



# Job Ticket Application Programming Interface

Version 0.20  
August 8, 2003

jt-api-spec--V0020-080803.sxw

## Authors (alphabetical)

<i>Authors</i>	<i>Company</i>	<i>Email</i>
Claudia Alimpich	International Business Machines	alimpich@us.ibm.com
Tom Hastings	Xerox	<a href="mailto:hastings@cp10.es.xerox.com">hastings@cp10.es.xerox.com</a>
Till Kampeter	Mandrake	<a href="mailto:till.kampeter@gmx.net">till.kampeter@gmx.net</a>
Ira McDonald	High North	<a href="mailto:imcdonald@sharplabs.com">imcdonald@sharplabs.com</a>
Glen Petrie (Editor)	EPSON	glen.petrie@eitc.epson.com

## Abstract

The Job Ticket Application Programming Interface (API) provides an abstract interface for applications to read, edit, and write industry standard document processing job tickets (defined outside this specification). This specification defines an abstract model of operations, objects (jobs, documents, etc.) and attributes for document processing. This specification also defines “C” and Java bindings of this abstract object model (including standard header files).

Copyright 2003, Free Standards Group

## Copyright Notice

[[ Editor: Is this the correct copyright notice for an FSG standards? ]]

Copyright (c) 2003 Free Standards Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this documentation files, to deal in the documentation without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the documentation, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the documentation.

THE DOCUMENTATION IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENTATION OR THE USE OR OTHER DEALINGS IN THE DOCUMENTATION.

## Table of Contents

- 1.INTRODUCTION.....4
- 2.TERMINOLOGY AND ACRONYMS <all>.....4
  - 2.1 TERMINOLOGY < All >.....4
  - 2.2 ACRONYMS < All >.....4
- 3. HIGH LEVEL MODEL < Claudia >.....5
  - 3.1 DIAGRAM < Claudia >.....5
  - 3.2 OBJECT DESCRIPTIONS< Claudia >.....8
- 4. “C” IMPLEMENTATION FRAMEWORK < who >.....8
  - 4.1 HEADER FILES RELATIONSHIPS < who >.....8
  - 4.2 HEADER FILE NAMING CONVENTION < who >.....8
  - 4.3 LIST OF HEADER FILES < who >.....8
- 5. JAVA IMPLEMENTATION FRAMEWORK .....8
- 6. CONFORMANCE.....8
- 7. INERNATIONALIZATION.....8
- 8. SECURITY.....9
- 9. NORMATIVE REFERENCES < All, Tom to write example >.....9
- 10. INFORMATIVE REFERENCES < All, Tom to write example >.....9
- APPENDIX: X CHANGES < Editor >.....9

### REFERENCES

[RFC2119]

## 1. INTRODUCTION

This specification describes and defines the Job Ticket Application Programming Interface (API) for the Free Standards Group. The Job Ticket API defines the abstract interface and does not imply or describe a job ticket or job ticket format. An implementation of the Job Ticket API (JTAPI) produces one or more the existing standard job tickets such as CIP4 Job Description Format (JDF) or the PWG Job Ticket Format.

[ Items to include: paragraph about JDF, CIP4 and PWG job ticket. Syntax neutral ] < Tom >

This specification has been developed and is intended for software developers, designers and architects.

The purpose of this specification is to define a Job Ticket Application Programming Interface (JTAPI) for the Linux/Unix/Posix Open Printing framework. The JTAPI creates and consumes job-tickets. The JTAPI is job-ticket syntax neutral. The JTAPI isolates an application from the content of a job-ticket. The JTAPI is programming language neutral.

## 2. TERMINOLOGY AND ACRONYMS <all>

### 2.1 CONFORMANCE TERMINOLOGY < All >

In this document, the uppercase terms “MUST”, “MUST NOT”, “REQUIRED”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are intended to be interpreted as described in [RFC2119]

In this document, each conformance statement uses one of the terms:

<i>Term</i>	<i>Meaning</i>
<b>MUST</b>	Implementation support of the element is REQUIRED.
<b>SHOULD</b>	Implementation support of the element is RECOMMENDED.
<b>MAY</b>	Implementation support of the element is OPTIONAL.

In this document, each operation, object, or attribute is defined as:

<i>Term</i>	<i>Meaning</i>
<b>REQUIRED</b>	Each implementation <b>MUST</b> support this element.
<b>RECOMMENDED</b>	Each implementation <b>SHOULD</b> support this element
<b>OPTIONAL</b>	Each implementation <b>MAY</b> support this element.

In this document, the term “support” is defined as:

<i>Term</i>	<i>Meaning</i>
<b>“support an operation”</b>	An implementation <b>MUST</b> accept a syntactically correct instance of the operation (includes all <b>REQUIRED</b> parameters) and <b>MUST</b> return one of the defined results for the invoked operation.
<b>“support an object”</b>	An implementation <b>MUST</b> accept a syntactically correct instance of the object (includes all <b>REQUIRED</b> attributes) and <b>MUST</b> return one of the defined results for the invoked operation.
<b>“support an attribute”</b>	An implementation <b>MUST</b> accept a syntactically correct instance of the attribute (includes a valid value) and <b>MUST</b> return one of the defined results for the invoked operation. Further, an implementation <b>MUST</b> accept at least one value (other than the empty value <b>NOT_SET</b> ) defined for the given attribute.

## 2.2 OTHER TERMINOLOGY <all>

<i>Term</i>	<i>Meaning</i>
<b>Job Ticket</b>	A set of one or more document processing activities, contained within a file or stream, serialized into some file format (for example, the XML instance textual encoding of a CIP4 JDF Job Ticket)..
<b>Job</b>	A set of one or more document processing activities, contained within a “Job Ticket”, described by “Job Description” attributes, and processed according to “Job Processing” attributes.
<b>Document</b>	A single document to be processed, contained within a “JOB”, described by “Document Description” attributes, and processed according to “Document Processing” attributes..
<b>Page</b>	A single logical page in a source document. Multiple source pages may be included in a single “Impression” on a “Side” of a “Sheet”.
<b>Sheet</b>	A single physical piece of media.
<b>Side</b>	A single side (front or back) of “Sheet” of media.
<b>Impression</b>	A single image marked (by some means) on a “Side” of a “Sheet”.

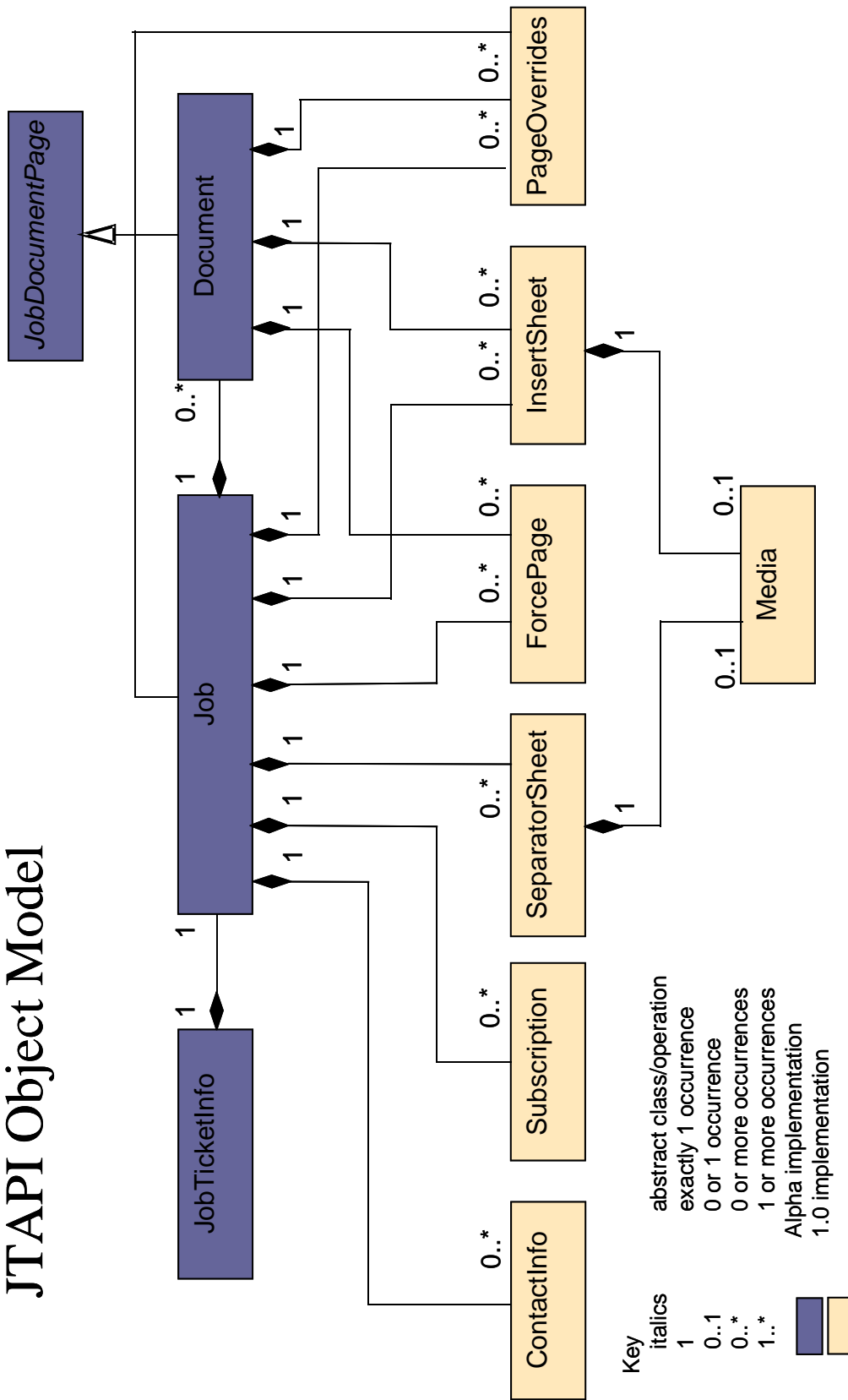
## 2.3 ACRONYMS < All >

<i>Acronyms</i>	<i>Meaning</i>	<i>Source</i>
<b>CIP4</b>	International Cooperation for the Integration of Processes in Prepress, Press and Postpress	<a href="http://www.cip4.org/">http://www.cip4.org/</a>
<b>JDF</b>		
<b>PWG</b>		
<b>FSG</b>		

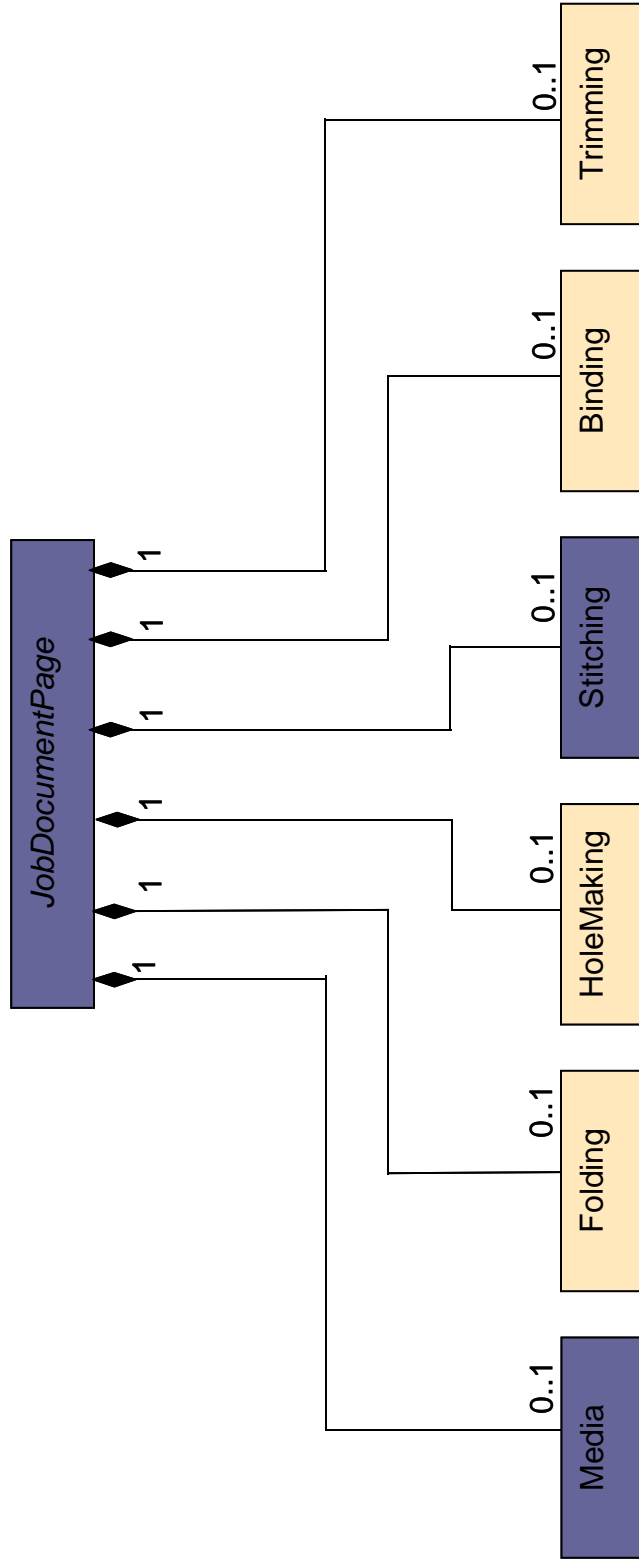
## 3. HIGH LEVEL MODEL < Claudia >

### 3.1 DIAGRAM < Claudia >

# JTAPI Object Model



# JTAPI Object Model (cont)



Complete set of UML diagrams:  
[ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/JTAPI\\_Diagrams/](ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/JTAPI_Diagrams/)



### **3.2 OBJECT DESCRIPTIONS < Claudia >**

## **4. “C” IMPLEMENTATION FRAMEWORK < who >**

### **4.1 HEADER FILES RELATIONSHIPS < who >**

### **4.2 HEADER FILE NAMING CONVENTION < who >**

### **4.3 LIST OF HEADER FILES < who >**

## **5. JAVA IMPLEMENTATION FRAMEWORK**

To be determined.

## **6. CONFORMANCE**

[[ Note: Should there be statement of the client/applications conformance using the API ? ]]

A conforming library implementation of the Job Ticket API v1.0:

- (1) **MUST** support the abstract object model defined in Section 3 of this specification;
- (2) **MUST** publish one or more sets of standard header files verbatim from Section 4 or Section 5 of this specification;
- (3) **MAY** publish one or more sets of vendor extension header files for each supported language binding defined in this specification;
- (4) **MUST** support every **REQUIRED** object and attribute (with at least one supported value) defined in this specification;
- (5) **MAY** support any **OPTIONAL** object or attribute (with at least one supported value) defined in this specification;
- (6) **MUST** support read/write access to one or more industry standard job ticket formats (defined outside this specification);
- (7) **MAY** support read/write access to one or more vendor-specific job ticket formats (defined outside this specification).

## **7. INERNATIONALIZATION**

Each library implementation of the Job Ticket API **MUST** accept a charset tag [RFC2978] to specify the character set and encoding for all text strings. Each library implementation of the Job Ticket API **MUST** default to the UTF-8 [RFC2279] transform of [ISO10646] for text strings.

**ISSUE:** The Job Ticket API does **NOT** support the transfer of a language tag [RFC3166] to specify the natural language of text strings, although all XML-based job ticket formats support a language tag. Therefore the Job Ticket API only partially conforms to the IETF Policy on Character Sets and Languages [RFC2277].

## **8. SECURITY**

The Job Ticket API does **NOT** support the transfer of any user security credentials. Each implementation of the Job Ticket API is a library that may be statically or dynamically linked with an application program. The application program itself may be authenticated and authorized by the native host operating system (by means outside the scope of this specification) for read and/or write access to job tickets stored on local or network file systems.

## **9. NORMATIVE REFERENCES < All, Tom to write example >**

## **10. INFORMATIVE REFERENCES < All, Tom to write example >**

## **APPENDIX: X CHANGES < Editor >**

# End of Document