
    - StrokePath(), FillPath(), StrokeFillPath() – make visible path
    - SetClipPath(), ResetClipPath() – defines clip region by current path



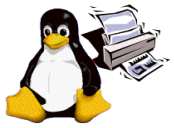
# Printer Driver API – Details (2)

- Text Operations
  - Still under investigation…
  - Current DrawBitmapText() will be removed.
  - Text Operations will includes:
    - Define and Query font metrics
    - Device Font Utilization
    - Font Downloading

- Bitmap and Scanline Operations
  - Bitmap is a bit oriented image data drawn in rectangle region
    - DrawImage()
    - StartDrawImage(), TransferDrawImage(), EndDrawImage()
  - Scanline is a horizontal line defined by start and end point pairs.
    - Used to draw graphics rendered by renderer
    - StartScanLine(), ScanLine(), EndScanLine()

- Raster Image Operations
  - StartRaster(), TransferRasterData(), EndRaster()
  - Set to be extended by Raster Team

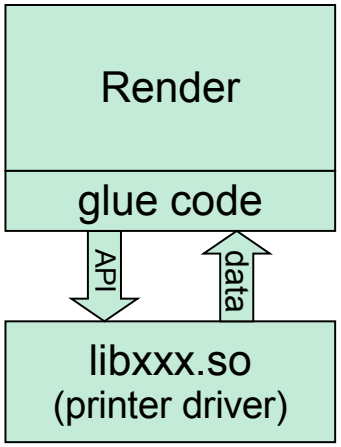
- Stream Data Operations
  - StartStream(), TransferStreamData(), EndStream()



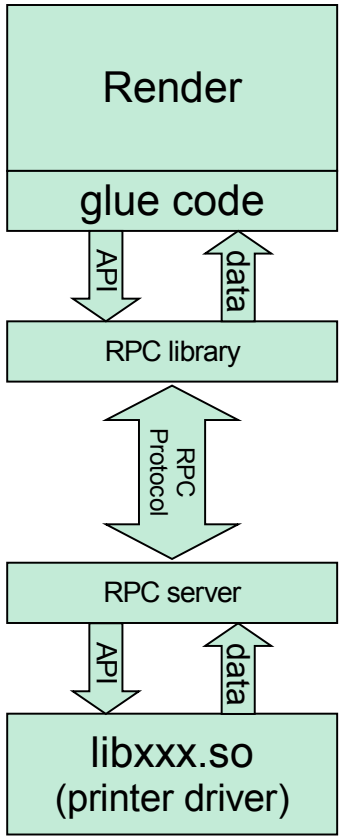
# Linking

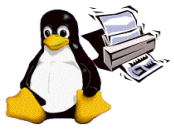
- Printer driver is provided as a dynamic library.
- Driver can be linked dynamically or via RPC.

direct linking
R: GPL
D: GPL
*or*
R: MIT
D: Closed or LGPL




RPC linking
R: any
D: any

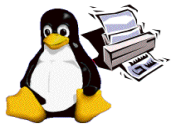




# PDAPI Raster- Common Job Properties

- Features
  - Standardized name for common features
  - Standardized keys and the values
  - An extensible paradigm for non-standard features
  - Coherence across the FSG OpenPrinting model

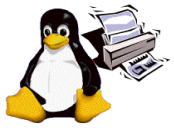
- ColorInput
- ColorOutput
- Copies
- Margins
- MediaBackCoating
- MediaColor
- MediaFrontCoating
- MediaInputTrayName
- MediaSizeName
- MediaType
- MediaUnprintableMargins
- NumberUp
- NumberUpPresentationDirection
- OutputBinName
- PrintQuality
- Resolution
- Rotation
- ScalingType
- ScalingPercentage
- SheetCollate
- Sides
- StitchingPosition
- StitchingReferenceEdge
- StitchingType
- StitchingCount
- StitchingAngle
- Trimming



# PD-Vector Working Group Information

- To subscribe to FSG Vector Printer Driver mailing list:
  - http://freestandards.org/mailman/listinfo/printing-japan
- To post a message to FSG Vector Printer Driver mailing list
  - printing-driver@freestandards.org
- To view FSG Vector Printer Driver mailing list archives
  - http://freestandards.org/mailman/listinfo/printing-japan
- To find FSG Vector Printer Driver documents
  - ftp://ftp.pwg.org/pub/pwg/fsg/vector/
- Participants

| | |
|---|---|
| Osamu Mihara | FUJI XEROX Printing Systems Co. Ltd. |
| Yamagishi Toshihiro | Turbolinux, Inc. |
| Koji Otani | AXE Inc. |
| Toratani Yasumasa | Canon Inc. |
| Ide Kentaro | SEIKO EPSON CORPORATION |
| Shinpei Kitayama | EPSON Avasys CORPORATION |



# Update for Version 1.0

- Currently working for formal release as Version 1.0.
- Changes from 0.2:
  - Document License: FDL to MIT
  - Symbols have “fsgpd” prefixes.
  - Tentative font operation is removed (no font support yet – sorry!)
  - OpenPrinter() now handles API spec version.
  - Change of parameters of raster functions (DrawImage(), StartDrawImage())
  - Scheme for Job/Doc/Page attribute: support of UPDF become mandatory.
  - Support of KRGB for inkjet devices
  - Many other fixes.
- GS meta driver (opvp) will be updated when Version 1.0 when it is available. Driver developers are encouraged to apply version 1.0.