

Linux Foundation
OpenPrinting

JTAPI

Job Ticket
Application Programming Interface

2011
Google Summer of Code
Proposed Activity



Glen W. Petrie

Senior Software Architect
EPSON IMAGING TECHNOLOGY CENTER
San Jose, CA, USA
glen.petrie@eitc.epson.com

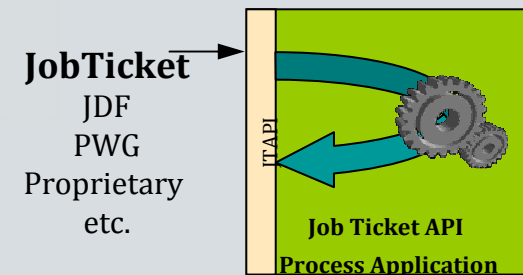
April 6, 2011

JTAPI GSOC 2011

1

JTAPI & Objectives

- **What is JTAPI ?**
 - **JTAPI stands for:**
 - Job Ticket Application Programming Interface
 - Pronounced “jay-tappy”, “Job Ticket API”, or “jay tee API”
- **Objectives**
 - To create and consume job tickets but not define a new job ticket
 - 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



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

State of JTAPI

- JTAPI Version 1.0 Specification/Header Release
 - Jan 2005 - Release Specification and headers
- 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)
 - Glen Petrie (Epson)

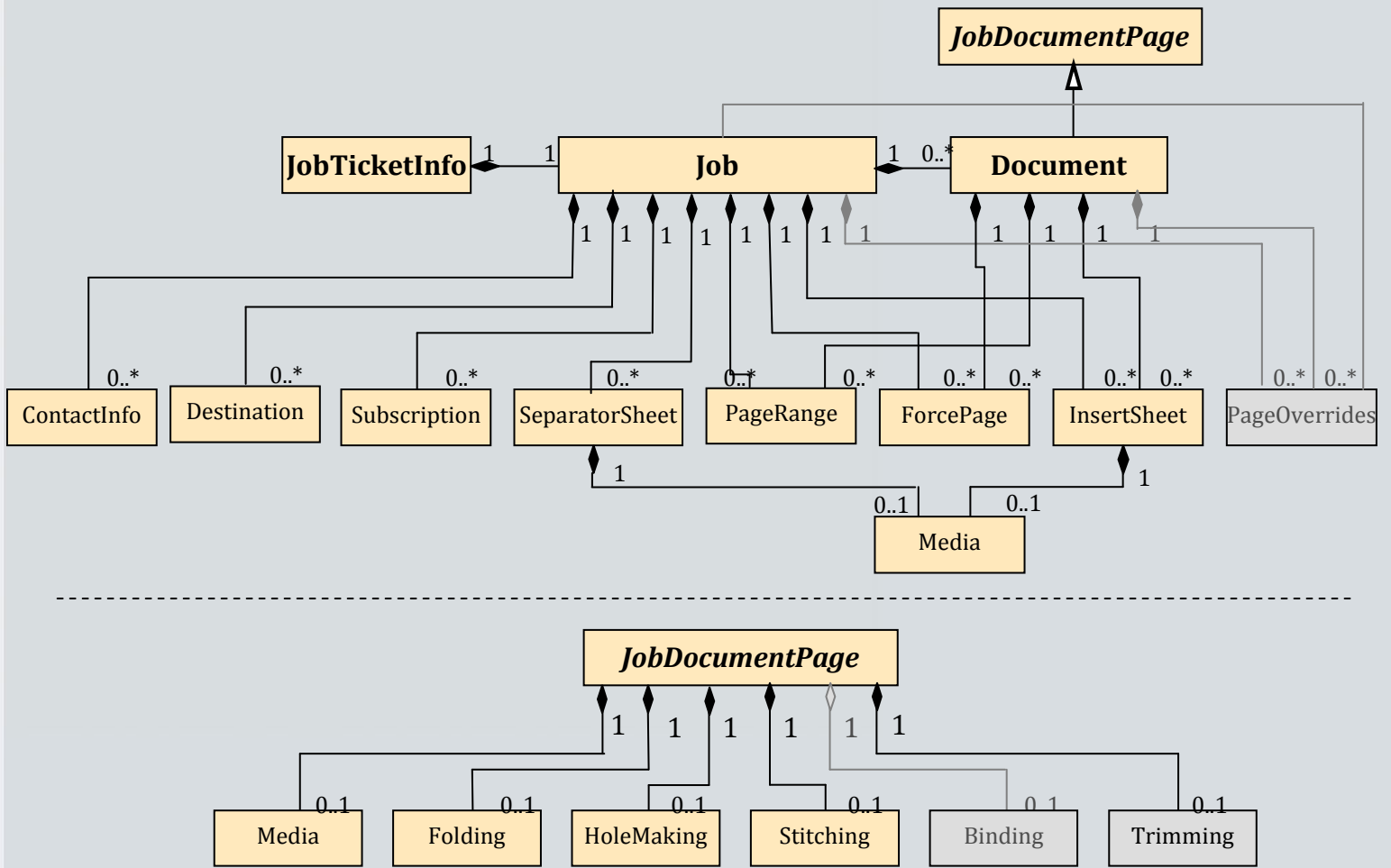


Existing Job Ticket Formats

- **PWG Job Ticket** Target Job Ticket for GSOC 2011
 - Defined by PWG Semantic Model 1.0
 - Is open, extensible, XML-based job ticket standard
- **CIP4 JDF (Job Definition Format) Job Ticket**
 - Defined by CIP4, a world wide standards body with over 150 members
 - Is open, extensible, XML-based job ticket standard
 - JDF Specification versions
 - 1.0 released April 2001
 - 1.1 released April 2002
 - 1.2 released May 2004
 - 1.3 to be released Mid 2005



JTAPI Object Model



JTAPI Version 1.0 Release

- **Completed C Header Files**
 - Each object 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 –Version 1.0 Release Details (1)



- 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.

JTAPI –Version 1.0 Release Details (2)

- **ContactInfo**
 - **fsgjtNewContactInfo**
 - Creates a ContactInfo object having the specified name.
- **Subscription**
 - **fsgjtNewSubscription**
 - Creates a Subscription object having the specified notification URI.
 - **fsgjtNewSubscriptionForEvent**
 - Creates a Subscription object having the provided notification URI and event.
- **SeparatorSheet**
 - **fsgjtNewSeparatorSheet**
 - Creates a SeparatorSheet object to be placed at the provided location.
- **ForcePage**
 - **fsgjtNewForcePage**
 - Creates a ForcePage object haing the specified page and sheet side.
- **InsertSheet**
 - **fsgjtNewInsertSheet**
 - Creates a InsertSheet object.
- **Destination**
 - **fsgjtNewDestination**
 - Creates a Destination object with a specific URI.



JTAPI –Version 1.0 Release Details (3)

- Media
 - **fsgjtNewMedia**
 - Creates a Media object using the specified name.
- Folding
 - **fsgjtNewFolding**
 - Creates a Folding object having the specified folding type.
- HoleMaking
 - **fsgjtNewHoleMaking**
 - Creates a HoleMaking object having the specified hole count and reference edge.
- PageRange
 - **fsgjtNewRangRange**
 - Creates a PageRange object.
- Stitching
 - **fsgjtNewStitching**
 - Creates a Stitching object having the specified stitch type.



JTAPI –Version 1.0 Release Details (4)

- Attribute

Generic support for all object/attributes

- **fsgjtNewAttribute**

- Creates a new Attribute object having the provided attribute name, value type, and value.

- **fsgjtDestroyAttribute**

- Free the memory used by the Attribute.

- **fsgjtAddValue**

- Add an additional value to this Attribute.

- **fsgjtGetName**

- Get the name of the Attribute.

- **fsgjtGetNextValue**

- Return the Attribute's next value.

- **fsgjtGetNumValues**

- Get the number of values that the Attribute contains.

- **fsgjtGetValueType**

- Returns the type of the Attribute's values.

- **fsgjtReplaceValue**

- Replaces the existing value(s) for this Attribute.

- **fsgjtResetToFirstValue**

- Reset the iterator to point to the first of the Attribute.



JTAPI –Version 1.0 Release Details (5)

- Miscellaneous / Helper
 - **fsgjtDestory**
 - Free the memory used by the an object.
 - **fsgjtGet**
 - Get the Attribute having the specific name.
 - **fsgjtSet**
 - Set the Attribute having the specific name.
 - **fsgjtSetIntegerAttribute**
 - Convenience function for setting an integer attribute.
 - **fsgjtSetObjectAttribute**
 - Convenience function for setting an object attribute.
 - **fsgjtSetObjectAttributeList**
 - Convenience function for setting an object list.
 - **fsgjtSetStringAttribute**
 - Convenience function for setting a string attribute.
 - **fsgjtSetStringAttributeList**
 - Convenience function for setting a string list.



JTAPI - Version 1.0 – Enumerations

- BindTypeEnum
- BooleanEnum*
- CollateEnum
- CompressionEnum
- ContactInfoRoleEnum
- FeedOrientationEnum
- FitPolicyEnum
- FoldTypeEnum
- HoldEnum
- ImageAlignmentXEnum
- ImageAlignmentYEnum
- InputTrayNameEnum*
- InsertSheetContentEnum
- JobTicketTypeVersionEnum*
- JogOffsetEnum
- LengthUnitEnum*
- MandatoryAttributesEnum
- MediaCoatingEnum
- MediaColorEnum
- MediaPrePrintedEnum
- MediaTypeEnum*
- OutputBinEnum
- PageDeliveryEnum
- PositionEnum
- PresentEnum
- PresentationDirectionEnum
- PrintContentOptimizeEnum
- PrintQualityEnum*
- ReferenceEdgeEnum
- RotationEnum*
- SeparatorSheetEnum
- SheetSideEnum
- SidesEnum*
- StitchingTypeEnum*
- SubscriptionEventEnum
- TrimmingTypeEnum
- ValueTypeEnum*



JTAPI

- **Job Ticket API (JTAPI)**

- All print jobs, large and small, require two components; namely, a job ticket and print content. Both of these take on many forms; for example, print content ranges from PDF to JPEG to TEXT to MS-WORD to LIBRE-WRITER to POWER POINT and countless others. All modern print solutions today either directly or indirectly (via the source application and/or transforms) support the various types of print content. Like print content, (print) job tickets come in numerous formats and even large ranges of content and capabilities. As OpenPrinting moves to provide coherence for printing and as printing moves to the Cloud where the source of the a print job is no longer the platform the user is working on, there needs to be a common solution for not only editing (print) job tickets but to also transform (print) job-tickets from one format to another while the print job moves through a Linux print solution (transforms, spoolers and print-managers) or through a federation of Clouds. A previous OpenPrinting activity created a specification and C code binding for a set of Application Programming Interfaces (APIs) called "JTAPI" to provide the necessary edit and transform functions. The JTAPI [specification](#) shows that it can be used to support mobile (cloud) devices, desktop, office and production printing solutions; demonstrating the eventual use for any print solution or printing need. "Behind" the JTAPI implementation are anywhere from one to many(-many-many) specific job-ticket format bindings (such as JDF, PWG Job-Ticket, etc) that will be added to this base implementation as a separate GSOC project.



JTAPI

JTAPI GSOC Project – (1)

- Objective
 - Using the existing JTAPI specification and C code header files the objective of this project is to develop, document, test, and release a platform-independent C code library for the reading of, modification of, and storage of a (print) job-ticket. Since creation of the base library will actually need a (print) job-ticket for development; the Printer Working Group's (PWG's) Micro-Job-Ticket (MJT) can be used as an illustrative job ticket for the purpose of developing the core implementation.



JTAPI GSOC Project – (2)

- **Approach:**
 - Review OP/JTWG Job-Ticket Application Programming Interface (JTAPI) header document
 - Review PWG Job Ticket specification
 - Create Test Job Ticket
 - Manually create a minimum of 3 representative Job Tickets (text files) to be used for testing and evaluation
 - Define the command-line Test Application to exercise the JTAPIs; include an initial set of commands
 - Create Thin-Thread implementation of the individual JTAPIs and the Test Application.
 - This will be the first demonstrational implementation and the start code for detailed development
 - This will include minimum documentation on how to use the Test Application
 - Enhance individual JTAPIs and the Test Application to provide full functionality
 - Provided update documentation as required
 - Project Demonstration



JTAPI GSOC Project – (3)



- Code License:
 - CPL
- Coding Language:
 - Platform Independent C (No platform or vendor-specific extensions, should work on as many platforms as possible)
- Coding Document:
 - In-line commenting must be sufficient to understand the flow and any section requiring extended understanding.
- Operating System:
 - Student's choice – Linux, Windows, Mac, ... (non-gui for either)
- Interface:
 - Command Line – GUI not required unless very simple (due to project time constraint)
- Document: Minimum:
 - How to build the JTAPI library
 - How to build the Test Application
 - The Test Application command-line instructions
 - Three examples of using the Test Application and exercising the JTAPIs