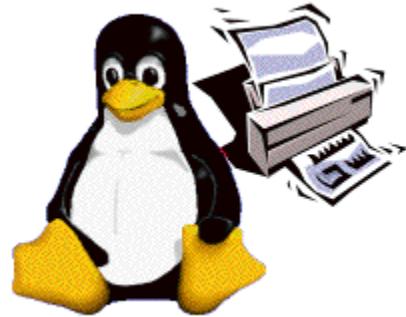# Job Ticket
Application Programming Interface
(JTAPI)

Version 0.99
March 14, 2005

Abstract

The Free Standards Group (FSG) OpenPrinting Job Ticket Application Programming Interface (JTAPI) provides an abstract interface for applications to read, edit, and write document processing job tickets (defined outside this specification). This specification defines an abstract model of objects (jobs, documents, etc.) and their operations and attributes for document processing (e.g. scanning, printing, copying, etc.). This specification also defines C language bindings of this abstract object model (including standard header files).

Copyright Notice

Table of Contents

1. Introduction

This specification describes and defines the Free Standards Group (FSG) Open Printing Job Ticket Application Programming Interface (JTAPI) for the Free Standards Group.   The JTAPI defines the abstract interface and does define a specific job ticket, job ticket file format or job ticket syntax. An implementation of the JTAPI produces and/or consumes one or more standard or vendor specific job tickets formats.

This specification is intended for software developers, designers and architects that need to read and/or write job tickets.

The purpose of this specification is to define an open standard, the Job Ticket Application Programming, for open printing on Linux/Unix/Posix/Windows/Macintosh/Embedded platforms. A JTAPI implementation produces and consumes job tickets.   The JTAPI is job ticket syntax neutral. The JTAPI isolates an application from the syntax of a job ticket to hind details and the structural complexity of specific job ticket along with interoperability between different job ticket file formats. The JTAPI abstract model is programming language neutral.

2. Public Standard Job Tickets

The International Cooperation for the Integration of Process in Prepress, Press and Postpress (CIP4) is a joint initiative of vendors for the graphical arts industry [cip4]. CIP4 has published a Job Definition Format (JDF) specification. JDF is a comprehensive XML-based file format proposed industry standard for end-to-end job ticket specifications combined with a message description standard and message interchange protocol to cover all aspects of the commercial printing workflows.

JDF / 1.0 was published April of 2001. JDF / 1.1 was published in August 2002 [jdf]. JDF / 1.2 was published in May 2004. JDF / 1.3 is due to be published the mid 2005. Yearly updates are foreseen as the needs of the printing industry evolve.

The Printer Working Group (PWG) is a joint initiative of printer vendors and print system providers to develop printing protocol standards for use on the Internet and within enterprises on their intra nets [pwg]. The PWG has published the Internet Printing Protocol (IPP) in September 2000 [rfc2910, rfc2911]

The PWG has published the PWG Semantic Model which summarizes the printing semantics common to a number of printing protocols, centered on the IPP semantics [pwg-sm]. The PWG Semantic Model includes an XML Schema definition. An XML Job Ticket is defined in the current PWG Semantic Model.

3. Overview

3.1   Print Environments

JTAPI has defined terms four distinct print environments, each having different user models, print scenarios and print requirements.   The following is a brief description of the four print environments.

3.1.1 Mobile

The mobile print environment is characterized by client devices that are small personal digital appliances used by a single user.   Examples include Personal Digital Assistants (PDA's), cell phones, digital cameras, Set Top Box's (STB) for TV applications and so forth.   All of these client devices have small system resource (memory) and slow processors.   Devices communication varies between low speed wireless connections (cellular, Bluetooth and 802.11) to direct USB or Firewire connections.   Typical application print sources include simple text editor, email, web browsers and/or camera.   In all cases, each print job consists of a single document with few pages and optional number of copies.   Typically, the print jobs are single sided (simplex) without layout or finishing options.

3.1.2 Desktop/Home

The desktop/home print environment is characterized by client devices that are single user desktop/laptop computers.   The client devices may have extended system resources (memory and disk) and faster processors depending on the user specific applications.   These client devices typically have directly connected printers but may support home (local-area) networks to centralized printers.   Print jobs can be black-and-white or color and be printed on low volume printers.   The printing speeds are typically less than 50 ppm (pages-per-minute) for black-and-white printing and less than 20 ppm for color printing.   This print environment supports multiple jobs usually having a single document.   Simple layout features includes simplex or manual duplex print jobs with no finishing options.

3.1.3 Office

The office print environment is similar to the desktop/home print environment except with extended job, network and print speed capabilities.   Although individual jobs still have simple layout options, they support multiple documents with non-manual duplexing.   Office print environment is extended to included local-area-networks (LAN) and wide-area-networks (WAN). Printing speeds range from 50 to 100 ppm for black-and-white printing and 20 to 60 ppm for color along with simple finishing options such as stapling, sorting and collating.

3.1.4 Production

The production print environment is characterized by large, high-speed printers (presses or digital printers) that are controlled by trained operators who typically receive jobs content and instructions from a pre-press printing process.   Productions printers are network connected within the print-shop and may supported external customer connectivity over secure channels.   Job content (print data)

can be static or variable consisting of single or multiple documents per print job printed on blank or pre-printed stock.   Printers support complex layout, duplex printing and large cut-sheet or continuous-feed media. Printing speeds exceed 100 ppm for black-and-white printing and 60 ppm color printing with complex layout and finishing options.

3.2   High Level Model

The JTAPI is object oriented extensible API. Figure 1.0 and 2.0 diagrams the JTAPI high level model and the relationship between JTAPI objects. A job ticket information object (JobTicketInfo) contains one job object (Job) where the job contains zero or more document objects (Document). The JobDocumentPage object is abstract and contains functionality that is common to jobs, documents, and specific pages in a job or document. Each of the other objects in the diagrams represents functionality that can be specified for the Job and Document objects. For example, the Media object represents the media that the job, document, or specific pages in the job or document is to be imaged/printed on. The section following the diagrams contains more detailed descriptions of each of the objects in the JTAPI model.
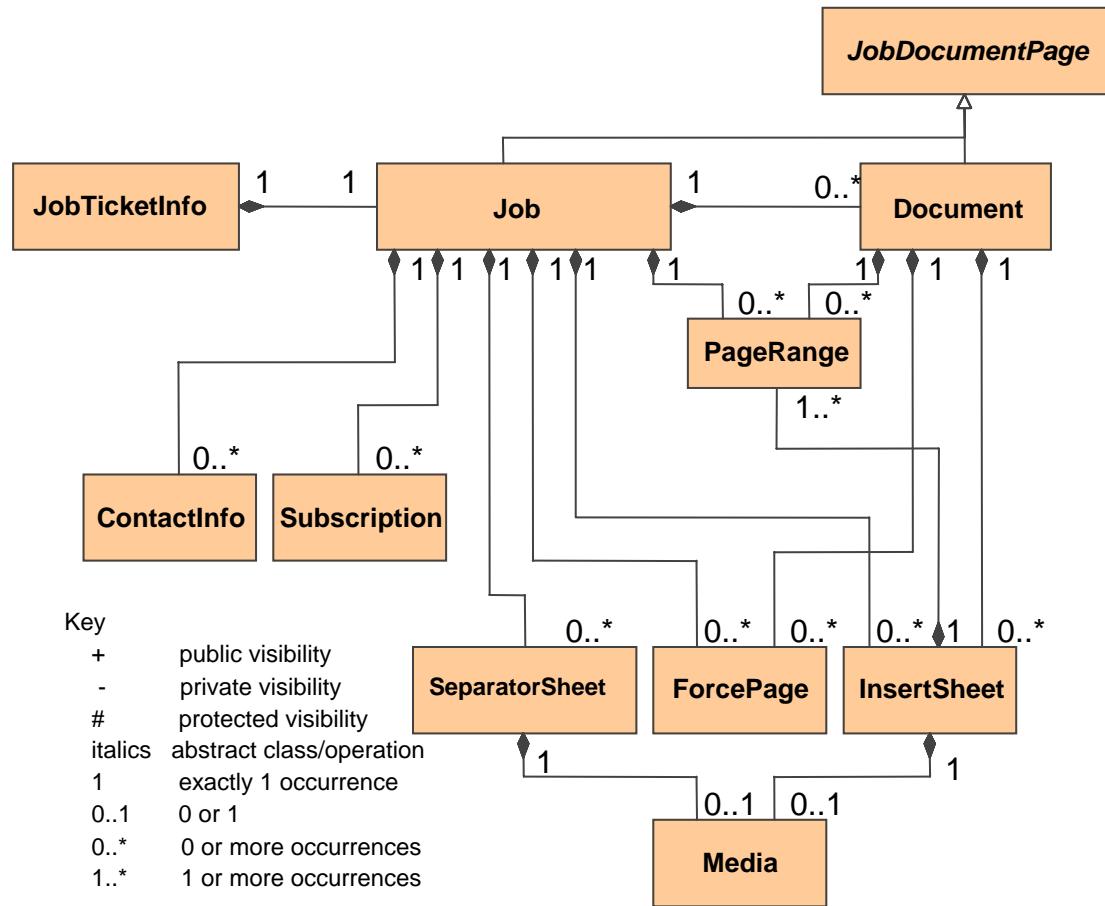
Figure 1.0   JTAPI High Level Object Model
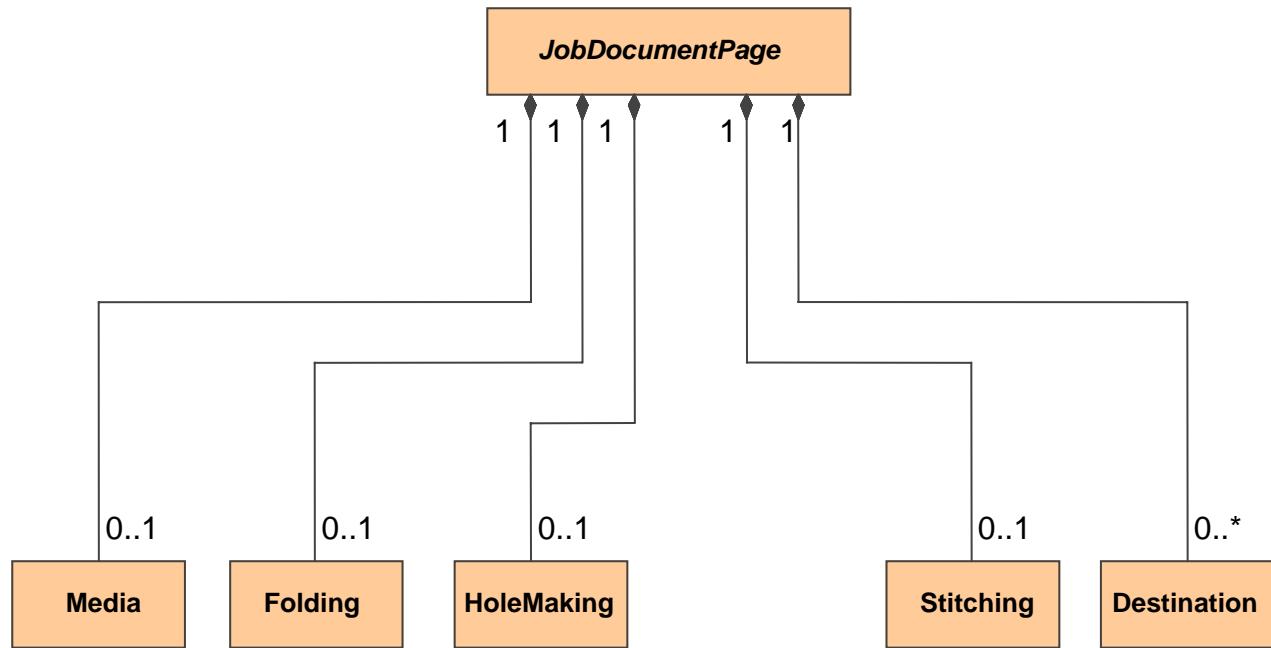
Figure 2.0   JTAPI JobDocumentPage High Level Object Model

3.3   Linking with JTAPI

The Job Ticket API can be provided in the form of a static or dynamic link library. Linking of library details are operating system dependent and will not be elaborated in this specification.

3.4   Conformance with JTAPI

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

(1) MUST support the abstract object model defined in Section 9 of this specification;

(2) MUST publish the set of standard header files verbatim of this specification.

(3) MAY publish one or more sets of vendor extension header files;

(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) MAY support read/write to one or more industry standard or vendor specific job ticket formats (defined outside this specification).

(7) MUST identify supported job ticket formats in any claim of conformance to this specification.

3.5   Internationalization

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 in the JT_Charset. Each library implementation of the Job Ticket API MUST default to the UTF-8 [RFC2279] transform of [ISO10646] for text strings.

The Job Ticket API supports the transfer of language tags [RFC3066] to specify the natural language of text strings.   Therefore the Job Ticket API fully conforms to the IETF Policy on Character Sets and Languages [RFC2277].

3.6   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. A conforming implementation MUST not store any sensitive information (password, private keys, etc). The JTAPI does not define or support encryption of a job ticket.

4. Application Programming Interface Summary

Object	Header	API Request	Argument	Page
---	fsgjt.h	fsgitOpenJTAPI	context	15
---	fsgjt.h	fsgitFlushJTAPI	context	16
---	fsgjt.h	fsgitCloseJTAPI	context	17
all	fsgjt.h	fsgjtDestroy	object	18
all	fsgjt.h	fsgjtGet	object, attributeName, attribute	19
all	fsgjt.h	fsgjtSet	object, attribute	20
all	fsgjt.h	fsgjtSetIntegerAttribute	object, attributeName, value	21
all	fsgjt.h	fsgjtSetObjectAttribute	object, attributeName, value	22
all	fsgjt.h	fsgjtSetObjectAttributeList	object, attributeName, list	23
all	fsgjt.h	fsgjtSetStringAttribute	object, attributeName, value	24
all	fsgjt.h	fsgjtSetStringAttributeList	object, attributeName, list	25
all	fsgjt_attribute.h	fsgjtNewAttribute	attribute, name, type, value	26
all	fsgjt_attribute.h	fsgjtDestoryAttribute	attribute	27
all	fsgjt_attribute.h	fsgjtAddValue	attribute, value	28
all	fsgjt_attribute.h	fsgjtGetName	attribute, name	29
all	fsgjt_attriubte.h	fsgjtGetNextValue	attribute, nextValue, moreValues	30
all	fsgjt_attribute.h	fsgjtGetNumValues	attribute, numValues	31
all	fsgjt_attribute.h	fsgjtGetValueType	attribute, type	32
all	fsgjt_attribute.h	fsgjtReplaceValue	attribute, value	33
all	fsgjt_attribute.h	fsgjtResetToFirstValue	attribute	34
contactInfo	fsgjt_contact_info.h	fsgjtNewContactInfo	contactInfo, name	35
destination	fsgjt_destination.h	fsgjtNewDestination	destination, targetUri	36
document	fsgjt_document.h	fsgjtNewDocument	document	37
document	fsgjt_document.h	fsgjtNewDocumentFromURI	document, uri	38
insert sheet	fsgjt_insert_sheet.h	fsgjtNewInsertSheet	insertSheet, position, pages	39
job	fsgjt_job.h	fsgjtNewJob	job	40
job	fsgjt_job.h	fsgjtNewJobFromDocument	job, document	41
job ticket info	fsgjt_job_ticket_info.h	fsgjtNewJobTicketInfo	jobTicketInfo, typeAndVersion	42
job ticket info	fsgjt_job_ticket_info.h	fsgjtNewJobTicketInfoFromURI	jobTicketInfo, uri, typeAndVersion	43
job ticket info	fsgjt_job_ticket_info.h	fsgjtNewJobTicketInfoFromBuffer	jobTicketInfo, buffer, bufferSize, typeAndVersion	44
job ticket info	fsgjt_job_ticket_info.h	fsgjtNewJobTicketInfoFromJob	jobTicketInfo, job, typeAndVersion	45
job ticket info	fsgjt_job_ticket_info.h	fsgjtWriteJobTicketInfoToBuffer	jobTicketInfo, buffer, bufferSize	46

Object	Header	API Request	Argument	Page
job ticket info	fsgjt_job_ticket_info.h	fsgjtWriteJobTicketInfoToURI	jobTicketInfo, uri	47
folding	fsgjt_folding.h	fsgjtNewFolding	folding, foldingType	48
force page	fsgjt_force_page.h	fsgjtNewForcePage	forcePage, page, sheetSide	49
hole making	fsgjt_hole_making.h	fsgjtNewHoleMaking	holeMaking, holeCount	50
media	fsgjt_media.h	fsgjtNewMedia	media, name	51
page range	fsgjt_page_range.h	fsgjtNewPageRange	pageRange, lower, upper	52
separator sheet	fsgjt_separator_sheet.h	fsgjtNewSeparatorSheet	separatorSheet, type	53
stitching	fsgjt_stitching.h	fsgjtNewStitching	stitching, stitchType	54
subscription	fsgjt_subscription	fsgjtNewSubscription	subscription, sendToUri	55

5. Top Level Application Programming Interface

The top level JTAPI is used for system level and/or a specific implementation of JTAPI to perform initialization and release of resources and/or system level protocols.

5.1 fsgjtOpenJTAPI

Syntax:	fsgjt_return_code_t fsgjtOpenJTAPI(context);		
Parameters:	void **	context	Pointer to opaque JTAPI interface structure context structure
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_MISSING_PARAM		
Impact Level:	Library		
Description:	Initializes the JTAPI library and may return a context structure used by the specific JTAPI library instance.		
Constraints, Limitation, Restrictions	This function must be called before calling any other JTAPI functions.		
Also See:	fsgjtCloseJTAPI		
Header:	fsgjt.h		

5.2   fsgjtFlushJTAPI

Syntax:	fsgjt_return_code_t fsgjtFlushJTAPI(context);		
Parameters:	void *	context	Opaque JTAPI structure context structure
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_MISSING_PARAM		
Impact Level:	Library		
Description:	Releases all objects and dynamic memory allocations and closes any I/O channels associated with this specific JTAPI library instance.		
Constraints, Limitation, Restrictions			
Also See:	fsgjtCloseJTAPI		
Header:	fsgjt.h		

5.3 fsgjtCloseJTAPI

Syntax:	fsgjt_return_code_t fsgjtCloseJTAPI(context);		
Parameters:	void *	context	Opaque JTAPI interface context structure
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_MISSING_PARAM		
Impact Level:	Library		
Description:	Releases this specific JTAPI library instance (and internally calls fsgjtFlushJTAPI).		
Constraints, Limitation, Restrictions	This function must be called after calling any other JTAPI functions.		
Also See:	fsgjtOpenJTAPI		
Header:	fsgjt.h		

6. Functional Level Application Programming Interface

Functional level JTAPI functions for used to create, manipulate and store job tickets. The API's are segmented into generic functions that are useable for any object, attribute functions that are useable for any attribute, and functions for specific objects.

6.1   fsgjtDestroy

Syntax:	fsgjt_return_code_t fsgjtDestroy(object);		
Parameters:	fsgjt_object_t	object	Object to free.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Free the memory used by the provided object.		
Constraints, Limitation, Restrictions	"fsghjtDestory MUST be called to free each object allocation during session, otherwise memory leaks will occur. Before calling fsgitCloseJTAPI, fsghjtDestory MUST be called to free each object allocation during session. Destroying higher-level objects will automatically destroy lower level objects.		
Also See:	FsgjtFlushJTAPI		
Header:	fsgjt.h		

6.2   fsgjtGet

Syntax:	fsgjt_return_code_t fsgjtGet(object, attributeName, attribute);		
Parameters:	fsgjt_object_t	object	Object from which to get the attribute.
	fsgjt_text_t	attributeName	Name of the attribute to get.
	fsgjt_attribute_t *	attribute	Attribute having provided name.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Gets the attribute having the provided name.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt.h		

6.3   fsgjtSet

Syntax:	fsgjt_return_code_t fsgjtSet(object, attribute);		
Parameters:	fsgjt_object_t	object	Object in which to set the attribute
	fsgjt_attribute_t	attribute	Attribute to set
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Sets the provided attribute for the provided object.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt.h		

6.4 fsgjtSetIntegerAttribute

Syntax:	fsgjt_return_code_t fsgjtSetIntegerAttribute(object, attributeName, value);		
Parameters:	fsgjt_object_t	object	Set the attribute for this object.
	fsgjt_text_t	attributeName	Name of the attribute to set.
	fsgjt_int32_t	value	New value for the integer-attribute.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Convenience function for setting an integer attribute to the provided value. This function creates a fsgjt_attribute_t, calls the fsgjtSet() function and then destroys the temporary Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt.h		

6.5   fsgjtSetObjectAttribute

Syntax:	fsgjt_return_code_t fsgjtSetObjectAttribute(object, attributeName, value);		
Parameters:	fsgjt_object_t	object	Set the attribute of this object.
	fsgjt_text_t	attributeName	Name of attribute to set.
	fsgjt_object_t	value	New value for the attribute.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Convenience function for setting an object Attribute to the provided value. This function creates a fsgjt_attribute_t, calls the fsgktSet() function and then destroys the temporary Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt.h		

6.6   **fsgjtSetObjectAttributeList**

Syntax:	fsgjt_return_code_t fsgjtObjectAttributeList(object, attributeName, list);		
Parameters:	fsgjt_object_t	object	Set the Attribute for this object
	fsgjt_text_t	attributeName	Name of the Attribute to set
	fsgjt_object_t*	list	List of values for the Attribute.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Convenience function for setting an object Attribute to the provided list of values. This function creates a fsgjt_attribute_t, adds the values, calls the fsgjtSet() function and then destroys the temporary Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsght.h		

6.7   fsgjtSetStringAttribute

Syntax:	fsgjt_return_code_t fsgjtSetStringAttribute(object, attributeName, value);		
Parameters:	fsgjt_object_t	object	Set the Attribute of this object
	fsgjt_text_t	attributeName	Name of Attribute to set
	fsgjt_text_t	value	New value for the Attribute.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Convenience function for setting a string Attribute to the provided value. This function creates a fsgjt_attribute_t, calls the fsgjtSet() function and then destroys the temporary Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt.h		

6.8 fsgjtSetStringAttributeList

Syntax:	fsgjt_return_code_t fsgjtSetStringAttributeList(object, attributeName, values);		
Parameters:	fsgjt_object_t	object	Set the attribute for this object
	fsgjt_text_t	attributeName	Name of attribute to set
	fsgjt_text_t*	values	List of new values for the attribute.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Convenience function for setting a text Attribute list to the provided values. This function creates a fsgjt_attribute_t, calls the fsgjtSet() function and then destroys the temporary Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt.h		

7. Attribute Level Application Programming Interfaces

Attribute level JTAPI functions for used to create, manipulate and store object attributes.

7.1   fsgjtNewAttribute

Syntax:	fsgjt_return_code_t fsgjtNewAttribute(attribute, name, type, value);		
Parameters:	fsgjt_attribute_t*	attribute	Pointer to the new Attribute returned
	fsgjt_text_t	name	Name for the new Attribute to create
	fsgjt_text_t	type	Type for the new Attribute to create
	void*	value	Pointer to the value for this attribute.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Creates a new Attribute object attribute having the provided attribute name, value type, and value.		
Constraints, Limitation, Restrictions	Multiple values should be added with the fsgjtAddValue() function.		
Also See:			
Header:	fsgjt_attribute.h		

7.2 fsgjtDestoryAttribute

Syntax:	fsgjt_return_code_t fsgjtDestoryAttribute(attribute);		
Parameters:	fsgjt_attribute_t	attribute	Attribute to free.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Free the memory used by the Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.3   fsgjtAddValue

Syntax:	fsgjt_return_code_t fsgjtAddValue(attribute, value);		
Parameters:	fsgjt_attribute_t	attribute	Attribute to add the provided value to.
	void*	value	Pointer to value to add.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_NOT_SUPPPORTED		
Impact Level:	Attribute		
Description:	Add an additional value to this Attribute		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.4 fsgjtGetName

Syntax:	fsgjt_return_code_t fsgjtGetName(attribute, name);		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose name to get.
	fsgjt_text_t *	name	Name of the Attribute
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Get the name of the Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.5   **fsgjtGetNextValue**

Syntax:	fsgjt_return_code_t fsgjtGetNextValue(attribute, nextValue, moreValues)		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose next value is to be returned.
	void**	nextValue	Pointer to the next value.
	fsgjt_present_t*	moreValues	"On" if there are more values for the attribute for the Attribute other than the value being returned.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Return the Attribute's next value.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.6   fsgjtGetNumValues

Syntax:	fsgjt_return_code_t fsgjtGetNumValues(attribute, numValues);		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose number of values to get.
	fsgjt_int32_t*	numValues	Number of values
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Get the number of values that the Attribute contains.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.7   fsgjtGetValueType

Syntax:	fsgjt_return_code_t fsgjtGetValueType(attribute, type);		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose value type to get.
	fsgjt_text_t *	type	Type of this Attribute's values (use Value Types).
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Attribute		
Description:	Returns the type of the Attribute's values.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.8   fsgjtReplaceValue

Syntax:	fsgjt_return_code_t fsgjtReplaceValue (attribute, value);		
Parameters:	fsgjt_attribute_t	attribute	Attribute to replace the existing value
	void*	value	Pointer to replacement value
Return Status:	FSGJT_OK		
	FSGJT_INTERNAL_ERROR		
	FSGJT_INVALID_PARAM		
	FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Replaces the existing value for the Attribute with the provided value.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_attribute.h		

7.9 fsgjtResetToFirstValue

Syntax:	fsgjt_return_code_t fsgjtResetToFirstValue(attribute);		
Parameters:	fsgjt_attribute_t	attribute	Attribute to reset
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Reset the iterator to point to the first value of the Attribute.		
Constraints, Limitation, Restrictions			
Also See:			
Header	fsgjt_attribute.h		

8. Object Level Application Programming Interfaces

Object level JTAPI functions for used to create, manipulate and store objects.

8.1   fsgjtNewContactInfo

Syntax:	fsgjt_return_code_t fsgjtNewContactInfo(contactInfo, name);		
Parameters:	fsgjt_object_t*	contactInfo	Pointer to the new ContactInfo object to create.
	fsgjt_text_t	name	Name of person to contact or NULL if ContactInfo name not being specified.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a ContactInfo object having the provided name.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_contact_info.h		

8.2 fsgjtNewDestination

Syntax:	fsgjt_return_code_t fsgjtNewDestination(destination, targetUri);		
Parameters:	fsgjt_object_t*	destination	Pointer to the new Destination object to create.
	fsgjt_uri_t	targetUri	URI that is the target location for the this destination
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Destination object having the provided target location URI.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_destination.h		

8.3   fsgjtNewDocument

Syntax:	fsgjt_return_code_t fsgjtNewDocument(document);		
Parameters:	fsgjt_object_t*	document	Pointer to the new document to be created.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Create a Document object.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_document.h		

8.4 fsgjtNewDocumentFromURI

Syntax:	fsgjt_return_code_t fsgjtNewDocumentFromURI(document, uri);		
Parameters:	fsgjt_object_t*	document	Pointer to the new document to be created.
	fsgjt_uri_t	uri	UIR of the document data or NULL if no data file is to be associated with job ticket.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Document object using the provided URI that contains the document data.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_document.h		

8.5   fsgjtNewInsertSheet

Syntax:	fsgjt_return_code_t fsgjtNewInsertSheet(insertSheet, position, pages);		
Parameters:	fsgjt_object_t*	insertSheet	Pointer to the new InsertSheet object to create
	fsgjt_position_t	position	Position of insert sheet (e.g., before)
	fsgjt_object_t*	pages	List of PageRange objects (scope) for the insert sheet operation.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a InsertSheet object having the provided position and page ranges.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_job.h		

8.6   fsgjtNewJob

Syntax:	fsgjt_return_code_t fsgjtNewJob(job);		
Parameters:	fsgjt_object_t*	job	Pointer to Job object to be created.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Job object without providing a Document.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_job.h		

8.7 fsgjtNewJobFromDocument

Syntax:	fsgjt_return_code_t fsgjtNewJobFromDocument(job, document);		
Parameters:	fsgjt_object_t*	job	Pointer to Job object to be created.
	fsgjt_object_t	document	Document to be contained in this job.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Job object containing the provided Document.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_job.h		

8.8   fsgjtNewJobTicketInfo

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfo(jobTicketInfo, typeAndVersion);		
Parameters:	fsgjt_object_t*	jobTicketInfo	Pointer to the new JobTicketInfo to create
	fsgjt_jobt_ticket_type_version_ t	typeAndVersion	Type and version of this Job Ticket
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a JobTicketInfo object. Used when a new JobTicketInfo is being created without a Job object and not from an external job ticket.		
Constraints, Limitation, Restrictions	Use JobTicketTypeVersionEnum		
Also See:			
Header:	fsgjt_job_ticket_info.h		

8.9   fsgjtNewJobTicketInfoFromURI

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfoFromURI (jobTicketInfo, uri, typeAndVersion);		
Parameters:	fsgjt_object_t*	jobTicketInfo	Pointer to the new JobTicketInfo to create
	fsgjt_uri_t	uri	URI where job ticket file resides
	fsgjt_job_ticket_type_version_t	typeAndVersion	Type and version of this job ticket.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	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.		
Constraints, Limitation, Restrictions	Use JobTicketTypeVersionEnum		
Also See:			
Header:	fsgjt_job_ticket_info.h		

8.10 fsgjtNewJobTicketInfoFromBuffer

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfoFromBuffer (jobTicketInfo, buffer, buffersize, typeAndVersion);		
Parameters:	fsgjt_object_t*	jobTicketInfo	Pointer to the new JobTicketInfo to create
	fsgjt_text_t	buffer	Buffer that contains the job ticket
	fsgjt_int32_t	bufferSize	Size of the buffer.
	fsgjt_job_ticket_type_version_t	typeAndVersion	Type and version of this job ticket
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	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.		
Constraints, Limitation, Restrictions	For charset use the tag from the IANA Charset Registry. Use JobTicketTypeVersionEnum		
Also See:			
Header:	fsgjt_job_ticket_info.h		

8.11 fsgjtNewJobTicketInfoFromJob

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfoFromJob		
			(jobTicketInfo, job, typeAndVersion);
Parameters:	fsgjt_object_t*	jobTicketInfo	Pointer to the new JobTicketInfo to create
	fsgjt_object_t	job	Job object that this JobTicketInfo is to contain
	fsgjt_job_ticket_type_version_t	typeAndVersion	Type and version of this job ticket
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a JobTicketInfo object using the provided Job object. Used when new job ticket is being created. The Job object must be created first.		
Constraints, Limitation, Restrictions	Use JobTicketTypeVersionEnum		
Also See:			
Header:	fsgjt_job_ticket_info.h		

8.12　fsgjtWriteJobTicketInfoToBuffer

Syntax:	fsgjt_return_code_t fsgjtWriteJobTicketInfoToBuffer (jobTicketInfo, buffer, buffersize);		
Parameters:	fsgjt_object_t	jobTicketInfo	JobTicketInfo from which the job ticket is to be written
	fsgjt_text_t	buffer	Buffer to write the job ticket to.　The buffer must be created by the caller.
	fsgjt_int32_t	bufferSize	Size of buffer that caller created.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Writes a job ticket to the provided buffer.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_job_ticket_info.h		

8.13 fsgjtWriteJobTicketInfoToURI

Syntax:	fsgjt_return_code_t fsgjtWriteJobTicketInfoToURI(jobTicketInfo, uri);		
Parameters:	fsgjt_object_t	jobTicketInfo	JobTicketInfo from which the job ticket is to be written
	fsgjt_uri_t	uri	URI of the job ticket that is to be created from the JobTicketInfo.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Write a job ticket at the provided URI.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_job_ticket_info.h		

8.14 fsgjtNewFolding

Syntax:	fsgjt_return_code_t fsgjtNewFolding (folding, foldingType);		
Parameters:	fsgjt_object_t*	folding	Pointer to the new Folding object to create
	fsgjt_folding_type_t	foldingType	Type of folding to be performed.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Folding object having the provided folding type.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_folding.h		

8.15 fsgjtNewForcePage

Syntax:	fsgjt_return_code_t fsgjtNewForcePage(forcePage, page, sheetSide);		
Parameters:	fsgjt_object_t*	forcePage	Pointer to the new ForcePage object to create
	fsgjt_int32_t	page	Page number where the force page is to occur.
	fsgjt_sheet_side_t	sheetSide	Sheet side to force at the provided page number.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a ForcePage object having the provided page and sheet side.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_force_page.h		

8.16 fsgjtNewHoleMaking

Syntax:	fsgjt_return_code_t fsgjtNewHoleMaking		
	(holeMaking, holeCount);		
Parameters:	fsgjt_object_t*	holeMaking	Pointer to the new HoleMaking object to create
	fsgjt_int32_t	holeCount	Count of the holes to make (punch, drill, etc.) for this HoleMaking object.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a HoleMaking object having the provided hole count.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_hole_making.h		

8.17 fsgjtNewMedia

Syntax:	fsgjt_return_code_t fsgjtNewMedia (media, name);		
Parameters:	fsgjt_object_t*	media	Pointer to the new Media to create
	fsgjt_text_t	name	Name of the media or NULL if Media name is not being specified. The name can be any string
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Media object using the provided name.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_media.h		

8.18 fsgjtNewPageRange

Syntax:	fsgjt_return_code_t fsgjtNewPageRange(pageRange, lower, upper);		
Parameters:	fsgjt_object_t*	pageRange	Pointer to the new PageRange object to create.
	fsgjt_int32_t	lower	Page number where the page range is to begin (inclusive).
	fsgjt_int32_t	upper	Page number where the page range is to end (inclusive).
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a PageRange object having the provided lower and upper bounds.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_page_range.h		

8.19 fsgjtNewSeparatorSheet

Syntax:	fsgjt_return_code_t fsgjtNewSeparatorSheet (separatorSheet, type);		
Parameters:	fsgjt_object_t*	separatorSheet	Pointer to the new SeparatorSheet object to create.
	fsgjt_separator_sheet_type_t	type	Type of separator sheet (e.g., blank)
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Separator Sheet object having the provided type and media.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_separator_sheet.h		

8.20 fsgjtNewStitching

Syntax:	fsgjt_return_code_t fsgjtNewStitching(stitching, stitchingType);		
Parameters:	fsgjt_object_t*	stitching	Pointer to the new Stitching object to create.
	fsgjt_stitching_type_t	stitchingType	Type (edge, corner, etc.) for this Stitching object.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Stitching object having the provided stitching count, reference edge and type.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_stitching.h		

8.21 fsgjtNewSubscription

Syntax:	fsgjt_return_code_t fsgjtNewSubscription(subscription, sendToUri);		
Parameters:	fsgjt_object_t*	subscription	Pointer to the new Subscription object to create.
	fsgjt_uri_t	sendToUri	URI for delivery of notification for this Subscription.
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Subscription object having the provided notification URi.		
Constraints, Limitation, Restrictions			
Also See:			
Header:	fsgjt_subscription.h		

9. Object Level Application Programming Interfaces Conformance Summary

Based on the characteristics, resource limitations and print capabilities of the separate print environments the level of support for individual JTAPI objects is outlined below.   The print environment type and the specific conformance version will be identified as part of the JTAPI initialization process.

JTAPI Object Conformance for Print Environments

Object	Conformance – 1.0 Release (Must, May)			
	Embedded/Mobile	Desktop/Home	Office/Network	Production
ContactInfo	May	May	May	**Must**
Destination	May	May	May	**Must**
Document	**Must**	**Must**	**Must**	**Must**
Folding	May	May	May	**Must**
ForcePage	May	May	May	**Must**
HoleMaking	May	May	May	**Must**
InsertSheet	May	May	May	**Must**
Job	**Must**	**Must**	**Must**	**Must**
JobDocument Page	**Must**	**Must**	**Must**	**Must**
JobTicketInfo	**Must**	**Must**	**Must**	**Must**
Media	**Must**	**Must**	**Must**	**Must**
PageRange	May	**Must**	**Must**	**Must**
SeparatorSheet	May	May	May	**Must**
Stitching	May	May	May	**Must**
Subscription	May	May	May	**Must**

10.   ContactInfo

A ContactInfo object contains information about a person who is to be contacted regarding the job. A ContactInfo object contains attributes that specify the name, address, company, URIs, and role of the person who is to be contacted. The role of the person identifies why the person is interested in the job. Some typical roles are Administrator, Customer, Delivery, and Approver.

10.1   ContactInfo Attributes & Conformance

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
ContactInfo	May	May	May	**Must**
contact-info-address	May	May	May	**Must**
contact-info-company	May	May	May	**Must**
contact-info-contact-uri	May	May	May	**Must** <Mail-To Scheme, Telephone Scheme>
contact-info-details	May	May	May	May
contact-info-name	**Must**	**Must**	**Must**	**Must**
contact-info-roles	**Must** {SENDER}	**Must** {SENDER}	**Must** {SENDER}	**Must** {SENDER, CUSTOMER, DELIVERY, PICKUP, ACCOUNTING }

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

11. Destination

A Destination object contains information about the target URI. A Destination object contains an attribute that specifies the target URI.

11.1 Destination Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
Destination	May	May	May	**Must**
destination-target-uri	May	May	May	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported

12. Document

Each Document object in a job references one or more files to be processed. A Document object contains attributes that describe the document and how it is to be processed.

Only one file per Document object is required to be supported.

12.1 Document Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
Document	Must	Must	Must	Must
document-data-uri	Must	Must	Must	Must

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

13. Folding

A Folding object contains information that specifies how a job, document, or range of pages in a job or document is to be folded. A Folding object contains attributes that specify the type of folding (such as, z-fold and saddle fold) and the reference edge from which to perform the folding operation.

13.1 Folding Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
Folding	May	May	May	**Must**
folding-reference-edge	May	May	May	May
folding-type	May	May	May	**Must** {SADDLE}

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

14.   ForcePage

A ForcePage object is used to specify that a single page, identified by an input page number, is imaged on a specific side (front or back) and/or specific cell of the sheet. Forcing a page is typically performed when the first page of each chapter is to be imaged on the front side of a sheet, also known as "chapter starts". A ForcePage object can also be used to only image on the back side of a back cover. ForcePage is only valid when two-sided printing is specified.

ForcePage will force a page to be imaged on the specified side even if the page would normally fall on the other side. For example, if the second page would normally be imaged on the back side of the sheet and it is to be forced to the front side, then the back side of the current sheet is left blank and the second page is imaged on the next sheet.

14.1   ForcePage Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
ForcePage	May	May	May	**Must**
force-page-cell-position-x	May	May	May	**Must**
force-page-cell-position-y	May	May	May	**Must**
force-page-page	May	May	May	**Must**
force-page-sheet-side	May	May	May	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

15. HoleMaking

A HoleMaking object specifies how a job, document, or range of pages in a job or document is to be punched or drilled. A HoleMaking object contains attributes that specify the number of holes to punch and the edge of the sheet to punch.

15.1 HoleMaking Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
HoleMaking	May	May	May	**Must**
hole-making-count	May	May	May	May
hole-making-reference-edge	May	May	May	**Must** {LEFT, RIGHT}

Not-Set is required for all attribute implemented. "{-}" indicates values to be supported.

16. InsertSheet

An InsertSheet object specifies information that describes an insert sheet that is to be inserted in the job or document. An insert sheet will not be imaged/printed with content from the document data stream.

An InsertSheet object contains attributes that specify where the insert sheet is to be placed in the job or a document (before or after specific page numbers), the media that is to be used for the insert sheet, the number of insert sheets to insert, and other insert sheet information.

16.1 InsertSheet Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
InsertSheet	May	May	May	**Must**
insert-sheet-content	May	May	May	**Must**
insert-sheet-count	May	May	May	May
insert-sheet-media	May	May	May	**Must**
insert-sheet-pages	**Must**	**Must**	**Must**	**Must**
insert-sheet-position	**Must**	**Must**	**Must**	**Must**

Not-Set is required for all attribute implemented. "{-}" indicates values to be supported.

17.　Job

A Job object specifies a description of a job and how it is to be processed. A Job object contains attributes that specify job-name (description) and job-copies (processing).

17.1　Job Attributes & Conformance

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
Job	**Must**	**Must**	**Must**	**Must**
job-billing-code	May	May	May	**Must**
job-binding	May	May	May	**Must**
job-client-id	May	May	**Must**	**Must**
job-collate	May	**Must**	**Must**	**Must**
job-comment	May	May	**Must**	**Must**
job-compression	**Must** {NONE}	**Must** {NONE}	**Must** {NONE}	**Must** {NONE}
job-contact-info	May	May	May	**Must**
job-copies	May	May	**Must**	**Must**
job-create-user-name	**Must**	**Must**	**Must**	**Must**
job-destinations	**Must** {Single-Valued}	**Must** {Single - Valued}	**Must** {Single-Valued}	**Must** {Single-Valued}
job-document-format	**Must** {application/octet-stream}	**Must** {application/octet-stream}	**Must** {application/octet-stream}	**Must** {application/octet-stream}
job-document-natural-language	May	**Must**	**Must**	**Must**
job-documents	**Must**	**Must**	**Must**	**Must**
job-feed-orientation	May	May	May	**Must**
job-fit-policy	**Must** {FIT_TO_PAGE}	**Must** {FIT_TO_PAGE}	**Must** {FIT_TO_PAGE}	**Must** {FIT_TO_PAGE}
job-folding	May	May	May	**Must**
job-force-pages	May	May	May	**Must**
job-hold	May	May	May	**Must**
job-hole-making	May	May	May	**Must**
job-image-alignment-x	May	May	May	**Must**
job-image-alignment-y	May	May	May	**Must**
job-image-shift-back-x	May	May	May	**Must**

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
job-image-shift-back-y	May	May	May	**Must**
job-image-shift-front-x	May	May	May	**Must**
job-image-shift-front-y	May	May	May	**Must**
job-include-pages	May	May	May	**Must**
job-insert-sheets	May	May	May	**Must**
job-jog-offset	May	May	May	**Must**
job-margin-bottom	**Must**	**Must**	**Must**	**Must**
job-margin-left	**Must**	**Must**	**Must**	**Must**
job-margin-right	**Must**	**Must**	**Must**	**Must**
job-margin-top	**Must**	**Must**	**Must**	**Must**
job-media	**Must**	**Must**	**Must**	**Must**
job-message-to-operator	May	May	May	**Must**
job-name	May	May	**Must**	**Must**
job-number-up-x	May	**Must**	**Must**	**Must**
job-number-up-y	May	**Must**	**Must**	**Must**
job-output-bin-name	May	**Must**	**Must**	**Must**
job-output-pages	**Must**	**Must**	**Must**	**Must**
job-page-delivery	May	May	May	**Must**
job-presentation-direction	May	May	May	**Must**
job-print-quality	**Must**	**Must**	**Must**	**Must**
job-priority	**Must**	**Must**	**Must**	**Must**
job-resolution-x	**Must**	**Must**	**Must**	**Must**
job-resolution-y	**Must**	**Must**	**Must**	**Must**
job-rotation	May	**Must**	**Must**	**Must**
job-scaling-factor-x	**Must**	**Must**	**Must**	**Must**
job-scaling-factor-y	May	May	May	**Must**
job-separator-sheets	May	May	**Must**	**Must**
job-sides	May	**Must**	**Must**	**Must**
job-stitching	May	May	**Must** {(CORNER,TOP-LEFT)}	**Must**
job-subscriptions	May	May	May	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

18. JobDocumentPage

JobDocumentPage is an abstract object that contains attributes that can be specified for the Job or Document objects. For example, one media can be specified for the entire job, which can be overridden by another media for a specific document in the job, which can be overridden by another media for specific pages in the job or document.

18.1 JobDocumentPage Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
JobDocument Page	Must	Must	Must	Must

Not-Set is required for all attribute implemented. "{-}" indicates values to be supported.

19.   JobTicketInfo

A JobTicketInfo object specifies job ticket information, such as a comment describing the job ticket, the version of the JTAPI, the type or syntax (JDF, PWG, etc.) of job ticket, and other job ticket information. A JobTicketInfo object does not contain job processing information.

19.1 JobTicketInfo Attributes & Conformance

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
JobTicketInfo	**Must**	**Must**	**Must**	**Must**
jt-api-charset	**Must** {"UTF8"}	**Must** {"UTF8"}	**Must** {"UTF8"}	**Must** {"UTF8"}
jt-api-version	**Must**	**Must**	**Must**	**Must**
jt-author-name	May	May	**Must**	**Must**
jt-charset	**Must**	**Must**	**Must**	**Must**
jt-comment	May	**Must**	**Must**	**Must**
jt-fidelity	**Must**	**Must**	**Must**	**Must**
jt-job	**Must**	**Must**	**Must**	**Must**
jt-length-units	**Must** {MICRO}	**Must** {MICRO}	**Must** {MICRO}	**Must** {MICRO}
jt-type-and-version	**Must**	**Must**	**Must**	**Must**
jt-version	May	May	May	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

20.   Media

A Media object specifies a media that is to be used when printing the job, document, or a range of pages in the job or document. A Media object contains attributes that specify a name that describes the media dimensions (for example, na_letter_8.5x11in), the color of the media, the actual media dimensions, and other media information.

20.1   Media Attributes & Conformance

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
Media	**Must**	**Must**	**Must**	**Must**
media-back-coating	May	May	May	**Must**
media-color	May	May	**Must**	**Must**
media-color-name	May	May	**Must**	**Must**
media-description	May	May	May	**Must**
media-front-coating	**Must**	**Must**	**Must**	**Must**
media-hole-count	May	May	May	**Must**
media-input-tray-name	May	May	**Must**	**Must**
media-manual-feed	**Must**	**Must**	**Must**	**Must**
media-name	May	May	May	**Must**
media-preprinted	May	May	May	**Must**
media-recycled-percentage	May	May	May	**Must**
media-set-count	May	May	May	**Must**
media-size-name	**Must**	**Must**	**Must**	**Must**
media-type	**Must**	**Must**	**Must**	**Must**
media-weight	May	May	May	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

21. Page Range

A Page Range object specifies the range of page that .

21.1 Page Range Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
PageRange	May	MUST	MUST	MUST
page-range-lower	May	MUST	MUST	MUST
page-range-upper	May	MUST	MUST	MUST

Not-Set is required for all attribute implemented. "{-}" indicates values to be supported.

22.   SeparatorSheet

A SeparatorSheet object specifies information that describes separator sheets. A separator sheet will not be imaged and/or printed with content from the document data stream.

A SeparatorSheet object contains attributes that specify where the separator sheets are to be placed (before or after the job or document, between copies, etc.), the media that is to be used for the separator sheet, and a message that is to be imaged/printed on the separator sheet.

22.1 SeparatorSheet Attributes & Conformance

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
SeparatorSheet	May	May	May	**Must**
separator-sheet-media	May	May	May	**Must**
separator-sheet-message	May	May	May	**Must**
separator-sheet-type	**Must** {BEFORE_JOB}	**Must** {BEFORE_JOB}	**Must** {BEFORE_JOB}	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

## 23.	Stitching

A Stitching object specifies how a job, document, or range of pages in a job or document is to be stapled and/or stitched. A Stitching object contains attributes that specify the number of stitches, the type of stitch (for example, corner, edge, or saddle), the edge of the sheet to stitch, and other stitching information.

### 23.1	Stitching Attributes & Conformance

Attribute	E/M	D/H	O/N	Prod
Stitching	May	May	May	**Must**
stitching-angle	May {0, 45, 90, 135}	May {0, 45, 90, 135}	May {0, 45, 90, 135}	**Must** {0, 45, 90, 135}
stitching-count	May	May	May	**Must** {1, 2}
stitching-reference-edge	May	May	May	**Must** {LEFT, RIGHT}
stitching-type	**Must** {CORNER, NONE}	**Must** {CORNER, NONE}	**Must** {CORNER, NONE}	**Must**

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

24.   Subscription

A Subscription object specifies subscription for notification of events that are to be sent to a specific destination. A Subscription object contains attributes that specify the events to be sent, the character set, the language, the comment text to be included in the notification, delivery method (for example, email), and other subscription information.

24.1 Subscription Attributes & Conformance

Attribute	*E/M*	*D/H*	*O/N*	*Prod*
Subscription	May	May	May	**Must**
subscription-charset	**Must** {UTF-8}	**Must** {UTF-8}	**Must** {UTF-8}	**Must** {UTF-8}
subscription-comment	May	May	May	**Must**
subscription-events	**Must** {JOB_CONFIG_ CHANGED, JOB_STATE_CH ANGED}	**Must** {JOB_CONFIG_ CHANGED, JOB_STATE_CH ANGED}	**Must** {JOB_CONFIG_ CHANGED, JOB_STATE_CH ANGED}	**Must** {JOB_CONFIG_ CHANGED, JOB_STATE_CH ANGED}
subscription-job-attributes	May	May	May	May
subscription-natural-language	May	May	May	May
subscription-send-to-uri	**Must** {SMNP:}	**Must** {SMNP:}	**Must** {SMNP: , MailTool:}	**Must** {SMNP:, MailTool:}

Not-Set is required for all attribute implemented.   "{-}" indicates values to be supported.

APPENDIX: A - Normative References

[ISO639] multi-part International Standard, presently consisting of [ISO639-1] and [ISO639-2].

[ISO639-1] Codes for the Representation of Names of Languages -- Part 1: Alpha-2 Code, ISO/IEC 639-1, 2000.

[ISO639-2] Codes for the Representation of Names of Languages -- Part 2: Alpha-3 Code, ISO/IEC 639-2, 1998.

[ISO3166] multi-part International Standard, presently consisting of [ISO3166-1] and [ISO3166-2].

[ISO3166-1] Codes for the Representation of Names of Countries and their Subdivisions, Part 1: Country Codes, ISO/IEC 3166-1, 1997.

[ISO3166-2] Codes for the Representation of Names of Countries and their Subdivisions, Part 2: Country Subdivision Codes, ISO/IEC 3166-2, 1998.

[ISO10646] multi-part International Standard, presently consisting of [ISO10646-1] and [ISO10646-2].

[ISO10646-1] Information Technology - Universal Multiple-Octet Code Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, ISO/IEC 10646-1, September 2000.

[ISO10646-2] Information Technology - Universal Multiple-Octet Code Character Set (UCS) - Part 2: Supplemental Planes, ISO/IEC 10646-2, January 2001.

[RFC2119] Bradner. Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, March 1997.

[RFC2396] Berners-Lee, Fielding, Masinter. URI Generic Syntax, RFC 2396, August 1998.

[RFC2910] Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.

[RFC2911] R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, " Internet Printing Protocol/1.10: Model and Semantics", RFC 2911, September 2000.

[RFC2978] Freed, Postel. IANA Charset Registration Procedures, RFC 2978, October 2000.

[RFC3066] Alvestrand. Tags for the Identification of Languages, RFC 3066, January 2001.

APPENDIX: B - Informative References

[cip4] The International Cooperation for the Integration of Processes in Prepress, Press and Postpress (CIP4) located at http://www.cip4.org/

[jdf] The Job Definition Format (JDF), version 1.1, August 2002. Sett the Document tab at: http://www.cip4.org

[pwg] The Printer Working Group located at http://www.pwg.org/

[pwg-sm] Zehler, P., Hastings, T., and Albright, S., Printer Working Group (PWG): Semantic Model, work in progress at ftp://ftp.pwg.org/pub/Semantic-Model/

[IANA-CHAR] IANA Registry of Character Sets ftp://ftp.iana.org/assignments/charset-reg/...

[IANA-MIME] IANA Registry of MIME Media Types ftp://ftp.iana.org/assignments/media-types/...

[RFC2277] Alvestrand. IETF Policy on Character Sets and Languages, RFC 2277, January 1998.

[RFC2279] Yergeau. UTF-8, a Transformation Format of ISO 10646, RFC 2279, January 1998.

APPENDIX: C - Terminology and Acronyms

Conformance Terminology

In this document, the uppercase terms "MUST", "MUST NOT" and "MAY" are intended to be interpreted as described in [RFC2119]

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

Table 1   Conformance Statement Terms

Term	Meaning
MUST	Implementation support is required for conformance to this specification.
MAY	Implementation support is optional for conformance to this specification.

In this document, the term "support" is defined as:

Table 2   Conformance Support Terms

Term	Meaning
"support an operation"	An implementation MUST accept a syntactically correct instance of the operation (includes all required parameters) and MUST return one of the defined results for the invoked operation.
"support an object"	An implementation MUST accept a syntactically correct instance of the object (includes all required attributes), MUST locally instantiate the object and MUST return one of the defined results for the invoked operation.
"support an attribute"	An implementation MUST accept a syntactically correct instance of the attribute (includes a valid value) and MUST locally instantiate the attribute. Further, an implementation MUST accept at least one value (other than the empty value NOT_SET) defined for the given attribute.
Element	Object operation or attribute.

Other Terminology

Table 3   Miscellaneous Terms

Term	*Meaning*
Job Ticket	A set of one or more job processing activities (e.g. stapling, binding, number of copies, insert sheet), contained within a file or stream, serialized into some file format (for example, the XML instance textual encoding of a CIP4 JDF Job Ticket).
Job	A set of one or more processing activities, contained within a "Job Ticket", described by "Job Description" attributes, and processed according to "Job Processing" attributes.
Document	A single document to be processed, contained within a "Job", described by "Document Description" attributes, and processed according to "Document Processing" attributes.
Document Processing	
Page	A single logical page in a source document. Multiple source pages may be included in a single "Impression" on a "Side" of a "Sheet".
Sheet	A single physical piece of media.
Side	A single side (front or back) of "Sheet" of media.
Impression	A single image marked (by some means) on a "Side" of a "Sheet".

Acronyms

Table 4   Acronyms

Acronyms	*Meaning*	*Source*
CIP4	International **C**ooperation for the **I**ntegration of **P**rocesses in **P**repress, **P**ress and **P**ostpress	http://www.cip4.org/
JDF	**J**ob **D**efinition **F**ormat	Version 1.1, August 2002. See documentation tab at http://www.cip4.org/
PWG	**P**rinter **W**orking **G**roup	http://www.pwg.org/

Acronyms	Meaning	Source
FSG	Free Standards Group	http://www.freestandards.org
FSG/OP	Free Standards Group – Open Printing	http://www.openprinting.org/
JTAPI	**J**ob **T**icket **A**pplication **P**rogramming **I**nterface	

Editors

Glen Petrie [glen.petrie@eitc.epson.com] - EPSON

Authors

Claudia Alimpich [alimpich@us.ibm.com] - International Business Machines
Till Kamppeter [`till.kamppeter@gmx.net`] – MandrakeSoft
Ira McDonald [`imcdonald@sharplabs.com`] - High North
Glen Petrie [glen.petrie@eitc.epson.com] - EPSON

Contributors

Tom Hastings [`hastings@cp10.es.xerox.com`] - Xerox

Changes

Date	Affected Version	Author	Change
08.12.03	0.30	G. Petrie	Added comments from T. Hastings in the introduction and reference sections.
08.12.03	0.30	G. Petrie	Added comments from I. McDonald in the reference section. Added comments and changes based on discussion from 08.12.03 weekly job-ticket meeting. Many section affected.
08.15.03	0.40	G. Petrie	Added comments and changes based on discussion from 08.12.03 weekly job-ticket meeting. Many section affected.
08.19.03	0.50	G. Petrie	Updated minor change to headings.
08.26.03	0.50 Claudia	G. Petrie	Added Sections 4.0, 4.1 and 4.2 from Claudia's version of 0.50, dated 08.19.03
08.28.03	0.55	G. Petrie	Edits and changes based on 08.28.03 weekly job-ticket meeting for Section 4.0, 4.1, 4.2. The addition table/figure labels and a table/figure table-of-content
10/15/03	0.77	G. Petrie	Updates to attribute compliance table based on 10.14.03 meeting. The addition of attribute #define/string-names to C language header section.
11.04.03	0.81	C. Alimpich	Added attributes/values to compliance table for SeparatorSheet, Stitching, Subscription, and Trimming objects. Removed header file contents (section 5.4).
11.11.03	0.82	G. Petrie	Re-arranged section 4.2. Added discussion table at end of file.
08.09.04	0.84	G. Petrie	Major rewrite of format
09.22.04	0.91	G. Petrie	Removed range, units, type, related columns from attribute tables. Fixed return code for top level api's. Added / deleted object and/or attributes based on group discussion.
10.25.04	0.93	G. Petrie	Fixed typo and errors.
01.18.05	0.96	G. Petrie	Incorporated UML and C Header updates