

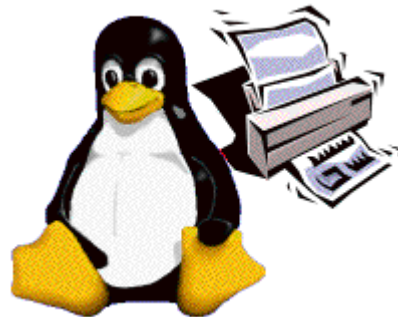


**Free
Standards
Group**

Open Printing



*By: Glen W. Petrie
Senior Software Architect
EPSON*





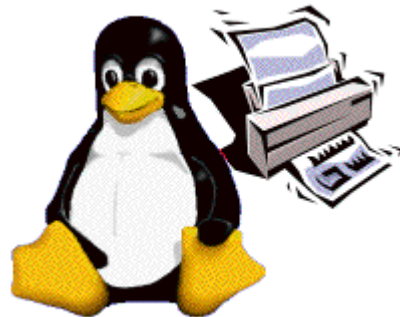
- Introduction
- Working Group Reviews
 - ✗ Architecture
 - ✗ Job Ticket
 - ✗ Application Interface
 - ✗ Driver – Vector/Raster
 - ✗ Status Monitoring
 - ✗ Print Channel Monitor
- Questions





**Free
Standards
Group**

Introduction





Organization

Introduction



Free Standards Group



Open Printing

America/Europe

Japan

Steering Committee

Steering Committee

Architecture

Vector Driver

Job Ticket

Status Monitoring

Application Interface

Print Channel Monitor

Raster Driver





Objectives/Mission

Introduction

“Standardizing on a Scalable Print Environment in Linux.”

Mission Statement

The goal of the Open Printing is to

develop and promote a set of standards

that will address the needs of

desktop to enterprise-ready printing;

including

management, reliability, security, scalability, printer feature access and network accessibility.

<http://www.openprinting.org/>





❏ OpenPrinting : Input

- ✘ OpenPrinting accepts input from interested parties in IT industry, government, education and the open source community

❏ OpenPrinting : Output

- ✘ Draft specification and implementation released together
- ✘ Public review of work
- ✘ Once implementation has been accepted by *both* upstream authors and at least two Linux distribution vendors, then it is “real”:

A published specification that points to a globally utilized, open source implementation



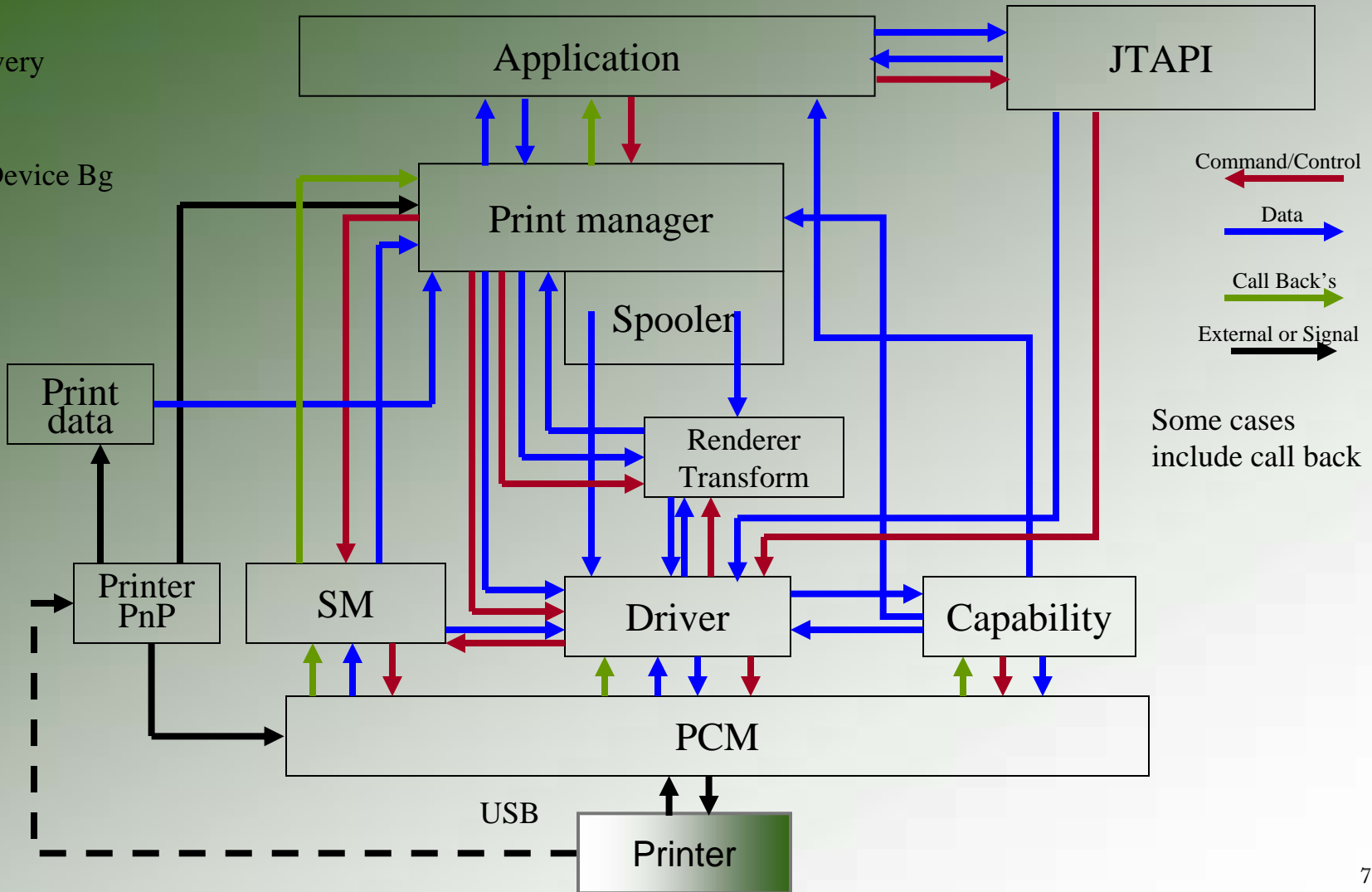


Open Printing Reference Model

Preliminary Reference Model

Introduction

Discovery
UPDF
PPD
FSG Device Bg





Working Groups and Objectives

Introduction

- Steering Committee
 - ✘ Provide overall priorities and top-level coordination.
- Architecture
 - ✘ Develop a modern Print System for Linux.
- Job Ticket API (JTAPI)
 - ✘ To create/consume job tickets; edit job tickets; write/export job tickets.
- Application/Program Interface (PAPI)
 - ✘ Provide applications print services neutral interface.
- Printer Driver (Raster/Vector) (PDAPI)
 - ✘ Provide a neutral interface for printing to any printers
- Status Monitoring API (SMAPI)
 - ✘ Provide a neutral interface for acquiring static and real-time printer status
- Print Channel Monitor API (PCMAPI)
 - ✘ Provide a neutral interface for any output portal/destination.





- Oct 25-26, 2001 – Print Summit Meeting
 - ✗ Began discussion of Open Printing and needs.

- Planned ■ June 10, 2003 – FSG Portland Face-to-Face Meeting
 - ✗ Defined Reference Model

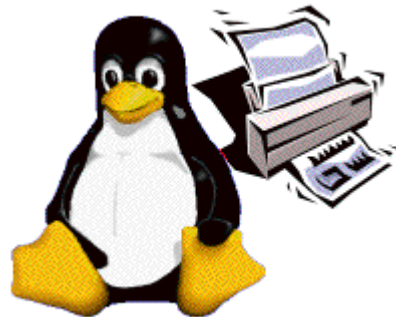
- Nov 21, 2004 – FSG San Antonio Face-to-Face Meeting





Free
Standards
Group

Steering Committee





Steering Committee & Objectives

Introduction

■ Steering Committee

- ✗ Divided into two major teams
 - ◆ US/Europe
 - ◆ Japan/Asia

■ Objectives

- ✗ Provide oversight to Open Printing Activities
- ✗ Provide direction setting to Open Printing Activities
- ✗ Provide coordination between Open Printing Working Groups
- ✗ Set priorities for Open Printing Activities
- ✗ Release Specification/API from Open Printing Working Groups
- ✗ Report to FSG Board on Open Printing Activities





Steering Committee Information

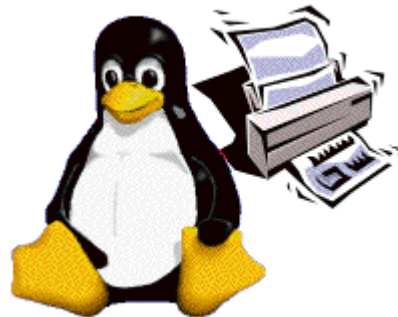
- Monthly FSG Steering Committee conference calls
 - ✦ First Monday of each month at 1:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Steering Committee mailing list:
 - ✦ <http://freestandards.org/mailman/listinfo/printing-sc>
- To post a message to FSG Steering Committee mailing list
 - ✦ printing-architecture@freestandards.org
- To view FSG Steering Committee mailing list archives
 - ✦ <http://freestandards.org/mailman/listinfo/printing-sc>
- To find FSG Architecture documents
 - ✦ ftp://ftp.pwg.org/pub/pwg/fsg/steering_committee
- Participants
 - Japan/Asia
 - ✦ Takaaki Higuchi (Sun)
 - ✦ Osamu Mihara (FUJI XEROX Printing Systems)
 - ✦ Keisho Shida (Canon)
 - ✦ Yasumasa Toratani (Canon)
 - US / Europe
 - ✦ Mark Hamzy (IBM)
 - ✦ Norm Jacobs (Sun)
 - ✦ Ira McDonald (High North Inc)
 - ✦ Glen Petrie (Epson)





**Free
Standards
Group**

Architecture





What is Open Printing Architecture ?

Introduction

■ Open Printing Architecture is a:

- ✗ Reference Model of the overall solution
- ✗ High-Level Architecture abstracting interfaces between components
- ✗ Detailed Architecture defining interfaces between components

■ Architecture elements contain:

- ✗ Reference Model (RM)
 - ◆ Overall system diagram
 - ◆ Functional decomposition into and description of subsystems.
 - ◆ Identification of data and/or control and/or interface between subsystems and/or external systems.
- ✗ High-Level Architecture (HLA)
 - ◆ A glossary of terminology.
 - ◆ Analysis and documented Use-Cases and requirements.
 - ◆ Identification of applicable/recommended standards for subsystem interfaces.
 - ◆ Identification of recommended infrastructure for integrating products.
- ✗ Detailed Architecture (DA)
 - ◆ Specification of system structure (classes, packages, associations – using UML).
 - ◆ Specification of system behavior (activity and sequence diagrams – using UML).
 - ◆ Identification of the system process structure.
 - ◆ Identification of inter-process communication mechanisms.

■ An Open Printing Architecture:

- ✗ Guides development of abstract interfaces.





Open Printing Architecture Objectives

Introduction

- Develop a modern Print System for Linux
- Document realistic Use-Models
- Extract and analyze requirements
- Identify applicable, existing, interface specifications for the requirements
- Apply to multiple print services or be print service neutral.
- Complete and document Reference Model, High-Level Architecture, and Detailed Architecture.





Use-Cases/Use-Models

Example Details Exceptions Diagram Requirements

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

1. Mobile Printing

Mobile printing by reference with document data transformation.

2. PDA Printing

PDA printing directly with document content.

3. Desktop Personal (Consumer) Printing

Print to low end inkjet printer from an application.

4. Desktop Small-Office/Home-Office Printing

Print to mid-volume laser printer, office inkjet and impact printers.

5. Office Desktop Printing

Document from an application is printed on a printer shared by 3-5 workplaces.

6. Central (Print Room) Office Printing

Mid to large volume printing from document repository with data transformation and post finishing

7. Desktop File (Direct) Printing

Print to print device directly form the file-manager/shell-script without going through an application.

8. Pay-for-Print Printing

Print using a job ticket from a print driver through a print spooler to a mid-range color laser printer.

9. Production Printing

Large volume transactional printing to high-speed production printer.

10. Graphics Art Printing

Large volume journal printing to high-speed printing press.





Mobile Printing Use-Model

1.0 Mobile Printing

Mobile printing by reference with document data transformation.

Example Use Model:

Alice goes shopping for a new digital camera to the Fine Camera store downtown. She takes along a magazine review of the new Bright 3000. The sales clerk at the Fine Camera store tells Alice that she would probably prefer the Orion 777. Alice uses her mobile hand-held to browse the Web site of Bright and find the URL for the detailed specs of the Bright 3000. She uses the public access printer in the Fine Camera store to print the Bright 3000 specs. Sure enough, the Orion 777 is a better choice.

Details: (1)

Alice turns on her mobile hand-held and hits the 'I' (Internet) button. The mobile hand-held starts a Web browser application, which connects to Alice's wireless Internet service provider (ISP) over a 14.4Kbps cellular modem. Alice types the Bright URL (from the magazine review) into her Web browser and hits the Enter key. The Web browser connects to Bright's home page over the Internet (via the HTTP proxy in the cellular ISP's firewall). Alice searches for the Bright 3000 specs (available in HTML) and copies the URL into her GUI clipboard.

Alice hits the 'P' (Print) button. The mobile hand-held starts a print application, which discovers the store's public access printer over Bluetooth using FSG/OP PAPI (which does Bluetooth device discovery) and then forms an ad-hoc wireless Personal Area Network (PAN) with the Target Printer.

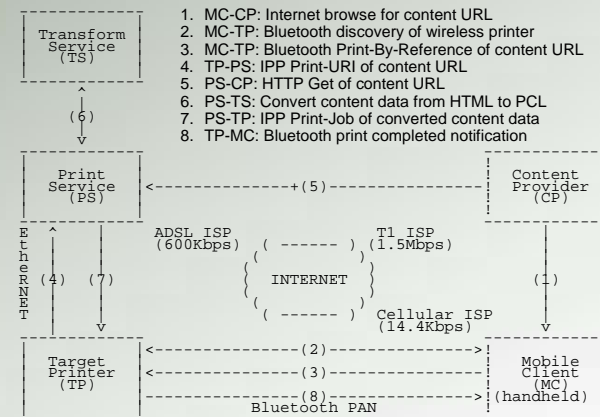


Requirements for this Use Model:

The FSP/OP Architecture MUST support:

- ◆ Administrative configuration of Print Services and their associations with Target Printers using FSG/OP PAPI;
- ◆ Administrative configuration of Target Printers and their associations with Print Services using FSG/OP PAPI;
- ◆ Dynamic discovery of network (IP, Bluetooth, IRDA, etc.) Print Services and Target Printers using FSG/OP PAPI;
- ◆ Multiple print protocol bindings of FSG/OP PAPI (for direct-connect and non-IP network printing);
- ◆ Print-by-reference operations (for example, printing from low-bandwidth mobile devices);
- ◆ Print-by-value operations (for example, printing of converted or local content);
- ◆ Content conversion by FSG/OP Transform Services (for example, reference printing);
- ◆ Job and Printer event notification (for example, job complete, printer intervention required, etc.).

Use Model Diagram:



1. MC-CP: Internet browse for content URL
2. MC-TP: Bluetooth discovery of wireless printer
3. MC-TP: Bluetooth Print-By-Reference of content URL
4. TP-PS: IPP Print-URI of content URL
5. PS-CP: HTTP Get of content URL
6. PS-TS: Convert content data from HTML to PCL
7. PS-TP: IPP Print-Job of converted content data
8. TP-MC: Bluetooth print completed notification



What's next - schedule

Next Steps

❏ Open Print Requirements Document

- ✗ Aug 2004 - Complete prioritization of identified sub-systems
- ✗ Sept 2004 - Complete Reference Model Alpha Release
- ✗ Nov 2004 - Integrate Use Models from FSG-Japan Use-Models
- ✗ Mar 2005 - Complete Detailed Description for Use Models
- ✗ May 2005 - Complete Requirements for Use Models
- ✗ Aug 2005 - Complete Common Requirements & Reference Model 1.0

❏ High-Level Architecture/Reference Model

- ✗ May 2006 - Completed document

How to accelerate the process and schedule?

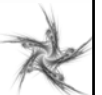
Funding ! More Participants !





Architecture Working Group Information

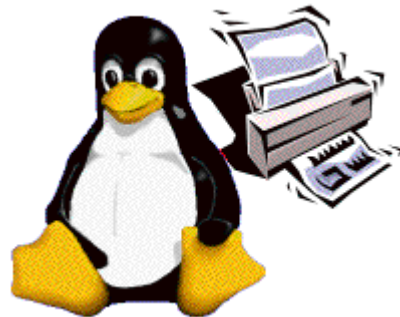
- Weekly FSG Architecture conference calls
 - ✦ Thursday at 3:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Architecture mailing list:
 - ✦ <http://freestandards.org/mailman/listinfo/printing-architecture>
- To post a message to FSG Architecture mailing list
 - ✦ printing-architecture@freestandards.org
- To view FSG Architecture mailing list archives
 - ✦ <http://freestandards.org/mailman/listinfo/printing-architecture>
- To find FSG Architecture documents
 - ✦ <ftp://ftp.pwg.org/pub/pwg/fsg/architecture/>
- Participants
 - ✦ Claudia Alimpich (IBM)
 - ✦ Jody Goldberg (Gnome)
 - ✦ Mark Hamzy (IBM) - chair
 - ✦ Tom Hastings (Xerox)
 - ✦ Norm Jacobs (Sun)
 - ✦ Till Kamppeter (MandrakeSoft)
 - ✦ Ira McDonald (High North Inc)
 - ✦ Glen Petrie (Epson)
 - ✦ Pete Zannucci (IBM)





**Free
Standards
Group**

Job Ticket





JTAPI & Objectives

Introduction

What is JTAPI ?

✗ JTAPI stands for:

- ◆ Job Ticket Application Programming Interface
- ◆ Pronounced “jay-tappy”, “Job Ticket API”, or “jay tee API”

✗ A job ticket contains:

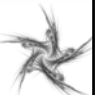
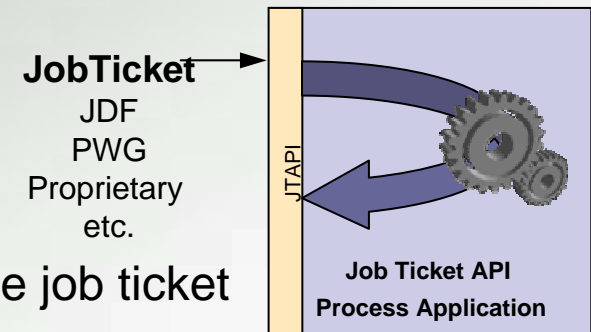
- ◆ **Instructions** describing how to process and/or print a job
- ◆ **Information** about the results of a job as it is processed and/or printed

✗ A JTAPI job ticket is:

- ◆ an electronic replacement of manual hard copy instructions and information

Objectives

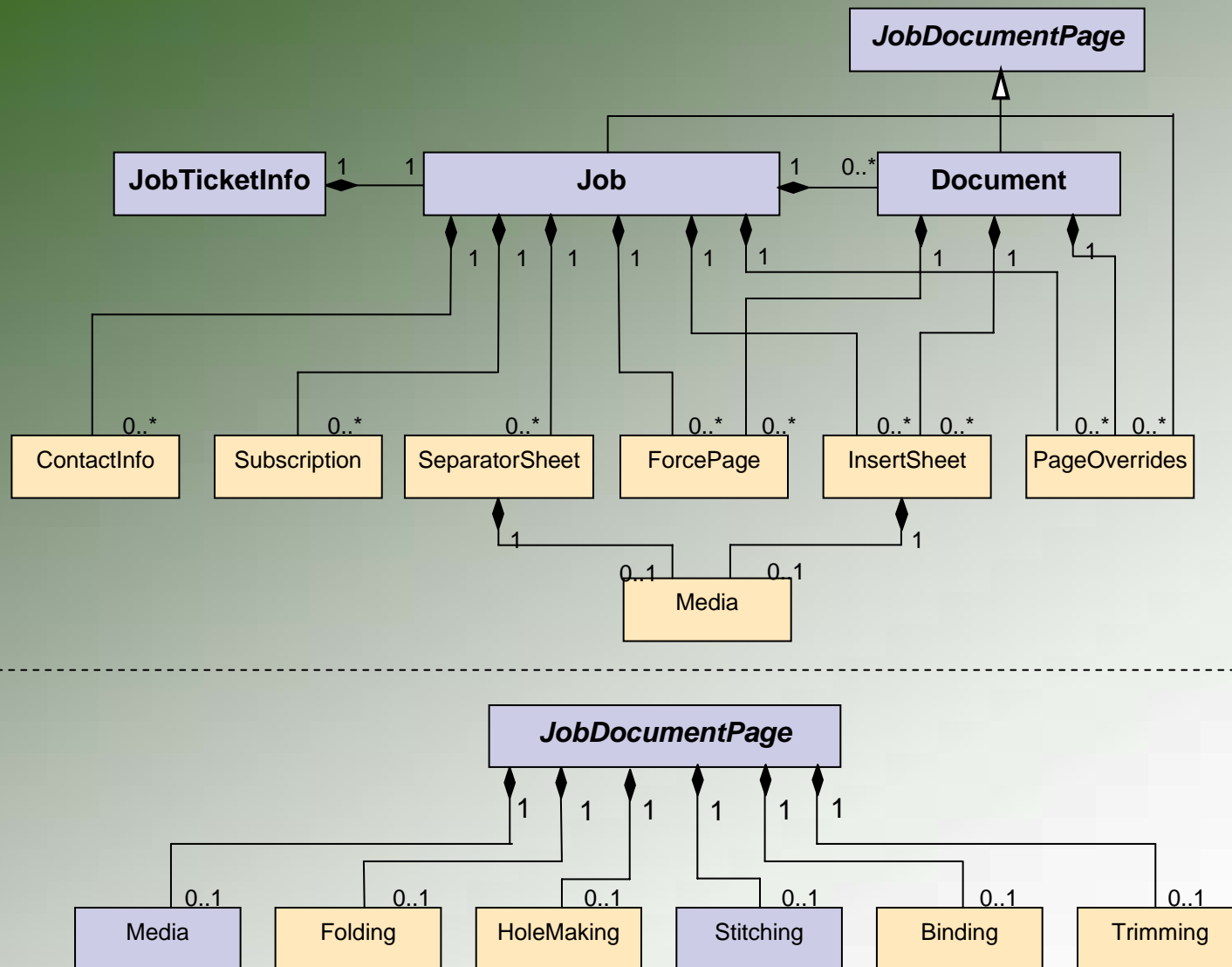
- ✗ To create and consume job tickets
- ✗ To be job ticket syntax neutral
- ✗ To isolate the application from the content of the job ticket
- ✗ To be programming language neutral
- ✗ To import and export multiple job ticket formats





JTAPI Object Model

Introduction





JTAPI Alpha Release

Introduction

■ Completed C Header Files

- ✗ 18 objects in separate files
- ✗ Common extensible method for attributes
- ✗ Data/object model that is object oriented
- ✗ Defines objects that are familiar to the printing industry
 - ◆ Job, Document, Insert Sheet, Media, Stitching, Hole Making, etc.
- ✗ Defines relationship between objects
- ✗ Defines operations to be performed on objects
- ✗ Defines attributes of objects
- ✗ Defines well-known enumerated values of all attributes





JTAPI Details

Introduction

JobTicketInfo

✘ fsgjtNewJobTicketInfo

- ◆ Creates a JobTicketInfo object. Used when a new job ticket is being created without a Job object and not from an existing job ticket.

✘ fsgjtNewJobTicketInfoFromURI

- ◆ Creates a JobTicketInfo object from the job ticket file at the provided URI. Other JTAPI objects are also created based on the information in the job ticket file.

✘ fsgjtNewJobTicketInfoFromBuffer

- ◆ Creates a JobTicketInfo object from the provided buffer that contains a job ticket. Other JTAPI objects are also created based on the information in the job ticket buffer.

✘ fsgjtNewJobTicketInfoFromJob

- ◆ Creates a JobTicketInfo object using the provided Job object. Used when a new job ticket is being created. The Job object must be created first.

✘ fsgjtWriteJobTicketToBuffer

- ◆ Writes a job ticket to the provided buffer.

✘ fsgjtWriteJobTicketToURI

- ◆ Writes a job ticket at the provided URI.

Job

✘ fsgjtNewJob

- ◆ Creates a Job without providing a Document.

✘ fsgjtNewJobFromDocument

- ◆ Creates a Job containing the provided Document.

Document

✘ fsgjtNewDocument

- ◆ Creates a Document object.

✘ fsgjtNewDocumentFromURI

- ◆ Creates a Document object using the provided URI that contains the document data.

Stitching

✘ fsgjtNewStitching

- ◆ Creates a Stitching object having the provided stitch type.

Media

✘ fsgjtNewMedia

- ◆ Creates a Media object using the provided name.





JTAPI Details

Attribute

Generic support for all object/attributes

✘ fsgjtNewAttribute

- ◆ Creates a new Attribute object having the provided attribute name, value type, and value. The value must not be an extension value.

✘ fsgjtDestroyAttribute

- ◆ Free the memory used by the Attribute.

✘ fsgjtAddExtensionValue

- ◆ Add an additional value, that is an extension, to this Attribute.

✘ fsgjtAddValue

- ◆ Add an additional value, that is not an extension, to this Attribute.

✘ fsgjtGetName

- ◆ Get the name of the Attribute.

✘ fsgjtGetNextExtensionValue

- ◆ Return the Attribute's next value that is an extension.

✘ fsgjtGetNextValue

- ◆ Return the Attribute's next value that is not an extension.

✘ fsgjtGetNumExtensionValues

- ◆ Get the number of values, that are extensions, that the Attribute contains.

✘ fsgjtGetNumValues

- ◆ Get the number of values, that are not extensions, that the Attribute contains.

✘ fsgjtGetValueType

- ◆ Returns the type of the Attribute's values.

✘ fsgjtisExtensible

- ◆ Returns true if this Attribute is extensible, which means that a value other than a pre-defined value can be added. Returns false if no value other than a pre-defined value can be added.

✘ fsgjtReplaceExtensionValue

- ◆ Replaces the existing extension value(s) for this Attribute with the provided value that is an extension. The non-extension value(s) of this Attribute are not affected.

✘ fsgjtReplaceValue

- ◆ Replaces the existing non-extension value(s) for this Attribute with the provided value that is not an extension. The extension value(s) of this Attribute are not affected.

✘ fsgjtResetToFirstValue

- ◆ Reset the iterators to point to the first non-extension value and the first extension value of the Attribute.





Alpha*/1.0 Version – Enumerations

■ Enumerations

- ✗ BindTypeEnum
- BooleanEnum*
- ✗ CollateEnum
- ✗ CompressionEnum
- ✗ ContactTypeEnum
- ✗ FeedOrientationEnum
- ✗ FitPolicyEnum
- ✗ FoldTypeEnum
- ✗ HoldEnum
- ✗ ImageAlignmentXEnum
- ✗ ImageAlignmentYEnum
- InputTrayNameEnum*
- ✗ InsertSheetContentEnum
- JobTicketTypeVersionEnum*
- ✗ JogOffsetEnum
- LengthUnitEnum*
- ✗ MandatoryAttributesEnum
- ✗ MediaCoatingEnum
- ✗ MediaColorEnum
- MediaTypeEnum*
- ✗ OutputBinEnum
- ✗ PageDeliveryEnum
- ✗ PositionEnum
- ✗ PreprintedEnum
- ✗ PresentEnum
- ✗ PresentationDirectionEnum
- ✗ PrintContentOptimizeEnum
- PrintQualityEnum*
- ✗ ReferenceEdgeEnum
- RotationEnum*
- ✗ SperatorSheetEnum
- ✗ SheetSideEnum
- SidesEnum*
- StitchingTypeEnum*
- ✗ SubscriptionEventEnum
- ✗ TrimTypeEnum
- ValueTypeEnum*





JTAPI What's Next

Next Steps

■ Final JTAPI Alpha Specification

- ✗ July 2004 - Release Alpha candidate for public review/comment
- ✗ August 2004 - Public review period ends
- ✗ Sept 2004 - Final Alpha release

■ Alpha Reference Implementation in C

- ✗ Dec 2004 - Requirement / Definition document
- ✗ Dec 2004 - Funding issues for code development and validation

■ JTAPI Version 1.0 Specification

- ✗ Jan 2005 - Draft version of Specification
- ✗ Jan 2005 - White Paper Proposal for code implementation





JT Working Group Information

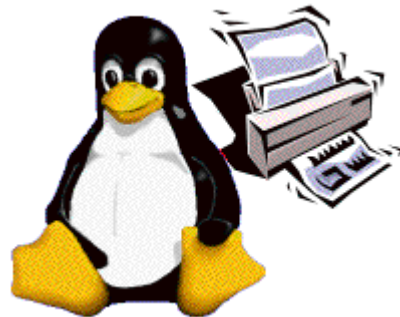
- Weekly FSG Job Ticket conference calls
 - ✦ Tuesdays at 3:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Job Ticket mailing list:
 - ✦ <http://freestandards.org/mailman/listinfo/printing-jobticket>
- To post a message to FSG Job Ticket mailing list
 - ✦ printing-jobticket@freestandards.org
- To view FSG Job Ticket mailing list archives
 - ✦ <http://freestandards.org/mailman/listinfo/printing-jobticket>
- To find FSG Job Ticket documents
 - ✦ <ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/>
- Participants
 - ✦ Claudia Alimpich (IBM) – chair
 - ✦ Jody Goldberg (Gnome)
 - ✦ Tom Hastings (Xerox)
 - ✦ Till Kampeter (MandrakeSoft)
 - ✦ Ira McDonald (High North Inc)
 - ✦ Glen Petrie (Epson)





**Free
Standards
Group**

Application Interface





PAPI & Objectives

Introduction

- **PAPI stands for:**
 - ✗ Printing Application Programming Interface
 - ✗ Pronounced “pappy” or “P.A.P.I.”

- **PAPI contains:**
 - ✗ **Data Structures** for interacting with basic print service objects.
 - ✗ **Functions** to perform various operations against a print service or it's objects.

- **PAPI is an API for applications to use to perform print service interaction.**

- **Objectives**
 - ✗ Provide applications a means of interacting with print services without being tied to a particular service or protocol
 - ✗ Allow a rich, extensible set of information to flow between application and print service
 - ✗ Support a rich enough set of operations to be useful to most applications with printing needs





PAPI Overview

Introduction

- Provides abstraction of server, queues (printers and classes), and jobs.
- High-level API focuses on providing a common interface to multiple printing systems and protocols.
- Uses extensible attribute sets based on IPP for representing queue and job information.
 - ✗ Multi-valued and typed
- Currently supports
 - ✗ Server connection (and authentication were applicable)
 - ✗ Basic queue operations:
 - ◆ enumerate, query, enable, disable, modify, enumerate/purge jobs
 - ✗ Basic job operations:
 - ◆ submit, stream, query, cancel, hold, release, modify





PAPI Details

Introduction

✚ Data types introduced by the PAPI:

- ✘ **papi_status_t** status/error code enumeration
- ✘ **papi_service_t** server/service connection
- ✘ **papi_printer_t** printer object/handle/context
- ✘ **papi_job_t** job object/handle/context
- ✘ **papi_attribute_t** object attributes
- ✘ **papi_stream_t** stream object/handle/context
- ✘ **papi_job_ticket_t** job ticket
- ✘ **papi_filter_t** filter for limiting printer lists

✚ Attribute API :

- ✘ **papiAttributeListAdd / papiAttributeListAdd***
 - ◆ Adds a single attribute (and value) to an attribute list
 - ◆ Supported types include- String, integer, boolean, range, resolution, datetime, collection, metadata
- ✘ **papiAttributeListGet***
 - ◆ Retrieve the value associates with an attribute
- ✘ **papiAttributeListFind**
 - ◆ Search for an attribute in a list
- ✘ **papiAttributeListGetNext**
 - ◆ Walk through an attribute list
- ✘ **papiAttributeListFromString**
 - ◆ create/append an attribute list using a text-based format
- ✘ **PapiAttributeListToString**
 - ◆ Convert an attribute list to a text base string form
- ✘ **papiAttributeListFree**
 - ◆ Deallocate resources associated with an attribute list

✚ Service API

- ✘ **papiServiceCreate**
 - ◆ create a new service object for submitting jobs, listing printers, etc
- ✘ **papiServiceDestroy**
 - ◆ destroys a service object and all associated resources
- ✘ **papiGet*() / papiSet*()**
 - ◆ Get or set the corresponding service object attributes (UserName, Password, Encryption, AuthCB, AppData)

✚ Miscellaneous API :

- ✘ **papiStatusString**
 - ◆ Return a textual representation of a papi_status_enumeration
- ✘ **papiLibrarySupportedCalls**
 - ◆ Enumerate all papiCalls that do not simply return PAPI_OPERATION_NOT_SUPPORTED
- ✘ **papiLibrarySupportedCall**
 - ◆ Determine if a specific function is supported by the implementation





PAPI Details

Introduction

Job API:

- ✘ **papiJobSubmit / papiJobSubmitByReference**
 - ◆ Submit a print job either immediately copying or postponing copy of the job data
- ✘ **papiJobValidate**
 - ◆ Verify that the document format and job attributes are valid for and supported by the designated print queue
- ✘ **papiJobStreamOpen / papiJobStreamWrite / papiJobStreamClose**
 - ◆ Open/write/close a stream for printing data on-the-fly
- ✘ **papiJobQuery**
 - ◆ Request job information from print service
- ✘ **papiJobModify**
 - ◆ Modifies the job attributes
- ✘ **papiJobCancel**
 - ◆ Cancels a print job
- ✘ **papiJobHold / papiJobRelease / papiJobRestart**
 - ◆ Hold, release, or restart a print job
- ✘ **papiJobGetAttributeList**
 - ◆ Retrieve the corresponding attribute list from a job object for further lookup or enumeration using the Attribute interface
- ✘ **papiJobGet**
 - ◆ Retrieve the corresponding PrinterName, Id or JobTicket from a job object
- ✘ **papiJobListFree / papiJobFree**
 - ◆ Free a job object of list of job objects and their associated resources

Printer API :

- ✘ **papiPrintersList**
 - ◆ Retrieves a list of printer/class queues
- ✘ **papiPrinterQuery**
 - ◆ Queries for detailed information about a printer object
- ✘ **papiPrinterModify**
 - ◆ Sets printer object attributes
- ✘ **papiPrinterPause / papiPrinterResume**
 - ◆ Stop and start job processing on a printer object
- ✘ **papiPrinterPurgeJobs**
 - ◆ Cancels all jobs on a printer
- ✘ **papiPrinterListJobs**
 - ◆ Lists jobs on a printer
- ✘ **papiPrinterGetAttributeList**
 - ◆ Retrieve an attribute list from printer object
- ✘ **papiPrinterListFree / papiPrinterFree**
 - ◆ Deallocate resources associated with a printer object or list of printer objects





PAPI Status & Plans

Introduction

■ Status

- ✘ Published v0.91 of specification
- ✘ Released PAPI based code on SourceForge
 - ◆ PAPI dynamic library
 - ◆ PAPI over CUPS
 - ◆ BSD/SYSV print commands implemented on top of the PAPI
- ✘ Released PAPI over RFC-1179 support

■ What's next

- ✘ Expand functionality in PAPI
 - ◆ Administrative operations
 - ◆ Document object
- ✘ Implement and Release PAPI over IPP support
- ✘ Release IPP server implemented over PAPI
- ✘ Integrate support for various Open Source projects:
 - ◆ GNOME, KDE, Open Office, Mozilla, Samba, etc.





PAPI Working Group Information

- To subscribe to FSG PAPI mailing list:
 - ✗ <http://freestandards.org/mailman/listinfo/printing-spool>

- To post a message to FSG PAPI mailing list
 - ✗ printing-spool@freestandards.org

- To view FSG PAPI mailing list archives
 - ✗ <http://freestandards.org/mailman/listinfo/printing-spool>

- To find FSG PAPI documents
 - ✗ <http://sourceforge.net/projects/openprinting>

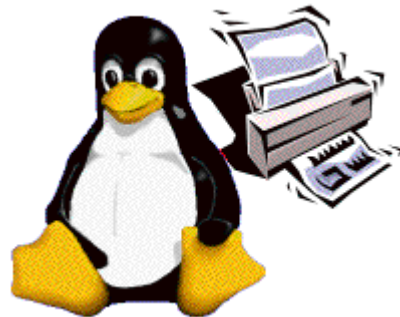
- Participants
 - ✗ Norm Jacobs (Sun) – chair
 - ✗ Alan Hlava (IBM)
 - ✗ Mike Sweet (Easy Software)
 - ✗ Ira McDonald (High North Inc)
 - ✗ Glen Petrie (Epson)





Free
Standards
Group

Printer Driver





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





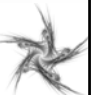
Printer Driver API

- 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

■ Raster Image Operations

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

■ Stream Data Operations

- ✘ StartStream(), TransferStreamData(), EndStream()

■ 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()



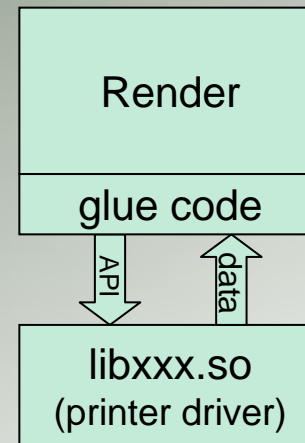


Linking

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

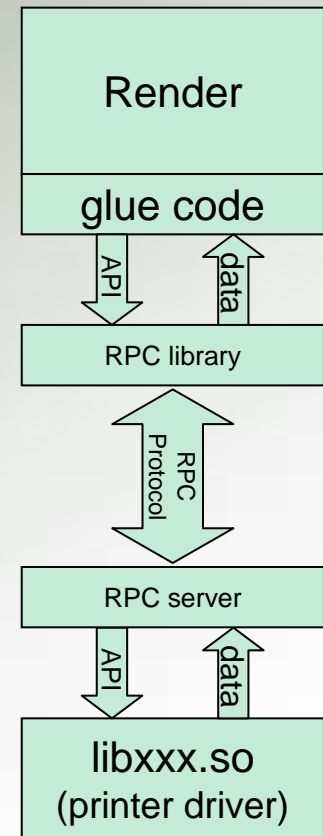
direct linking

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





PD-Raster Issues / Concerns

- Small number of contributors in this group
- Contributors from across the printing spectrum
- Integration of the vector API into this group
- Reconciliation with other standards
- Expansion into other printing areas
 - ◆ - Device fonts
- Funding for people on this group





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)





PD-Raster Working Group Information

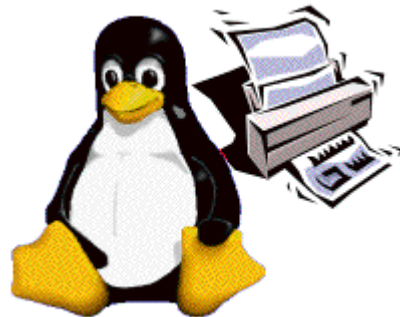
- Weekly FSG Printer Driver conference calls
 - ✘ Mondays at 2:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Printer Driver mailing list:
 - ✘ <http://freestandards.org/mailman/listinfo/printing-driver>
- To post a message to FSG Printer Driver mailing list
 - ✘ printing-driver@freestandards.org
- To view FSG Printer Driver mailing list archives
 - ✘ <http://freestandards.org/mailman/listinfo/printing-driver>
- To find FSG Printer Driver documents
 - ✘ <ftp://ftp.pwg.org/pub/pwg/fsg/driver/>
- Participants
 - ✘ Mark Hamzy (IBM) – chair
 - ✘ Till Kamppeter (MandrakeSoft)
 - ✘ Glen Petrie (Epson)





Free
Standards
Group

Status Monitoring





Status Monitoring (SM) & Objectives

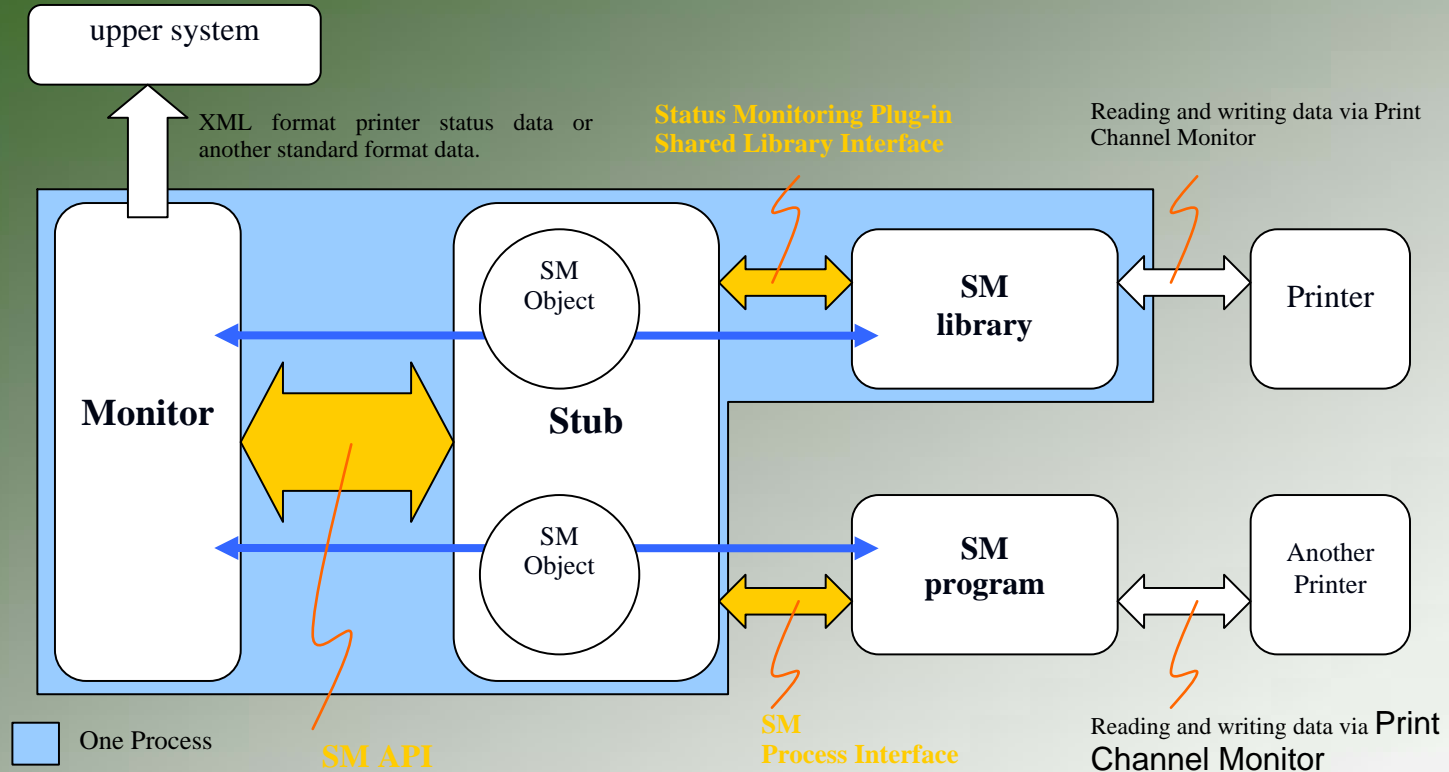
- What is the Status Monitoring API?
 - ✘ An interface for obtaining the printer status and/or information about the printer.

- Objective
 - ✘ Define a programming neutral API for obtaining the printer status and information.





Architecture Overview



- ✦ “Monitor” is:
 - ✦ A program which calls the SM API entries provided by the Stub.
- ✦ “Stub” is:
 - ✦ A library which connects the Monitor and the SM module, and provides the SM API entries to the Monitor.





Status Monitoring Module

■ “Status Monitoring module” is:

- ✗ A shared library or an individual program which provides a set of the SM API functions.

■ “Status Monitoring library” is:

- ✗ A SM module as a shared library being linked by the Stub.
 - ◆ Run as the same process as the Stub and Monitor.
 - Obtains the printer status/information **passively**.
 - Can hook signals of the Monitor.

■ “Status Monitoring program” is:

- ✗ A SM module as an individual program which communicates with the Stub via an inter-process communication interface.
 - ◆ Obtain the printer status individually from the Stub and Monitor.
 - Obtain the printer status/information **actively**.
 - Can deal with signals individually from the Monitor.





Printer Status Data Format

- Current version's API supports:
 - ✘ The XML data according to the Printer MIB v2 Schema being considered in the PWG WBMM group.
<ftp://ftp.pwg.org/pub/pwg/wbmm/schemas/>
 - ✘ The filtering mechanism to reduce a high data processing/transfer rate.
 - ◆ idReadMode of the SMstartRead() function
 - SM_READ_PRT_MIB_ALL
 - SM_READ_PRT_MIB_SUMMARY





SM Working Group Information

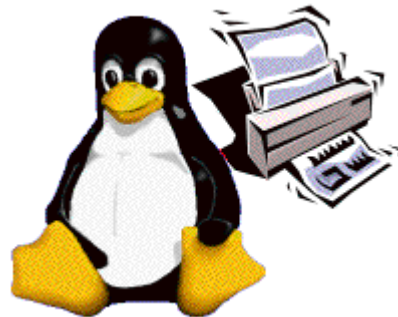
- To subscribe to FSG SM mailing list:
 - ✦ <http://freestandards.org/mailman/listinfo/printing-japan>
- To post a message to FSG SM mailing list
 - ✦ printing-japan@freestandards.org
- To view FSG SM mailing list archives
 - ✦ <http://freestandards.org/mailman/listinfo/printing-japan>
- To find FSG PCM documents
 - ✦ ftp://ftp.pwg.org/pub/pwg/fsg/status_monitoring/
- Participants
 - TORATANI Yasumasa (Canon Inc.)
 - Osamu MIHARA (FUJI XEROX Printing Systems)
 - KANJO Hidenori (BBR INC.)
 - YOSHIDA Mikio (BBR INC.)
 - Shinpei KITAYAMA (EPSON KOWA)
 - YAMAGISHI Toshihiro (Turbolinux)
 - Hisao NAKAMURA (E&D)
 - Koji OTANI (AXE)





**Free
Standards
Group**

Print Channel Monitor





Print Channel Monitor Objective

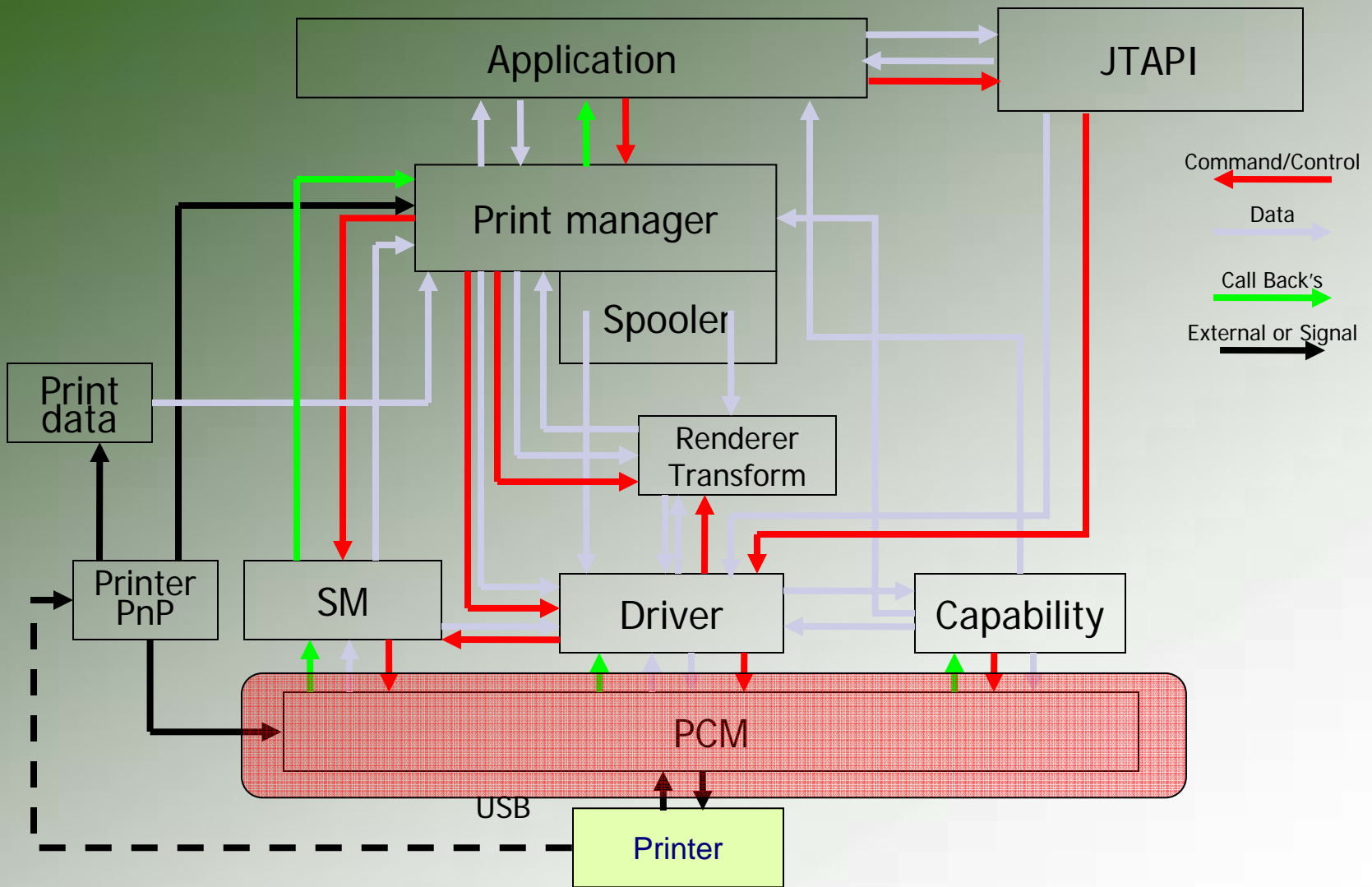
- Print Channel Monitor (PCM) will wrap the difference of physical layer
 - ✗ Upper layer do not need to care about the difference of physical layer like USB, Parallel, Ethernet.
 - ✗ Wrap the difference of transaction protocol used by printer venders.

- PCM will focus on communication layer
 - ✗ Job control and device management are left to upper layer.



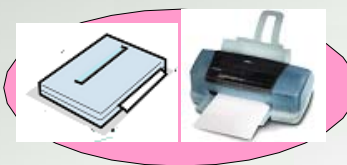
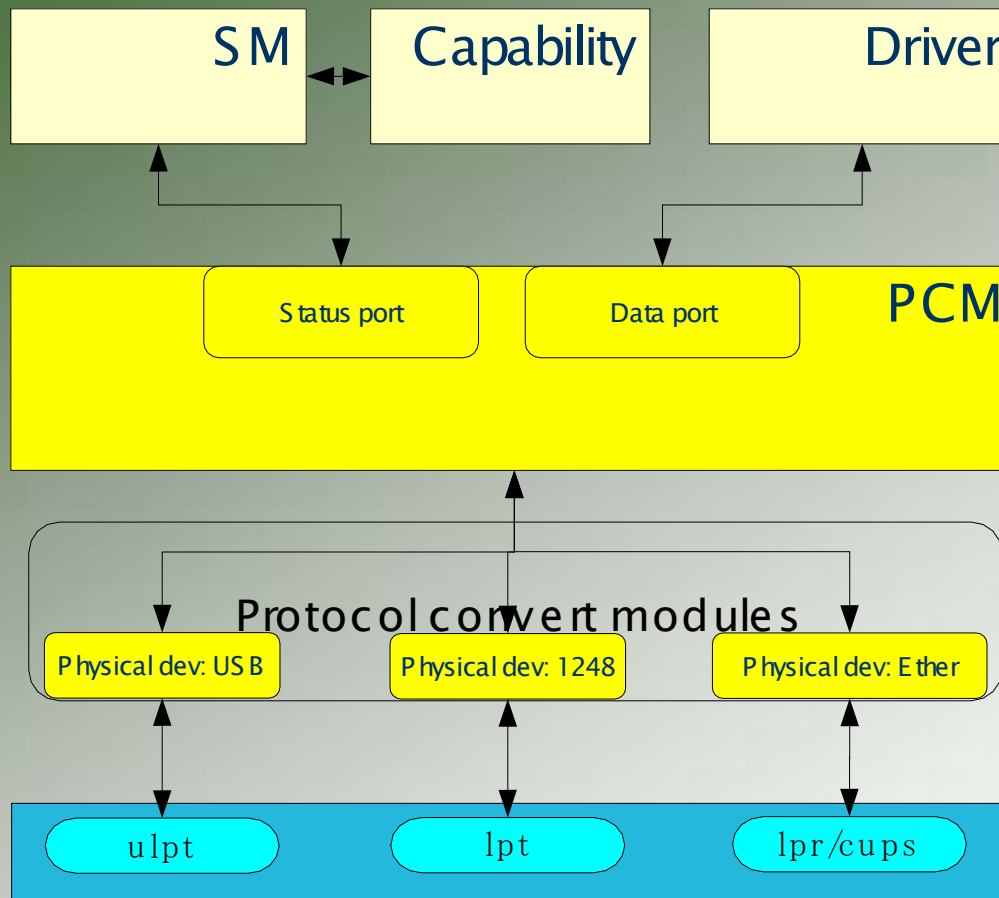


PCM outline





PCM outline





- Draft – working now!

 - 2004/5/31 Pre-draft of the PCM specifications.
(written only the main point.)

 - 6/ F Open to working group.

 - 6/ F Start discussion and writing the draft.

- Discussion about PCM

 - ✗ Printing-Japan ML
 - ✗ Face to Face meeting – Monthly

- Milestone

 - ✗ 2004/8/27 Create a model/architecture
 - ✗ 2005/1/30 Release 1.0 specification

- To-Do

 - ✗ Consider the other existing Printing Systems so PCM could be easily adopted
 - ✗ Consider the future extension
 - ◆ Support new I/F or new device like MFP





PCM Working Group Information

- To subscribe to FSG PCM mailing list:
 - ✗ <http://freestandards.org/mailman/listinfo/printing-japan>

- To post a message to FSG PCM mailing list
 - ✗ printing-japan@freestandards.org

- To view FSG PCM mailing list archives
 - ✗ <http://freestandards.org/mailman/listinfo/printing-japan>

- To find FSG PCM documents
 - ✗ To be determined

- Participants
 - ✗ Kentaro Ide (Seiko Epson Corp)
 - ✗ Shinpei Kitayama (Epson Kowa Corp.)
 - ✗ Fumio Nagasaka (Seiko Epson Corp.)
 - ✗ Kazuo Nomura (Seiko Epson Corp.)
 - ✗ Koji Otani (AXE Inc)





FSG Open Printing

THANKS

