



Job Ticket Application Programming Interface (JTAPI)

Version 0.93
October 25, 2004



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 2004, Free Standards Group

Copyright Notice

Copyright (c) 2004 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.

Linux is a trademark of Linus Torvalds.

UNIX is a registered trademark of the Open Group in the United States and other countries.

Table of Contents

1.	INTRODUCTION.....	5
2.	INDUSTRY STANDARD JOB TICKETS.....	6
3.	OVERVIEW.....	7
3.1	PRINT ENVIRONMENTS.....	7
3.2	HIGH LEVEL MODEL.....	7
3.3	LINKING WITH JTAPI.....	9
3.4	CONFORMANCE WITH JTAPI.....	9
3.5	INTERNATIONALIZATION.....	9
3.6	SECURITY.....	9
4.	APPLICATION PROGRAMMING INTERFACE SUMMARY.....	10
5.	TOP LEVEL APPLICATION PROGRAMMING INTERFACE.....	11
5.1	FSGJTOPENJTAPI.....	11
5.2	FSGJTFLUSHJTAPI.....	12
5.3	FSGJTCLOSEJTAPI.....	13
6.	FUNCTIONAL LEVEL APPLICATION PROGRAMMING INTERFACE.....	14
6.1	FSGJTDESTROY.....	14
6.2	FSGJTGET.....	15
6.3	FSGJTSET.....	16
6.4	FSGJTSETINTEGERATTRIBUTE.....	17
6.5	FSGJTSETOBJECTATTRIBUTE.....	18
6.6	FSGJTSETOBJECTATTRIBUTEList.....	19
6.7	FSGJTSETSTRINGATTRIBUTE.....	20
6.8	FSGJTSETSTRINGATTRIBUTEList.....	21
7.	ATTRIBUTE LEVEL APPLICATION PROGRAMMING INTERFACES.....	22
7.1	FSGJTNEWATTRIBUTE.....	22
7.2	FSGJTDESTORYATTRIBUTE.....	23
7.3	FSGJTADDVALUE.....	24
7.4	FSGJTGETNAME.....	25
7.5	FSGJTGETNEXTVALUE.....	26
7.6	FSGJTGETNUMVALUES.....	27
7.7	FSGJTGETVALUETYPE.....	28
7.8	FSGJTREPLACEVALUE.....	29
7.9	FSGJTRESETTOFIRSTVALUE.....	30
8.	OBJECT LEVEL APPLICATION PROGRAMMING INTERFACES CONFORMANCE SUMMARY....	44
9.	CONTACTINFO.....	45
9.1	CONTACTINFO ATTRIBUTES & CONFORMANCE.....	45
10.	DESTINATION.....	46
10.1	DESTINATION ATTRIBUTES & CONFORMANCE.....	46
11.	DOCUMENT.....	47
11.1	DOCUMENT ATTRIBUTES & CONFORMANCE.....	47

12.	FOLDING	48
12.1	FOLDING ATTRIBUTES & CONFORMANCE	48
13.	FORCEPAGE	49
13.1	FORCEPAGE ATTRIBUTES & CONFORMANCE	49
14.	HOLEMAKING	50
14.1	HOLEMAKING ATTRIBUTES & CONFORMANCE	50
15.	INSERTSHEET	51
15.1	INSERTSHEET ATTRIBUTES & CONFORMANCE	51
16.	JOB	52
16.1	JOB ATTRIBUTES & CONFORMANCE.....	52
17.	JOBDOCUMENTPAGE	54
17.1	JOBDOCUMENTPAGE ATTRIBUTES & CONFORMANCE.....	54
18.	JOBTICKETINFO	55
18.1	JOBTICKETINFO ATTRIBUTES & CONFORMANCE	55
19.	MEDIA	56
19.1	MEDIA ATTRIBUTES & CONFORMANCE.....	56
20.	SEPARATORSHEET	57
20.1	SEPARATORSHEET ATTRIBUTES & CONFORMANCE	57
21.	STITCHING	58
21.1	STITCHING ATTRIBUTES & CONFORMANCE.....	58
22.	SUBSCRIPTION	59
22.1	SUBSCRIPTION ATTRIBUTES & CONFORMANCE.....	59
APPENDIX: A - NORMATIVE REFERENCES		60
APPENDIX: B - INFORMATIVE REFERENCES		61
APPENDIX: C - TERMINOLOGY AND ACRONYMS		62
	CONFORMANCE TERMINOLOGY	62
	OTHER TERMINOLOGY.....	63
	ACRONYMS.....	63
CHANGES		66

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 not 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 ticket 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 hide 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. Industry 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 is due to be published the end of 2003. 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 intranets [pwg]. The PWG has published the Internet Printing Protocol (IPP) in September 2000 [rfc2910, rfc2911]

The PWG is in the process of publishing 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. Therefore, an XML Job Ticket using the semantics of the PWG Semantic Model is possible.

3. Overview

3.1 Print Environments

[Describe the Embedded Mobile, Desktop/Home, Office/Network, and Production environments.]

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.

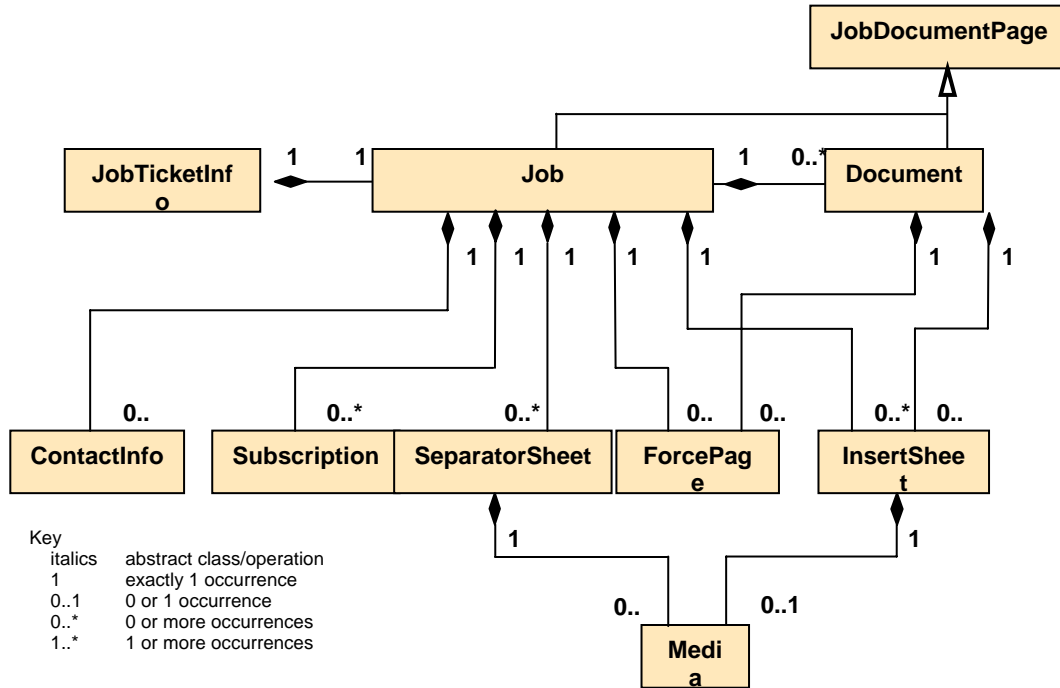


Figure 1.0 JTAPI High Level Object Model

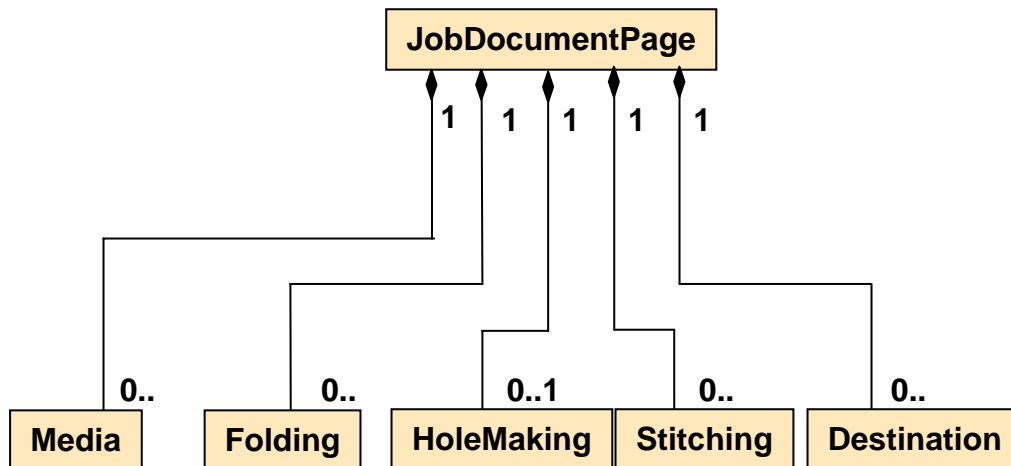


Figure 2.0 JTAPI JobDocumentPage High Level Object Model

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 8 of this specification;
- (2) MUST publish one or more sets of standard header files verbatim from Section 23, Section 24 or Section 25 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	
---	fsgjt.h	fsgjtOpenJTAPI	void**
---	fsgjt.h	fsgjtFlushJTAPI	void*
---	fsgjt.h	fsgjtCloseJTAPI	void*
all	fsgjt.h	fsgjtDestroy	Object
all	fsgjt.h	fsgjtGet	object, attribute
all	fsgjt.h	fsgjtSet	object, attribute
all	fsgjt.h	fsgjtSetIntegerAttribute	object, attribute, value
all	fsgjt.h	fsgjtSetObjectAttribute	object, attribute, value
all	fsgjt.h	fsgjtSetObjectAttributeList	object, attribute, list
all	fsgjt.h	fsgjtSetStringAttribute	object, attribute, value
all	fsgjt.h	fsgjtSetStringAttributeList	object, attribute, values
all	fsgjt_attribute.h	fsgjtNewAttribute	attribute, name, type, value
all	fsgjt_attribute.h	fsgjtDestoryAttribute	Attribute
all	fsgjt_attribute.h	fsgjtAddValue	attribute, value
all	fsgjt_attribute.h	fsgjtGetName	attribute, name
all	fsgjt_attribute.h	fsgjtGetNumValues	attribute, numValues
all	fsgjt_attribute.h	fsgjtGetValueType	attribute, type
all	fsgjt_attribute.h	fsgjtReplaceValue	attribute, value
all	fsgjt_attribute.h	fsgjtResetToFirstValue	Attribute
all	fsgjt_destination.h	fsgjtNewDestination	destinationObject, uri
document	fsgjt_document.h	fsgjtNewDocument	documentObject
document	fsgjt_document.h	fsgjtNewDocumentFromURI	documentObject, uri
stitching	fsgjt_stitching.h	fsgjtNewStitching	stitchingObject, stitchType
media	fsgjt_media.h	fsgjtNewMedia	mediaObject, name
job	fsgjt_job.h	fsgjtNewJob	jobObject
job	fsgjt_job.h	fsgjtNewJobFromDocument	jobObject, documentObj
job ticket info	fsgjt_job_ticket_info.h	fsgjtNewJobTicketInfo	jobTicketInfoObject, typeAndVersion
job ticket info	fsgjt_job_ticket_info.h	fsgJtNewJobTicketFromURI	jobTicketInfoObject, uri, typeAndVersion
job ticket info	fsgjt_job_ticket_info.h	fsgJtNewJobTicketFromBuffer	jobTicketInfoObject, buffer, bufferSize, typeAndVersion
job ticket info	fsgjt_job_ticket_info.h	fsgJtNewJobTicketFromJob	jobTicketInfoObject, job, typeAndVersion
job ticket info	fsgjt_job_ticket_info.h	fsgJtNewJobTicketToBuffer	jobTicketInfoObject, buffer, bufferSize
job ticket info	fsgjt_job_ticket_info.h	fsgJtNewJobTicketToURI	jobTicketInfoObject, uri

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(jtapi_struct);		
Parameters:	void **	jtapi_struct	An optional opaque structure that to be returned by the specific instantiation of the JTAPI
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_MISSING_PARAM		
Impact Level:	Library		
Description:	Initializes the JTAPI library and returns a structure that is utilized by the specific JTAPI library instantiation.		
Constraints, Limitation, Restrictions	This API call must be called before calling any other JTAPI calls.		
Also See:	fsgjtCloseJTAPI		

5.2 fsgjtFlushJTAPI

Syntax:	fsgjt_return_code_t fsgjtFlushJTAPI(jtapi_struct);		
Parameters:	void *	jtapi_struct	The opaque structure that was returned from the fsgjtOpenJTAPI call for this specific instantiation of the JTAPI
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 associate with this session.		
Constraints, Limitation, Restrictions			
Also See:	fsgjtCloseJTAPI		

5.3 fsgjtCloseJTAPI

Syntax:	fsgjt_return_code_t fsgjtCloseJTAPI(jtapi_struct);		
Parameters:	void *	jtapi_struct	The opaque structure that was returned from the fsgjtOpenJTAPI call for this specific instantiation of the JTAPI
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_REQUEST FSGJT_MISSING_PARAM		
Impact Level:	Library		
Description:	Releases the JTAPI library, removes any dynamic memory allocations and closes any I/O channels. This call will execute an fsgjtFlushJTAPI call.		
Constraints, Limitation, Restrictions	This API call must be called after calling any other JTAPI calls.		
Also See:	fsgjtOpenJTAPI		

6. Functional Level Application Programming Interface

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

6.1 fsgjtDestroy

Syntax:	fsgjt_return_code_t fsgjtDestroy(object);		
Parameters:	fsgjt_object_t	Object	Pointer to the 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	"fsgjtDestory MUST be called to free each object allocation during session, otherwise memory leaks will occur. Before calling fsgitCloseJTAPI, fsgjtDestory MUST be called to free each object allocation during session. Destroying higher level objects will automatically destroy lower level objects.		
Also See:	fsgjtFlushJTAPI		

6.2 fsgjtGet

Syntax:	fsgjt_return_code_t fsgjtGet(object, attributeName, attribute);		
Parameters:	fsgjt_object_t	object	Pointer to the object from which to get the attribute.
	char *	attributeName	Name of the attribute to get.
	fsgjt_attribute_t	attribute	Attribute having provided name. (returned)
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:			

6.3 fsgjtSet

Syntax:	fsgjt_return_code_t fsgjtSet(object, attribute);		
Parameters:	fsgjt_object_t	object	Pointer to the 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:			

6.4 fsgjtSetIntegerAttribute

Syntax:	fsgjt_return_code_t fsgjtSet(object, attribute, value);		
Parameters:	fsgjt_object_t	object	Set the attribute for this object.
	char*	attribute	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 attribute to the provided value. This function creates a fsgjt_attribute_t, calls the set() function and then destroys the attribute.		
Constraints, Limitation, Restrictions			
Also See:			

6.5 fsgjtSetObjectAttribute

Syntax:	fsgjt_return_code_t fsgjtSetObjectAttribute(object, attribute, value);		
Parameters:	fsgjt_object_t	object	Set the attribute of this object.
	char *	attribute	Name of the 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 set() function and then destroys the attribute.		
Constraints, Limitation, Restrictions			
Also See:			

6.6 fsgjtSetObjectAttributeList

Syntax:	fsgjt_return_code_t fsgjtObjectAttributeList(object, attribute, list);		
Parameters:	fsgjt_object_t	object	Set the attribute for this object
	char*	attribute	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 set() function and then destroys the attribute.		
Constraints, Limitation, Restrictions			
Also See:			

6.7 fsgjtSetStringAttribute

Syntax:	fsgjt_return_code_t fsgjtSetStringAttribute(object, attribute, value);		
Parameters:	fsgjt_object_t	object	Set the attribute of this object
	char*	attribute	Name of attribute to set
	char*	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 set() function and then destroys the attribute.		
Constraints, Limitation, Restrictions			
Also See:			

6.8 fsgjtSetStringAttributeList

Syntax:	fsgjt_return_code_t fsgjtSetStringAttributeList(object, attribute, list);		
Parameters:	fsgjt_object_t	object	Set the attribute for this object
	char*	attribute	Name of attribute to set
	char**	list	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 string attribute list to the provided values. This function creates a fsgjt_attribute_t, calls the set() function and then destroys the attribute.		
Constraints, Limitation, Restrictions			
Also See:			

7. Attribute Level Application Programming Interfaces

Attribute level JTAPI calls 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 to return
	char*	name	Name for the new attribute to create
	char*	type	Type for the new attribute
	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 object attribute having the provided attribute name, value type, and value.		
Constraints, Limitation, Restrictions	Multiple values should be added with the add() function.		
Also See:			

7.2 fsgjtDestoryAttribute

Syntax:	fsgjt_return_code_t fsgjtDestory(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:			

7.3 fsgjtAddValue

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

7.4 fsgjtGetName

Syntax:	fsgjt_return_code_t fsgjtGetName(attribute, name);		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose name to get.
	char**	name	Name of the attribute to return
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:			

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.
	char**	nextValue	Pointer to the next value.
	fsgjt_present_t*	moreValues	“On” if there are more values 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:			

7.6 fsgjtGetNumValues

Syntax:	fsgjt_return_code_t fsgjtNumValues(attribute, numValues);		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose number of values to get.
	fsgjt_int32_t*	numValues	Number of values to return
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:			

7.7 fsgjtGetValueType

Syntax:	fsgjt_return_code_t fsgjtGetValueType(attribute, type);		
Parameters:	fsgjt_attribute_t	attribute	Attribute whose value type to get.
	char**	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:			

7.8 fsgjtReplaceValue

Syntax:	fsgjt_return_code_t fsgjtReplaceValue (attribute, value);		
Parameters:	fsgjt_attribute_t	attribute	Attribute to replace the provided value(s)
	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(s) for the attribute with the provided value.		
Constraints, Limitation, Restrictions			
Also See:			

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 iterators to point to the first value.		
Constraints, Limitation, Restrictions			
Also See:			

7.10 fsgjtNewDestination

Syntax:	fsgjt_return_code_t fsgjtNewDestination(destination, uri);		
Parameters:	fsgjt_object_t*	destination	Pointer to the new Destination object to create.
	fsgjt_uri_t	uri	UIR 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 provide target location URI.		
Constraints, Limitation, Restrictions			
Also See:			

7.11 fsgjtNewDocument

Syntax:	fsgjt_return_code_t fsgjtNewDocument(document);		
Parameters:	fsgjt_object_t*	Document	Pointer to the new document to create.
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:			

7.12 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:			

7.13 fsgjtNewJob

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

7.14 fsgjtNewJobFromDocument

Syntax:	fsgjt_return_code_t fsgjtNewJobFromDocument(job, document);		
Parameters:	fsgjt_object_t*	job	Pointer to job 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 containing the provided Document.		
Constraints, Limitation, Restrictions			
Also See:			

7.15 fsgjtNewJobTicketInfo

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfo(jobTicketInfo, typeAndVersion);		
Parameters:	fsgjt_object_t*	jobTicketInfo	Pointer to the new JobTicketInfo to create
	char*	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 job ticket is being created without a Job object and not from an existing job ticket.		
Constraints, Limitation, Restrictions	Use JobTicketTypeVersionEnum		
Also See:			

7.16 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
	char*	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 UIR. Other JTAPI objects are also created based on the information in the job ticket.		
Constraints, Limitation, Restrictions	Use JobTicketTypeVersionEnum		
Also See:			

7.17 fsgjtNewJobTicketInfoFromBuffer

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketFromBuffer (jobTicketInfo, buffer, buffersize, typeAndVersion);		
Parameters:	fsgjt_object_t*	jobTicketInfo	Pointer to the new JobTicketInfo to create
	char**	buffer	Buffer that contains the job ticket
	size_t	bufferSize	Size of the buffer.
	char*	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.		
Constraints, Limitation, Restrictions	Use JobTicketTypeVersionEnum		
Also See:			

7.18 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
	char*	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:			

7.19 fsgjtWriteJobTicketInfoToBuffer

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfoToBuffer (jobTicketInfo, buffer, buffersize);		
Parameters:	fsgjt_object_t	jobTicketInfo	JobTicketInfo from which the job ticket is to be written
	char**	buffer	Pointer to buffer to write the job ticket to. The buffer must be created by the caller.
	size_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:			

7.20 fsgjtWriteJobTicketInfoToURI

Syntax:	fsgjt_return_code_t fsgjtNewJobTicketInfoToURI(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:			

7.21 fsgjtNewMedia

Syntax:	fsgjt_return_code_t fsgjtNewMedia (media, name);		
Parameters:	fsgjt_object_t*	Media	Pointer to the new Media to create
	char*	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:			

7.22 fsgjtNewStitching

Syntax:	fsgjt_return_code_t fsgjtNewStitching(stitching, stitchType);		
Parameters:	fsgjt_object_t*	stitching	Pointer to the new Stitching object to create.
	fsgjt_stitch_type_t	stitchType	Position where the stitching is to occur
Return Status:	FSGJT_OK FSGJT_INTERNAL_ERROR FSGJT_INVALID_PARAM FSGJT_MISSING_PARAM		
Impact Level:	Object		
Description:	Creates a Stitching object having the provided stitch type.		
Constraints, Limitation, Restrictions			
Also See:			

8. 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 in Table 5.0. 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

<i>Object</i>	<i>Conformance – 1.0 Release (Must, May)</i>			
	<i>Embedded/Mobile</i>	<i>Desktop/Home</i>	<i>Office/Network</i>	<i>Production</i>
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
SeparatorSheet	May	May	May	Must
Stitching	May	May	May	Must
Subscription	May	May	May	Must

9. 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.

9.1 ContactInfo Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
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>
contact-info-detail	May	May	May	May
contact-info-name	Must	Must	Must	Must
contact-info-types	Must {SENDER}	Must {SENDER}	Must {SENDER}	Must {SENDER, CUSTOMER, DELIVERY, PICKUP, ACCOUNTING }

Not-Set is required for all attribute implemented. "{-}" indicates value the must be supported.

10. Destination

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

10.1 Destination Attributes & Conformance

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

Not-Set is required for all attribute implemented. “{-}” indicates value the must be supported

11. 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.

11.1 Document Attributes & Conformance

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

Not-Set is required for all attribute implemented. “{-}” indicates value the must be supported.

12. 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.

12.1 Folding Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
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 value the must be supported.

13. 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.

13.1 ForcePage Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
ForcePage	May	May	May	Must

Not-Set is required for all attribute implemented. "{-}" indicates value the must be supported.

14. 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.

14.1 HoleMaking Attributes & Conformance

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

Not-Set is required for all attribute implemented. “{-}” indicates value the must be supported.

15. 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.

15.1 InsertSheet Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
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 value the must be supported.

16. 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).

16.1 Job Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
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-uri	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
job-image-shift-back-y	May	May	May	Must

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
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-top	Must	Must	Must	Must
job-margin-right	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-rotation	May	Must	Must	Must
job-presentation-direction	May	May	May	Must
job-priority	Must	Must	Must	Must
job-print-quality	Must	Must	Must	Must
job-resolution-x	Must	Must	Must	Must
job-resolution-y	Must	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
job-trimming	May	May	May	Must

Not-Set is required for all attribute implemented. “{-}” indicates value the must be supported.

17. 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.

17.1 JobDocumentPage Attributes & Conformance

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

Not-Set is required for all attribute implemented. “{-}” indicates value the must be supported.

18. 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.

18.1 JobTicketInfo Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
<i>JobTicketInfo</i>	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-job	Must	Must	Must	Must
jt-mandatory-attributes	Must	Must	Must	Must
jt-type-and-version	Must	Must	Must	Must
jt-length-units	Must {MICRO}	Must {MICRO}	Must {MICRO}	Must {MICRO}
jt-version	May	May	May	Must

Not-Set is required for all attribute implemented. “{-}” indicates value the must be supported.

19. 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.

19.1 Media Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
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	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 value the must be supported.

20. 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.

20.1 SeparatorSheet Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
<i>SeparatorSheet</i>	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 value the must be supported.

21. 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.

21.1 Stitching Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
<i>Stitching</i>	May	May	May	Must
stitching-angle	May {tbd}	May {tbd}	May {tbd}	Must {tbd}
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 value the must be supported.

22. 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.

22.1 Subscription Attributes & Conformance

<i>Attribute</i>	<i>E/M</i>	<i>D/H</i>	<i>O/N</i>	<i>Prod</i>
<i>Subscription</i>	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 {tbd}	Must {tbd}	Must {tbd}	Must {tbd}
subscription-job-attributes	May	May	May	May
subscription-natural-language	May	May	May	May
subscription-send-to-uri	Must {tbd}	Must {tbd}	Must {tbd}	Must {tbd}

Not-Set is required for all attribute implemented. “{-}” indicates value the must 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. Set 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/...](ftp://ftp.iana.org/assignments/charset-reg/)

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

<i>Term</i>	<i>Meaning</i>
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

<i>Term</i>	<i>Meaning</i>
“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

<i>Term</i>	<i>Meaning</i>
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

<i>Acronyms</i>	<i>Meaning</i>	<i>Source</i>
CIP4	International Cooperation for the Integration of Processes in Prepress, Press and Postpress	http://www.cip4.org/
JDF	Job Definition Format	Version 1.1, August 2002. See documentation tab at http://www.cip4.org/
PWG	Printer Working Group	http://www.pwg.org/

<i>Acronyms</i>	<i>Meaning</i>	<i>Source</i>
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 Kampeter [till.kampeter@gmx.net] - MandrakeSoft

Ira McDonald [imcdonald@sharp-labs.com] - High North

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

Contributors

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

Changes

<i>Date</i>	<i>Affected Version</i>	<i>Author</i>	<i>Change</i>
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.