Subj: Minutes, IETF/DMTF PWG Job Monitoring Project (JMP), November 8, 1996, New Orleans

From: Tom Hastings
Date: 11/08/96
File: jm961108.doc .pdf .ps .txt

The sections follow the agenda outline. Action items are indicated as:

ACTION ITEM (name): ....

The action items are repeated at the end of the minutes for convenience. They will be covered on the agenda for the next meeting.

There are 5 issues indicated by:

<u>ISSUE nn: ....</u>

Reviewers are encouraged to comment on the issues. If unresolved, the issues will be carried over in the updated Job Monitoring MIB Spec and MIB, until resolved.

Attendees:

Ron Bergman - Dataproducts
Keith Carter, IBM
Andy Davidson - Tektronix
Lee Farrell - Canon Info Systems
Tom Hastings - Xerox Corp.
Bob Herriot, Sun Microsystems
Scott Isaacson, Novell
David Kellerman - Northlake Software
Rick Landau, Digital
Harry Lewis, IBM
Jay Martin, Underscore
Bob Pentecost - HP
David Roach - Unisys
Bob Setterbo, Adobe
Mike Timperman - Lexmark
Randy Turner, Sharp
Bill Wagner - DPI
Jim Walker - DAZEL Corp.
Chris Wellens, InterWorking Labs
Don Wright - Lexmark
Lloyd Young - Lexmark
Atsushi Yuki, Kyocera

1. Presentation by Keith Carter on OS/2 WARP Job Monitoring

The group was pleased to have an OS representative, since we have been trying to attract participation from the OS venders. Keith Carter presented the OS/2 WARP output device architecture and how job monitoring is done in OS/2 WARP. The concepts of "Protocol Converter" and "Port Driver" are separate and distinct. The Port Driver permits bi-directional support of printers.

He also presented some comments on the current Job Monitoring MIB Specification. OS/2 WARP supports many of the objects/attributes that are in the current Job Monitoring MIB specification. He indicated that OS/2 WARP might implement a Job Monitoring MIB agent, so that his comments were directed from that point of view:

1. The JMP agreed to change the data type of the **jmJobCurrentId** from a 255 byte printable ASCII string to a 32-bit integer.

2. The JMP agreed to change the name from **jmJobCurrentId** to **jmJobLocalId** to make it clearer that the object is the job-identifier generated locally by the printer or server that accepted the job over the job submission channel.

3. The JMP did *not* agree to add a text object to hold the server/queue/printer name that is part of the **job-identifier** in some systems. The reasoning was that the management application knows what agent it is talking to, so that the name of the server/queue/printer that the agent is instrumenting would be redundant. Also the **jmJobDeviceNameRequested** object helps distinguish jobs in a server that supports multiple printers.

4. Keith (and Scott - see below) proposed and the JMP accepted adding a **jmQueueNumberOfInterveningJobs** object to the **jmQueueTable** that is the number of jobs ahead of the job.

5. Keith proposed adding a **jmJobNotify** object that contained the address of the user to be notified if the job had trouble. The JMP felt that such an object was *not* needed in a MIB that monitors jobs.

6. Bob Herriot suggested clarifying that the **jmJobOwner** contain the authenticated owner of the job. This will help an operator and the accounting system know which user is interested in the job and which user should be charged, if the **jmJobAccountName** is an empty string.

7. Keith proposed adding **jmJobTotalPagesSpooled**, **jmJobTotalPagesSent** (to printer), and **jmJobTotalPagesPrinted**. For applications using the OS/2 GDI, the number of pages is known by the system. However, for files that are submitted by other means the number of pages is not known by the system (but only by the printer interpreter). We did not fully discuss this proposal. Also we did *not* fully discuss whether such objects should be distinct objects or should be enums in the **jmResourceTable.** Such a decision depends on whether a management application needs to poll these objects without getting the entire **jmResourceTable**. The number of impressions completed and the number of sheets completed are already enums in the **jmResourceType** object in the **jmResourceTable**.

8. Keith suggested changing the conformance of the **jmJobAccountName** object from Mandatory to Conditionally Mandatory since not all job submission protocols have a concept of a job account name that is distinct from the **jmJobOwner**. Instead, the JMP agreed that string objects, such as **jmJobAccountName**, shall always be implemented and that an agent shall return an empty string (a string with zero length, not an ASN.1 NULL) for implementations that do not have the concept of an account name or when the job does not specify an account name.

2. Job Submission Models

1. In reviewing the new goal agreed to in New York and the two configurations (1 and 2a), there was agreement that the Job Monitoring MIB is being designed for a single instance of a Job Monitoring MIB agent, rather than attempting to design the MIB to support multiple agents being queried from the same management application.

   ACTION ITEM (Tom Hastings): Delete the other more complicated pictures from the specification, leaving just configurations 1 and 2a.

2. Randy Turner suggested that if the job model includes the concept of multiple documents per job, then the data structures and tables should reflect that concept. Randy felt that having a separate table with an extra index would not be much of an extra burden, since implementations that only had one document per job, would only require an extra byte in the OID for per-document objects. There has been concern that providing separate document tables with an additional index of document-sequence-number would complicate the MIB. Also the **jmResourceTable** allows for multiple resources whether per job or per document, so that those implementations that do have multiple documents can represent multiple resources. Also the resource table allows resources to be represented when there are multiple resources per document, such as media, fonts, even PDLs, etc..

3. Ron Bergman's comments

1. The JMP agreed with Ron's definitions of spooling and queuing. That *spooling* is the act of a printer or server of (1) accepting jobs, (2) writing the job's attributes and document data on secondary storage and (3) ordering (*queuing*) the jobs for the purpose of scheduling them to be processed. That *queuing* is a subset of *spooling*, namely, ordering the jobs for the purpose of scheduling them to be processed; the job's attributes and document data are stored elsewhere and fetched as part of processing the job.

2. Ron suggested that we don't need the upstream and downstream identifiers. The JMP agreed to get rid of two of the four identifiers: **jmJobClientId** and **jmJobDownstreamId**. Ron feels that the **jmJobClientId** is valuable. In fact, 7 of the 9 job submission protocols have a **jmJobClientId**. The JMP felt that the other objects in the MIB would help a human locate his jobs amongst a list of jobs. The ISO DPA protocol had a **job-client-id** attribute, but none of the ISO DPA client implementations actually submit the **job-client-id** attribute, correct? The idea in ISO

DPA was to allow, but not require, a client to supply an identifier generated by the client.

ACTION ITEM (Ron): submit a proposal for reworking the job identification objects and sent to the pwg@pwg.org DL. Depending on the comments, hold a teleconference before the next meeting.

3. The JMP agreed to change the name from **jmJobCurrentId** to **jmJobLocalId** to make it clearer that the object is the job-identifier generated locally by the printer or server that accepted the job over the job submission channel. See above.

4. There was also the suggestion to change the name of the **jmJobDownstreamId** to **jmJobPrinterId**, but my notes don't show whether we agreed to this or not. I'm confused about this object, since if the MIB agent is in the printer then the **jmJobPrinterId** would be the same value as the **jmJobLocalId**, wouldn't it? And if the agent is in the server that is connected to the printer, then of what value in our MIB is the id that the printer generates and returns to the server when the server submits the job to the printer, since we've agreed not to try to support access to different agents from the same management application or end-user client?

ISSUE 01 - 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?

4. Review of Scott's proposal for structure of the MIB

We reviewed Scott's proposal for the structure of the MIB (**jm951005.doc**) that Tom Hastings had annotated with comments (**jm961107.doc**). The following agreements were reached:

1. We agreed that the **jmJobTable** should be doubly indexed. The first index allows an implementation to have multiple job monitoring MIBs and so the first index will be a job MIB instance index (**jmMIBInstanceIndex**). However, instead of the proposed **hrDeviceIndex** from the host resource MIB, it should just be an instance index, since representing a logical printer in a server using an **hrDeviceIndex** would be a problem because there wouldn't be a printer MIB for a logical printer, only for a physical printer. Therefore, the Job Monitoring MIB will also have table that enumerates the instances so that a management application can determine which instance(s) to examine. That table will also need an object that is the name of the instance.

ISSUE 02: Is the name **jmMIBInstanceIndex** O.K., or should it be something like **jmJobDeviceIndex**?

2. The second index of the **jmJobTable** will be the **jmJobLocalId** which now is a Integer32, instead of a text string, making the OIDs much shorter.

3. The JMP accepted the proposal for a new group that is indexed only by Job MIB instance: **jmGeneralGroup** with the following new object: **jmGeneralJobRetentionPolicy** (renamed from **jmGeneralPolicy** and **jmGeneralJobRetentionDefault**) object that indicates the site-settable value for the

time in seconds that jobs are retained after they are completed for the accounting programs to copy the data.

ISSUE 03: Ok that the **jmGeneralGroup** is indexed by **jmMIBInstanceIndex** so that there can be different values for different printers?

4. Additional new objects for the **jmGeneralGroup**: **jmGeneralMaxNumberOfJobs** in the **jmJobTable** and **jmGeneralCurrentNumberOfJobs** in the **jmJobTable**.

5. The JMP agreed to add the new object: **jmSpooling** to the **jmGeneralGroup** (instead of the **jmSpoolingGroup**), to change the name from **jmSpooling** to **jmQueuing** and to make **jmQueuing** a simple Boolean, rather than **PresentOnOff**, since it seemed unlikely for queuing to be turned off in an implementation.

6. The JMP agreed to add the **jmQueuingSchedulingAlgorithm** object as a scalar to the **jmGeneralGroup** which has the enums: **none(3)[**when no queuing]**, fifo(4)**, **shortestJobFirst(5)**.

7. The JMP agreed to add the **jmQueueTable** which contains the **jmJobLocalId** values for the jobs that have not completed. The JMP agreed to move the following objects from the **jmJobTable** to the **jmQueueTable**, since their values are not needed, even by the accounting system, after the jobs have finished processing: **jmJobPriority** and **jmJobProcessAfterTime**. Probably we should move the **jmJobMessageToOperator** from the **jmJobTable** to the **jmQueueTable** too**.**

8. The JMP agreed to add the new **jmQueueNumberOfInterveningJobs** to the **jmQueueTable** which contains the number of jobs ahead of this job. (Also suggested by Keith Carter).

9. The JMP agreed to add the **jmCompletedTable** which contains the **jmJobLocalId** values for the jobs that have finished being processed.

10. The JMP agreed that there is not a need to have a **jmDocTable** which is indexed by document sequence number for implementations that have multiple documents per job. See discussion of Job Submission Models above.

11. The JMP agreed to have a **jmResourceTable** which is indexed by: (1) **jmMIBInstanceIndex**, (2) **jmJobLocalId**, and (3) **jmResourceIndex**.

12. The JMP rejected the suggestion to have **jmDevicesAssigned** and **jmDeviceStateOfDevicesAssigned**. Instead, a single new job state: **needsAttention()** will be added to the **jmJobCurrentState** object. A management application will have to use other means to find out what is wrong with the device that the job is using, such as consulting with the associated Printer MIB. Also it was felt that end-users only need to know that the device that their job is using when their job is processing needs attention, not what kind of attention. The operator has the Printer MIB to tell him what the printer needs, in gory detail.

ISSUE 04 - We need to explain how the job monitoring MIB management application locates the corresponding Printer MIB from the objects in the Job Monitoring MIB. Ok for it to have to search the Resources Table for the **physicalDevices(8)** enum to

find the corresponding **hrDeviceIndex**? Or should there be a distinct object so that a management application can determine the Printer MIB more quickly?

ISSUE 05 - We also need to explain how an operator using the Job Monitoring MIB uses the objects displayed to locate the corresponding Printer MIB and to learn the name of the printer, in case there isn't a Printer MIB implementation.

13. The JMP also rejected the **jmJobStateReasonsMessage** table and even as a single per-job object. Again, the user only needs to know which job state his job is in.
14. The JMP rejected the proposal to add a **jmAccountTable** to contain account entries. The resource table seem sufficient.
15. The JMP rejected adding an error and warning counter and message table. Instead the JMP agree to keep the **processingMessage(13)** enum in the **jmResourceTable** for the purposes of logging messages encountered during the processing of the job.

ACTION ITEM (Scott Isaacson): Update the picture of the tables for the MIB.

5. Review of object/attribute specifications

We ran out of time to review the specifications of each object/attribute that is proposed for the MIB (the text before the ISO DPA attribute name).

ACTION ITEM (all): Make comments via the **pwg@pwg.org** DL about the object specifications (the text before the ISO DPA attribute name) in the current draft spec (jmp-spec.doc .pdf).

6. Collected Action Items

The action items are repeated here for convenience and for tracking at the next JMP meeting:

1. **ACTION ITEM (Tom Hastings)**: Draft a definition for **job** based on previous meeting agreements and send to the DL. Also **document** and concept of **proxy**.
2. **ACTION ITEM (Tom Hastings)**: Contact Martin Kirk of X/Open to see if the PSIS draft can be made available to the PWG. If not the whole document, at least the Appendices that document the LPR/LPD extensions that have been implemented by DEC, HP, IBM, Sun, and Xerox.
3. **ACTION ITEM (Tom Hastings)**: Delete the other more complicated pictures from the specification, leaving just configurations 1 and 2a.
4. **ACTION ITEM (Ron)**: submit a proposal for reworking the job identification objects..
5. **ACTION ITEM (Scott Isaacson)**: Update the picture of the tables for the MIB.
6. **ACTION ITEM (*all*)**: Make comments via the **pwg@pwg.org** DL about the object specifications (the text before the ISO DPA attribute name) in the current draft spec (jmp-spec.doc .pdf).

7. **ACTION ITEM (Ron Bergman):** Contact Diedre about becoming chartered as an IETF working group.

7. Completed Action Items

The following action items were completed:

1. **ACTION ITEM (Tom Hastings and Ron Bergman)**: Rework the security goal along the above agreements and send to the DL for comment.
   DONE

2. **ACTION ITEM (Tom Hastings and Ron Bergman)**: Rework the charter to make it more generic, but with some print-specific parts as well. Propose to the DL.
   DONE

3. **ACTION ITEM (*responsible persons* indicated in the job submission protocol table):** Map the brainstorming information objects/attributes to their job submission protocol, so that we can weed out information objects that do not have wide support in job submission protocols.
   DONE.

4. **ACTION ITEM (Harry Lewis)**: contact the DMTF OS person (Paul A Rocio?) for help in getting OS/NOS vendor participation, since the DMTF has OS developers participating.
   DONE. - Keith Carter from OS/2 WARP presented their comments on our work on 11/08/96.

8. Object/Attribute totals

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

Group	Table	Description	No. of objects	Conformance
jmGeneralGroup	N/A	General attributes that apply to all jobs in the MIB instance.	5	Mandatory
jmQueueGroup	**jmQueueTable**	Ordered list of jobs that have *not* finished and job attributes that only matter until the job has finished processing.	7	Conditionally Mandatory
jmCompletedGroup	**jmCompletedTable**	Ordered list of jobs that have finished processing.	2	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.	5	Mandatory
		Mandatory Totals:	32	
		Conditionally Mandatory Totals:	7	
		Totals:	39	

9. List of objects/attributes for the Job Monitoring MIB/MIF

NOTE - the revision marks indicate the changes agreed to at the PWG meeting on 11/8/96 in New Orleans.

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 **DataType** column indicates the data type of the object. Enums are given distinct names that start with a capital letter.

The **Conformance** 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 **jmMIBInstance~~hrDeviceIndex~~** and **hrJobLocal~~Current~~Id**.
- **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 **jmMIBInstance~~hrDeviceIndex~~**, **hrJobLocal~~Current~~Id**, and a running instance index

The **Protocols** column in the number of job submission protocols that this object/attribute appears out of our su~~e~~rvey of 9 job submission protocols. The 9 job submission protocols are: **ISO DPA, Apple PAP, IPDS, LPR/LPD, NDPS, PJL, PSERVER, SMB,** and **TIPSI.**

9.1 The General Group (Mandatory)

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

1. **jmMIBInstanceIndex -** a running index of MIB instances supported by this printer or server. If only one MIB instance is supported, the value of this index shall be 1.

General (G)	DataType	Conformance	Cardinality	Protocols
1. jmMIBInstanceIndex - a running index of MIB instances supported by this printer or server.	**Index**	**M**	**1**	
2. **jmGeneralJobRetentionPolicy** - time in seconds that jobs are retained after completion.	**Integer32(0..)**	**M**	**1**	
3. **jmGeneralMaxNumberOfJobs** - the maximum number of job; -1 means no limit.	**Integer32(0..)**	**M**	**1**	
4. jmGeneralCurrentNumberOfJobs - the total number of jobs currently in the Job Table (pending and completed).	**Integer32(0..)**	**M**	**1**	
5. jmGeneralQueuing - a Boolean. **TRUE** means that the printer or server can accept more than one job at a time and, therefore, that the conditionally mandatory **jmQueueGroup** is implemented.	**Boolean**	**M**	**1**	

9.2 The Queue Group (Conditionally Mandatory)

The **jmQueueGroup** is a group that is implemented only by servers and printers that support job queuing and so is conditionally mandatory. The **jmQueueGroup** consists of objects/attributes that are *not* needed after the job has completed processing. The **jmQueueGroup** is made up of the **jmQueueSchedulingAlgorithm** object indexed only by **jmMIBInstanceIndex** and the **jmQueueTable** which is an ordered list of jobs that have not completed processing.

jmQueueGroup (Q)	DataType	Conformance	Cardinality	Protocol
1. jmQueueSchedulingAlgorithm- a type 2 enum indicating the type of scheduling algorithm being used by this server or printer: **fifo(3), shortestJobFirst(4)**	enum	**CM**	**1**	

The **jmQueueTable** is indexed by:

1. **jmMIBInstanceIndex -** a running index of MIB instances supported by this printer or server. If only one MIB instance is supported, the value of this index shall be 1.
2. **jmQueueIndex** - a running index of the jobs that have *not* finished processing.

jmQueueEntry (Q)	DataType	Conformance	Cardinality	Protocol
1. jmQueueIndex - a running index of the jobs that have *not* finished processing.	Index	**CM**	**1**	
2. **jmQueueJobLocalId** - a copy of the job's identifier (**jmJobLocalId**) generated locally by the printer or server implementing this JM MIB	Integer32(0..) ~~A(255)~~	**CM**	**1**	6
3. **jmQueueNumberOfInterveningJobs** - the number of jobs in front of this job	Integer32(0..)	**CM**	**1**	1
4. **jmQueue~~Job~~Priority** - Schedling~~Job~~ priority for the job.	Integer32(0..100)	**CM**	**1**	3
5. **jmQueue~~Job~~ProcessAfterTime** - date/time after which the job is to become a candidate for processing.	GenTime	**CM**	**1**	1
6. **jmQueue~~Job~~MessageToOperator** - job-message-to-operator from submitting user or device	T(255)	**CM**	**1**	1

9.3 The Completed Group (Mandatory)

The **jmCompletedGroup** consists *entirely* of the **jmCompletedTable** which is an ordered list of the job that have completed processing. The **jmCompletedTable** is indexed by:

1. **jmMIBInstanceIndex -** a running index of MIB instances supported by this printer or server. If only one MIB instance is supported, the value of this index shall be 1.
2. **jmCommpletedIndex -** a running index of the jobs that have finished processing.

Completed (C)	DataType	Conformance	Cardinality	Protocol
1. jmCompletedIndex - a running index of the jobs that have finished processing.	Index	**M**	**1**	
2. **jmCompletedJobLocalId** - a copy of the job's identifier (**jmJobLocalId**) generated by the printer or server implementing this JM MIB	Integer 32(0..) ~~A(255)~~	**M**	**1**	6

9.4 The Job Group (Mandatory)

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. If only one MIB instance is supported, the value of this index shall be 1.
2. **jmJobLocalId -** the job identifier that was generated locally by the server or printer that accepted the job.

jmJobEntry - Identification (I)	DataType	Conformance	Cardinality	Protocol
1. ~~**jmJobClientId** - Job client id (on the original client)~~	~~T(255)~~	~~**M**~~	~~**1**~~	~~7~~
2. ~~**jmJobUpstreamId** - Job upstream id (upstream from the server implementing this JM MIB)~~	~~A(255)~~	~~**CM**~~	~~**1**~~	~~6~~
3. **jmJobLocal~~Current~~Id** - the ~~J~~job's ~~current~~ identifier generated locally by ~~(on~~ the server or printer implementing this JM MIB~~)~~	Integer 32(0..) ~~A(255)~~	**M**	**1**	6
4. **jmJobDownstreamId** - Job downstream id (downstream from the server implementing this JM MIB)	Integer 32(0..) ~~A(255)~~	**CM**	**1**	6
5. **jmJobTypes** - Job types (print, fax, scan, etc.) - bit vector to get multiple values in a single object	enum encoded as bits	**M**	**1**	3
6. **jmJobOwner** - Job owner (User name that originally submitting print job)	T(63~~255~~)	**M**	**1**	7
7. **jmJobSourceChannel** - Source channel (index of channel row in Printer MIB)	PrtChannelIndex	**M**	**1**	3
8. **jmJobSourceChannelInformation** - Job Channel Information	T(255)	**M**	**1**	3
9. **jmJobName** - Job name (assigned by job owner)	T(63~~255~~)	**M**	**1**	5
10. **jmJobSubmissionTime** - Date/Time of job submission by job owner	DateAndTime	**CM**	**1**	4

jmJobEntry - Identification (I)	DataType	Conformance	Cardinality	Protocol
11. **jmJobComment** - Job comment	T(63~~25~~ ~~5~~)	**M**	**1**	5
12. **jmJobDeviceNameRequested** - Device name (Device-specific name of device)	T(63~~25~~ ~~5~~)	**M**	**1**	4

Job Group - Parameters (J)	DataType	Conformance	Cardinality	Protocol
14. jmJobTotalOctetsHigh - total octets to be processed in the job - high order 31 bits	Integer 32	**M~~?~~**	**1**	1
15. jmJobTotalOctetsLow - total octets to be processed in the job - low order 31 bits; -2 if unknown	Integer 32	**M~~?~~**	**1**	1

Job Group - Status and Accounting (S)	DataType	Conformance	Cardinality	Protocol
16. **jmJobCurrentState** - Job state (~~**held**,~~ **pending**, **processing**, **completed**, etc.)	JobState	**M**	**1**	7
17. **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).~~	bit vector	**M**	**1**	5
18. ~~**jmJobStateReasonsMessage** - Job State Reason Messages the explain current job state reasons - multi-valued~~	~~**T(255)**~~	~~**CM**~~	~~**n**~~	
19. ~~**jmDevicesAssigned** - Devices assigned to this job~~	~~**HrDeviceIndex**~~	~~**M**~~	~~**n**~~	~~1~~
20. ~~**jmDeviceStateOfDevicesAssigned** - state of each of the devices assigned to the job~~	~~**enum**~~	~~**M**~~	~~**n**~~	~~1~~
21. **jmJobOctetsCompletedHigh** - Octets completed -high order part	**Integer 32**	**M**	**1**	3
22. **jmJobOctetsCompletedLow** - Octets completed -low order part	**Integer 32**	**M**	**1**	3

Job Group - Status and Accounting (S)	DataType	Conformance	Cardinality	Protocol
23. **jmJobStartedProcessingTime** - Date/Time of day job started processing on device	**DateAndTime**	**CM**	**1**	3
24. **jmJobCompletionTime** - Date/Time of day job finished using the device	**DateAndTime**	**CM**	**1**	1
25. **jmJobAccountName** - Account Name	**T~~O~~(63 ~~255~~)**	**M**	**1**	3

9.5 The Resources Group (Mandatory)

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. If only one MIB instance is supported, the value of this index shall be 1.
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

jmResourceEntry (R)	DataType	Conformance	Cardinality	Protocol
1. jmResourceIndex - a running index of the resources requested and/or used by the job.	**Index**	**M**	**1**	
2. **jmResourceType** - Resources required/used (table):	**enum**	**M**	**n**	
a) **documentName(3) -** Document name(s) (or file-names)		**CM**	**n**	7
b) **jobCopiesRequested(4) -** Number of job copies requested		**CM**	**1**	4
c) **jobCopiesProduced(5) -** Number of job copies produced		**CM**	**1**	1
d) **documentCopiesRequested(5) -** Number of document copies requested		**CM**	**1**	4
e) **documentCopiesProduced(5) -** Number of document copies produced		**CM**	**1**	1
f) **sides(6) -** Number of sides requested (one-sided, two-sided)		**CM**	**1**	5
g) **interpreters(7) -** PDLs requested/used		**M**	**n**	5
h) **physicalDevices(8) -** physical devices requested/used		**CM**	**n**	6
i) **faxPhoneNumbers(9) -** FAX phone number(s) requested/used		**CM**	**n**	

jmResourceEntry (R)	DataType	Conformance	Cardinality	Protocol
j) **impressions(10) -** Impressions (sides) completed		**CM**	**1**	3
k) **sheets(11) -** Sheets completed		**M**	**1**	2
l) **processingTime(12) -** Processing time so far		**M**	**1**	2
m) **processingMessage(13) -** Processing Messages		**CM**	**n**	
3. **jmResourceName -** resource required/usage name	**T(63~~255~~)**	**M**	**n**	
4. jmResourceUnits - resource required/used usage-unit	**Units**	**M**	**n**	
5. jmResourceAmount - resource amount requested/used; -2 - unknown	**Integer 32**	**M**	**n**	