Information Objects/Attributes for the Job Monitoring MIB/MIF

From: Tom Hastings
Date: 12/30/96
Version: 0.5
File: ftp://ftp.pwg.org/pub/pwg/snmpmib/jobs-mib/jmp-spec.doc .ps

Status: Fourth draft of proposed specification for each of the 44 objects/attributes for the Job Monitoring MIB/MIF that the JMP has listed at its 11/08/96 meeting, plus a few suggestions made at the IETF meeting itself. See jmp-list.doc.

The actual specifications of each object needs line-by-line review. We did *not* have time for such review at the 11/08/96 meeting as indicated in the minutes.

The greatly simplified specifications of each object is derived from the ISO DPA attribute specifications in most cases. I've move the full ISO DPA specifications to an Appendix and indicated the corresponding ISO DPA attribute-name in the DPA column in the body of this specification. . Revision marks show the agreements reached at the November meeting where we were able to finish the entire document. I've indicated ISSUES in the text that we have identified as issued but have not resolved. These issues are also listed at the end of the Table of Contents with the page number of the issue. I've also copied in map-summ.doc into this document and moved it to an appendix so we can more easily compare the Job Monitoring objects with the job submission protocols and keep the object names updated in that summary.

In moving the descriptions to becoming "resources needed/used", I've copied them in the order of occurrence in the document to a new table at the end of the job status/accounting section as the descriptions for each resource enum. I've not used revision marks for such moves, but only for changes within each description of what had been an object and what now is an enum. I may have "stretched" the meaning of a resource a bit. For example, I've even made the documents within a job be a resource, so that an implementer could enumerate the names of each of the documents if he/she wants to, since we have an enum for such a "resource", but an implementation is not required to do so, since it is an enum, not an object.

I've added MIB-like names, using the MIB names that Scott included in his table structure memo (jm961005.doc). In the companion jmp-list.doc, I also added a **conformance** column so we can track what is mandatory and what is conditionally mandatory and a **Cardinality** column, so that we can see which objects occur once per job and which have multiples occurrences per job.

Finally, I took Ron's short descriptions and put them first in the description column.

1. STATEMENT OF CHARTER 5

2. GOALS 5

2.1 Schedule: 7

3. TERMINOLOGY (NEEDS MORE WORK) 7

4. SYSTEM CONFIGURATIONS FOR THE JOB MONITORING MIB 11

4.1 Configuration 1 - client-printer 11

4.2 Configuration 2 - client-server-printer - printer doesn't have an agent 12

5. OBJECT/ATTRIBUTE GROUPS AND TABLES 15

6. LEGEND USED IN THE JOB MONITORING MIB SPECIFICATION TABLES 15

6.1 The MIB/MIF Datatype column explanation 16

7. OBJECT/ATTRIBUTE SPECIFICATION 18

7.1 The MIB Instance Group 18

7.2 The General Group 19

7.3 The Queue Group 21

7.4 The Completed Group 24

7.5 The Job Group 25

7.5.1 Job Identification (I) object/attribute specification 25

7.5.2 Job Parameters (P) object/attribute specification 30

7.5.3 Job Status and Accounting (S) 31

7.6 The Resource Group 46

8. APPENDIX A - COMPARISON WITH ISO DPA 51

8.1 The MIB Instance Group - comparison with ISO DPA 51

8.2 The General Group - comparison with ISO DPA 51

8.3 The Queue Group - comparison with ISO DPA 52

8.4 The Completed Group - comparison with ISO DPA 54

8.5 The Job Group - comparison with ISO DPA 55

8.6 The Resource Group - comparison with ISO DPA 63

9. APPENDIX B - MAPPING FROM JOB SUBMISSION PROTOCOLS TO JMP OBJECTS/ATTRIBUTES 69

66 TABLE OF ISSUES

67 Issue 1 - Are the definitions of job and document ok? 7
68 Issue 2 - Change terms from printer to device? 7
69 Issue 3 - How are T(63) strings localized? 17
70 Issue 4 - How is code conversion done for strings, if at all? 17
71 Issue 3 - How are T(63) strings localized? 17
72 Issue 4 - How is code conversion done for strings, if at all? 17
73 Issue 5 - Is the name jmMIBInstanceIndex ok? 18
74 Issue 6 - Do we really need the jmJobDownstreamId/jmJobPrinterId? 26
75 Issue 7 - Should jmJobTypes be something that a job submitter can specify or not? 27
76 Issue 8 - Is the parent job type the or of the child jobs? 27
77 Issue 9 - What is the difference between jmJobName and jmClientId? 29
78 Issue 10 - Combine jmJobTotalOctetsHigh and Low and count by K? 30
79 Issue 11 - Make jmJobStateReasons a bit string? 35
80 Issue 12 - Combine jmJobOctetsCompletedHigh and Low and count by K? 45
81 Issue 13 - Add jmJobTotalPagesSpooled, jmJobTotalPagesSentToDevice, and jmJobTotalPagesPrinted? 45
82 Issue 14 - Always have separate resource types for requested versus used? 46
83 Issue 15 - Ok to use the string names, instead of enums for the printer interpreter language family? 48

1. Statement of Charter

The Job Monitoring Project (JMP) shall develop a set of objects for monitoring of the status and progress of print jobs and to obtain resource accounting data at the completion of a job. The object set will initially be developed independent from the management system format. The object set development shall not preclude use of the set with either an SNMP MIB, a DMI MIF, or the SENSE protocol.

Use of the object set shall not be limited to printers. To obtain complete information regarding a print job in many printing systems will require the object set to be implemented into a file server, spooler, and/or a printer management system software.

When the object set is finalized, an SNMP MIB will be developed. The object set may also be ported to other formats, such as SENSE and a DMI MIF, as an extension to this project.

2. Goals

The following goals are copied from the charter.doc approved by JMP on 6/19/96.

1. The major goals of this MIB are to satisfy the needs of an agent in the printer and secondarily an agent in the server. Implementations should place the agent as close to the processing of the print job as possible. This MIB applies to printers that spool as well as those that don't. This MIB is all necessary so that servers are able find out about jobs in a printer that may have been submitted by other servers using other job submission protocols. In most environments that support high function job submission/job control protocols, like ISO DPA, those protocols would be used to monitor and manage print jobs rather than using the Job Monitoring MIB.

   The following specific text was agreed to at the October 1996 JMP meeting:

   "The job monitoring MIB is for agents in the printer (spooling or non-spooling) or the first server closest to the printer where the printer is either:

   a. directly connected to the server only

   --or--

   b. the printer does *not* contain the job monitoring MIB agent."

2. The job MIB is intended to provide the following information for the indicated Role Models (see Appendix D - Roles of Users in the Printer MIB draft update to RFC 1759). A limited set of *mandatory* job and document attributes for a printer, plus a set of *conditionally mandatory* attributes, will be developed to provide this information.

User		
	(U1)	A timely notification that his job has completed and where.
	(U2)	The current status of the user's job (user queries).
	(U3)	Error and diagnostic information for jobs that did not successfully complete.
	(U4)	Ability to identify the least busy printer.
Operator		
	(OP1)	A presentation of the state of all the jobs in the print system.
	(OP2)	Which users submitted each job.
	(OP3)	What resources does each job need.

	(OP4)	For which physical printers are the jobs candidates.
	(OP5)	Some idea of how long each job will take.
Capacity Planner		
	(C1)	How busy are printers.
	(C2)	What time of day are they used.
	(C3)	How long do users' jobs wait before starting to print.
Accountant		
	(A1)	A record of resources used and printer usage data for charging users or groups for resources used.

The MIB will provide for printers that can contain more than one job at a time, but still be usable for low end printers that only contain a single job at a time. In particular, the MIB shall meet the needs of Windows and other PC environments for managing low-end networked devices without unnecessary overhead or complexity, while also providing for higher end systems and devices.

2. The job MIB may also be transformed into a companion DMTF MIF. The job MIB/MIF should meet the needs for managing devices using the DMTF DMI version 2, when the devices are directly connected to PCs.

3. The MIB will provide job resource accounting information after the printer has finished printing the job. This resource accounting information is intended to be used by:

    - A management station that is co-located with the printer to provide an enhanced console capability.
    - End user job monitoring programs that provide status on progress and completion of jobs during the complete life cycle of the job, including a defined period after the job completes.
    - System accounting programs that copy the completed job statistics to an accounting system. It is recognized that depending on accounting programs to copy MIB data during the job-retention period is somewhat unreliable, since the accounting program may not be running (or may have crashed).

4. System usage statistics gathering programs that copy the completed job statistics to system usage logs.

5. The MIB will provide a compatible subset of job and document attributes of the ISO DPA standard, so that coherence is maintained between the two protocols and information presented to end users and system operators. However, the printer job monitoring MIB is intended to be used with printers that implement other job submitting and management protocols, such as IEEE 1284.1 (TIPSI), as well as with ones that do implement ISO DPA. So nothing in the printer job monitoring MIB shall require implementation of the ISO DPA protocol.

6. The MIB will be designed so that an additional MIB(s) can be specified in the future for monitoring multi-function (scan, FAX, copy) jobs. The job MIB will be designed such that a future multi-function job monitoring MIB will be able to use entire groups from this MIB.

7. The MIB will not address any security issues. Security provisions will be limited only to those provided by SNMP, in current or future versions.

8. The MIB will provide for the monitoring of jobs that clients submit directly to devices, to supervisor control programs, or to spooling systems.

9. The MIB will provide SNMP MIB access to jobs submitted to the device by any protocol, including devices that accept jobs using multiple protocols.

2.1 Schedule:

Submit a request for an IETF charter for this project at the December IETF meeting. The request will not be submitted until a final draft of the Job Monitoring object set has been completed.

3. Terminology (needs more work)

The following terms are used in this specification[1]:

A *job* is a unit of work whose results are expected together without interjection of unrelated results. A *client* is able to specify job instructions that apply to the job as a whole. Proscriptive instructions specify how, when, and where the job is to be printed. Descriptive instructions describe the job. A job contains one or more *documents*.

A *document* is a sub-section within a job. A document contains print data and document instructions that apply to just the document. The *client* is able to specify document instructions separately for each document in a job. Proscriptive instructions specify how the document is to be processed and printed by the *server*. Descriptive instructions describe the document. Server implementation of more than one document per job is optional.

Issue 1 - Are the definitions of job and document ok?

I've had action items to improve the definitions of job and document. Lets review what is above and see what the problems are. See footnote as well.

A *client* is the network entity that End Users use to submit jobs to *spoolers*, *servers*, or *printers*, depending on the configuration, using any job submission protocol. The **client** may or may not also use SNMP and the Job Monitoring MIB to monitor jobs, depending on implementation.

A *server* is a network entity that accepts jobs from clients. A server may be a *printer*, a printer *supervisor* control program, or a print *spooler*.

A *printer* is a device that puts marks on media.

Issue 2 - Change terms from printer to device?

Should we change term from printer to device throughout, so that the MIB can be used with other devices, such as scanners, and fax machines. Then we would be meeting our goal number 6. The object names are already changed, but not all descriptions.

A *supervisor* is a server that contains a control program that controls a printer. A supervisor is a client to the printer.

A *spooler* is a server that accepts jobs, spools the data, and decides when and on which printer to print the job. A spooler is a client to a printer or a printer supervisor, depending on implementation.

Spooling is the act of a *printer* or *server* of (1) accepting jobs, (2) writing the job's attributes and document data on to secondary storage and (3) ordering (*queuing*) the jobs for the purpose of scheduling the jobs to be processed.

---

[1] Existing systems use conflicting terms, so these terms are drawn from the ISO 10175 Document Printing Application (DPA) standard. For example, PostScript systems use the term *session* for what we call a job in this paper and the term *job* to mean what we call a document in this paper. PJL systems use the term ...

Queuing is the act of a *printer* or *server* of (1) accepting jobs and (2) ordering the jobs for scheduling. The job's attributes and document data are kept elsewhere. Thus *spooling* implies *queuing*, but *queuing* does not imply *spooling*. In other words, *queuing* is a sub-set of *spooling*.

A *monitor* is the network entity that End Users, System Operators, Accountants, Asset Managers, and Capacity Planners use to monitor jobs using SNMP. The **monitor** logically separate from the **client**, though a single network entity could implement both a client and a monitor.

An *agent* is the network entity that accepts SNMP requests from a *network management station* (NMS) and implements the Job Monitoring MIB.

A *network management station (NMS)* is a network entity that submits SNMP operations to agents.

A *proxy* is an agent that acts as a concentrator one or more other agents by accepting SNMP operations on the behalf of one or more other agents, forwarding them on to those other agents, gathering responses from those other agents and returning them to the original requesting monitor.

A *user* is a person that uses a client or an NMS.

An *end user* is a user that uses a client to submit a print job.

A *system operator* is a user that uses an NMS to monitor the system and carries out tasks to keep the system running.

A *system administr*ator is a user that specifies policy for the system.

An *attribute* is a name, value-pair that specifies an action, a status, or a condition in a job or a document object and in a DMTF MIF.

An *SNMP information object* is a name, value-pair that specifies an action, a status, or a condition in an SNMP MIB.

Job monitoring using SNMP is (1) identifying jobs within the serial streams of data being processed by the printer, (2) creating "rows" in the job table for each job, and (3) recording information, known by the agent, about the processing of the job in that "row".

Job accounting is recording what happens to the job during the processing and printing of the job.

There are two approaches that implementers may use to address the problems of the end-user using the Job Monitoring MIB:

1. The **client** also supports SNMP and the Job Monitoring MIB for status/notification to the submitting user

2. The **monitor** supports SNMP and the Job Monitoring MIB for status/notification to *any* user, including the job-submitting end user; for example, the Windows Print Manager.

The following diagram illustrates the relationships between the defined entities.

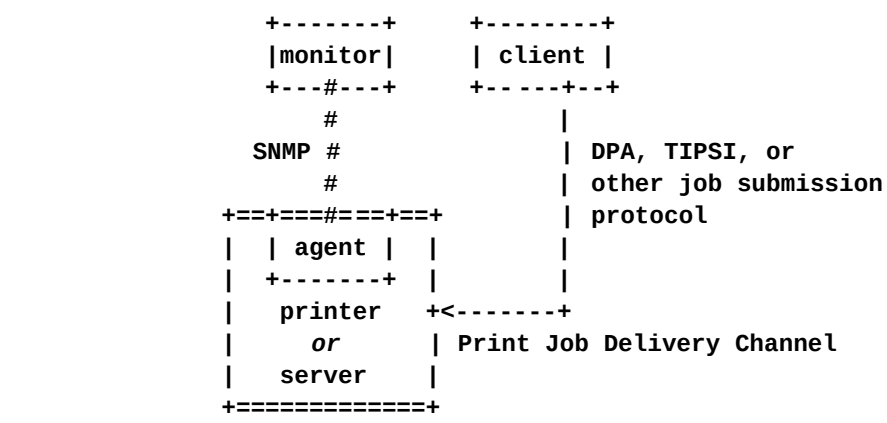

Figure 1 - Relationship between client, printer/server, management station, and agent

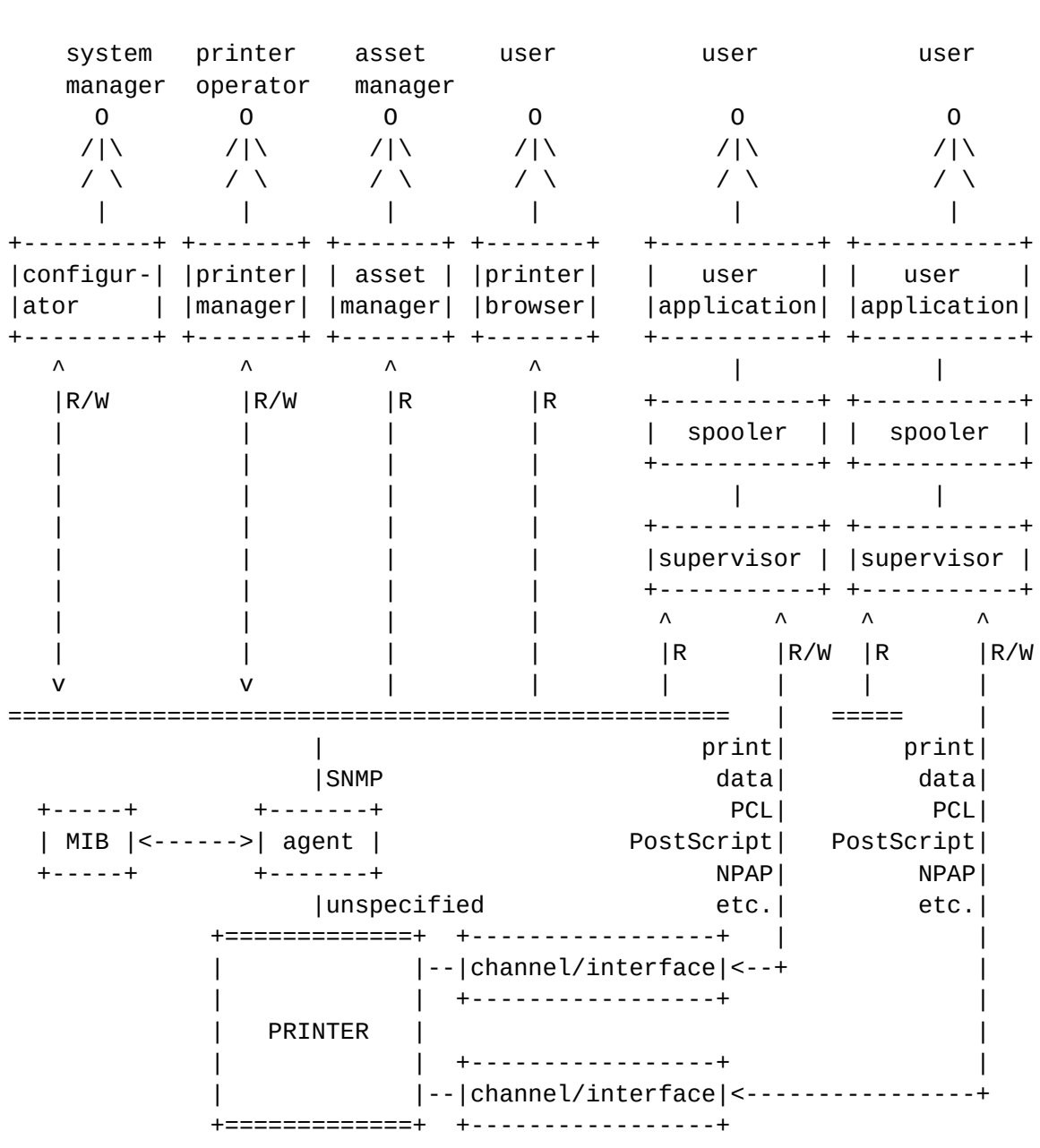

Figure 2 - One Printer's View of the Network (extracted from RFC 1759)

4. System Configurations for the Job Monitoring MIB

This section enumerates the two configurations for which the Job Monitoring MIB is intended to be used. See Goals section.

4.1 Configuration 1 - client-printer

In the **client-printer** configuration, the **client**(s) submit jobs directly to the printer, either by some direct connect, or by network connection. The **client-printer** configuration can accommodate multiple job submitting **clients** in either of two ways:

1. if each **client** relinquishes control of the Print Job Delivery Channel after each job (or after a number of jobs)
2. if the printer supports more than one Print Job Delivery Channel

The job submitting **client** and/or **monitor** communicates directly with an agent that is either part of the printer or is a front for the **printer** using some printer-specific management protocol between the agent and the **printer** (for example the management commands in the TIPSI protocol). Since the means of communication between an agent implementation and the **printer** is of no concern to us, the agent is shown as part of the **printer** in all of the configurations.

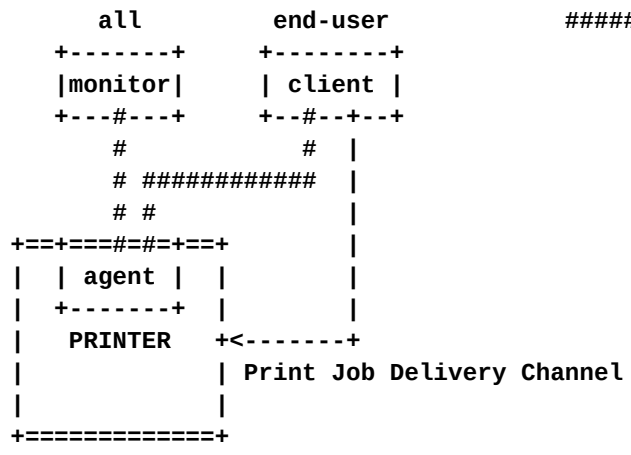

Figure 3 - Configuration 1 - client-printer

4.2 Configuration 2 - client-server-printer - printer doesn't have an agent

In the **client-server-printer** configuration 2, the **client**(s) submit jobs to an intermediate **server** by some network connection, *not* directly to the **printer**.

The job submitting **client** and/or **monitor** communicates directly with:

1. an agent that is part of the **server** (or a front for the server)

There is no SNMP agent in the printer in configuration 2.

Configuration 2 and Configuration 1 are the configurations that are the primary goal of the Job Monitoring MIB. The additional configurations are shown for completeness. However, the complexity of developing a Job Monitoring MIB that will work with the various agents in the various boxes where agents are shown, is overwhelming.

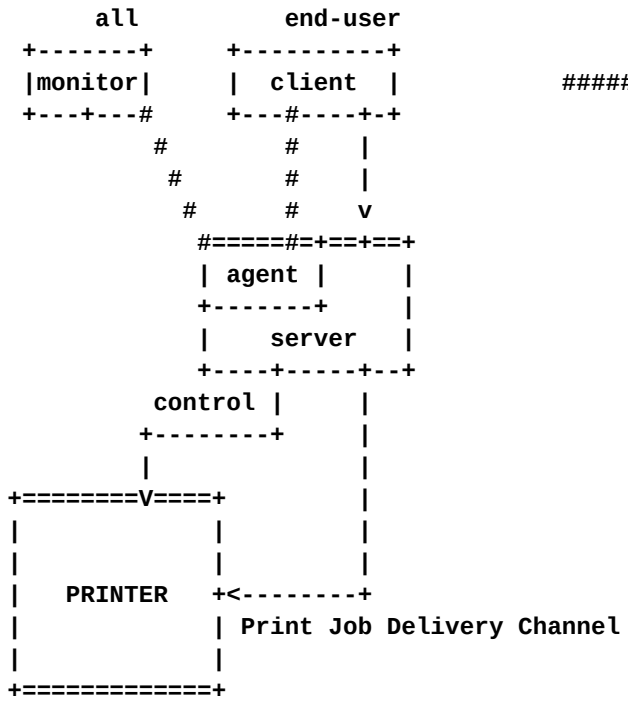

Figure 4 - Configuration 2 - client-server-printer - printer doesn't have an agent

5. Object/Attribute Groups and Tables

There is a one to one relationship between tables and groups as follows:

Group	Table	Description	No. of objects	Conformance
jmMIBInstanceGroup	**jmMIBInstanceTable**	A table of indexes to each Job Monitoring MIB instance.	1	Mandatory
jmGeneralGroup	N/A	General attributes that apply to all jobs in the MIB instance.	56	Mandatory
jmQueueGroup	**jmQueueTable**	Ordered list of jobs that have *not* finished and job attributes that only matter until the job has finished processing. Mandatory only if queuing (or spooling).	7	Conditionally mandatory
jmCompletedGroup	**jmCompletedTable**	Ordered list of jobs that have finished processing.	3	Mandatory
jmJobGroup	**jmJobTable**	Per job objects.	20	Mandatory
jmResourceGroup	**jmResourceTable**	Resources requested and/or used by the job. Can have more than one per job.	7	Mandatory
		Mandatory Totals:	36	
		Conditionally Mandatory Totals:	7	
		Totals:	43	

6. Legend used in the Job Monitoring MIB Specification Tables

The following legend is used in the following tables:

Job Identification, Job Parameters, Status/Accounting	The first column is the MIB name followed by a descriptive name for the object/attribute that is applicable to both MIB and MIF. Names for the MIB have a prefix of "**jm**" and mixed case with each word starting with an upper case letter and no intervening spaces or hyphens. For the MIF the descriptive name will have intervening spaces and *no* hyphens. We will keep the names in this section the same as the summary file (jm-list.doc).
Data type	The data type of the object/attribute. Enums are given distinct names that start with a capital letter. See below.
Specification	The specification of the MIB/MIF object/attribute.

DPA	The page number of the attribute in ISO DPA, if there is one, and the name of that ISO DPA attribute. See the Appendix for the ISO DPA specification of that corresponding attribute. If the entry if empty, this object/attribute has no ISO DPA counterpart.

6.1 The MIB/MIF Datatype column explanation

The Datatype column shows the data type for the MIB/MIF object/attribute.

A(63)	**DisplayString** (ASCII) 63 octets maximum
I(63)	**InternationalDisplayString** of 63 octets maximum, localized as specified by the genCurrentLocalizationIndex. See General MIB (file: gen.doc, .ps, .txt) and the General textual conventions (file: gentc.doc, .ps, .txt).
T(63)	A single Text (maximum of 63 octets) attribute specified as **CodeIndexedStringIndex**, e.g., in one of the coded character sets that the server supports. See General MIB (file: /pub/pwg/snmpmib/jobs-mib/gen.doc, .ps, .txt) and General MIB textual-conventions (file: gentc.doc, .ps, .txt). The server may store the attributes in the coded character set supplied by the client when the job was submitted, or the server may convert the coded character set to one of the other coded character sets that the server supports. NOTE 1 -Code conversion from the character repertoire of ISO Latin 1 to Unicode representation is easy (add a leading zero octet). However, code conversion from Unicode to ASCII or Latin1 when there are characters outside the repertoire of the destination coded character set is harder. Alternatives: 1. Return the characters that the server cannot represent in the requested coded character set using a closely related character, such as the unaccented Latin letter, or some error condition such as * or ? for those characters that don't have obvious closely related characters. 2. Don't return anything; make the client request one of the other coded character representations (by writing a different value into the genCurrentCodedCharSet object). 3. Convert Unicode into the two character mnemonic representation contained in RFC 1345 which has a two character ASCII representation for all characters of Unicode Do the same for conversion of ISO Latin1 into ASCII. For example, RFC 1345 represents LATIN SMALL LETTER A WITH ACUTE as a' and MICRO SIGN as My. 4. RFC 1345 was designed for use by software and implementors and, therefore, avoids the use of so-called national-use characters (ACCENT GRAVE (`), CIRCUMFLEX ACCENT (^), TILDE (~). This fourth alternative is to use better two-character approximations than those in RFC 1345 that would be recognized by end-users without special training and would use the obvious national-use characters I such approximations.

	Issue 3 - How are T(63) strings localized? ***How are strings localized? [See General MIB proposal from a year ago (file: /pub/pwg/snmpmib/jobs-mib/gen.doc, .ps, .txt) and General MIB textual-conventions (file: gentc.doc, .ps, .txt).*** **Issue 4 - How is code conversion done for strings, if at all?** ***How is code conversion done for strings, if at all? Where is such code conversion done? In the agent, in the management application? See also description under T(63) data type explanation]***
O(63)	**Octet String**(of 63 octets maximum length).
Integer32	32-bit **Integer** with explicit range indicated - for unsigned quantities, the range is specified as 0..2^31-1 to avoid using the sign bit which avoids implementation problems with signed vs. unsigned representation.
Counter32	32-bit unsigned counter.
DateAndTime	**DateAndTime** from SMIv2 textual-conventions
Index16	16-bit index into another table with a range: 0..2^15-1 to avoid using the sign bit, which causes problems in some implementations. Need to specify whether a 0 value is allowed and, if so, what is its significance.
Index32	32-bit index into another table with a range: 0..2^31-1 to avoid using the sign bit, which causes problems in some implementations. Need to specify whether a 0 value is allowed and, if so, what is its significance.
bit-vector	32-bit integer with each bit being significant

7. Object/Attribute specification

The following tables lists the 43 objects agreed to at the 11/08/96 JMP meeting to be candidates for inclusion in the Job Monitoring MIB. The object/attributes are arranged by MIB/MIB groups. The explanation of each column in described in the previous section. After review of the specifications, additional objects/attributes may be removed from the specification.

7.1 The MIB Instance Group

The jm**MIBInstanceGroup** consists of objects that are for *all* Job Monitoring MIB instances, not just a single instance.

The **jmMIBInstanceGroup** consists entirely of the **jmMIBInstanceEntry** which is indexed by:

1. **jmMIBInstanceIndex -** a running index of Job Monitoring MIB instances supported by this printer or server.

Every object/attribute in the Job Monitoring MIB is indexed by a jmMIBInstanceIndex, so that a single server can support multiple Job Monitoring MIB instances, one for each set of jobs that are separately queuing. Therefore, the cost of permitting a server to support multiple Job Monitoring MIB instances for a printer that only supports one MIB instance is a single extra octet for each OID.

Issue 5 - Is the name jmMIBInstanceIndex ok?

Is the name jmMIBInstanceIndex O.K., or should the name be something like jmJobDeviceIndex?

jmMIBInstanceGroup (M)	Datatype	Specification	DPA
jmMIBInstanceIndex	**Index16(0..2^15-1)**	The index of a Job Monitoring MIB instance. Agents implementing a single Job Monitoring MIB instance shall use an index value of 1 for this object/attribute.	

7.2 The General Group

The **jmGeneralGroup** consists of objects of a general nature that are *not* per-job. The **jmGeneralGroup** consists entirely of the **jmGeneralEntry** which is indexed by:

1. **jmMIBInstanceIndex -** a running index of Job Monitoring MIB instances supported by this printer or server.

jmGeneralGroup (G)	Datatype	Specification	DPA
1. **jmMIBInstanceIndex**	**Index16(0..2^15-1)**	The index of a Job Monitoring MIB instance. Agents implementing a single Job Monitoring MIB instance shall use an index value of 1 for this object/attribute.	
2. **jmGeneralJobRetentionPolicy**	**Integer32(0..2^31-1)**	The default time in seconds that the printer or server retains jobs in the retained state after completion as specified by the system administrator for this instance of the Job Monitoring MIB. While the job is in the retained state, some job submission protocols permit a requester to issue a **ResubmitJob** operation that submits a copy of the job to the same or different printer on the same or different server. Some job submission protocols permit the requester to specify a different value than the system administrator's specified default value. For job submission protocols that do not, this value is the fixed policy specified by the system administrator (by means outside this MIB specification). Printers or servers that do not support the concept of retained jobs shall return a value of 0 for this object/attribute.	118 job-retention-period
3. **jmGeneralMaxNumberOfJobs**	**Integer32(0..2^31-1)**	The maximum number of queued and completed jobs that this server or print can support at the same time. The value **(-1)** shall indicate that there is no fixed limit.	
4. jmGeneralCurrentNumberOfJobs	**Integer32(0..2^31-1)**	The total number of jobs currently in the **jmJobTable**, i.e., the total number of jobs that are in any state: queued (**pre-processing**, **held**, **pending**), **processing**, **needs-attention**, **paused**, **interrupted**, or finished (**retained**, **completed**).	
5. jmGeneralQueuingAlgorithm	**JMQueuingAlgorithm**	The current queuing algorithm being used by the server or printer. The value is either fixed by the implementation or is settable by the system administrator (by means outside this MIB specification).	269 scheduler-identifier

jmGeneralGroup (G)	Datatype	Specification	DPA
	none(3)	This printer or server is not capable of queuing jobs or is not currently configured to accept more than one job at a time; the server or printer shall accept one job at a time and shall reject subsequent jobs until the current job completes processing. The **none(3)** value also indicates that the **jmQueuingGroup** is not implemented.	
	smallestJobFirst(4)	The server or printer shall process the smallest job of equal priority first, independent of the time that the job was received by the server or printer. However, jobs that require resources that require human intervention need not be processed in size order, depending on implementation and/or system administrator specified policy.	
	timeReceived(5)	The server or printer shall process the jobs of equal priority based on time that the server or printer received the job; the earliest received job shall be processed first. However, jobs that require resources that require human intervention need not be processed in order, depending on implementation and/or system administrator specified policy.	

7.3 The Queue Group

The **jmQueueGroup** is made up entirely of the **jmQueueTable** which is an ordered list of jobs that have not completed processing. The **jmQueueGroup** consists of objects/attributes that are not needed after the job has completed processing. The **jmQueueGroup** is conditionally mandatory and shall be implemented by a server or print that performs queuing (or spooling). The **jmQueueGroup** shall *not* be implemented if the value of **jmGeneralQueuingAlgorithm** is **none**. The **jmQueueTable** is indexed by:

1. **jmMIBInstanceIndex -** a running index of Job Monitoring MIB instances supported by this printer or server.
2. **jmQueueIndex** - a running index of the jobs that have *not* finished processing.

jmQueueGroup (Q)	Datatype	Specification	DPA
1. **jmMIBInstanceIndex**	**Index16(0..2^15-1)**	The 16-bit index of a Job Monitoring MIB instance. Agents implementing a single Job Monitoring MIB instance shall use an index value of 1 for this object/attribute.	
2. jmQueueIndex	**Index32(0..2^31-1)**	The 32-bit index of the jobs that have not finished processing. The index values need not be monatonic increasing. However, the order of the jobs specified by the value of this index shall conform to the order specified for the **jmGeneralQueuingAlgorithm** object/attribute.	
3. jmQueueLocalId	**Integer32(0..2^31-1)**	The job's identifier generated locally by the server or printer when that server or printer accepted the job. This value permits the management application to access the other tables to obtain the job-specific objects/attributes. The value 0 need not be generated.	104 job-identifier
4. jmQueueNumberOfInterveningJobs	**Integer32(0..2^31-1)**	The number of jobs that would be processed before this job is processed according to the queuing algorithm specified by the value of the **jmGeneralQueuingAlgorithm** object if no other jobs were to be submitted. The server or printer shall set the value of this object to 0 when the job starts processing.	134 intervening-jobs

jmQueueGroup (Q)	Datatype	Specification	DPA
5. jmJobPriority	**Integer32(0..100)**	This attribute specifies a priority for scheduling the job. It is used by servers and printers that employ a priority-based scheduling algorithm. A higher value specifies a higher priority. The value **1** is defined to indicate the lowest possible priority (a job which a priority-based scheduling algorithm shall pass over in favor of higher priority jobs). The value **100** is defined to indicate the highest possible priority. Priority is expected to be evenly or 'normally' distributed across this range. The mapping of vendor-defined priority over this range is implementation-specific. The omission of this attribute implies that the submitting user places no constraints concerning priority on the scheduling of the job. The server or printer shall fill in the default priority value. A value of **0** shall be returned by implementations that do *not* have a priority-based queuing algorithm.	117 Job-priority
6. jmJobProcessAfterTime	**GenTime**	This object specifies the calendar date and time of day after which the job shall become a candidate to be scheduled for processing. If the value of this attribute is in the future, the server shall set the value of the job's **jmJobCurrentState** to **held** and add the **jobProcessAfterSpecified** bit value to the job's **jmJobStateReasons** object and shall not schedule the job for processing until the specified date and time has passed. When the specified date and time arrives, the server shall remove the **jobProcessAfterSpecified** bit value from the job's **jmJobStateReasons** object and, if no other reasons remain, shall change the job's **jmJobCurrentState** to **pending** so that the job becomes a candidate for being scheduled on devices(s). The server shall assign an empty value to the **jmJobProcessAfterTime** object when no process after time has been specified, so that the job shall be a candidate for processing immediately.	116 Job-print-after

jmQueueGroup (Q)	Datatype	Specification	DPA
7. jmJobMessageToOperator	**T(63)**	This object contains a message from the user to the operator to indicate something about the processing of this job. This message is *not* necessarily related to other job-scheduling attributes. The server shall make this message available to the operator when the job has been accepted.	117 Job-message-to-operator

7.4 The Completed Group

The **jmCompletedGroup** consists entirely of the **jmCompletedTable** which is an ordered list of the job that have completed processing, i.e., jobs that are in the **terminating**, **retained** or **completed** state. The **jmCompletedTable** is indexed by:

1. **jmMIBInstanceIndex -** a running index of Job Monitoring MIB instances supported by this printer or server.
2. **jmCommpletedIndex -** a running index of the jobs that have finished processing.

jmCompletedGroup (C)	Datatype	Specification	DPA
1. **jmMIBInstanceIndex**	**Index16(0..2^15-1)**	The 16-bit index of a Job Monitoring MIB instance. Agents implementing a single Job Monitoring MIB instance shall use an index value of 1 for this object/attribute.	
2. jmCompletedIndex	**Index32(0..2^31-1)**	The 32-bit index of the jobs that have completed processing, i.e., jobs that are in the **terminating**, **retained**, or **completed** states. The index values need not be monatonic increasing. However, the order of the jobs specified by the value of this index shall be the order in which the jobs finished processing.	
3. jmJobLocalId	**Integer32(0..2^31-1)**	The job's identifier generated locally by the server or printer when that server or printer accepted the job. This value permits the management application to access the other tables to obtain the job-specific objects/attributes. The value 0 need not be generated.	104 job-identifier

7.5 The Job Group

The **jmJobGroup** consists of (1) job identification, (2) job parameters, and (3) job status and accounting objects/attributes that have a *single* value per job. The **jmJobGroup** consists entirely of the **jmJobTable** which is indexed by:

1. **jmMIBInstanceIndex -** an instance index to distinguish separate sets of tables when a server supports more than one printer.
2. **jmJobLocalId** - the job identifier that was generated locally by the server or printer that accepted the job.

7.5.1 Job Identification (I) object/attribute specification

The following **jmJobGroup** objects/attributes identify the job to the user of the management application which may be acting in the role of an end-user or a system operator:

jmJobGroup - identification (I)	**Datatype**	**Specification**	**DPA**
1. jmMIBInstanceIndex	**Index16(0..2^15-1)**	The 16-bit index of a Job Monitoring MIB instance. Agents implementing a single Job Monitoring MIB instance shall use an index value of 1 for this object/attribute.	
2. jmJobLocalId	**Integer32(0..2^31-1)**	The identifier of the job on the device or server. The job's identifier is generated locally by the server or device when that server or device accepted the job. However, if the device does not generate a job identifier for each job, then the Job Monitoring MIB agent shall generate the job identifier for the job. The value 0 need not be generated.	104 job-identifier
3. jmJobDownstreamId	**Integer32(0..2^31-1)**	This object identifies the job on the next downstream (downstream from the server implementing this Job Monitoring MIB) server or device. **Issue 6 - Do we really need the jmJobDownstreamId/jmJobPrinterId?** ***Do we really need the jmJobDownstreamId/jmJobPrinterId or can we get away with having only a single id in our MIB, namely the jmJobLocalId object?***	104 Job-identifier-on-printer

jmJobGroup - identification (I)	**Datatype**	**Specification**	**DPA**
4. jmJobTypes	**JMJobTypes**	Specifies the type(s) of service to which the job has been submitted (print, fax, scan, etc.). The service type is represented as an enum that is bit encoded with each job service type so that more general and arbitrary services can be created, such as services with more than one destination type, or ones with only a source or only a destination. One of the purposes of this object/attribute is to permit a requester to filter out jobs that are not of interest. For example, a printer operator may only be interested in jobs that include printing. That is why the object/attribute is in the job identification category. **Issue 7 - Should jmJobTypes be something that a job submitter can specify or not?** ***Should the jmJobTypes object/attribute be read-only, rather than something that the requester specifies through the job-submission protocol? If yes, then move it to the job status section.*** **Issue 8 - Is the parent job type the or of the child jobs?** ***If the sub-jobs are of different types, what type is the parent job? The inclusive or of the sub-job bits?*** The following service component types are defined and are assigned a separate bit value in the enum for use with the jmJobTypes object/attribute:	
	print(0)	The job contains some document production instructions that specify printing	
	scan(1)	The job contains some document production instructions that specify scanning	
	faxIn(2)	The job contains some document production instructions that specify receive fax	
	faxOut(4)	The job contains some document production instructions that specify sending fax	
	getFile(8)	The job contains some document production instructions that specify accessing files or documents	

jmJobGroup - identification (I)	**Datatype**	**Specification**	**DPA**
	putFile(16)	The job contains some document production instructions that specify storing files or documents	
	mailList(32)	The job contains some document production instructions that specify distribution of documents using an electronic mail system.	
5. jmJobOwner	**T(63)**	The name of the user that submitted the job. The method of assigning this user name will be system and/or site specific but the method must insure that the name is unique to the network that is visible to the client and target device. This value should be the *authenticated* name of the user submitting the job.	106 Job-owner
6. jmJobDeviceNameRequested	**T(63)**	The administratively defined name of the target device. Its value corresponds to the new object being added to the Printer MIB: **prtGeneralDeviceName**. NOTE - while this object/attribute could be considered a resource and could be allocated in the **jmResourceTable**, it has been allocated as a separate object, since management applications are likely to want to get this value each time they access a job, rather than have to copy the entire **jmResourceTable** to search for it.	118 Printer-name-requested
7. jmDeviceIndex	**Index32**	The device index (hrDeviceIndex) of the associated Printer or other device MIB that contains the state of the device to which the job was submitted or has been assigned by the server. See Host Resources MIB.	
8. jmJobSourceChannel	**PrtChannelIndex**	The index of the row in the associated Printer MIB of the channel which is the source of the print job. See RFC 1759.[2]	

[2] RFC 1759 does not have a textual convention for **PrtChannelIndex**, so that the job monitoring MIB is not able to import these enum definitions from the Printer MIB. When the PWG progresses the Printer MIB to draft standard status we are introducing a **PrtChannelIndex** textual convention in the Printer MIB, so that the job monitoring MIB can import **PrtChannelIndex**. Also, now that the Job Monitoring MIB points to the Channel row in the Printer MIB, there is no need for a port object/attribute in the Job Monitoring MIB, since the PWG is adding a **prtChannelInformation** object to the Channel table of the draft Printer MIB.

jmJobGroup - identification (I)	**Datatype**	**Specification**	**DPA**
9. jmJobName	**T(63)**	This object is the human readable string name of the job as assigned by the submitting user to help the user distinguish between his/her various jobs. This name does not need to be unique. This attribute is intended for enabling a user or the user's application to convey a job name that may be printed on a start sheet, returned in a **ListObjectAttributes** result, or used in notification or logging messages. If this attribute is not specified, no job name is assumed, but implementation specific defaults are allowed, such as the value of the **document-name** attribute of the first document in the job or the file-name of the first document in the job. **Issue 9 - What is the difference between jmJobName and jmClientId?** ***Is there really a difference between the job name and the job client id? If so, what is that difference? Lack of uniqueness and whether assigned by humans or by software?***	105 Job-name
10. jmJobSubmissionTime	**DateAndTime**	This object indicates the date and time the job was submitted by the client to the first server or printer.	130 Submission-time
11. jmJobComment	**T(63)**	This object contains the arbitrary human-readable text string supplied by the submitting user or the application program for any purpose. For example, a user might indicate what he/she is going to do with the printed output or might indicate how the document was produced.	106 Job-comment

391 ### 7.5.2 Job Parameters (P) object/attribute specification

392 The Job Monitoring MIB agent derives the following **jmJobGroup** objects/attributes from parameters
393 submitted in the job submission protocol and/or embedded in the PDL:

jmJobGroup - job parameters (P)	**Datatype**	**Specification**	**DPA**

jmJobGroup - job parameters (P)	**Datatype**	**Specification**	**DPA**
12. jmJobTotalOctetsHigh	**Integer32(0..2^31-1)**	The high order 31 bits of the total number of octets to be processed in the job. The total octets to be processed can be used in the denominator with the **jmJobOctetsCompletedHigh/jmJobOctetsCompletedLow** object/attributes in the numerator in order to produce a "thermometer" of progress of the job. Only 31 bits are used in each half, so that the sign bit is avoided, since unsigned arithmetic is problematic on some platforms. NOTE - The total octets to be processed is represented in two objects/attributes, so that SNMPv1 implementations can implement, since Integer64 requires SNMPv2. The value (-2) means unknown.	138 total-job-octets
13. jmJobTotalOctetsLow	Integer32**(0..2^31-1)**	This object contains the low 31 bits of the total number of octets to be processed in the job. The value (-2) means unknown. **Issue 10 - Combine jmJobTotalOctetsHigh and Low and count by K?** ***Should we combine these two 31-bit objects into a single 31-bit object and count by K octets, instead of octets?***	138 total-job-octets

7.5.3 Job Status and Accounting (S)

The **jmJobCurrentState** object specifies the standard job states. The legal job state transitions are shown in the state transition diagram presented in Figure 5.

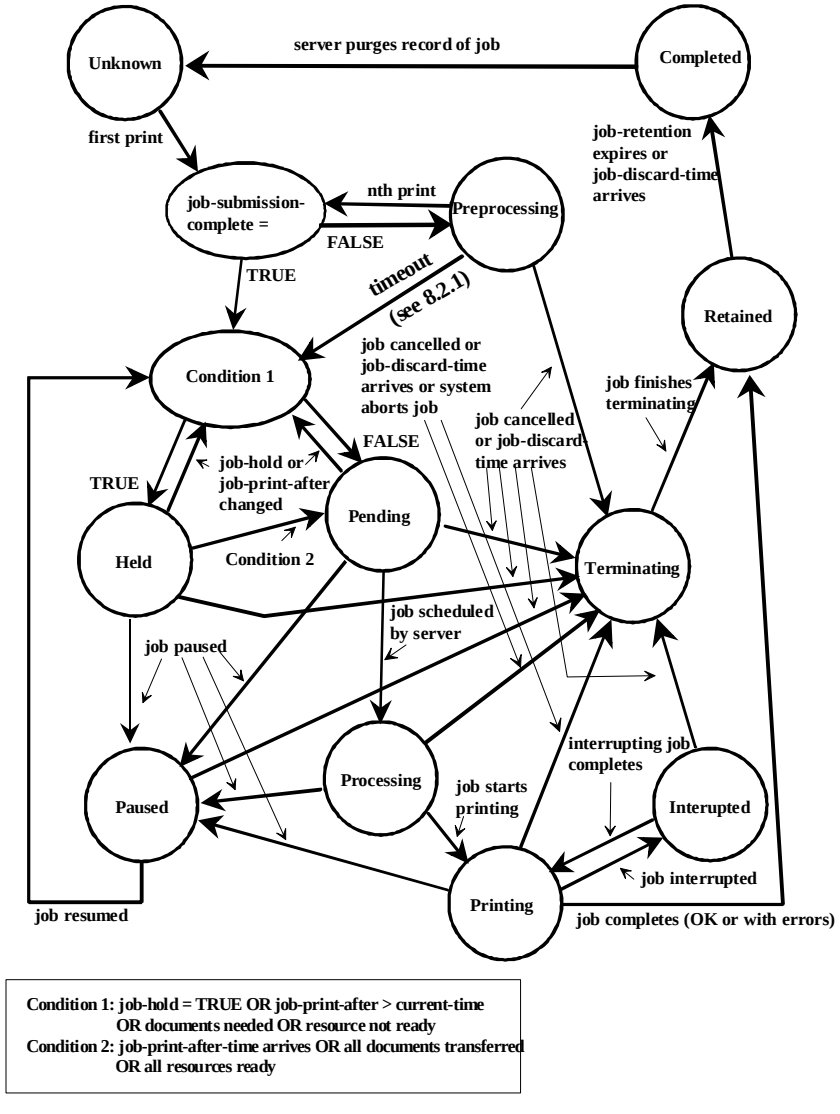

Figure 9-1 - Job state transition diagram

Figure 5 - Job State Transition Diagram

The following **jmJobGroup** objects/attribute are used to provide status information to users and accounting data for accounting applications:

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
14. jmJobCurrentState	**JMJobState**	This object identifies the current state of the job (pending, processing, held, etc.) Management applications shall be prepared to receive all the standard job states. Servers and devices are not required to generate all job states, only those which are appropriate for the particular implementation. The following job state standard values are defined by adding +2 to the last arc of the ISO DPA OBJECT IDENTIFIER value of the **job-current-state** job attribute:	124 Current-job-state
	unknown(2)	The job state is not known, or is indeterminate.	
	pre-processing(3)	The job has been created on the server by the **create-job** sub-operation of the print-request, but a print-request with a **TRUE** value for the **job-submission-complete** component of the PrintArgument has not yet been received and no document has started processing. The job maybe in the process of being checked by the server for attributes, defaults being applied, a printer being selected, etc.	
	held(12)	The job is waiting to be released for scheduling for any number of reasons as specified by the value of the job's **job-state-reasons** attribute.	
	pending(6)	The job's **job-submission-complete** attribute is **TRUE** since the server has received a print-request with the **job-submission-complete** parameter **TRUE** and the job is waiting to start processing on a printer.	

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	processing(7)	The server is processing the job, or has made the job ready for printing, but the output device is not yet printing it, either because the job hasn't reached the output device or because the job is queued in the output device or some other spooler, awaiting the output device to print it. If a server implementation or policy is to start processing documents before the last print-request (with a **TRUE** value for the **job-submission-complete** parameter) and the value of the job's **job-scheduling** attribute is not **after-complete**, the server shall change the job's **current-job-state** from **pre-processing** directly to the **processing** state when the server begins processing any of the job's documents.	
	needs-attention(9)	A device that the job is using requires human attention. Usually devices indicate their condition in human readable form locally at the device. The management application can obtain more complete device status by querying the appropriate device MIB using the **jmDeviceIndex** object.	not in ISO DPA[3]
	paused(13)	The job has been paused as a result of a PauseJob operation.	
	interrupted (8)	The job was interrupted by the InterruptJob request for an intervening job, and shall resume processing automatically once the intervening job has completed.	
	terminating (14)	The job has been cancelled by a CancelJob request or aborted by the server and is in the process of terminating. The job's **job-state-reasons** attribute contains the reasons that the job is being terminated.	

[3] Instead of the needs-attention job state, ISO DPA uses the multi-valued **printer-state-of-printers-assigned** job attribute, so that the state of each device that a job is using can be accurately represented. However, for the Job Monitoring MIB, the simpler approach is used of adding a single **needs-attention** job state if any device that the job is using needs attention and relying on the device MIB for more information. The representation of jobs that use more than one device is not handled by the Job Monitoring MIB, since only one hrDeviceIndex value is allowed per job.

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	retained(11)	The job is being retained at the server as a result of the job's **job-retention-period** being non-zero. The job has (1) completed successfully or with warnings or errors, (2) been aborted while printing by the server, or (3) been cancelled by the **CancelJob** request before or during processing. The job's **job-state-reasons** attribute contains the reasons that the job has been retained. While in the **retained** state, all of the job's document data (and resources, if any) shall be retained by the server or printer; thus a copy of the job in the **retained** state could be processed again.	
	completed(17)	The job has: (1) completed successfully or with warnings or errors, (2) been aborted by the server while printing, or (3) been cancelled by the **CancelJob** request, AND the job's: (1) **job-retention-period** was zero or has expired, or (2) **job-discard-time** has arrived. The job's **job-state-reasons** attribute contains the reason(s) that the job has been completed. While in the **completed** state, a job's document data (and resources if any) need not be retained by the server; thus a job in the **completed** state could not be reprinted. The length of time that a job may be in this state, before transitioning to **unknown**, is implementation-dependent. However, servers that implement the **completed** job-state shall retain all of the job's Job Monitoring MIB objects, except the jmQueuingGroup objects, so that a management application accounting program can copy them to an accounting log.	

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
15. jmJobStateReasons	**JMJobStateReasons - enum encoded as a bit vector**	This object provides additional information regarding the **jmJobCurrentState** object. This object identifies the reason or reasons that the job is in the **held, pending, processing, terminating**, **retained**, or **completed** state. The server shall indicate the particular reason(s) by setting the value of the **job-state-reasons** attribute. When the job is not in any of these states, the server shall set the value of the **job-state-reasons** attribute to the empty set. **Issue 11 - Make jmJobStateReasons a bit string?** ***There are so many reasons that 31 bits will not be enough. Adding additional objects, just postpones the problem. Should we make this object be a bit string, of, say, 63 octets, with each reason encoded as a separate bit, 8 bit reasons per octet?*** The following standard values are defined as 2 ** (last arc of DPA id-val-reasons-xxx OID - 1):	127 Job-state-reasons
	documents-needed(1)	The complete job has been accepted by the server (the value of the **job-submission-complete** element was **TRUE** in the last Print request for the job), but the server is waiting for its files to start and/or finish being transferred before the job can be scheduled to be printed.	yes
	job-hold-set(2)	The value of the job's **job-hold** attribute is **TRUE**.	yes
	job-print-after-specified(4)	The value of the job's **job-print-after** attribute has specified a time specification that has not yet occurred.	yes
	required-resources-not-ready(8)	At least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.	yes
	successful completion(16)	The job completed successfully.	yes

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	completed-with-warnings(32)	The job completed with warnings.	yes
	completed-with-errors(64)	The job completed with errors (and possibly warnings too).	yes
	cancelled-by-user(128)	The job was cancelled by the user using the CancelJob request.	yes
	cancelled-by-operator(256)	The job was cancelled by the operator using the CancelJob request.	yes
	aborted-by-system(512)	The job was aborted by the system.	yes
	logfile-pending(1024)	The job's logfile is pending file transfer.	yes
	logfile-transferring(2048)	The job's logfile is being transferred.	yes
	cascaded(0)	After the outbound gateway retrieves all job and document attributes and data, it stores the information into a spool directory. Once it has done this, it sends the supervisor a job-processing event with this **job-state-reason** which tells the supervisor to transition to a new job state.	no
	deleted-by-administrator(1)	The administrator has issued a **Delete** operation on the job or a **Clean** operation on the server or queue containing the job; therefore the job may have been cancelled before or during processing, and will have no retention-period or completion-period.	no
	discard-time-arrived(2)	The job has been deleted (cancelled with the **job-retention-period** set to 0) due to the fact that the time specified by the job's **job-discard-time** has arrived [if the job had already completed, the only action that would have occurred is that the **job-retention-period** would be set to 0 and the job is deleted].	no

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	post-processing-failed(3)	The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has been placed into retained state for a system-defined period of time (Printxchange, 5 minutes), so the administrator can examine it, resubmit it, etc. The post-processing agent is a plug-and-play mechanism which the system and the customer uses to add functionality that is executed after a job has finished processing.	no
	submission-interrupted (4)	Indicates that the job was not completely submitted for the following reasons: (1) the server has crashed before the job was closed by the client. The server shall put the job into the **completed** state (and shall not print the job). (2) the server or the document transfer method has crashed in some non-recoverable way before the document data was entirely transferred to the server. The server shall put the job into the completed state (and shall not print the job). (3) the client crashed or failed to close the job before the time-out period (Printxchange, 20 minutes). The server shall close the job and put the job into the held state with **job-state-reasons** of **submission-interrupted** and **job-hold-set** and with the job's **job-hold** attribute set to **TRUE**. The user may release the job for scheduling by issuing the **ReleaseJob** operation.	no
	max-job-fault-count-exceeded(5)	The job has been faulted and returned by the server several times and that the **job-fault-count** exceeded the device's (or server's, if not defined for the device) **cfg-max-job-fault-count**. The job is automatically put into the **held** state regardless of the **hold-jobs-interrupted-by-device-failure** attribute. This j**ob-state-reasons** value is used in conjunction with the **job-interrupted-by-device-failure** value.	no
	devices-need-attention-time-out(6)	One or more document transforms that the job is using needs human intervention in order for the job to make progress, but the human intervention did not occur within the site-settable time-out value and the server/device has transitioned the job to the **held** state.	no

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	needs-key-operator-time-out(7)	One or more devices or document transforms that the job is using need a specially trained operator (who may need a key to unlock the device and gain access) in order for the job to make progress, but the key operator intervention did not occur within the site-settable time-out value and the server/device has transitioned the job to the **held** state.	no
	job-start-wait-time-out(8)	The server/device has stopped the job at the beginning of processing to await human action, such as installing a special cartridge or special non-standard media, but the job was not resumed within the site-settable time-out value and the server/device has transitioned the job to the **held** state. Normally, the job is resumed by means outside the job submission protocol, such as some local function on the device.	no
	job-end-wait-time-out(9)	The server/device has stopped the job at the end of processing to await human action, such as removing a special cartridge or restoring standard media, but the job was not resumed within the site-settable time-out value and the server/device has transitioned the job to the **retained** state. Normally, the job is resumed by means outside the job submission protocol, such as some local function on the device, whereupon the job shall transition immediately to the **terminating** state.	no
	job-password-wait-time-out(10)	The server/device has stopped the job at the beginning of processing to await input of the job's password, but the human intervention did not occur within the site-settable time-out value and the server/device has transitioned the job to the **held** state. Normally, the password is input and the job is resumed by means outside the job submission protocol, such as some local function on the device.	no
	device-timed-out(11)	A device that the job was using has not responded in a period specified by the device's site-settable **device-timeout-period**[4] attribute.	no

[4] The **device-timeout-period** attribute is a rename of the ISO DPA **printer-timeout-period** printer attribute.

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	connecting-to-device-time-out(12)	The server is attempting to connect to one or more devices which may be dial-up, polled, or queued, and so may be busy with traffic from other systems, but server was unable to connect to the device within the site-settable time-out value and the server has transitioned the job to the **held** state.	no
	transferring(13)	**T**he job is being transferred to a down stream server or device.	no
	queue-in-device(14)	The job has been queued in a down stream server or device.	no
	job-cleanup(15)	**T**he server/device is performing cleanup activity as part of ending normal processing.	no
	processing-to-stop-point(16)	The requester has issued an **InterruptJob** operation and the server/device is processing up until the specified stop point occurs.	no
	job-password-wait(17)	The server/device has selected the job to be next to process, but instead of assigning resources and started the job processing, the server/device has transitioned the job to the **held** state to await entry of a password (and dispatched another job, if there is one). The user resumes the job either locally or by issuing a **ReleaseJob** supplying a **job-password=*secret-code*** input parameter that must match the job's **job-password** attribute.	no
	validating(18)	The server/device is validating the job after a **CreateJob** operation. The job state may be **creating, held, pending,** or **processing.**	no
	queue-held(19)	The operator has held the entire queue by means outside the scope of the Job model.	no
	job-proof-print-wait(20)	The job has produced a single proof copy and is in the **held** state waiting for the requester to issue the **ReleaseJob** operation to release the job to print normally, obeying the **job-copies** and **copy-count** job and document attributes that were originally submitted.	no
	held-for-diagnostics(21)	The system is running intrusive diagnostics, so the all jobs are being held.	no
	service-off-line(22)	The service/document transform is off-line and accepting no jobs. All **pending** jobs are put into the **held** state. This could be true if its input is impaired or broken.	no

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	no-space-on-server(23)	The job is held because there is no room on the server to store all of the job. For example, there is no room for the document data or a scan-to-file job.	no
	pin-required(24)	The SA-settable device policy is (1) to require PINs, and (2) to hold jobs that do not have a pin supplied as an input parameter to **CreateJob.** The requester shall either (1) enter a pin locally at the device or issue the **ReleaseJob** operation supplying the PIN in order for the job to be able to proceed.	no
	exceeded-account-limit(25)	The account for which this job is drawn has exceeded its limit. This condition should be detected before the job is scheduled so that the user does not wait until his/her job is scheduled only to find that the account is overdrawn. This condition may also occur while the job is processing either as processing begins or part way through processing. An overdraft mechanism should be included to be user-friendly, so as to minimize the chances that the job cannot finish or that media is wasted. For example, the server/device should finish the current copy for a job with collated document copies, rather than stopping in the middle of the current document copy.	no
	held-for-retry(26)	The job encountered some errors that the server/device could not recover from with its normal retry procedures, but the error is worth trying the job later, such as phone number busy or remote file system in-accessible. For such a situation, the server/device shall add the **held-for-retry** value to the job's **job-state-reasons** attribute and transition the job from the **processing** to the **held**, rather than to the **retained** state.	no
	cancelled-by-shutdown	The job was cancelled because the server or device was shutdown before completing the job. The job shall be placed in the **pending** state [if the job was not started, else the job shall be placed in the **terminating** state].	PSIS
	device-unavailable	This job was aborted by the system because the device is currently unable to accept jobs. This reason [shall be] used in conjunction with the reason aborted-by-system. The job shall be placed in the **pending** state.	PSIS

jmJobGroup - job status and accounting (S)	**Datatype**	**Specification**	**DPA**
	wrong-device	This job was aborted by the system because the device is unable to handle this particular job; the spooler should try another device. This reason [shall be] used in conjunction with the reason aborted-by- system. The job shall be **pending** if the queue contains other physical devices that the job could print on, and the spooler is capable of not sending the job back to a physical device that has rejected the job for this job-state-reasons value. Otherwise, [the job] shall be **retained**.	PSIS
	bad-job	This job was aborted by the system because this job has a major problem, such as an ill-formed PDL; the spooler should not even try another device. This reason shall be used in conjunction with the reason aborted-by-system. The job shall be placed in the **terminating** state.[5]	PSIS
	job-interrupted-by-device-failure	A device or the print system software that the job was using has failed while the job was processing. The device/printer is keeping the job in the **held** state until an operator can determine what to do with the job.	PSIS

399 The following table shows job-state-reasons and the job states for which they are applicable. The ISO
400 DPA job state reasons are shown along with additional job-state-reasons that should be considered to give
400
401 users additional feedback on the progress of their job:

	Job States							
	held	pending	processing	paused	interrupted	terminating	retained	completed
Descriptive Name	ISO DPA values							
documents-needed	x							
job-hold-set	x							
job-process-after-specified	x							
required-resources-not-ready	x							
successful-completion							x	x
completed-with-warnings							x	x

[5] This PSIS definition of bad-job has been clarified in the Job Monitoring MIB. The PSIS version reads (with the additions shown inside []: "This job was aborted by the system because this job has a major problem[, such as an ill-formed PDL]; the spooler should not even try another printer. This reason [shall be] used in conjunction with the reason aborted-by-system. The job shall be placed in the [**terminating]** state.

	Job States							
	held	pending	processing	paused	interrupted	terminating	retained	completed
Descriptive Name	ISO DPA values							
completed-with-errors							x	x
cancelled-by-user						x	x	x
cancelled-by-operator						x	x	x
aborted-by-system						x	x	x
logfile-pending						x	x	
logfile-transferring						x	x	

401

Additional reasons								
Descriptive Name	held	pending	processing	paused	interrupted	terminating	retained	completed
cascaded(0)						x	x	x
deleted-by-administrator(1)						x	x	x
discard-time-arrived(2)						x	x	x
postprint-failed(3)						x	x	x
submission-interrupted(4)						x	x	x
max-job-fault-count-exceeded(5)						x	x	x
devices-need-attention-time-out(6)	x					x	x	x
needs-key-operator-time-out(7)	x					x	x	x
job-start-wait-time-out(8)	x					x	x	x
job-end-wait-time-out(9)						x	x	x
job-password-wait-time-out(10)	x	x						
device-timed-out(11)	x					x	x	x
connecting-to-device-time-out(12)	x					x	x	x
transferring(13)			x					
queued-in-device(14)			x					
job-cleanup(15)			x					
processing-to-stop-			x					

Additional reasons								
Descriptive Name	held	pending	processing	paused	interrupted	terminating	retained	completed
point(16)								
job-password-wait(17)		x				x	x	x
validating(18)	x	x	x					
queue-held(19)	x							
job-proof-print-wait(20)	x							
held-for-diagnostics(21)	x							
service-off-line(22)	x							
no-space-on-server(23)	x							
pin-required(24)	x					x	x	x
exceeded-account-limit(25)	x					x	x	x
held-for-retry(26)	x							

402

X/Open PSIS job-state-reasons extension values								
Descriptive Name	held	pending	processing	paused	interrupted	terminating	retained	completed
cancelled-by-shutdown						x	x	x
printer-unavailable		x						
wrong-printer						x	x	x
bad-job						x	x	x
job-interrupted-by-printer-failure	x							

403

jmJobGroup - job status and accounting (S) - continued	**Datatype**	Specification	DPA

jmJobGroup - job status and accounting (S) - continued	**Datatype**	Specification	DPA
16. jmJobOctetsCompletedHigh	**Integer32(0..2^31-1)**	This object and the **jmJobOctetsCompletedLow** object together define the number of octets currently processed by the device. For multiple copies generated from a single data stream, the value shall be incremented as if each copy was printed from a new data stream without resetting the count between copies. NOTE - Keep this as its own separate object, rather than including this in the resources mechanism, because this object is the only "resource" that need 64 bits. Also management applications will want to access it frequently to make a job progress thermometer and such management applications would not want to have to read all of the resource rows to find the one that keeps octets completed. This object is the high order 31-bits of 62-bit quantity. This objects avoid the use of the sign bit, because unsigned arithmetic is problematic on some platforms. The value of this object/attribute shall be **0** if processing has not started for this job.	132 Octets-completed

jmJobGroup - job status and accounting (S) - continued	**Datatype**	Specification	DPA
17. jmJobOctetsCompletedLow	**Integer32(0..2^31-1)**	Low 31 bits of the octets processed for this job. **Issue 12 - Combine jmJobOctetsCompletedHigh and Low and count by K?** ***Should we combine these two 31-bit objects into a single 31-bit object and count by K octets, instead of octets?*** **Issue 13 - Add jmJobTotalPagesSpooled, jmJobTotalPagesSentToDevice, and jmJobTotalPagesPrinted?** ***Add jmJobTotalPagesSpooled, jmJobTotalPagesSentToDevice, and jmJobTotalPagesPrinted?***	132 job-octets-completed
18. jmJobStartedProcessingTime	**DateAndTime**	The date and time at which this job started processing.	130 Started-printing-time
19. jmJobCompletionTime	**DateAndTime**	The date and time at which this job completed processing.	135 Completion-time
20. jmJobAccountName	**T(63)**	Information supplied by the submitting user for use by accounting services to allocate or categorize charges for services provided, such as a customer account name. If the implementation does not support the concept of an account name and/or the submitting user does not supply one, the value returned shall be a zero-length string.	120 Accounting-information

7.6 The Resource Group

The **jmResourceGroup** consists of requested and used resources objects/attributes that can have multiple values per job. The **jmResourceGroup** consists entirely of the **jmResourceTable** which is indexed by:

1. **jmMIBInstanceIndex -** an instance index to distinguish separate sets of tables when a server supports more than one printer.
2. **jmJobLocalId** - the job identifier that was generated locally by the server or printer that accepted the job.
3. **jmResourceIndex -** a running index of resources for each job

The following table is a per job table with an extra index for each resource that a job can require and consume. The enum type indicates the type of resource. Some resource types are used both for requested and used values, while others have distinct types for requested versus used. The agent is able to discover the resources required either from the job submission protocol itself or from the document PDL. As the documents are interpreted, the interpreter may discover additional resources that are required and so adds additional rows to this table. As the resources are actually used, the usage counter is incremented according to the units indicated by the **jmResourceUnits** object.

Issue 14 - Always have separate resource types for requested versus used?

Always have separate resource types for requested versus used?

jmResourceGroup (R)	Datatype	Specification	DPA
1. jmMIBInstanceIndex	**Index16(0..2^15-1)**	The 16-bit index of a Job Monitoring MIB instance. Agents implementing a single Job Monitoring MIB instance shall use an index value of 1 for this object/attribute.	
2. jmJobLocalId	**Integer32(0..2^31-1)**	The identifier of the job on the device or server. The job's identifier is generated locally by the server or device when that server or device accepted the job. However, if the device does not generate a job identifier for each job, then the Job Monitoring MIB agent shall generate the job identifier for the job. The value 0 need not be generated.	104 job-identifier
3. jmResourceIndex	**Index16(0..2^15-1)**	A running index of the resources requested and/or used by the job.	
4. **jmResourceType**	**JMResourceType**	The type of resource, e.g., medium, ink, staples, processing-time, color-impressions, etc. required to process the job before the job start processing. This value is updated while the job processes to indicate the amount of the resource that is being used while the job is processing. After the job completes processing, this value indicates the total usage of this resource made by the job. The accounting system will copy this value to a suitable longer term storage.	

Some resources are mandatory for conformance, and the rest are optional. The mandatory resources are:

- **interpreters(9)**
- **sheetsCompleted(13)**
- **processingTime(14)**

Some resource instances can occur more than once in the **jmResourceTable** while most resources shall have only a single entry in the **jmResourceTable**, if they are present at all. The resources that can appear multiple times, are flagged with an ***** in the **Datatype** column. However, there shall be no duplicate rows. For example, if a document or job uses multiple types of media, there shall be only one row in the **jmResourceTable** for each media type, not one row for each document that uses that medium type.

Resource type name	enum value	jmResourceAmount Datatype	Description	DPA
a) Document name(s) (or file-names)	**documentName(3)**	**T(63) ***	The name of the document. If the document does not have a name, the file name may be used as a default.	139 Document-name
b) Number of copies of the entire job requested	**jobCopiesRequested(4)**	**Integer32(0..2^31-1)**	The number of copies of the entire job that are to be produce A value of -2 means unknown.	108 job-copies
c) Number of copies of the entire job produced so far	**jobCopiesProduced(5)**	**Integer32(0..2^31-1)**	The number of copies of the entire job that the entire job has produced so far. A value of -2 means unknown.	108 total-job-copies
d) Number of document copies requested	**documentCopiesRequested(6)**	**Integer32(0..2^31-1)**	The total count of the number of document copies requested. If there are documents A, B, and C, and document B is specified to produce 4 copies, the number of document copies requested is 6 for the job.	167 copy-count
e) Number of document copies produced so far	**documentCopiesProduced(7)**	**Integer32(0..2^31-1)**	The total count of the number of document copies produced so far. If there are documents A, B, and C, and document B is specified to produce 4 copies, the number of document copies starts a 0 and runs up to 6 for the job as the job processes.	180 copies-completed
f) Number of sides requested/used (one-sided, two-sided)	**sides(7)**	**Integer32(1..2)**	The number of sides that any document in this job will require or did use.	163 sides
g) **interpreters** - PDLs requested/used	**interpreters(8)**	**T(63) ***	The interpreter language family, level, and version, corresponding to the Printer MIB **prtInterpreterLangFamily**, **prtInterpreterLangLevel**, and	167 document-format

Resource type name	enum value	jmResourceAmount Datatype	Description	DPA
			prtInterpreterLangVersion objects, that this job requires and uses. A document or a job may use more than one PDL. However, there shall be only one distinct row for each distinct PDL; there shall be no duplicates. For the MIB, instead of using the Printer MIB enums for Interpreter Language Family (which map 1-to-1 to ISO DPA document-format OIDs), use the name of the PDL, along with the level and version strings to identify the PDLs required/used. This object/attribute will be mainly used by the operator to redirect jobs. **Issue 15 - Ok to use the string names, instead of enums for the printer interpreter language family?** ***Ok to use the string names, instead of enums for the printer interpreter language family?***	
h) Physical devices requested/used	**physicalDevices(9)**	**Index32 ***	The index of the physical device MIB instance requested/used. This value is an hrDeviceIndex value. See the Host Resource MIB.	119 Physical-printers-requested Printers-assigned
i) FAX phone number(s) requested/used	**faxPhoneNumbers(10)**	**T(63) ***	The FAX phone number(s) requested/used.	
j) Impressions (sides) completed	**impressionsCompleted(11)**	**Counter32 (0..2^31-1)**	The number of impressions processed by this job. For a print job, an impression is the marking of the entire side of a sheet. Two-sided processing involves two impressions	131 Impressions-completed

Resource type name	enum value	jmResourceAmount Datatype	Description	DPA
			per sheet. The value of this attribute shall be 0 if processing has not started for this job.	
k) Sheets completed	**sheetsCompleted(12)**	**Counter32 (0..2^31-1)**	The number of medium sheets that have been processed whether those sheets have been processed on one side or on both.	131 Media-sheets-completed
l) Processing time completed	**processingTime(13)**	**Integer32(0..2^31-1)**	The amount of time that the job has been processing in seconds.	135 Processing-time
m) Processing Messages	**processingMessage(14)**	**T(63) ***	A message that is generated during the processing of the job as a simple form of processing log to show progress and any problems.	

430

jmResourceGroup - job status and accounting (R) - continued	Datatype	Specification	DPA
5. **jmResourceName** - resource required/usage name	**T(63)**	The name of the resource, e.g., iso-a4-white, red-ink, number 2 staples, the document name, etc..	

jmResourceGroup - job status and accounting (R) - continued	Datatype	Specification	DPA
6. **jmResourceUnits** - resource required/used usage-unit	**JMResourceUnits**	The units in which this resource is measured, e.g., sheets, feet, meters, micrometers, tenThosandthsOfInches, seconds, count, etc. Start with the Printer MIB units and add: tenThousandthsOfInches(3),--.0001 micrometers(4), characters(5), lines(6), impressions(7), sheets(8), dotRow(9), hours(11), thousandthsOfOunces(12), tenthsOfGrams(13), hundrethsOfFluidOunces(14), tenthsOfMilliliters(15) feet(16), meters(17), seconds(18), count(19), -- Integer32(0..2^31-1) text(20), -- T(63) hrDeviceIndex(21) -- Index32	
7. **jmResourceAmount** - resource amount requested/used	**T(63), Integer32, Index32, OR Counter32**	The amount of the resource requested or used so far in the units specified by the "resource required/used usage unit" object. Since resources can be added to this table while the job is waiting to be processed, which can be a long time before any of the resource is actually used, the amount-used is set to 0. The counter is divided into two objects, so that SNMPv1 implementations can support the Job Monitoring MIB.	

431

Appendix A - ISO DPA attribute specifications

The ISO DPA attribute specifications have been moved from the JMP object/attribute specifications to this appendix for reference. The corresponding JMP object/attribute is indicated in the first column. If the second column is empty, there is no corresponding ISO DPA attribute.

8. Appendix A - Comparison with ISO DPA

The order of the groups is the same as the specification.

8.1 The MIB Instance Group - comparison with ISO DPA

JMP object/attribute	Corresponding ISO DPA specification
1. jmMIBInstanceIndex - a running index of Job Monitoring MIB instances supported by this printer or server.	

8.2 The General Group - comparison with ISO DPA

jmGeneralGroup (G)	Corresponding ISO DPA specification
1. jmMIBInstanceIndex - a running index of Job Monitoring MIB instances supported by this printer or server.	
2. **jmGeneralJobRetentionPolicy** - *default* time in seconds that jobs are retained after completion.	
3. **jmGeneralMaxNumberOfJobs** - the maximum number of job; -1 means no limit.	
4. jmGeneralCurrentNumberOfJobs - the total number of jobs currently in the Job Table (pending and completed).	
5. jmGeneralQueuingAlgorithm - the current scheduling algorithm being used or **none** (no queuing is possible)**.**	**scheduling-identifier** This attribute uniquely identifies the scheduling-algorithm. The following standard values are defined: **fifo**, **shortest-job-first**

8.3 The Queue Group - comparison with ISO DPA

jmQueueGroup (Q)	Corresponding ISO DPA specification
1. jmMIBInstanceIndex - a running index of Job Monitoring MIB instances supported by this printer or server.	
2. jmQueueIndex - a running index of the jobs that have *not* finished processing.	
3. **jmJobLocalId** - the job's identifier generated locally by the printer or server implementing this JM MIB	**Job-identifier** See below.
4. **jmQueueNumberOfInterveningJobs** - the number of jobs in front of this job	Intervening-jobs This attribute indicates the number of other jobs to be printed before this job may be scheduled for printing. The server shall set the value of this attribute to **0** when the job begins printing.
5. **jmJobPriority** - Job priority	Job-priority This attribute specifies a priority for scheduling the print-job. It is used by servers that employ a priority-based scheduling algorithm. A higher value specifies a higher priority. The value **1** is defined to indicate the lowest possible priority (a job which a priority-based scheduling algorithm shall pass over in favor of higher priority jobs). The value **100** is defined to indicate the highest possible priority. Priority is expected to be evenly or 'normally' distributed across this range. The mapping of vendor-defined priority over this range is implementation-specific. The omission of this attribute implies that the user places no constraints concerning priority on the scheduling of the print-job.
6. **jmJobProcessAfterTime** - process-after-time	Job-print-after This attribute specifies the calendar date and time of day after which the print-job shall become a candidate to be scheduled for printing. If the value of this attribute is in the future, the server shall set the value of the job's current-job-state to held and add the job-print-after-specified value to the job's job-state-reasons attribute and shall not schedule the print-job for printing until the specified date and time has passed. When the specified date and time arrives, the server shall remove the job-print-after-specified value from the job's job-state-reason attribute and, if no other reasons remain, shall change the job's current-job-state to pending so that the job becomes a candidate for being scheduled on printer(s). The server shall assign an empty value (see 9.1.2) to the job-print-after attribute when no print after time has been assigned, so that the job shall be a candidate for scheduling immediately.

jmQueueGroup (Q)	Corresponding ISO DPA specification
7. **jmJobMessageToOperator** - job-message-to-operator from submitting user or device	Job-message-to-operator This attribute carries a message from the user to the operator to indicate something about the processing of this print-job. This message, unlike the **job-start-message,** is not necessarily related to other job-scheduling attributes. The server shall make this message available to the operator when the job has been accepted.

442

8.4 The Completed Group - comparison with ISO DPA

jmCompletedGroup (C)	Corresponding ISO DPA specification
1. jmMIBInstanceIndex - a running index of Job Monitoring MIB instances supported by this printer or server.	
2. jmCompletedIndex - a running index of the jobs that have finished processing.	
3. **jmJobLocalId** - the job's identifier generated by the printer or server implementing this JM MIB	**Job-identifier** See below.

8.5 The Job Group - comparison with ISO DPA

jmJobGroup - Identification **(I)**	**Corresponding ISO DPA specification**
1. jmMIBInstanceIndex - a running index of Job Monitoring MIB instances supported by this printer or server.	

2. **jmJobLocalId** - the job's identifier generated locally by the server or printer implementing this JM MIB	Job-identifier This attribute provides the job-identifier for this job on the server. The server shall generate a job-identifier value that is unique on that server, but need not be unique across the distributed environment. The value of the **job-identifier** attribute shall be returned by the server as part of the **PrintResult** in the first Print operation for the job (see 8.2.1). The client shall pass its value as part of the **PrintArgument** in subsequent Print operations for the same job.
3. **jmJobDownstreamId -** Job downstream id (downstream from the server implementing this JM MIB)	Job-identifier-on-printer This attribute provides the job-identifier for this job to uniquely identify it on the printer or next server downstream, and it shall be assigned by that printer or server. The value of the **job-identifier-on-printer** attribute shall be the job-identifier that the printer or the next server downstream generates at the time the server submits the job to the printer or the next server downstream. If the printer or the next server downstream does not generate its own job identifier, the server shall leave the value of this attribute unspecified. NOTE – This attribute is intended for jobs submitted to other servers, such as a gateway to a legacy server, or to printers which have their own internal queue and assign their own job identifiers.
4. **jmJobTypes** - Job types (print, fax, scan, etc.) - bit vector to get multiple values in a single object	

5. **jmJobOwner** - Job owner (User name that originally submitting print job)	Job-owner This attribute supplies the name of the human owner of the print-job, i.e., the name of the user who submitted the job originally, not the user who most recently (re)submitted the job. The value of **job-owner** will often be the same as **job-originator**. The **job-owner** will be different from **job-originator** when the job has been submitted by the originator on behalf of the owner. This attribute is not to take the place of the security parameters or the access-and-accounting attributes. If this attribute is not specified, the value of **user-name** or **job-originator** should be used for any circumstances which require a value for **job-owner**.
6. jmDeviceIndex - the host resources index of the corresponding Printer MIB that the job was submitted to or has been assigned to be printed on by the server.	
7. **jmJobSourceChannel** - Source channel on which the job was submitted (index of channel row in the Printer MIB)	
8. **jmJobName** - Job name (assigned by job owner)	Job-name This attribute supplies a human readable string for the print-job. This string is used for naming the print-job in human-readable “free-form” fashion. This attribute is intended for enabling a user or the user’s application to convey a job name that may be printed on a start sheet, returned in a ListObjectAttributes result, or used in notification or logging messages. If this attribute is not specified, no job name is assumed, but implementation specific defaults are allowed, such as the value of the **document-name** attribute of the first document in the job.
9. **jmJobSubmissionTime** - Date/Time of job submission by job owner	Submission-time This attribute indicates the time at which the original Print request for this job was accepted by the first server. For all subsequent servers, this object/attribute shall be a string of zero digits indicating no submission time.
10. **jmJobComment** - Job comment	Job-comment This attribute supplies an arbitrary human-readable text string associated with the print-job. This attribute is intended for enabling a user to convey a text string that may be printed on a job start sheet, for example, in an implementation-dependent manner.

11. **jmJobDeviceNameRequested** - Device name (Device-specific name of device)	Printer-name-requested This attribute identifies the printer to be used for printing the job. The client shall specify the value of this attribute with the first invocation of the **Print** operation for the print-job as the explicit **printer-name** component of the **PrintArgument**, rather than as an attribute (see 8.2.1.1). NOTES 1 To cause a server to select a printer according to other attributes, the system administrator should define a logical printer that supports ALL of the physical printers supported by the server. 2 For the server that supports only a single printer, the logical printer name may be the same as the server name, as long as they cannot be confused for each other in the name service directory. 3 Initial-value-job objects should have the value of their **printer-name-requested** attribute specified as an empty value in order to indicate that no printer-name is defaulted.

447

jmJobGroup - Parameters (J)	**Corresponding ISO DPA specification**

jmJobGroup - Parameters (J)	**Corresponding ISO DPA specification**

12. jmJobTotalOctetsHigh - total octets to be processed in the job - high order 31 bits	**total-job-octets**
	This attribute indicates the size of the job in octets, including document and job copies.
	`total-job-octets ATTRIBUTE`
	`WITH ATTRIBUTE-SYNTAX cardinal64Syntax`
	`SINGLE VALUE`
	`::= id-att-total-job-octets`
	The server may update the value of this attribute after each document has been transferred to the server or the server may provide this value after all documents have been transferred to the server, depending on implementation. In other words, while the job is in the **pre-processing** state and when the job is in the **held** state with the **job-state-reasons** containing a **document-needed value**, the value of the **total-job-octets** job status attribute depends on implementation and may not correctly reflect the size of the job.
	In computing this value, the server shall include the multiplicative factors contributed by the (1) **copy-count** document attribute, (2) the **results-profile**.**job-copies** job attribute element and (3) multiple values of the **results-profile** job attribute, independent of whether the printer can process multiple copies of the job or document without making multiple passes over the job or document data and independent of the value of the **output** document attribute (**page-collate** vs. **no-page-collate**). Thus the server computation is independent of the printer implementation and shall be:
	1. Document contribution: Multiply each **copy-count** by the size of the document in octets.
	2. Add each document contribution together
	3. Job result contribution: Multiply the job size by the number **job-copies** in the result set.
	4. Add each job result contribution together
	Multiply the value by the number of values in the job's **result-profile** attribute.

13. jmJobTotalOctetsLow - total octets to be processed in the job - low order 31 bits; -2 if unknown	Low part of **total-job-octets.**

448

jmJobGroup - Status and Accounting (S)	**Corresponding ISO DPA specification**
14. **jmJobCurrentState** - Job state (**pending**, **processing**, **completed**, etc.)	Current-job-state This attribute identifies the current state of the job (pending, printing, held, etc.) The following job state standard values are defined:

449

Descriptive Name	Descriptor Text
unknown	The job state is not known, or is indeterminate.
pre-processing	The job has been created on the server by the **create-job** sub-operation of the print-request, but a print-request with a **TRUE** value for the **job-submission-complete** component of the PrintArgument has not yet been received and no document has started processing. The job maybe in the process of being checked by the server for attributes, defaults being applied, a printer being selected, etc.
held	The job is waiting to be released for scheduling for any number of reasons as specified by the value of the job's **job-state-reasons** attribute.
pending	The job's **job-submission-complete** attribute is **TRUE** since the server has received a print-request with the **job-submission-complete** parameter **TRUE** and the job is waiting to start processing on a printer.
processing	The server is processing the job, or has made the job ready for printing, but the output device is not yet printing it, either because the job hasn't reached the output device or because the job is queued in the output device or some other spooler, awaiting the output device to print it.
paused	The job has been paused as a result of a PauseJob operation.
interrupted	The job was interrupted by the InterruptJob request for an intervening job, and shall resume processing automatically once the intervening job has completed.
terminating	The job has been cancelled by a CancelJob request or aborted by the server and is in the process of terminating. The job's **job-state-reasons** attribute contains the reasons that the job is being terminated.
retained	The job is being retained at the server as a result of the job's **job-retention-period** being non-zero. The job has (1) completed successfully or with warnings or errors, (2) been aborted while printing by the server, or (3) been cancelled by the CancelJob request before or during processing. The job's **job-state-reasons** attribute contains the reasons that the job has been retained. While in the retained state, all of the job's document data (and resources, if any) shall be retained by the server; thus a job in the **retained** state could be reprinted, using some means outside the scope of ISO/IEC 10175-Part 1.

Descriptive Name	Descriptor Text
completed	The job has: (1) completed successfully or with warnings or errors, (2) been aborted by the server while printing, or (3) been cancelled by the **CancelJob** request, AND the job's: (1) **job-retention-period** was zero or has expired, or (2) **job-discard-time** has arrived. The job's **job-state-reasons** attribute contains the reason(s) that the job has been completed. While in the **completed** state, a job's document data (and resources if any) need not be retained by the server; thus a job in the **completed** state could not be reprinted. The length of time that a job may be in this state, before transitioning to **unknown**, is implementation-dependent. However, servers that implement the **completed** job-state shall retain, as a minimum, the following attributes for any job in the completed state: **job-identifier**, **job-owner**, **job-name**, **current-job-state**, **printers-assigned**, and **job-state-reasons**.

Print clients and DP-Servers shall be prepared to receive all the standard job states. DP-Servers are not required to generate all job states, only those which are appropriate for the particular implementation.

If a server implementation or policy is to start processing documents before the last print-request (with a **TRUE** value for the **job-submission-complete** parameter) and the value of the job's **job-scheduling** attribute is not **after-complete**, the server shall change the job's **current-job-state** from **pre-processing** directly to the **processing** state when the server begins processing any of the job's documents.

15. **jmJobStateReasons** - Job state reasons - additional information about the job state: reasons being held, additional completed information such as successful, warnings, or errors.	Job-state-reasons This attribute identifies the reason or reasons that the job is in the **held, terminating**, **retained**, or **completed** state. The server shall indicate the particular reason(s) by setting the value of the **job-state-reasons** attribute. When the job is not in any of these states, the server shall set the value of the **job-state-reasons** attribute to the empty set. The following [DPA] standard values are defined: **documents-needed, job-hold-set, job-print-after-specified, required-resources-not-ready, successful completion, completed-with-warnings, completed-with-errors, cancelled-by-user, cancelled-by-operator, aborted-by-system, logfile-pending ,** and **logfile-transferring.**

16. **jmJobOctetsCompletedHigh** - Octets completed -high order part	Octets-completed This attribute indicates the number of octets of the job that the printer(s) have completed printing. The server shall not reset its value during the processing of multiple copies of documents or the job. Since this attribute is intended to measure the progress of a job, the value shall include repeated pages due to multiple copies. The accuracy of this value is implementation-dependent. It may be approximated by the number of octets conveyed to the printer. This attribute may not be supported for all printers and all page description languages. The value of this attribute shall be **0** if printing has not started for this job.
17. **jmJobOctetsCompletedLow** - Octets completed -low order part	Low part of **octets-completed**
18. **jmJobStartedProcessingTime** - Date/Time of day job started processing on device	Started-printing-time This attribute indicates the time at which this job started printing.
19. **jmJobCompletionTime** - Date/Time of day job finished using the device	Completion-time This attribute indicates the time at which this job completed. Providing this time is useful for jobs which are retained after printing.
20. **jmJobAccountName** - Account Name	Accounting-information This attribute specifies information required by accounting services (e.g. the account to be charged for any services rendered). Accounting information is intended to be interpreted by an accounting system, and may be opaque to the print service.

8.6 The Resource Group - comparison with ISO DPA

jmResourceGroup (R)	Corresponding ISO DPA specification
1. jmMIBInstanceIndex - a running index of Job Monitoring MIB instances supported by this printer or server.	
2. **jmJobLocalId -** the job's current identifier generated locally by the server or printer implementing this JM MIB	**job-identifier** See above.
3. jmResourceIndex - a running index of the resources requested and/or used by the job.	
4. **jmResourceType -** Resources required/used (table):	
a) **documentName(3) -** Document name(s) (or file-names)	Document-name This attribute supplies a human readable string for the document. This string is used for naming the document in a human-readable "free-form" fashion. This attribute is intended for enabling a user or the user's application to convey a document name that may be printed on a start sheet, returned in a ListObjectAttributes result, or used in notification or logging messages. If this attribute is not specified, no document name is assumed, but implementation specific defaults are allowed, such as the simple-name part of the value of the **document-file-name** attribute. It is suggested, however, that the server not supply additional text for this attribute when printing its value (e.g. on a start sheet). This string only has meaning to the clients and can therefore take several forms, e.g. the name of a mail folder, name of a revisable document, the file specification minus the file path, the title of a document, etc.
b) **jobCopiesRequested(4) -** Number of job copies requested	job-copies Total number of job copies in the job, i.e., number of job copies summed across the job-result-sets. Whether job copies are collated or not depends on implementation. NOTE - In ISO DPA, job-copies is a separate value for each job result set, not the summation. But it didn't seem worth the effort to make job-copies a table for the MIB.
c) **jobCopiesProduced(5) -** Number of job copies produced	total-job-copies Total number of job copies in the job, i.e., number of job copies summed across the job-result-sets. Whether job copies are collated or not depends on implementation. NOTE - In ISO DPA, job-copies is a separate value for each job result set, not the summation. But it didn't seem worth the effort to make job-copies a table for the MIB.

jmResourceGroup (R)	Corresponding ISO DPA specification
d) **documentCopiesRequested(5) -** Number of document copies requested	copy-count This attribute specifies the number of copies of the documents, or of the selected pages of the document, to be printed. In ISO DPA, there is a **copy-count** attribute for each document in the job. The proposal here is to have a single per-job count of the number of copies of documents, in order to avoid a per-document table.
e) **documentCopiesProduced(6) -** Number of document copies produced	copies-completed In ISO DPA, there is a **copy-count** attribute for each document in the job. The proposal here is to have a single per-job count of the number of copies of documents, in order to avoid a per-document table.
f) **sides(7) -** Number of sides requested (one-sided, two-sided)	Sides This attribute specifies the number of printable surfaces of the medium to be imaged.

jmResourceGroup (R)	Corresponding ISO DPA specification
g) **interpreters(8) -** PDLs requested/used	Document-format This attribute identifies the overall print document format used for the document. It consists of three elements, a **document-format**, a **document-format-variants** and a **document-format-version**. The latter two elements are optional. The **document-format** element identifies a particular family of document formats, of which there may exist several versions or variants. The **document-format-variants** and **document-format-version** elements identify a specific instance of a document format. The variant refers to a particular functional subset of a format. For example, the format PostScript has variants of level 1 and level 2, and the format PCL has several variants, including PCL4 and PCL5. The version distinguishes among successive releases of the same basic format and variant. For example, successive versions of Xerox Interpress include versions 2.0, 2.1, 3.0, 3.1, etc. Put in a separate table so can have multiple values, one for each document.

jmResourceGroup (R)	Corresponding ISO DPA specification
h) **physicalDevices(9) -** physical devices requested/used	Physical-printers-requested This attribute identifies the physical printer or printers that shall be the only candidates for printing the job. **physical-printers-requested ATTRIBUTE** **WITH ATTRIBUTE-SYNTAX simpleNameSyntax** **MULTI VALUE** **::= id-att-physical-printers-requested** The value of this attribute shall be a subset of the value of the **physical-printers-supported** for the logical printer specified as the explicit **printer-name-requested** component of the **PrintArgument** (else the server shall reject the job or ignore this attribute, depending on whether this attribute is specified as compulsory or non-compulsory). NOTE – This attribute is provided for implementations and/or system administration policies that require that clients submit only to logical printers, i.e., that require that the **printer-name-requested** component of **PrintArgument** be a logical printer, since ISO 10175 permits **printer-name-requested** to be a logical or physical printer. Printers-assigned[6] This attribute identifies the physical device or devices to which this job has been assigned, if any. When the job is first submitted and the server has not yet assigned any devices to the job, the **SEQUENCE** shall be empty. If the server intends to use a single device for the job, and the server has assigned a device to the job, the **SEQUENCE** shall contain just that device. If a server has split the job into multiple pieces and assigned each piece to a different device, the **SEQUENCE** shall contain n elements, one for each assigned device. A job with multiple job-result-sets is an example of a job that would be easy to split into multiple pieces. A **SEQUENCE** with no elements shall be returned if this attribute is supported, but this job has not yet been assigned to any physical device. The number of elements in the **SEQUENCE** for this attribute shall be the same as the number of elements in the **SEQUENCE** for the associated job attribute **device-state-of-devices-assigned**. In addition, the *i*th element of each of this associated attributes shall be a value that pertains to the printer named by the *i*th element of **devices-assigned**. The **devices-assigned** value shall not be the same as the printer requested by the user if the job's **printer-name-requested** attribute specified a logical printer that supports one or more different physical printers. The **devices-assigned** value might differ also if the job has been re-assigned by an operator to ensure successful completion of the job, allowing the user to find out where a job has been re-assigned (when necessary). The value of the job's **devices-assigned** attribute shall remain after the job has completed, so that users can determine the physical devices(s) on which the job was printed.

jmResourceGroup (R)	Corresponding ISO DPA specification
i) **faxPhoneNumbers(10) -** FAX phone number(s) requested/used	
j) **impressions(11) -** Impressions (sides) completed	Impressions-completed This attribute indicates the number of impressions that the printer engine(s) have placed on the media for the job. See the note in the **pages-completed** attribute for the relationship of the **pages-completed**, **impressions-completed** and **media-sheets-completed** attributes. The server shall not reset its value during the processing of multiple copies of documents or the job. Since this attribute is intended to measure the progress of a job, the value shall include repeated pages due to multiple copies. When the job completes, this attribute should contain the value of the total number of impressions that the printer made for the print-job. The accuracy of this value is implementation-dependent. It is expected that the value reported is never greater than the actual value. This attribute may not be supported for all printers and all page description languages. The value of this attribute shall be 0 if printing has not started for this job.
k) **sheets(12) -** Sheets completed	Media-sheets-completed This attribute indicates the number of sheets of media that the printer(s) have completed printing for the job. See the note in the **pages-completed** attribute for the relationship of the **pages-completed**, **impressions-completed** and **media-sheets-completed** attributes. The server shall not reset its value during the processing of multiple copies of documents or the job. Since this attribute is intended to measure the progress of a job, the value shall include repeated pages due to multiple copies. When the job completes, this attribute should contain the value of the total number of sheets of media used for the print-job. The accuracy of this value is implementation-dependent. It is expected that the value reported is never greater than the actual value. This attribute may not be supported for all printers and all page description languages. The value of this attribute shall be 0 if printing has not started for this job.
l) **processingTime(13) -** Processing time so far	Processing-time This attribute indicates how long an individual job has been processing [in seconds].
m) **processingMessage(14) -** Processing Messages	
5. **jmResourceName** - resource required/usage name	
6. jmResourceUnits - resource required/used usage-unit	

jmResourceGroup (R)	Corresponding ISO DPA specification
7. **jmResourceAmount** - resource amount requested/used; -2 - unknown	

9. APPENDIX B - Mapping from Job Submission Protocols to JMP Objects/attributes

The JMP object/attributes are divided into the following categories:

1. Job Identification (I)
2. Job Parameters (P)
3. Job Status and Accounting (S)

The following table lists each JMP object/attribute and indicates in each column whether there is a corresponding attribute in the indicated job submission protocol. Objects marked with a leading ** are ones that were not in our original brainstorming list. Combined objects from the original 69 are not marked with **, since they were counted in the job protocol survey.

The first column contains the MIB name followed by a descriptive name for the object/attribute that is applicable to both MIB and MIF. Eventually, we will need to pick MIB names for the MIB which will have a prefix of "jm" and mixed case with each word starting with an upper case letter and no intervening spaces or hyphens. For the MIF the descriptive name will have intervening spaces and no hyphens. We will keep the names in this section the same as the specification section.

The **Conf**. column specifies the conformance:

M means **Mandatory** for conformance to this MIB specification

CM means **Conditional** Mandatory (for spooling systems, and systems with day and time clocks, etc.).

The **Cardinality** columns contains:

1 meaning there is only **one** of these objects per job, so that the object can be in a table that is indexed by **jmMIBInstanceIndex** and **hrJobCurrentId**.

n meaning that there may be **more than one** of these objects per job, so that that the object must be in another table that in indexed by **jmMIBInstanceIndex**, **hrJobLocalId**, and a running instance index

Appendix B - Mapping from Job Submission Protocols to Job Monitoring MIB

Job Identification (I)	Conf.	Cardinality	ISO DPA	Appl e PAP	IPDS	LPR /LPD	NDPS	PJL	PSERVER	SMB	TIPSI
1. **jmJobClientId** - Job client id (on the original client)	**M**	**1**	x	x		x	x		x	x	*x*
2. **jmJobUpstreamId** - Job upstream id (upstream from the server implementing this JM MIB)	**CM**	**1**	x		x	x	x	x		x	
3. **jmJobLocalId** - Job current id (on the server implementing this JM MIB)	**M**	**1**	x		x	x	x	x		x	
4. **jmJobDownstreamId** - Job downstream id (downstream from the server implementing this JM MIB)	**CM**	**1**	x		x		x	x	x		**x**
5. **jmJobTypes** - Job types (print, fax, scan, etc.) - bit vector to get multiple values in a single object	**M**	**1**			x		x			x	
6. **jmJobOwner** - Job owner (User name that originally submitting print job)	**M**	**1**	x	x	x		x		x	x	*x*
7. **jmJobSourceChannel** - Source channel (index of channel row in Printer MIB)	**M**	**1**		x		x					**x**
8. **jmJobName** - Job name (assigned by job owner)	**M**	**1**	x		x		x	x	x		
9. **jmJobSubmissionTime** - Date/Time of job submission by job owner	**CM**	**1**	x				x		x	x	
10. **jmJobComment** - Job comment	**M**	**1**	x				x	x	x		*x*
11. **jmJobDeviceNameRequested** - Device name (Device-specific name of device)	**M**	**1**	x		x		x				**x**

Appendix B - Mapping from Job Submission Protocols to Job Monitoring MIB

476

Job Parameters (J)	Conf.	Cardinality	ISO DPA	Apple PAP	IPDS	LPR/LPD	NDPS	PJL	PSERVER	SMB	TIPSI
1. **jmJobPriority** - Job priority	**CM**	**1**	x				x				*x*
2. **jmJobProcessAfterTime** - process-after-time	**CM**	**1**	x								
3. **jmJobMessageToOperator** - job-message-to-operator from submitting user or device	**CM**	**1**	x								
4. jmJobTotalOctetsHigh - total octets to be processed in the job - high order 31 bits	**M?**	**1**	x								
5. jmJobTotalOctetsLow - total octets to be processed in the job - low order 31 bits	**M?**	**1**	x								

Appendix B - Mapping from Job Submission Protocols to Job Monitoring MIB

477

Job Status and Accounting (S)	Conf.	Cardinality	ISO DPA	Appl e PAP	IPDS	LPR /LPD	NDPS	PJL	PSERVER	SMB	TIPSI
1. **jmJobCurrentState** - Job state (held, pending, processing, completed, etc.)	**M**	**1**	x	x		x	x	x		x	**x**
2. **jmJobStateReasons** - Job state reasons - additional information about the job state: reasons being held, additional executing information such as device(s) needs attention, additional completed information such as successful, warnings, or errors. (whether bits or text string is TBD).	**M**	**1**	x		x		x	**x**			**x**
3.											
4.											
5.											
6. **jmJobOctetsCompletedHigh** - Octets completed -high order part	**M**	**1**	x				x				**x**
7. **jmJobOctetsCompletedLow** - Octets completed -low order part	**M**	**1**	x				x				**x**
8. **jmJobStartedProcessingTime** - Date/Time of day job started processing on device	**CM**	**1**	x				x				**x**
9. **jmJobCompletionTime** - Date/Time of day job finished using the device	**CM**	**1**									
10. **jmJobAccountName** - Account Name	**M**	**1**	x				x				**x**
11. **jmResourceType** - Resources required/used (table):	**M**	**n**									
a) Document name(s) (or file-names)	**CM**	**n**	x	x	x	x	x		x		x
b) Number of job copies requested	**CM**	**1**	x				x	x	x		
c) Number of job copies produced	**CM**	**1**	x								
d) Number of document copies requested	**CM**	**1**	x				x	x	x		

Appendix B - Mapping from Job Submission Protocols to Job Monitoring MIB

Job Status and Accounting (S)	Conf.	Cardinality	ISO DPA	Apple PAP	IPDS	LPR/LPD	NDPS	PJL	PSERVER	SMB	TIPSI
e) Number of document copies produced	**CM**	**1**	x								
f) Number of sides requested (one-sided, two-sided)	**CM**	**1**	x		x		x	x			*x*
g) PDLs requested/used	**M**	**n**	x			x	x	x			*x*
h) physical devices requested/used	**CM**	**n**	x		x		x	x	x		x
i) FAX phone number(s) requested/used	**CM**	**n**									
j) Impressions (sides) completed	**CM**	**1**	x				x	x			
k) Sheets completed	**M**	**1**	x				x				
l) Processing time so far	**M**	**1**	x				x				
m) Processing Messages	**CM**	**n**									
12. **jmResourceName** - resource required/usage name	**M**	**n**									
13. jmResourceUnits - resource required/used usage-unit	**M**	**n**									
14. jmResourceAmount - resource amount requested/used	**M**	**n**									

478