OPVP

— *in relation to other printing related technology*

Hin-Tak Leung

htl10@users.sourceforge.net

Epson EPL project

Introduction

- Who am I?
  - Bought “wrong” Epson printer one year. . . Long story. . .
  - History with LaTeX/CJK, knows a bit about fonts and postscript
  - WTS, and HP CLJ 3600
- familiar with IJS, and interested in OPVP because of the Japanese connection.

Apologies if there are mistakes in this slide set!

ghostscript + printer-specific code

Approaches:

- patch
- uniprint (Postscript? snipplets)
- omni (XML, shared libraries) URL2
- pipe (unidirectional)
  - pnm2xxx
  - CUPS Raster
- ijs (bi-directional pipes) URL2
- shared libraries
  - X11, lvga256 and vgalib in ESP Ghostscript
  - opvp/oprp (+bidi pipes in rpc use)

What about Wine and/or using Windows drivers?

Why?

- Extra dependencies

  X11: libXt.so, libSM.so, libICE.so, libXext.so, libX11.so, libXau.so, libXdmcp.so

  lvga256, vgalib: libvga.so, libvgagl.so

  hpijs: libstc++.so
- faster and independent development and release cycle
- separate IP (closed-source, incompatible licensing terms, etc)
- cancellation / retry / status /consumable-level query

Is GPL/LGPL appropriate?

Shared Libraries

- Disadvantages:
  - not portable to non-unix platforms
  - compiler/linker specific (gcc)
  - C entry point (slightly awkward C++ support)
  - API-version specific
  - ABI/alignment sensitive
- Advantages:
  - Same memory space – no unnecessary copying

recompile for compatibility (need source code!)

IJS

- Disadvantages:
  - copying scan-lines (even white ones!)
- Advantages:
  - possibility of widely different build, or even non-C/C++ – based driver
  - separate memory space — protect client process

SkipRaster, CopyRaster?

CUPS Raster

> “Because doing a CUPS raster driver allows your driver to take full advantage of the CUPS filter architecture and work on more platforms. It is also faster (both at run-time and when developing drivers...)”
>
> *Michael Sweet, Wed Jan 4 2006*

- rastertolabel (rastertodymo)
- rastertoepson
- rastertohp

Dependencies versus convenience. Issue with pipes: buffering, lack of co-ordination between renderer and driver. (needs rest of cups or foomatic)

IJS

“IJS does offer the advantage of a very small print client which can be directly embedded in applications (for embedded solutions). The IJS service concept allows for complete separation of proprietary driver software from GNU open-source - important to some developers. The IJS service concept also allows for simple update by driver software independent of the client application and/or print spoolers.
I hope that even though IJS is not yet version 1.0, that it continues to be supported by the Linux print community.”
Glen Petrie, Mon Jan 30 2006

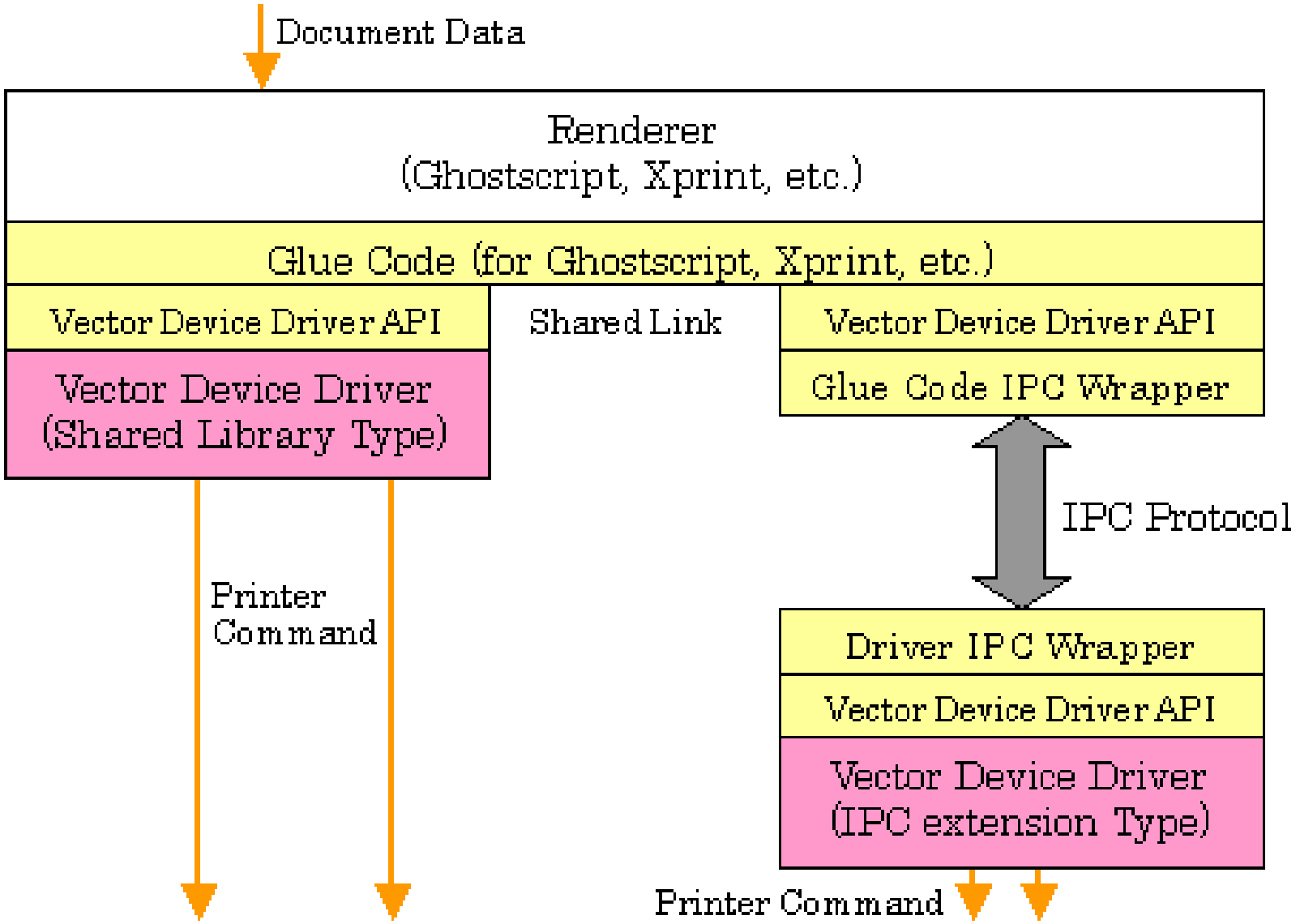
X11, lvga256, vgalib

Load every library from a built-in location (`/usr/local/lib/ghostscript`):

espgs/src/gp_unix.c

```
dir = opendir(GS_DEVS_SHARED_DIR);
if (dir == 0) return;

while ((dirent = readdir(dir)) != 0) {
  strncpy(pbuff, dirent->d_name, sizeof(buff) - (pbuff - buff) - 1);
  if ((handle = dlopen(buff, RTLD_NOW)) != 0) {
    if ((gs_shared_init = dlsym(handle, "gs_shared_init")) != 0) {
  (*gs_shared_init)();
    } else {
    }
  }
}
```

OPVP


Document Data
Renderer
(Ghostscript, Xprint, etc.)
Glue Code (for Ghostscript, Xprint, etc.)
Vector Device Driver API
Vector Device Driver
(Shared Library Type)
Shared Link
Vector Device Driver API
Glue Code IPC Wrapper
IPC Protocol
Printer
Command
Driver IPC Wrapper
Vector Device Driver API
Vector Device Driver
(IPC extension Type)
Printer Command

OPVP

Loading from sys locations (or $LD_LIBRARY_PATH):

espgs/addons/opvp/gdevopvp.c

```
if ((h = dlopen(list[i],RTLD_NOW))) {
        OpenPrinter = dlsym(h,"OpenPrinter");
        errorno = dlsym(h,"errorno");
        if (OpenPrinter && errorno) {
                handle = h;
                break;
        }
        OpenPrinter = NULL;
        errorno = NULL;
}
```

Similiar to Windows GDI — overloaded API methods in the same memory space.

DLL loading in Gsview

	windows	unix
gsdll_open()	LoadLibrary()	dlopen()
gsdll_sym()	GetProcAddress()	dlsym()
gsdll_close()	FreeLibrary()	dlclose()

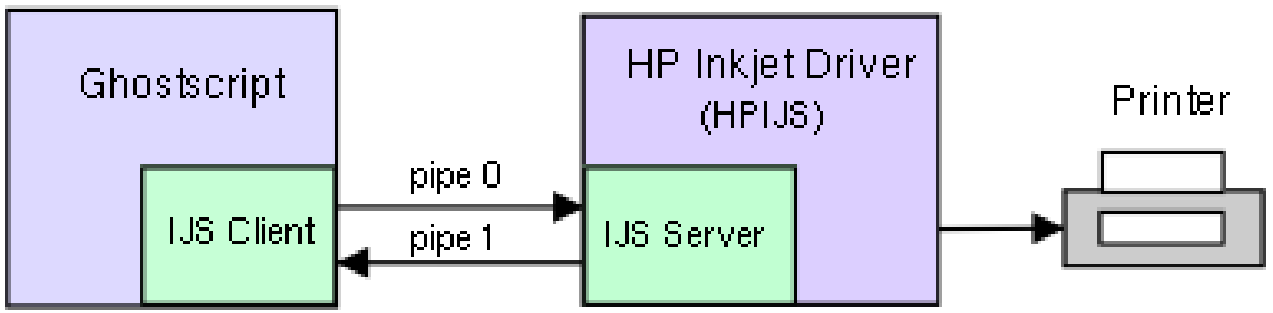
gsview/src/cdll.c

```
/* Load GS DLL if not already loaded */
int gsdll_load(GSDLL *dll, const char *name)
{
    gsapi_revision_t rv;
    ...
    if (gsdll_open(dll, name)) {
    ...
    }

    /* Get pointers to functions */
    dll->revision = (PFN_gsapi_revision) gsdll_sym(dll, "gsapi_revision");
    ...

    /* check DLL version */
    if (dll->revision(&rv, sizeof(rv)) != 0) {
        gs_addmessf("\nUnable to identify Ghostscript DLL revision...\n");
        ...
    }
    dll->revision_number = rv.revision;
    ...
}
```

IJS — Architecture

V. 0.35, attempts to 1.0:

- IJS 1.0 spec August 2003 draft
- IJS 1.0 spec Sept 2003 draft at EITC (defunc)



Duplicating file descriptors for 2 pipes and `fork()` — somewhat non-portable.

IJS on windows

ijs_exec_win.c

```
...
CreatePipe(&hChildStdinRd, &hPipeTemp, ...);
DuplicateHandle(GetCurrentProcess(), hPipeTemp,
        GetCurrentProcess(), &hChildStdinWr, ...);
...
CreatePipe(&hPipeTemp, &hChildStdoutWr, ...);
DuplicateHandle(GetCurrentProcess(), hPipeTemp,
        GetCurrentProcess(), &hChildStdoutRd, ...);
...
CreateProcess(server_cmd, ...);
...
```

IJS on unix

ijs_exec_unix.c

```
...
pipe (fds_to);
...
pipe (fds_from);
...
child_pid = fork ();
...
if (child_pid == 0)
  {
    ...
    dup2 (fds_to[0], STDIN_FILENO);
    dup2 (fds_from[1], STDOUT_FILENO);
    ...
    status = execvp (argv[0], argv);
    ...
  }
```

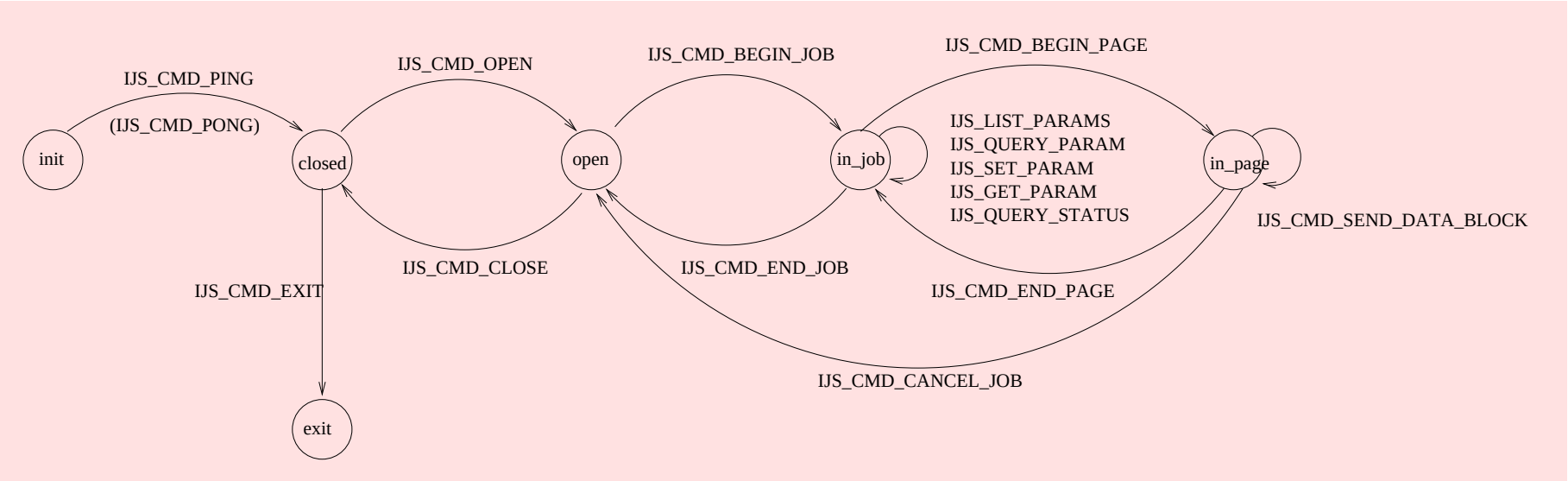
OPVP — RPC mode

opvp_rpc_client.c

```
/* communication pipe */
...
static int outPipe[2];
static int inPipe[2];
...
int OpenPrinter(int outputFD, char *printerModel, int *nApiEntry,
    OPVP_api_procs **apiEntry)
{
   ...
   pipe(outPipe);
   ...
   pipe(inPipe);
   ...
   serverPid = fork();
   ...
   if (serverPid == 0) {
      close(outPipe[1]);
      close(inPipe[0]);
        execlp (SERVERNAME,SERVERNAME,...);
   } else {
        close(outPipe[0]);
        close(inPipe[1]);
        close(outputFD);
   }
   ...
   oprpc_waitReady(rpcHandle);
   ...
}
```

OPVP/RPC uses shared memory, pull/push. Very similiar to IJS

IJS — protocol



ijs.h

```
typedef enum {
  IJS_CMD_ACK,            IJS_CMD_NAK,    IJS_CMD_PING,           IJS_CMD_PONG,
  IJS_CMD_OPEN,           IJS_CMD_CLOSE,  IJS_CMD_BEGIN_JOB,      IJS_CMD_END_JOB,
  IJS_CMD_CANCEL_JOB,
  IJS_CMD_QUERY_STATUS,
  IJS_CMD_LIST_PARAMS,    IJS_CMD_ENUM_PARAM,  IJS_CMD_SET_PARAM, IJS_CMD_GET_PARAM,
  IJS_CMD_BEGIN_PAGE,     IJS_CMD_SEND_DATA_BLOCK,                IJS_CMD_END_PAGE,
  IJS_CMD_EXIT
} IjsCommand;
```

Vector Formats

- Adobe Postscript
- HP-GL/2 + RTL
- PCL XL (a.k.a. PCL 6)
- PDF
- Windows Metafile
- Macintosh PICT
- SVG

and more. . .

PCL XL Reference Protocol Class 3.0

91 instructions

BeginSession	ReadStream	SetFillMode	SetPathToClip	Pie
EndSession	EndStream	SetFont	SetCharSubMode	PiePath
BeginPage	ExecStream	SetLineDash	BeginUserDefinedLineCaps	Rectangle
EndPage	RemoveStream	SetLineCap	EndUserDefinedLineCaps	RectanglePath
Comment	PopGS	SetLineJoin	CloseSubPath	RoundRectangle
OpenDataSource	PushGS	SetMiterLimit	NewPath	RoundRectanglePath
CloseDataSource	SetClipReplace	SetPageDefaultCTM	PaintPath	Text
BeginFontHeader	SetBrushSource	SetPageOrigin	ArcPath	TextPath
ReadFontHeader	SetCharAngle	SetPageRotation	SetColorTrapping	BeginImage
EndFontHeader	SetCharScale	SetPageScale	BezierPath	ReadImage
BeginChar	SetCharShear	SetPatternTxMode	SetAdaptiveHalftoning	EndImage
ReadChar	SetClipIntersect	SetPenSource	BezierRelPath	BeginRastPattern
EndChar	SetClipRectangle	SetPenWidth	Chord	ReadRastPattern
RemoveFont	SetClipToPage	SetROP	ChordPath	EndRastPattern
SetCharAttributes	SetColorSpace	SetSourceTxMode	Ellipse	BeginScan
SetDefaultGS	SetCursor	SetCharBoldValue	EllipsePath	EndScan
SetColorTreatment	SetCursorRel	SetNeutralAxis	LinePath	ScanLineRel
BeginStream	SetHalftoneMethod	SetClipMode	LineRelPath	Passthrough
				VenderUnique

PDAPI SPEC 1.0RC20060711

74 instructions

Creating and Managing Print Contexts [2]

fsgpdOpenPrinter	fsgpdClosePrinter

Job Control Operations [9]

fsgpdStartJob	fsgpdAbortJob	fsgpdStartDoc	fsgpdStartPage	fsgpdQueryDeviceCapability
fsgpdEndJob		fsgpdEndDoc	fsgpdEndPage	fsgpdQueryDeviceInfo

Graphics State Object Operations [35]

fsgpdResetCTM	fsgpdQueryROP	fsgpdSetLineDash	fsgpdSetLineJoin
fsgpdSetCTM	fsgpdSetROP	fsgpdGetLineDash	fsgpdGetLineJoin
fsgpdGetCTM	fsgpdGetROP	fsgpdSetLineDashOffset	fsgpdSetMiterLimit
fsgpdInitGS	fsgpdSetFillMode	fsgpdGetLineDashOffset	fsgpdGetMiterLimit
fsgpdSaveGS	fsgpdGetFillMode	fsgpdSetLineStyle	fsgpdSetPaintMode
fsgpdRestoreGS	fsgpdSetAlphaConstant	fsgpdGetLineStyle	fsgpdGetPaintMode
fsgpdQueryColorSpace	fsgpdGetAlphaConstant	fsgpdSetLineCap	fsgpdSetStrokeColor
fsgpdSetColorSpace	fsgpdSetLineWidth	fsgpdGetLineCap	fsgpdSetFillColor
fsgpdGetColorSpace	fsgpdGetLineWidth		fsgpdSetBgColor

Path Operations [14]

fsgpdNewPath	fsgpdFillPath	fsgpdSetCurrentPoint	fsgpdRectanglePath
fsgpdEndPath	fsgpdStrokeFillPath	fsgpdLinePath	fsgpdRoundRectanglePath
fsgpdStrokePath	fsgpdSetClipPath	fsgpdPolygonPath	fsgpdBezierPath
	fsgpdResetClipPath		fsgpdArcPath

Bitmap Image Operations [4]

fsgpdDrawImage	fsgpdStartDrawImage	fsgpdTransferDrawImage	fsgpdEndDrawImage

Scan Line Operations [3]

fsgpdStartScanline	fsgpdScanline	fsgpdEndScanline

Raster Image Operations [4]

fsgpdStartRaster	fsgpdTransferRasterData	fsgpdSkipRaster	fsgpdEndRaster

Stream Data Operations [3]

fsgpdStartStream	fsgpdTransferStreamData	fsgpdEndStream

Implemented in `opvp_common.h`, `OPVP_api_procs`

PCL XL Reference Protocol Class 3.0

91 instructions

BeginSession	ReadStream	SetFillMode	SetPathToClip	Pie
EndSession	EndStream	SetFont	SetCharSubMode	PiePath
BeginPage	ExecStream	SetLineDash	BeginUserDefinedLineCaps	Rectangle
EndPage	RemoveStream	SetLineCap	EndUserDefinedLineCaps	RectanglePath
Comment	PopGS	SetLineJoin	CloseSubPath	RoundRectangle
OpenDataSource	PushGS	SetMiterLimit	NewPath	RoundRectanglePath
CloseDataSource	SetClipReplace	SetPageDefaultCTM	PaintPath	Text
BeginFontHeader	SetBrushSource	SetPageOrigin	ArcPath	TextPath
ReadFontHeader	SetCharAngle	SetPageRotation	SetColorTrapping	BeginImage
EndFontHeader	SetCharScale	SetPageScale	BezierPath	ReadImage
BeginChar	SetCharShear	SetPatternTxMode	SetAdaptiveHalftoning	EndImage
ReadChar	SetClipIntersect	SetPenSource	BezierRelPath	BeginRastPattern
EndChar	SetClipRectangle	SetPenWidth	Chord	ReadRastPattern
RemoveFont	SetClipToPage	SetROP	ChordPath	EndRastPattern
SetCharAttributes	SetColorSpace	SetSourceTxMode	Ellipse	BeginScan
SetDefaultGS	SetCursor	SetCharBoldValue	EllipsePath	EndScan
SetColorTreatment	SetCursorRel	SetNeutralAxis	LinePath	ScanLineRel
BeginStream	SetHalftoneMethod	SetClipMode	LineRelPath	Passthrough
				VenderUnique

PDAPI SPEC 1.0RC20060711

74 instructions

Creating and Managing Print Contexts [2]

fsgpdOpenPrinter	fsgpdClosePrinter

Job Control Operations [9]

fsgpdStartJob	fsgpdAbortJob	fsgpdStartDoc	fsgpdStartPage	fsgpdQueryDeviceCapability
fsgpdEndJob		fsgpdEndDoc	fsgpdEndPage	fsgpdQueryDeviceInfo

Graphics State Object Operations [35]

fsgpdResetCTM	fsgpdQueryROP	fsgpdSetLineDash	fsgpdSetLineJoin
fsgpdSetCTM	fsgpdSetROP	fsgpdGetLineDash	fsgpdGetLineJoin
fsgpdGetCTM	fsgpdGetROP	fsgpdSetLineDashOffset	fsgpdSetMiterLimit
fsgpdInitGS	fsgpdSetFillMode	fsgpdGetLineDashOffset	fsgpdGetMiterLimit
fsgpdSaveGS	fsgpdGetFillMode	fsgpdSetLineStyle	fsgpdSetPaintMode
fsgpdRestoreGS	fsgpdSetAlphaConstant	fsgpdGetLineStyle	fsgpdGetPaintMode
fsgpdQueryColorSpace	fsgpdGetAlphaConstant	fsgpdSetLineCap	fsgpdSetStrokeColor
fsgpdSetColorSpace	fsgpdSetLineWidth	fsgpdGetLineCap	fsgpdSetFillColor
fsgpdGetColorSpace	fsgpdGetLineWidth		fsgpdSetBgColor

Path Operations [14]

fsgpdNewPath	fsgpdFillPath	fsgpdSetCurrentPoint	fsgpdRectanglePath
fsgpdEndPath	fsgpdStrokeFillPath	fsgpdLinePath	fsgpdRoundRectanglePath
fsgpdStrokePath	fsgpdSetClipPath	fsgpdPolygonPath	fsgpdBezierPath
	fsgpdResetClipPath		fsgpdArcPath

Bitmap Image Operations [4]

fsgpdDrawImage	fsgpdStartDrawImage	fsgpdTransferDrawImage	fsgpdEndDrawImage

Scan Line Operations [3]

fsgpdStartScanline	fsgpdScanline	fsgpdEndScanline

Raster Image Operations [4]

fsgpdStartRaster	fsgpdTransferRasterData	fsgpdSkipRaster	fsgpdEndRaster

Stream Data Operations [3]

fsgpdStartStream	fsgpdTransferStreamData	fsgpdEndStream

PCLXL vs OPVP

- PCL XL is not C-like (not function-prototype based): variable number and types of arguments; Arguments named and typed.
- PCL XL is unidirectional (no `Get*` functions).
- PCL XL contains fonts related APIs.
- PCL XL contains operators for composites, masks, patterns, and can use these for Fill and Stroke.
- opvp_draw_image()/TransferDrawImage() Don't support TIFF?

espgs/addons/opvp/opvp_common.h

```
/* Image Formats */
typedef enum    _OPVP_ImageFormat {
        OPVP_iformatRaw                 = 0,
        OPVP_iformatRLE                 = 1,
        OPVP_iformatJPEG                = 2,
        OPVP_iformatPNG                 = 3
} OPVP_ImageFormat;
```

OPVP — missing gs hooks

espgs/addons/opvp/gdevopvp.c, #define opvp_procs{}

```
        NULL,   /* sync_output */\
        NULL,   /* tile_rectangle OBSOLETE */\
        NULL,   /* draw_line OBSOLETE */\
        NULL,   /* get_bits */\
        NULL,   /* map_cmyk_color */\
        NULL,   /* get_xfont_procs */\
        NULL,   /* get_xfont_device */\
        NULL,   /* map_rgb_alpha_color */\
        NULL,   /* get_alpha_bits OBSOLETE */\
        NULL,   /* copy_alpha */\
        NULL,   /* get_band */\
        NULL,   /* copy_rop */\
        NULL,   /* draw_thin_line */\
        NULL,   /* image_data */\
        NULL,   /* end_image */\
        NULL,   /* strip_tile_rectangle */\
        NULL,   /* strip_copy_rop */\
        NULL,   /* get_clipping_box */\
        NULL,   /* begin_typed_image */\
        NULL,   /* get_bits_rectangle */\
        NULL,   /* map_color_rgb_alpha */\
        NULL,   /* create_compositor */\
        NULL,   /* get_hardware_params */\
        NULL,   /* text_begin */\
        NULL,   /* finish_copydevice */\
        NULL,   /* begin_transparency_group */\
        NULL,   /* end_transparency_group */\
        NULL,   /* begin_transparency_mask */\
        NULL,   /* end_transparency_mask */\
        NULL    /* discard_transparency_layer */\
}
```

OPVP — missing gs hooks

gs/src/gdevpsdf.h

```
...
/* Complete distiller parameters. */
typedef struct psdf_distiller_params_s {
/* General parameters */ ...
/* Color processing parameters */ ...
/* Color sampled image parameters */ ...
/* Grayscale sampled image parameters */ ...
/* Monochrome sampled image parameters */ ...
/* Font embedding parameters */ ...
} psdf_distiller_params;
...
/* Define the extended device structure. */
#define gx_device_psdf_common\
        gx_device_vector_common;\
        psdf_version version;\
        bool binary_ok;         /* derived from ASCII85EncodePages */\
        bool HaveCFF;\
        bool HaveTrueTypes;\
        bool HaveCIDSystem;\
        double ParamCompatibilityLevel;\
        psdf_distiller_params params
...
}
```

New high-level device `ps2write`.

X11/Xprint

OPVP does not have:

- Widgets (N/A)
- Events/Callbacks (N/A)
- Composite/Transparency
- Font metric, font enumeration of built-in fonts
- client side fonts (xfs/fontconfig)

Examples

```
gs -r600 -sDEVICE=opvp -sDriver=libopvpnull.so -sModel=model_name -sJobInfo="job_info" \
-q -dBATCH -dSAFER -dQUIET -dNOPAUSE -sOutputFile=- \$1

gs -sPAPERSIZE=a4 -dFIXEDMEDIA \
-sProcessColorModel=DeviceGray -dBitsPerSample=1 \
-sDEVICE=ijs -sIjsServer=ijs_server_epsonepl \
-sDeviceManufacturer=Epson -sDeviceModel=EPL5700L \
-sIjsParams="" \
-dIjsUseOutputFD \
-dNOPAUSE -dSAFER -dBATCH \
-sOutputFile=epl_test_5700L.epl epl_test.ps
```

Reference Client Implementations

- OPVP:
  - ghostscript (ps $\Longrightarrow$ opvp)
  - openprinting xprint (X11 $\Longrightarrow$ opvp) dev halted?
  - opvppdf (pdf $\Longrightarrow$ opvp)
- IJS:
  - ghostcript (ps $\Longrightarrow$ ijs)
  - ijs_client_example (pnm $\Longrightarrow$ ijs)

Reference Server Implementations

- OPVP:
  - opvpnull (opvp $\Longrightarrow$ Text Description)
  - opfc-ModuleHP (opvp $\Longrightarrow$ PCL 5C): HP ColorLaserJet 4600 and 5500
- IJS:
  - ijs_server_example (ijs $\Longrightarrow$ ps)

IJS-based drivers

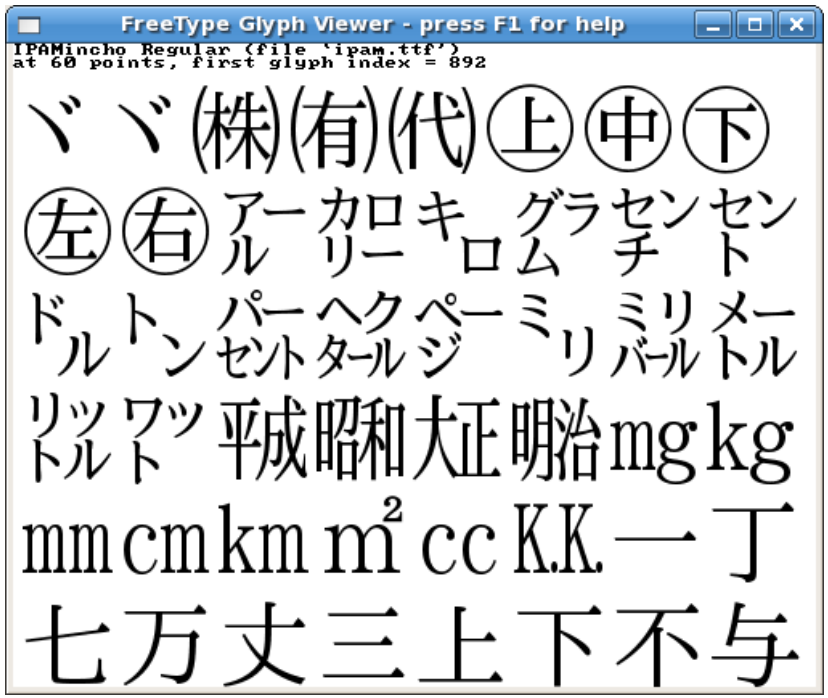
- HP Linux Imaging and Printing (HPLIP)
- Gutenprint/Gimp-Print
- Epson 5x00L/6x00L Printer Driver
-
- Viewplus Tech. Tiger Braille Embosser series for the blind?
- Official Epson IJS (defunct)

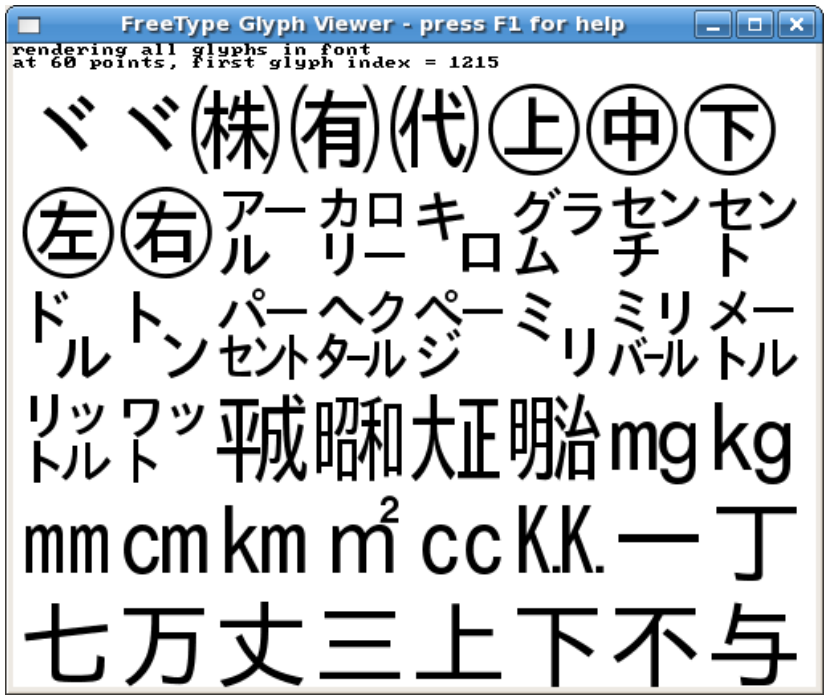
OpenPrinting: Supported printers

- Canon LIPS
- Epson ESC-Page Color
- HP PCL5C
-
- Fuji Xerox/NEC NPDL RPC type.
- Canon (LIPS4 Printers/Copiers)
- Canon (LIPS LX Printers/Copiers)
- Canon (CAPT Printers)

IPA Japanese fonts

5 good quality fonts available for use.





- Postscript, PCL, ESC/Page printers have built-in font sets.
- Printers catering for the japanese market have extra built-in fonts.

Summary

OPVP needs:

- Cross-platform DLL loading
  - e.g. Russell Lang's Gsview uses `libgs.so` on unix and `gsdll32.dll` on windows.
  - e.g. Apache Portable Runtime
  - standardizing on library locations
- font related APIs c.f. SVG
- protocol version negotiation?
- UI/Capability DLLs?
- OPRP does 300dpi x 300 dpi only??

Summary

IJS can benefit from:

- skip-scan-line (oprp has SkipRaster)
- copy-scan-line
- faster transport mechanisms (shared memory?)

Prologs: debugging

What do people do for debugging?

- IJS used to use stdin/stdout for IO.
  - Epson EPL uses `stderr`.
  - HPIJS uses `syslog()`, and some part can also writes to `/tmp`.
- shared libraries shouldn't write to stderr?

References

Avasys (Epson)'s OpenPrinting Project page
Opfc on Sourceforge Japan
Freestandards printing-driver mailing list archive
IJS at linuxprinting.org
Sourceforge OpenPrinting
PDAPI Specs
HP Linux Inkjet Driver Project
FSG OpenPrinting Initiative
Ghostscript
foo2zjs: a linux printer driver for ZjStream protocol
HP PPA GhostScript printfilter
ghostscript documentations on Device drivers