1 INTERNET-DRAFT R. Bergman 2 Dataproducts Corp. 3 T. Hastings 4 Xerox Corporation 5 S. Isaacson б Novell, Inc. 7 H. Lewis 8 IBM Corp. 9 October 2November 8, 1998 10 Job Monitoring MIB - V1.32 11 <draft-ietf-printmib-job-monitor-08.txt> 12 13 Status of this Memo 14 This document is an Internet-Draft. Internet-Drafts are working 15 documents of the Internet Engineering Task Force (IETF), its 16 areas, and its working groups. Note that other groups may also 17 distribute working documents as Internet-Drafts. 18 Internet-Drafts are draft documents valid for a maximum of six 19 months and may be updated, replaced, or obsoleted by other 20 documents at any time. It is inappropriate to use Internet-Drafts 21 as reference material or to cite them other than as "work in 22 progress." 23 To learn the current status of any Internet-Draft, please check the "lid-abstracts.txt" listing contained in the Internet-Drafts 24 25 Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net 26 (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East 27 Coast), or ftp.isi.edu (US West Coast). 28 This Internet-Draft expires on April 2May 8, 1998. 29 30 Abstract 31 This document has been developed and approved by the Printer Working Group (PWG) as a PWG standard. It is intended to be 32 33 distributed as an Informational RFC. This document provides a 34 printer industry standard SNMP MIB for (1) monitoring the status 35 and progress of print jobs (2) obtaining resource requirements 36 before a job is processed, (3) monitoring resource consumption 37 while a job is being processed and (4) collecting resource 38 accounting data after the completion of a job. This MIB is 39 intended to be implemented (1) in a printer or (2) in a server 40 that supports one or more printers. Use of the object set is not 41 limited to printing. However, support for services other than printing is outside the scope of this Job Monitoring MIB. Future 42 43 extensions to this MIB may include, but are not limited to, fax 44 machines and scanners.

Bergman, Hastings, Isaacson, Lewis Informational

[Page 2]

	INTERNET-DRAFT Job Monitoring MIB, V1.3 <u>November 8</u> , 1	998
45 46	TABLE OF CONTENTS	
47	1 INTRODUCTION	7
48	1.1 Types of Information in the MIB	7
49	1.2 Types of Job Monitoring Applications	9
50	2 TERMINOLOGY AND JOB MODEL	10
51 52 53 54 55 56	<pre>2.1 System Configurations for the Job Monitoring MIB 2.1.1 Configuration 1 - client-printer 2.1.2 Configuration 2 - client-server-printer - agent in the server 14 2.1.3 Configuration 3 - client-server-printer - client monitor printer agent and server</pre>	13 13 s 15
57	3 MANAGED OBJECT USAGE	17
58 59 60 61 62 63 64	<pre>3.1 Conformance Considerations 3.1.1 Conformance Terminology 3.1.2 Agent Conformance Requirements 3.1.2.1 MIB II System Group objects 3.1.2.2 MIB II Interface Group objects 3.1.2.3 Printer MIB objects 3.1.3 Job Monitoring Application Conformance Requirements</pre>	17 17 <b>18</b> <b>18</b> <b>18</b> 18
65	3.2 The Job Tables and the Oldest Active and Newest Active Indexes	19
66 67 68 69 70 71	3.3 The Attribute Mechanism and the Attribute Table(s) 3.3.1 Conformance of Attribute Implementation 3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes 3.3.3 Index Value Attributes 3.3.4 Data Sub-types and Attribute Naming Conventions	21 22 22 23 23
72 73 74 75 76 77 78 79 80 81	<pre>3.3.5 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes 3.3.6 Requested Objects and Attributes 3.3.7 Consumption Attributes 3.3.8 Attribute Specifications 3.3.9 Job State Reason bit definitions 3.3.9.1 JmJobStateReasons1TC specification 3.3.9.2 JmJobStateReasons2TC specification 3.3.9.4 JmJobStateReasons4TC specification</pre>	24 24 25 44 <b>45</b> <b>53</b>
82	3.4 Monitoring Job Progress	53
83	3.5 Job Identification	58
	Bergman, Hastings, Isaacson, Lewis Informational [Page	3]

	INTERNET-DRAFT Job Monitoring MIB, V1. <u>3</u> <u>November 8</u> ,	1998
84	3.5.1 The Job Submission ID specifications	59
85 86 87 88	<pre>3.6 Internationalization Considerations     3.6.1 Text generated by the server or device     3.6.2 Text supplied by the job submitter     3.6.3 'DateAndTime' for representing the date and time</pre>	63 63 64 65
89 90 91 93 93 95 95 96 97	3.7 IANA and PWG Registration Considerations 3.7.1 PWG Registration of enums 3.7.1.1 Type 1 enumerations 3.7.1.2 Type 2 enumerations 3.7.1.3 Type 3 enumeration 3.7.2 PWG Registration of type 2 bit values 3.7.3 PWG Registration of Job Submission Id Formats 3.7.4 PWG Registration of MIME types/sub-types for document-formats 68	65 66 <b>66</b> 67 67 67
98 99 100	<b>3.8 Security Considerations</b> 3.8.1 Read-Write objects 3.8.2 Read-Only Objects In Other User's Jobs	68 68 68
101	3.9 Notifications	68
102	4 MIB SPECIFICATION	69
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123	Textual conventions for this MIB module JmUTF8StringTC JmJobStringTC JmAturalLanguageTagTC JmTimeStampTC JmJobSourcePlatformTypeTC JmFinishingTC JmPrinteResolutionTC JmPrinterResolutionTC JmTonerEconomyTC JmBooleanTC JmJobCollationTypeTC JmJobStateTC JmJobStateTC JmJobStateReasons1TC JmJobStateReasons2TC JmJobStateReasons3TC JmJobStateReasons4TC	71 71 71 72 73 74 75 75 75 75 76 77 85 88 90 99 99
124 125 126 127	The General Group (MANDATORY) jmGeneralJobSetIndex (Int32(132767)) jmGeneralNumberOfActiveJobs (Int32(0)) jmGeneralOldestActiveJobIndex (Int32(0))	101 102 102 103

Bergman, Ha	astings,	Isaacson,	Lewis	Informational	[Page 4]
-------------	----------	-----------	-------	---------------	----------

	INTERNET-DRAFT Job Monit	oring MIB,	V1. <u>3</u>	November	<mark>8</mark> , 1998
128 129 130 131	jmGeneralNewestActiveJobInde jmGeneralJobPersistence (I jmGeneralAttributePersistenc jmGeneralJobSetName (UTF8S	int32(15) e (Int32	)		103 104 104 105
132 133 134 135	The Job ID Group (MANDATORY) jmJobSubmissionID (OCTET S jmJobIDJobSetIndex (Int32( jmJobIDJobIndex (Int32(0	032767))	(48)))		105 107 108 108
136 137 138 139 140 141 142 143 144 145	The Job Group (MANDATORY) jmJobIndex (Int32(1)) jmJobState (JmJobStateTC) jmJobStateReasons1 (JmJobS jmNumberOfInterveningJobs jmJobKOctetsPerCopyRequested jmJobKOctetsProcessed (Int jmJobImpressionsPerCopyReque jmJobImpressionsCompleted jmJobOwner (JobString63)	(Int32(-2. l (Int32( 232(-2)) ested (Int	.)) -2)) t32(-2))		108 110 111 111 112 112 112 113 113 114
146 147 148 149 150	The Attribute Group (MANDATORY jmAttributeTypeIndex (JmAt jmAttributeInstanceIndex ( jmAttributeValueAsInteger jmAttributeValueAsOctets (	tributeType Int32(13	2767))		114 117 117 118 119
151 152 153 154 155	The Mirror Attribute Group (OF jmMirrorAttrTypeIndex (JmA jmMirrorAttrInstanceIndex jmMirrorAttrValueAsInteger jmMirrorAttrValueAsOctets	ttributeTy (Int32(1	32767))		120 121 121 121 121
156	5 APPENDIX A - IMPLEMENTING TH	IE JOB LIFE	CYCLE		126
157	6 APPENDIX B - SUPPORT OF JOB	SUBMISSION	PROTOCOLS		127
158	7 REFERENCES				127
159	8 AUTHOR'S ADDRESSES				129
160	9 CHANGE HISTORY				131
161	9.1 Changes to produce version	1.3, date	d November 8	, 1998	131
162	9.2 Changes to produce version	1.2, date	d October 2,	1998	131
163	9.3 Changes to produce version	1.1, date	d October 1,	1998	132

Bergman, Hastings, Isaacson, Lewis Informational [Page 5]

133

164 10 INDEX

165

Bergman, Hastings, Isaacson, Lewis Informational

[Page 6]

[Page 7]

166

Job Monitoring MIB

#### 167 1 Introduction

This specification defines an official Printer Working Group (PWG) 168 [PWG] standard SNMP MIB for the monitoring of jobs on network printers. 169 This specification is being published as an IETF Information Document 170 for the convenience of the Internet community. In consultation with 171 172 the IETF Application Area Directors, it was concluded that this MIB specification properly belongs as an Information document, because this 173 174 MIB monitors a service node on the network, rather than a network node 175 proper.

176 The Job Monitoring MIB is intended to be implemented by an agent within 177 a printer or the first server closest to the printer, where the printer 178 is either directly connected to the server only or the printer does not 179 contain the job monitoring MIB agent. It is recommended that 180 implementations place the SNMP agent as close as possible to the 181 processing of the print job. This MIB applies to printers with and 182 without spooling capabilities. This MIB is designed to be compatible with most current commonly-used job submission protocols. In most 183 184 environments that support high function job submission/job control 185 protocols, like ISO DPA[iso-dpa], those protocols would be used to 186 monitor and manage print jobs rather than using the Job Monitoring MIB.

187 The Job Monitoring MIB consists of a General Group, a Job Submission ID 188 Group, a Job Group, and an Attribute Group. Each group is a table. All accessible objects are read-only. The General Group contains 189 general information that applies to all jobs in a job set. The Job 190 191 Submission ID table maps the job submission ID that the client uses to 192 identify a job to the jmJobIndex that the Job Monitoring Agent uses to identify jobs in the Job and Attribute tables. The Job table contains 193 194 the MANDATORY integer job state and status objects. The Attribute 195 table consists of multiple entries per job that specify (1) job and document identification and parameters, (2) requested resources, and (3) consumed resources during and after job processing/printing. A 196 197 larger number of job attributes are defined as textual conventions that 198 199 an agent SHALL return if the server or device implements the 200 functionality so represented and the agent has access to the 201 information. The Attribute table provides access to job attributes by 202 job index. An OPTIONAL Mirror Attribute table is defined which 203 provides access to the same job attributes by attribute.

# 204 1.1 Types of Information in the MIB

205 The job MIB is intended to provide the following information for the 206 indicated Role Models in the Printer MIB[print-mib] (Appendix D - Roles 207 of Users).

Job Monitoring MIB, V1.3 November 8, 1998

- 208 User:
- 209 Provide the ability to identify the least busy printer. The user 210 will be able to determine the number and size of jobs waiting for 211 each printer. No attempt is made to actually predict the length 212 of time that jobs will take.
- 213 Provide the ability to identify the current status of the user's job (user queries). 214
- 215 Provide a timely indication that the job has completed and where it can be found. 216
- 217 Provide error and diagnostic information for jobs that did not 218 successfully complete.
- Operator: 219
- 220 Provide a presentation of the state of all the jobs in the print 221 system.
- 222 Provide the ability to identify the user that submitted the print 223 iob.
- 2.2.4 Provide the ability to identify the resources required by each 225 iob.
- 226 Provide the ability to define which physical printers are 227 candidates for the print job.
- 228 Provide some idea of how long each job will take. However, exact 229 estimates of time to process a job is not being attempted. 230 Instead, objects are included that allow the operator to be able 231 to make gross estimates.
- 232 Capacity Planner:
- 233 Provide the ability to determine printer utilization as a 234 function of time.
- Provide the ability to determine how long jobs wait before 235 236 starting to print.
- 237 Accountant:
- 238 Provide information to allow the creation of a record of 239 resources consumed and printer usage data for charging users or 240 groups for resources consumed.
- 241 Provide information to allow the prediction of consumable usage 2.4.2 and resource need.

Job Monitoring MIB, V1.3 November 8, 1998

243 The MIB supports printers that can contain more than one job at a time, 244 but still be usable for low end printers that only contain a single job 245 at a time. In particular, the MIB supports the needs of Windows and 246 other PC environments for managing low-end direct-connect (serial or 247 parallel) and networked devices without unnecessary overhead or 248 complexity, while also providing for higher end systems and devices.

# 1.2 Types of Job Monitoring Applications 249

250 The Job Monitoring MIB is designed for the following types of 251 monitoring applications:

- 252 1. Monitor a single job starting when the job is submitted and 253 ending a defined period after the job completes. The Job 254 Submission ID table provides the map to find the specific job 255 to be monitored.
- 256 2. Monitor all 'active' jobs in a queue, which this specification 257 generalizes to a "job set". End users may use such a program when selecting a least busy printer, so the MIB is designed for 258 259 such a program to start up quickly and find the information 260 needed quickly without having to read all (completed) jobs in order to find the active jobs. System operators may also use 261 such a program, in which case it would be running for a long 262 period of time and may also be interested in the jobs that have 263 264 completed. Finally such a program may be used to provide an 265 enhanced console and logging capability.
- 266 3. Collect resource usage for accounting or system utilization 267 purposes that copy the completed job statistics to an accounting system. It is recognized that depending on 268 269 accounting programs to copy MIB data during the job-retention 270 period is somewhat unreliable, since the accounting program may 271 not be running (or may have crashed). Such a program is also expected to keep a shadow copy of the entire Job Attribute 272 273 table including completed, canceled, and aborted jobs which the 274 program updates on each polling cycle. Such a program polls at 275 the rate of the persistence of the Attribute table. The design 276 is not optimized to help such an application determine which 277 jobs are completed, canceled, or aborted. Instead, the application SHOULD query each job that the application's shadow 278 279 copy shows was not complete, canceled, or aborted at the 280 previous poll cycle to see if it is now complete or canceled, 281 plus any new jobs that have been submitted.

282 The MIB provides a set of objects that represent a compatible subset of 283 job and document attributes of the ISO DPA standard[iso-dpa] and the 284 Internet Printing Protocol (IPP)[ipp-model], so that coherence is 285 maintained between these two protocols and the information presented to 286 end users and system operators by monitoring applications. However, the job monitoring MIB is intended to be used with printers that 287 288 implement other job submitting and management protocols, such as IEEE 1284.1 (TIPSI)[tipsi], as well as with ones that do implement ISO DPA. 289

Bergman, Hastings, Isaacson, Lewis Informational [Page 9]

290 Thus the job monitoring MIB does not require implementation of either 291 the ISO DPA or IPP protocols.

292 The MIB is designed so that an additional MIB(s) can be specified in 293 the future for monitoring multi-function (scan, FAX, copy) jobs as an 294 augmentation to this MIB.

2 Terminology and Job Model 295

296 This section defines the terms that are used in this specification and 297 the general model for jobs in alphabetical order.

298 NOTE - Existing systems use conflicting terms, so these terms are drawn from the ISO 10175 Document Printing Application (DPA) 299 300 standard[iso-dpa]. For example, PostScript systems use the term 301 session for what is called a *job* in this specification and the term 302 job to mean what is called a *document* in this specification.

303 Accounting Application: The SNMP management application that copies 304 job information to some more permanent medium so that another 305 application can perform accounting on the data for Accountants, Asset 306 Managers, and Capacity Planners use.

307 Agent: The network entity that accepts SNMP requests from a monitor or 308 accounting application and provides access to the instrumentation for 309 managing jobs modeled by the management objects defined in the Job 310 Monitoring MIB module for a server or a device.

311 Attribute: A name, value-pair that specifies a job or document 312 instruction, a status, or a condition of a job or a document that has 313 been submitted to a server or device. A particular attribute NEED NOT 314 be present in each job instance. In other words, attributes are present in a job instance only when there is a need to express the 315 value, either because (1) the client supplied a value in the job 316 317 submission protocol, (2) the document data contained an embedded 318 attribute, or (3) the server or device supplied a default value. An agent MAY represent an attribute as an entry (row) in the Attribute 319 320 table in this MIB in which entries are present only when necessary. 321 Attributes are identified in this MIB by an enum.

322 Client: The network entity that end users use to submit jobs to 323 spoolers, servers, or printers and other devices, depending on the configuration, using any job submission protocol over a serial or parallel port to a directly-connected device or over the network to a 324 325 326 networked-connected device.

Device: A hardware entity that (1) interfaces to humans, such as a 327 328 device that produces marks on paper or scans marks on paper to produce 329 an electronic representation, (2) accesses digital media, such as CD-330 ROMs, or (3) interfaces electronically to another device, such as sends 331 FAX data to another FAX device.

Bergman, Hastings, Isaacson, Lewis Informational [Page 10]

332 Document: A sub-section within a job that contains print data and 333 document instructions that apply to just the document.

334 Document Instruction: An instruction specifying how to process the 335 document. Document instructions MAY be passed in the job submission 336 protocol separate from the actual document data, or MAY be embedded in the document data or a combination, depending on the job submission 337 338 protocol and implementation.

339 End User: A user that uses a client to submit a print job. See 340 "user".

341 Impression: For a print job, an impression is the passage of the 342 entire side of a sheet by the marker, whether or not any marks are made 343 and independent of the number of passes that the side makes past the 344 marker. Thus a four pass color process counts as a single impression, 345 as does highlight color. Impression counters count all kinds: 346 monochrome, highlight color, and full process color, while full color counters only count full color impressions, and high light color 347 348 counters only count high light color impressions.

349 One-sided processing involves one impression per sheet. Two-sided 350 processing involves two impressions per sheet. If a two-sided document has an odd number of pages, the last sheet still counts as two 351 352 impressions, if that sheet makes two passes through the marker or the 353 marker marks on both sides of a sheet in a single pass. Two-up 354 printing is the placement of two logical pages on one side of a sheet 355 and so is still a single impression. See "page" and "sheet".

356 NOTE - Since impressions include blank sides, it is suggested that 357 accounting application implementers consider charging for sheets, 358 rather than impressions, possibly using the value of the sides 359 attribute to select different charges for one-sided versus two-sided 360 printing, since some users may think that impressions don't include 361 blank sides.

362 Internal Collation: The production of the sheets for each document copy performed within the printing device by making multiple passes over 363 364 either the source or an intermediate representation of the document.

365 Job: A unit of work whose results are expected together without 366 interjection of unrelated results. A job contains one or more 367 documents.

368 Job Accounting: The activity of a management application of accessing 369 the MIB and recording what happens to the job during and after the 370 processing of the job.

Job Monitoring MIB, V1.3 November 8, 1998

371 Job Instruction: An instruction specifying how, when, or where the job 372 is to be processed. Job instructions MAY be passed in the job 373 submission protocol or MAY be embedded in the document data or a 374 combination depending on the job submission protocol and 375 implementation.

376 Job Monitoring (using SNMP): The activity of a management application of accessing the MIB and (1) identifying jobs in the job tables being 377 processed by the server, printer or other devices, and (2) displaying 378 379 information to the user about the processing of the job.

380 Job Monitoring Application: The SNMP management application that End 381 Users, and System Operators use to monitor jobs using SNMP. A monitor 382 MAY be either a separate application or MAY be part of the client that 383 also submits jobs. See "monitor".

384 Job Set: A group of jobs that are queued and scheduled together 385 according to a specified scheduling algorithm for a specified device or set of devices. For implementations that embed the SNMP agent in the 386 387 device, the MIB job set normally represents all the jobs known to the 388 device, so that the implementation only implements a single job set. 389 If the SNMP agent is implemented in a server that controls one or more devices, each MIB job set represents a job queue for (1) a specific 390 391 device or (2) set of devices, if the server uses a single queue to load 392 balance between several devices. Each job set is disjoint; no job 393 SHALL be represented in more than one MIB job set.

394 Monitor: Short for Job Monitoring Application.

395 Page: A page is a logical division of the original source document. 396 Number up is the imposition of more than one page on a single side of a 397 sheet. See "impression" and "sheet" and "two-up".

398 Proxy: An agent that acts as a concentrator for one or more other 399 agents by accepting SNMP operations on the behalf of one or more other 400 agents, forwarding them on to those other agents, gathering responses 401 from those other agents and returning them to the original requesting 402 monitor.

403 Queuing: The act of a *device* or *server* of ordering (queuing) the jobs 404 for the purposes of scheduling the jobs to be processed.

405 Printer: A device that puts marks on media.

406 Server: A network entity that accepts jobs from clients and in turn 407 submits the jobs to *printers* and other *devices* that may be directly 408 connected to the server via a serial or parallel port or may be on the 409 network. A server MAY be a printer supervisor control program, or a 410 print spooler.

411 Sheet: A sheet is a single instance of a medium, whether printing on 412 one or both sides of the medium. See "impression" and "page".

Bergman, Hastings, Isaacson, Lewis Informational [Page 12]

413 SNMP Information Object: A name, value-pair that specifies an action, 414 a status, or a condition in an SNMP MIB. Objects are identified in 415 SNMP by an OBJECT IDENTIFIER.

416 Spooler: A server that accepts jobs, spools the data, and decides when 417 and on which printer to print the job. A spooler is a client to a printer or a printer supervisor, depending on implementation. 418

- 419 Spooling: The act of a *device* or *server* of (1) accepting jobs and (2) 420 writing the job's attributes and document data on to secondary storage.
- 421 Stacked: When a media sheet is placed in an output bin of a device.

422 Supervisor: A server that contains a control program that controls a 423 printer or other device. A supervisor is a client to the printer or 424 other device.

- 425 System Operator: A user that uses a monitor to monitor the system and 426 carries out tasks to keep the system running.
- 427 System Administrator: A user that specifies policy for the system.
- 428 Two-up: The placement of two pages on one side of a sheet so that each 429 side or impressions counts as two pages. See "page" and "sheet".

430 User: A person that uses a client or a monitor. See "end user".

#### 431 2.1 System Configurations for the Job Monitoring MIB

432 This section enumerates the three configurations in which the Job 433 Monitoring MIB is intended to be used. To simplify the pictures, the devices are shown as printers. See section 1.1 entitled "Types of 434 435 Information in the MIB".

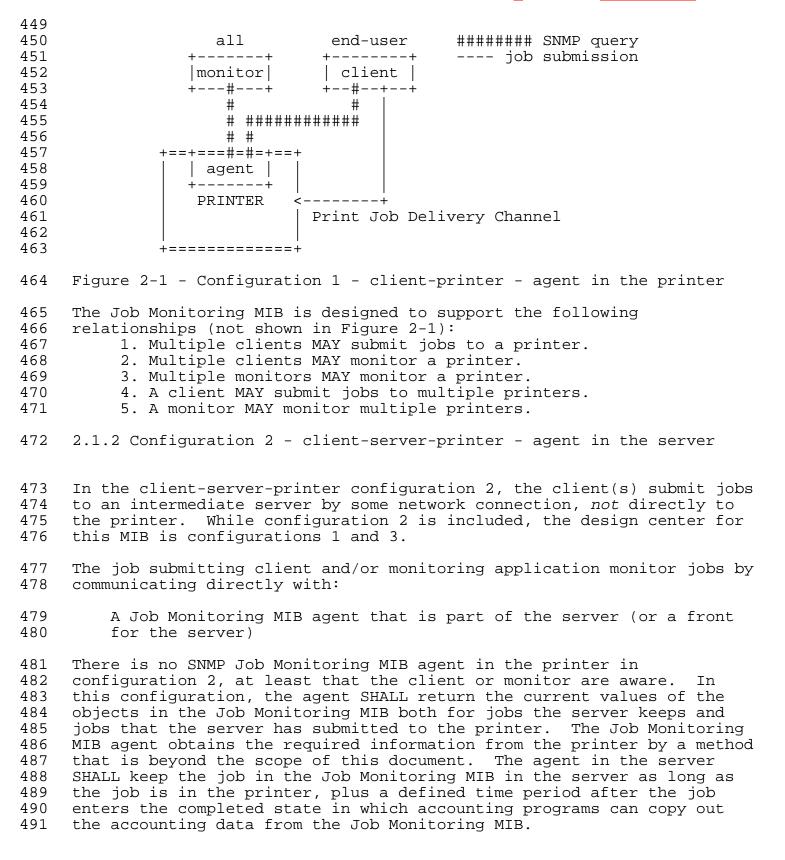
436 The diagram in the Printer MIB[print-mib] entitled: "One Printer's View 437 of the Network" is assumed for this MIB as well. Please refer to that 438 diagram to aid in understanding the following system configurations.

439 2.1.1 Configuration 1 - client-printer

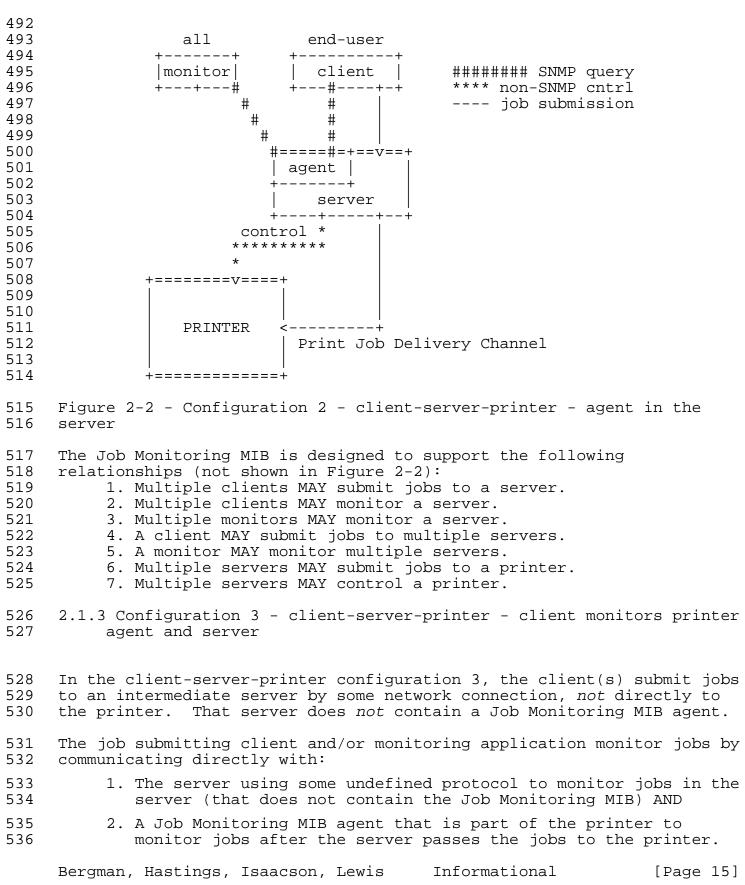
440 In the client-printer configuration 1, the client(s) submit jobs 441 directly to the printer, either by some direct connect, or by network 442 connection.

443 The job submitting client and/or monitoring application monitor jobs by 444 communicating directly with an agent that is part of the printer. The agent in the printer SHALL keep the job in the Job Monitoring MIB as 445 long as the job is in the printer, plus a defined time period after the 446 447 job enters the completed state in which accounting programs can copy 448 out the accounting data from the Job Monitoring MIB.

Bergman, Hastings, Isaacson, Lewis Informational [Page 13]



Bergman, Hastings, Isaacson, Lewis Informational [Page 14]



552

Job Monitoring MIB, V1.3 November 8, 1998

In such configurations, the server deletes its copy of the job 537 538 from the server after submitting the job to the printer usually 539 almost immediately (before the job does much processing, if 540 any).

541 In configuration 3, the agent (in the printer) SHALL keep the values of the objects in the Job Monitoring MIB that the agent implements updated 542 for a job that the server has submitted to the printer. The agent 543 544 SHALL obtain information about the jobs submitted to the printer from the server (either in the job submission protocol, in the document 545 546 data, or by direct query of the server), in order to populate some of the objects the Job Monitoring MIB in the printer. The agent in the 547 printer SHALL keep the job in the Job Monitoring MIB as long as the job 548 is in the Printer, and longer in order to implement the completed state 549 550 in which monitoring programs can copy out the accounting data from the 551 Job Monitoring MIB.

554				
553	all	end-user		
554	++ +		-+	
555	monitor	client	######	## SNMP query
556	++ +	*+	-+ **** n	on-SNMP query
557	# *	*	j	ob submission
558	# *	*		
559	# *	*		
560	# *==	===v====v	==+	
561	#			
562	#	server		
563	#			
564	# +	#+	+	
565	# option	al#		
566	# #######	####		
567	# #			
568	+==+=v===v=+==+			
569	agent			
570	++			
571	PRINTER <-	+		
572		Print Job	Delivery Cha	nnel
573				
574	+==========+			

575 Figure 2-3 - Configuration 3 - client-server-printer - client monitors 576 printer agent and server

577 The Job Monitoring MIB is designed to support the following relationships (not shown in Figure 2-3): 578 579 1. Multiple clients MAY submit jobs to a server. 580 2. Multiple clients MAY monitor a server. 581 3. Multiple monitors MAY monitor a server. 4. A client MAY submit jobs to multiple servers. 582 583 5. A monitor MAY monitor multiple servers. Multiple servers MAY submit jobs to a printer.
 Multiple servers MAY control a printer. 584 585

Bergman, Hastings, Isaacson, Lewis Informational [Page 16]

- 586 3 Managed Object Usage
- 587 This section describes the usage of the objects in the MIB.

#### 588 3.1 Conformance Considerations

589 In order to achieve interoperability between job monitoring applications and job monitoring agents, this specification includes the 590

- conformance requirements for both monitoring applications and agents. 591
- 592 3.1.1 Conformance Terminology

593 This specification uses the verbs: "SHALL", "SHOULD", "MAY", and "NEED 594 NOT" to specify conformance requirements according to RFC 2119 [req-595 words] as follows:

- 596 "SHALL": indicates an action that the subject of the sentence must 597 implement in order to claim conformance to this specification
- 598 "MAY": indicates an action that the subject of the sentence does not 599 have to implement in order to claim conformance to this specification, in other words that action is an implementation option 600
- 601 "NEED NOT": indicates an action that the subject of the sentence does not have to implement in order to claim conformance to this specification. The verb "NEED NOT" is used instead of "may not", 602 603 604 since "may not" sounds like a prohibition.
- 605 "SHOULD": indicates an action that is recommended for the subject of 606 the sentence to implement, but is not required, in order to claim 607 conformance to this specification.
- 608 3.1.2 Agent Conformance Requirements
- 609 A conforming agent:
- 610 1. SHALL implement all MANDATORY groups in this specification.
- 611 2. SHALL implement any attributes if (1) the server or device 612 supports the functionality represented by the attribute and (2) the information is available to the agent. 613
- 614 3. SHOULD implement both forms of an attribute if it implements an attribute that permits a choice of INTEGER and OCTET STRING 615 forms, since implementing both forms may help management 616 applications by giving them a choice of representations, since 617 618 the representation are equivalent. See the JmAttributeTypeTC 619 textual-convention.
- 620 NOTE - This MIB, like the Printer MIB, is written following the subset 621 of SMIv2 that can be supported by SMIv1 and SNMPv1 implementations.

Bergman, Hastings, Isaacson, Lewis Informational [Page 17] 622 3.1.2.1 MIB II System Group objects

The Job Monitoring MIB agent SHALL implement all objects in the System 623 624 Group of MIB-II[mib-II], whether the Printer MIB[print-mib] is 625 implemented or not.

3.1.2.2 MIB II Interface Group objects 626

627 The Job Monitoring MIB agent SHALL implement all objects in the Interfaces Group of MIB-II[mib-II], whether the Printer MIB[print-mib] 628 629 is implemented or not.

630 3.1.2.3 Printer MIB objects

631 If the agent is providing access to a device that is a printer, the 632 agent SHALL implement all of the MANDATORY objects in the Printer 633 MIB[print-mib] and all the objects in other MIBs that conformance to 634 the Printer MIB requires, such as the Host Resources MIB[hr-mib]. Ιf 635 the agent is providing access to a server that controls one or more 636 direct-connect or networked printers, the agent NEED NOT implement the 637 Printer MIB and NEED NOT implement the Host Resources MIB.

- 638 3.1.3 Job Monitoring Application Conformance Requirements
- 639 A conforming job monitoring application:
- 640 1. SHALL accept the full syntactic range for all objects in all 641 MANDATORY groups and all MANDATORY attributes that are required 642 to be implemented by an agent according to Section 3.1.2 and 643 SHALL either present them to the user or ignore them.
- 644 2. SHALL accept the full syntactic range for all attributes, 645 including enum and bit values specified in this specification 646 and additional ones that may be registered with the PWG and 647 SHALL either present them to the user or ignore them. In 648 particular, a conforming job monitoring application SHALL not 649 malfunction when receiving any standard or registered enum or bit values. See Section 3.7 entitled "IANA and PWG 650 651 Registration Considerations".
- 652 3. SHALL NOT fail when operating with agents that materialize 653 attributes after the job has been submitted, as opposed to when 654 the job is submitted.
- 655 4. SHALL, if it supports a time attribute, accept either form of 656 the time attribute, since agents are free to implement either 657 time form.

# 658 3.2 The Job Tables and the Oldest Active and Newest Active Indexes

659 The jmJobTable and jmAttributeTable contain objects and attributes,

- respectively, for each job in a job set. These first two indexes are: 660
- 1. jmGeneralJobSetIndex which job set 661
- 662 2. jmJobIndex - which job in the job set

663 In order for a monitoring application to quickly find that active jobs 664 (jobs in the pending, processing, or processingStopped states), the MIB 665 contains two indexes:

- 666 1. jmGeneralOldestActiveJobIndex - the index of the active job 667 that has been in the tables the longest.
- 668 2. jmGeneralNewestActiveJobIndex - the index of the active job 669 that has been most recently added to the tables.

670 The agent SHALL assign the next incremental value of jmJobIndex to the job, when a new job is accepted by the server or device to which the 671 agent is providing access. If the incremented value of jmJobIndex 672 673 would exceed the implementation-defined maximum value for jmJobIndex, 674 the agent SHALL 'wrap' back to 1. An agent uses the resulting value of 675 jmJobIndex for storing information in the jmJobTable and the 676 jmAttributeTable about the job.

677 It is recommended that the largest value for jmJobIndex be much larger than the maximum number of jobs that the implementation can contain at 678 a single time, so as to minimize the premature re-use of a jmJobIndex 679 680 value for a newer job while clients retain the same 'stale' value for 681 an older job.

682 It is recommended that agents that are providing access to 683 servers/devices that already allocate job-identifiers for jobs as 684 integers use the same integer value for the jmJobIndex. Then management applications using this MIB and applications using other 685 protocols will see the same job identifiers for the same jobs. 686 Agents 687 providing access to systems that contain jobs with a job identifier of 0 SHALL map the job identifier value 0 to a jmJobIndex value that is 688 one higher than the highest job identifier value that any job can have 689 690 on that system. Then only job 0 will have a different job-identifier value than the job's jmJobIndex value. 691

692 NOTE - If a server or device accepts jobs using multiple job submission 693 protocols, it may be difficult for the agent to meet the recommendation 694 to use the job-identifier values that the server or device assigns as 695 the jmJobIndex value, unless the server/device assigns job-identifiers 696 for each of its job submission protocols from the same job-identifier 697 number space.

698 Each time a new job is accepted by the server or device that the agent 699 is providing access to AND that job is to be 'active' (pending,

700 processing, or processingStopped, but not pendingHeld), the agent SHALL copy the value of the job's jmJobIndex to the 701

jmGeneralNewestActiveJobIndex object. If the new job is to be 702

703 'inactive' (pendingHeld state), the agent SHALL not change the value of jmGeneralNewestActiveJobIndex object (though the agent SHALL assign the 704 705 next incremental jmJobIndex value to the job).

706 When a job transitions from one of the 'active' job states (pending, 707 processing, processingStopped) to one of the 'inactive' job states (pendingHeld, completed, canceled, or aborted), with a jmJobIndex value 708 that matches the jmGeneralOldestActiveJobIndex object, the agent SHALL 709 710 advance (or wrap) the value to the next oldest 'active' job, if any. See the JmJobStateTC textual-convention for a definition of the job 711 712 states.

713 Whenever a job transitions from one of the 'inactive' job states to one 714 of the 'active' job states (from pendingHeld to pending or processing), 715 the agent SHALL update the value of either the

716

jmGeneralOldestActiveJobIndex or the jmGeneralNewestActiveJobIndex 717 objects, or both, if the job's jmJobIndex value is outside the range

718 between jmGeneralOldestActiveJobIndex and

719 jmGeneralNewestActiveJobIndex.

720 When all jobs become 'inactive', i.e., enter the pendingHeld, 721 completed, canceled, or aborted states, the agent SHALL set the value 722 of both the jmGeneralOldestActiveJobIndex and

723 jmGeneralNewestActiveJobIndex objects to 0.

724 NOTE - Applications that wish to efficiently access all of the active 725 jobs MAY use jmGeneralOldestActiveJobIndex value to start with the 726 oldest active job and continue until they reach the index value equal 727 to jmGeneralNewestActiveJobIndex, skipping over any pendingHeld, 728 completed, canceled, or aborted jobs that might intervene.

729 If an application detects that the jmGeneralNewestActiveJobIndex is 730 smaller than jmGeneralOldestActiveJobIndex, the job index has wrapped. 731 In this case, the application SHALL reset the index to 1 when the end 732 of the table is reached and continue the GetNext operations to find the 733 rest of the active jobs.

NOTE - Applications detect the end of the jmAttributeTable table when 734 the OID returned by the GetNext operation is an OID in a different MIB. 735 There is no object in this MIB that specifies the maximum value for the 736 737 jmJobIndex supported by the implementation.

738 When the server or device is power-cycled, the agent SHALL remember the 739 next jmJobIndex value to be assigned, so that new jobs are not assigned 740 the same jmJobIndex as recent jobs before the power cycle.

Bergman, Hastings, Isaacson, Lewis Informational [Page 20]

Job Monitoring MIB, V1.3 November 8, 1998

## 741 3.3 The Attribute Mechanism and the Attribute Table(s)

742 Attributes are similar to information objects, except that attributes 743 are identified by an enum, instead of an OID, so that attributes may be 744 registered without requiring a new MIB. Also an implementation that 745 does not have the functionality represented by the attribute can omit the attribute entirely, rather than having to return a distinguished 746 747 value. The agent is free to materialize an attribute in the jmAttributeTable as soon as the agent is aware of the value of the 748 749 attribute.

- 750 The agent materializes job attributes in a four-indexed 751 jmAttributeTable:
- 752 1. jmGeneralJobSetIndex - which job set
- 753 2. jmJobIndex - which job in the job set
- 3. jmAttributeTypeIndex which attribute 754
- 755 4. jmAttributeInstanceIndex - which attribute instance for those 756 attributes that can have multiple values per job.

757 With this order of table indexing, an application can obtain all of the 758 attributes of a particular job using SNMPv1 GetNext or SNMPv2 GetBulk.

759 An OPTIONAL mirror table, called jmMirrorAttrTable, provides access to the same job attributes, but with a different order to the indexes: 760

- 761 1. jmAttributeTypeIndex - which attribute
- 762 2. jmGeneralJobSetIndex - which job set
- 763 3. jmJobIndex - which job in the job set
- 764 4. jmAttributeInstanceIndex - which attribute instance for those attributes that can have multiple values per job. 765

766 With this order of table indexing, an application can obtain selected 767 attributes of a number of jobs using SNMPv1 GetNext or SNMPv2 GetBulk.

768 Some attributes represent information about a job, such as a file-name, 769 a document-name, a submission-time or a completion time. Other 770 attributes represent resources required, e.g., a medium or a colorant, 771 etc. to process the job before the job starts processing OR to indicate 772 the amount of the resource consumed during and after processing, e.g., 773 pages completed or impressions completed. If both a required and a 774 consumed value of a resource is needed, this specification assigns two separate attribute enums in the textual convention. 775

776 NOTE - The table of contents lists all the attributes in order. This 777 order is the order of enum assignments which is the order that the SNMP 778 GetNext operation returns attributes. Most attributes apply to all 779 three configurations covered by this MIB specification (see section 2.1 780 entitled "System Configurations for the Job Monitoring MIB"). Those

Bergman, Hastings, Isaacson, Lewis Informational [Page 21]

781 attributes that apply to a particular configuration are indicated as 782 'Configuration n:' and SHALL NOT be used with other configurations.

783 3.3.1 Conformance of Attribute Implementation

784 An agent SHALL implement any attribute if (1) the server or device supports the functionality represented by the attribute and (2) the 785 786 information is available to the agent. The agent MAY create the 787 attribute row in the jmAttributeTable when the information is available 788 or MAY create the row earlier with the designated 'unknown' value 789 appropriate for that attribute. See next section.

790 If the server or device does not implement or does not provide access 791 to the information about an attribute, the agent SHOULD NOT create the 792 corresponding row in the jmAttributeTable.

3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes 793

Some attributes have a 'useful' Integer32 value, some have a 'useful' 794 795 OCTET STRING value, some MAY have either or both depending on implementation, and some MUST have both. See the JmAttributeTypeTC 796 797 textual convention for the specification of each attribute.

798 SNMP requires that if an object cannot be implemented because its 799 values cannot be accessed, then a compliant agent SHALL return an SNMP 800 error in SNMPv1 or an exception value in SNMPv2. However, this MIB has been designed so that 'all' objects can and SHALL be implemented by an 801 agent, so that neither the SNMPv1 error nor the SNMPv2 exception value 802 803 SHALL be generated by the agent. This MIB has also been designed so 804 that when an agent materializes an attribute, the agent SHALL 805 materialize a row consisting of both the jmAttributeValueAsInteger and 806 jmAttributeValueAsOctets objects.

807 In general, values for objects and attributes have been chosen so that 808 a management application will be able to determine whether a 'useful', 809 'unknown', or 'other' value is available. When a useful value is not 810 available for an object, that agent SHALL return a zero-length string for octet strings, the value 'unknown(2)' for enums, a '0' value for an 811 812 object that represents an index in another table, and a value '-2' for 813 counting integers.

814 Since each attribute is represented by a row consisting of both the jmAttributeValueAsInteger and jmAttributeValueAsOctets MANDATORY 815 816 objects, SNMP requires that the agent SHALL always create an attribute 817 row with both objects specified. However, for most attributes the 818 agent SHALL return a "useful" value for one of the objects and SHALL 819 return the 'other' value for the other object. For integer only 820 attributes, the agent SHALL always return a zero-length string value 821 for the jmAttributeValueAsOctets object. For octet string only

Bergman, Hastings, Isaacson, Lewis Informational [Page 22]

822 attributes, the agent SHALL always return a '-1' value for the 823 jmAttributeValueAsInteger object.

824 3.3.3 Index Value Attributes

825 A number of attributes are indexes in other tables. Such attribute names end with the word 'Index'. If the agent has not (yet) assigned an index value for a particular index attribute for a job, the agent 826 827 SHALL either: (1) return the value 0 or (2) not add this attribute to 828 829 the jmAttributeTable until the index value is assigned. In the interests of brevity, the semantics for 0 is specified once here and is 830 not repeated for each index attribute specification and a DEFVAL of 0 831 832 is implied, even though the DEFVAL for jmAttributeValueAsInteger is -2.

833 3.3.4 Data Sub-types and Attribute Naming Conventions

834 Many attributes are sub-typed to give a more specific data type than 835 Integer32 or OCTET STRING. The data sub-type of each attribute is 836 indicated on the first line(s) of the description. Some attributes 837 have several different data sub-type representations. When an attribute has both an Integer32 data sub-type and an OCTET STRING data 838 839 sub-type, the attribute can be represented in a single row in the 840 jmAttributeTable. In this case, the data sub-type name is not included 841 as the last part of the name of the attribute, e.g., documentFormat(38) which is both an enum and/or a name. When the data sub-types cannot be 842 843 represented by a single row in the jmAttributeTable, each such 844 representation is considered a separate attribute and is assigned a separate name and enum value. For these attributes, the name of the 845 846 data sub-type is the last part of the name of the attribute: Name, 847 Index, DateAndTime, TimeStamp, etc. For example, 848 documentFormatIndex(37) is an index.

849 NOTE: The Table of Contents also lists the data sub-type and/or data 850 sub-types of each attribute, using the textual-convention name when 851 such is defined. The following abbreviations are used in the Table of 852 Contents as shown:

853

'Int32(-2)' 'Int32(0)' 'Int32(1)' 'Int32(mn)'	<pre>Integer32 (-22147483647) Integer32 (02147483647) Integer32 (12147483647) For all other Integer ranges, the lower</pre>
	and upper bound of the range is
	indicated.
'UTF8String63'	JmUTF8StringTC (SIZE(063))
'JobString63'	JmJobStringTC (SIZE(063))
′Octets63′	OCTET STRING (SIZE(063))
'Octets(mn)'	For all other OCTET STRING ranges, the
	exact range is indicated.

854

Bergman, Hastings, Isaacson, Lewis Informational [Page 23] INTERNET-DRAFT Job Monitoring MIB, V1.<u>3</u> November 8, 1998

855 3.3.5 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes

Most attributes have only one row per job. However, a few attributes 856 857 can have multiple values per job or even per document, where each value 858 is a separate row in the jmAttributeTable. Unless indicated with 'MULTI-ROW:' in the JmAttributeTypeTC description, an agent SHALL 859 860 ensure that each attribute occurs only once in the jmAttributeTable for a job. Most of the 'MULTI-ROW' attributes do not allow duplicate 861 862 values, i.e., the agent SHALL ensure that each value occurs only once for a job. Only if the specification of the 'MULTI-ROW' attribute also says "There is no restriction on the same xxx occurring in multiple 863 864 865 rows" can the agent allow duplicate values to occur for the job.

NOTE - Duplicates are allowed for 'extensive' 'MULTI-ROW' attributes, such as fileName(34) or documentName(35) which are specified to be 'per-document' attributes, but are *not* allowed for 'intensive' 'MULTI-ROW' attributes, such as mediumConsumed(171) and documentFormat(38) which are specified to be 'per-job' attributes.

871 3.3.6 Requested Objects and Attributes

872 A number of objects and attributes record requirements for the job. 873 Such object and attribute names end with the word 'Requested'. In the interests of brevity, the phrase 'requested' means: (1) requested by 874 the client (or intervening server) in the job submission protocol and 875 may also mean (2) embedded in the submitted document data, and/or (3) 876 877 defaulted by the recipient device or server with the same semantics as if the requester had supplied, depending on implementation. Also if a 878 879 value is supplied by the job submission client, and the server/device 880 determines a better value, through processing or other means, the agent 881 MAY return that better value for such object and attribute.

882 3.3.7 Consumption Attributes

A number of objects and attributes record consumption. Such attribute 883 names end with the word 'Completed' or 'Consumed'. If the job has not 884 yet consumed what that resource is metering, the agent either: (1) 885 SHALL return the value 0 or (2) SHALL not add this attribute to the 886 887 jmAttributeTable until the consumption begins. In the interests of brevity, the semantics for 0 is specified once here and is not repeated 888 for each consumption attribute specification and a DEFVAL of 0 is 889 890 implied, even though the DEFVAL for jmAttributeValueAsInteger is -2.

Job Monitoring MIB, V1.3 November 8, 1998

[Page 25]

891 3.3.8 Attribute Specifications

892 This section specifies the job attributes.

893 In the following definitions of the attributes, each description indicates whether the useful value of the attribute SHALL be 894 represented using the jmAttributeValueAsInteger or the 895 896 jmAttributeValueAsOctets objects by the initial tag: 'INTEGER:' or 897 'OCTETS:', respectively.

898 Some attributes allow the agent implementer a choice of useful values 899 of either an integer, an octets representation, or both, depending on 900 implementation. These attributes are indicated with 'INTEGER:' AND/OR 901 'OCTETS:' tags.

902 A very few attributes require both objects at the same time to represent a pair of useful values (see mediumConsumed(171)). These 903 904 attributes are indicated with 'INTEGER:' AND 'OCTETS:' tags. See the 905 jmAttributeGroup for the descriptions of these two MANDATORY objects.

906 NOTE - The enum assignments are grouped logically with values assigned 907 in groups of 20, so that additional values may be registered in the 908 future and assigned a value that is part of their logical grouping.

909 Values in the range 2\*\*30 to 2\*\*31-1 are reserved for private or 910 experimental usage. This range corresponds to the same range reserved 911 in IPP. Implementers are warned that use of such values may conflict 912 with other implementations. Implementers are encouraged to request registration of enum values following the procedures in Section 3.7.1. 913

914 NOTE: No attribute name exceeds 31 characters.

915 916	The standard attribute types are:	
917	jmAttributeTypeIndex	Datatype
918		
919		
920	other(1),	Integer32 (-22147483647)
921		AND/OR
922		OCTET STRING(SIZE(063))
923		attribute that is not in the
924	list and/or that has not been	approved and registered with
925	the PWG.	

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

926	+++++++++++++++++++++++++++++++++++++++
927	+ Job State attributes
928	+
929	
	+ The following attributes specify the state of a job.
930	+++++++++++++++++++++++++++++++++++++++
931	
932	jobStateReasons2(3), JmJobStateReasons2TC
933	INTEGER: Additional information about the job's current
934	state that augments the jmJobState object. See the
935	description under the JmJobStateReasons1TC textual-
936	convention.
937	
938	jobStateReasons3(4), JmJobStateReasons3TC
939	INTEGER: Additional information about the job's current
940	state that augments the jmJobState object. See the
941	description under JmJobStateReasons1TC textual-convention.
942	
943	jobStateReasons4(5), JmJobStateReasons4TC
944	INTEGER: Additional information about the job's current
945	state that augments the jmJobState object. See the
946	description under JmJobStateReasons1TC textual-convention.
947	
948	processingMessage(6), JmUTF8StringTC (SIZE(063))
949	OCTETS: MULTI-ROW: A coded character set message that is
950	generated by the server or device during the processing of
951	the job as a simple form of processing log to show progress
952	and any problems. The natural language of each value is
953	specified by the corresponding
954	processingMessageNaturalLangTag(7) value.
955	
956	NOTE - This attribute is intended for such conditions as
957	interpreter messages, rather than being the printable form
958	of the jmJobState and jmJobStateReasons1 objects and
959	jobStateReasons2, jobStateReasons3, and jobStateReasons4
960	attributes. In order to produce a localized printable form
961	of these job state objects/attribute, a management
962	application SHOULD produce a message from their enum and
963	bit values.
964	
965	NOTE - There is no job description attribute in IPP/1.0
966	that corresponds to this attribute and this attribute does
967	not correspond to the IPP/1.0 'job-state-message' job
968	description attribute, which is just a printable form of
969	the IPP 'job-state' and 'job-state-reasons' job attributes.
	the III job state and job state reasons job attributes.
970	
971	There is no restriction for the same message occurring in
972	multiple rows.

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

973	
974 975 976 977 978 979	<pre>processingMessageNaturalLangTag(7), OCTET STRING(SIZE(063)) OCTETS: MULTI-ROW: The natural language of the corresponding processingMessage(6) attribute value. See section 3.6.1, entitled 'Text generated by the server or device'.</pre>
980 981 982 983 984 985 986	If the agent does not know the natural language of the job processing message, the agent SHALL either (1) return a zero length string value for the processingMessageNaturalLangTag(7) attribute or (2) not return the processingMessageNaturalLangTag(7) attribute for the job.
980 987 988 989 990 991	There is no restriction for the same tag occurring in multiple rows, since when this attribute is implemented, it SHOULD have a value row for each corresponding processingMessage(6) attribute value row.
991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006	<pre>jobCodedCharSet(8), CodedCharSet INTEGER: The MIBenum identifier of the coded character set that the agent is using to represent coded character set objects and attributes of type 'JmJobStringTC'. These coded character set objects and attributes are either: (1) supplied by the job submitting client or (2) defaulted by the server or device when omitted by the job submitting client. The agent SHALL represent these objects and attributes in the MIB either (1) in the coded character set as they were submitted or (2) MAY convert the coded character set to another coded character set or encoding scheme as identified by the jobCodedCharSet(8) attribute. See section 3.6.2, entitled 'Text supplied by the job submitter'.</pre>
1006 1007 1008 1009 1010 1011 1012 1013	These MIBenum values are assigned by IANA [IANA-charsets] when the coded character sets are registered. The coded character set SHALL be one of the ones registered with IANA [IANA] and the enum value uses the CodedCharSet textual- convention from the Printer MIB. See the JmJobStringTC textual-convention.
1014 1015 1016 1017 1018	If the agent does not know what coded character set was used by the job submitting client, the agent SHALL either (1) return the 'unknown(2)' value for the jobCodedCharSet(8) attribute or (2) not return the jobCodedCharSet(8) attribute for the job.

	INTERNET-DRAFT	Job Monitori	ing MIB, V1. <u>3</u>	November 8, 1998
1019 1020 1021 1022 1023 1024 1025 1026	OCTETS: by the for the by the mediumRe	The natural lar job submitter or job, i.e., all	nguage of the job defaulted by the objects and attr textual-conventi See Section 3.6.	RING(SIZE(063)) o attributes supplied ne server or device ributes represented on, such as jobName, 2, entitled 'Text
1027 1028 1029 1030 1031 1032	by the return a jobNatu:	job submitting o a zero length st calLanguageTag(9		2) not return
1032 1033 1034 1035		ification attrik		-++++++++++++++++++++++++++++++++++++++
1036 1037 1038 1039	+ The follow + operator,	or an accountir	help an end user ng program identi	
1040 1041 1042 1043 1044		ier (URI) [RFC-1	OCTET STR ne job's Universa 1738]. See IPP [	
1045 1046 1047 1048	SNMP Get		n smaller values,	te this value on each rather than having
1049 1050 1051 1052				SHALL use multiple in the second value,
1053 1054 1055 1056	a URI, i			t maximum length for and HTTP/1.1 specify
1057 1058 1059 1060 1061 1062 1063	characte submitt or cate custome	Arbitrary bina er set data or e ing user for use gorize charges f r account name o	ary information we encrypted data su by accounting s for services prov or number.	services to allocate vided, such as a
1064 1065	NOTE: T	iis attribute NH	ED NOT be printa	able characters.

	INTERNET-DRAFT	Job Monitoring MIB, V1. <u>3</u>	November 8, 1998
1066 1067 1068 1069 1070 1071	OCTETS: name, nu that sub	nedJobName(22), JmJobStrin Configuration 3 only: The huma umber, or ID of the job as assign omitted the job to the device than ng access to with this MIB.	an readable string ned by the server
1072 1073 1074 1075 1076 1077	find his either t	This attribute is intended for en s/her job that a server submitted the client does not support the p ver does not pass the jmJobSubmis lee.	d to a device when jmJobSubmissionID or
1077 1078 1079 1080 1081 1082 1083	assigned distingu	JmJobStrin The human readable string name I by the submitting user to help wish between his/her various jobs I to be unique.	the user
1084 1085 1086 1087 1088	user's a on a sta notifica	cribute is intended for enabling application to convey a job name art sheet, returned in a query re ation or logging messages.	that MAY be printed esult, or used in
1089 1090 1091 1092 1093 1094 1095	submissi the ager time spe	to assist users to find their on protocols that don't supply a nt SHOULD maintain the jobName at ecified by the jmGeneralJobPersis than the (shorter) jmGeneralAttre	a jmJobSubmissionID, ttribute for the stence object,
1095 1096 1097 1098 1099 1100 1101	submitte specific document	attribute is not specified when ed, no job name is assumed, but is c defaults are allowed, such as t Name attribute of the first docu Name attribute of the first docu	implementation the value of the ument in the job or
1102 1103 1104 1105 1106 1107 1108 1109 1110 1111	attribut permit t jobs tha is inter user mig such as indicate	Name attribute is distinguished for the submitting user to distinguished for the submitting user to distinguish at he/she has submitted. The jok anded to be free form additional for the submission of the submission of the different set of input parameter different job submissions.	e is intended to sh between different oComment attribute information that a ith himself/herself, he results or to

T

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123	<pre>jobServiceTypes(24), JmJobServiceTypesTC INTEGER: Specifies the type(s) of service to which the job has been submitted (print, fax, scan, etc.). The service type 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. For example, a job service might scan, faxOut, and print a single job. In this case, three bits would be set in the jobServiceTypes attribute, corresponding to the hexadecimal values: 0x8 + 0x20 + 0x4, respectively, yielding: 0x2C.</pre>
1123 1124 1125 1126 1127 1128 1129 1130	Whether this attribute is set from a job attribute supplied by the job submission client or is set by the recipient job submission server or device depends on the job submission protocol. This attribute SHALL be implemented if the server or device has other types in addition to or instead of printing.
1130 1131 1132 1133 1134 1135	One of the purposes of this 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.
1136 1137 1138 1139 1140	<pre>jobSourceChannelIndex(25), Integer32 (02147483647) INTEGER: The index of the row in the associated Printer MIB[print-mib] of the channel which is the source of the print job.</pre>
1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151	<pre>jobSourcePlatformType(26), JmJobSourcePlatformTypeTC INTEGER: The source platform type of the immediate upstream submitter that submitted the job to the server (configuration 2) or device (configuration 1 and 3) to which the agent is providing access. For configuration 1, this is the type of the client that submitted the job to the device; for configuration 2, this is the type of the client that submitted the job to the server; and for configuration 3, this is the type of the server that submitted the job to the device.</pre>
1152 1153 1154 1155 1156 1157	<pre>submittingServerName(27), JmJobStringTC (SIZE(063)) OCTETS: For configuration 3 only: The administrative name of the server that submitted the job to the device. submittingApplicationName(28), JmJobStringTC (SIZE(063)) OCTETS: The name of the client application (not the server</pre>
1158 1159 1160	in configuration 3) that submitted the job to the server or device.

Job Monitoring MIB, V1.3 November 8, 1998

1161 jobOriginatingHost(29), JmJobStringTC (SIZE(0..63)) OCTETS: The name of the client host (not the server host 1162 1163 name in configuration 3) that submitted the job to the 1164 server or device. 1165 1166 JmJobStringTC (SIZE(0..63)) deviceNameRequested(30), OCTETS: The administratively defined coded character set 1167 name of the target device requested by the submitting user. 1168 1169 For configuration 1, its value corresponds to the Printer 1170 MIB[print-mib]: prtGeneralPrinterName object. For configuration 2 and 3, its value is the name of the logical 1171 or physical device that the user supplied to indicate to 1172 1173 the server on which device(s) they wanted the job to be 1174 processed. 1175 1176 queueNameRequested(31), JmJobStringTC (SIZE(0..63)) 1177 OCTETS: The administratively defined coded character set 1178 name of the target queue requested by the submitting user. 1179 For configuration 1, its value corresponds to the queue in 1180 the device for which the agent is providing access. For 1181 configuration 2 and 3, its value is the name of the queue 1182 that the user supplied to indicate to the server on which 1183 device(s) they wanted the job to be processed. 1184 1185 NOTE - typically an implementation SHOULD support either 1186 the deviceNameRequested or queueNameRequested attribute, 1187 but not both. 1188 1189 physicalDevice(32), hrDeviceIndex 1190 AND/OR 1191 JmUTF8StringTC (SIZE(0..63)) 1192 INTEGER: MULTI-ROW: The index of the physical device MIB 1193 instance requested/used, such as the Printer MIB[print-1194 mib]. This value is an hrDeviceIndex value. See the Host 1195 Resources MIB[hr-mib]. 1196 1197 AND/OR 1198 1199 OCTETS: MULTI-ROW: The name of the physical device to 1200 which the job is assigned. 1201 1202 numberOfDocuments(33), Integer32 (-2..2147483647) INTEGER: The number of documents in this job. 1203 1204 1205 The agent SHOULD return this attribute if the job has more 1206 than one document. 1207

	INTERNET-DRAFT	Job Monito:	ring MIB,	V1. <u>3</u>	<u>November 8</u> , 1998
1208 1209 1210 1211			The coded		TC (SIZE(063)) set file name or
1211 1212 1213 1214	There is multiple		on on the	same file 1	name occurring in
1215 1216 1217 1218	documentName OCTETS: document	MULTI-ROW: '			<pre>FC (SIZE(063)) set name of the</pre>
1218 1219 1220 1221	There is in multip		on on the	same docume	ent name occurring
1222 1223 1224 1225 1226 1227 1228 1229	string submittin submittin example, with the	An arbitrary upplied by the ng application a user might	human-rea e submitt: n program indicate ut or the	adable coded ing user or for any pur what he/she job submitt	rpose. For e is going to do ting application
1230 1231 1232		omment attribu ame attribute		t intended (	to be a name; see
1233 1234 1235 1236 1237 1238 1239	INTEGER: in the Pr language job requ:	rinter MIB[pr: (PDL) or con	The index int-mib] ( trol lange document	x in the pro of the page uage interp	D2147483647) InterpreterTable description reter that this AY use more than
1240 1241 1242 1243	are allo		ALL be on	ly one dist:	nere multiple rows inct row for each uplicates.
1243 1244 1245 1246 1247 1248	agent that if the ag	at implements	the Prin implemen	ter MIB and t the Printe	be used with an SHALL not be used er MIB. Such an ute instead.

T

1249 documentFormat(38), PrtInterpreterLangFamilyTC 1250 AND/OR 1251 OCTET STRING(SIZE(0..63)) 1252 INTEGER: MULTI-ROW: The interpreter language family 1253 corresponding to the Printer MIB[print-mib] 1254 prtInterpreterLangFamily object, that this job requires/uses. A document or a job MAY use more than one 1255 PDL or control language. 1256 1257 1258 AND/OR 1259 1260 OCTETS: MULTI-ROW: The document format registered as a 1261 media type[iana-media-types], i.e., the name of the MIME content-type/subtype. Examples: 'application/postscript', 1262 1263 'application/vnd.hp-PCL', 'application/pdf', 'text/plain' 1264 (US-ASCII SHALL be assumed), 'text/plain; charset=iso-8859-1', and 'application/octet-stream'. The IPP 'document-1265 format' job attribute uses these same values with the same 1266 1267 semantics. See the IPP [ipp-model] 'mimeMediaType' 1268 attribute syntax and the document-format attribute for 1269 further examples and explanation. 1270 1271 1272 + Job Parameter attributes 1273 + 1274 + The following attributes represent input parameters 1275 + supplied by the submitting client in the job submission 1276 + protocol. 1277 1278 1279 jobPriority(50), Integer32 (-2..100) 1280 INTEGER: The priority for scheduling the job. It is used 1281 by servers and devices that employ a priority-based 1282 scheduling algorithm. 1283 1284 A higher value specifies a higher priority. The value 1 is 1285 defined to indicate the lowest possible priority (a job which a priority-based scheduling algorithm SHALL pass over 1286 in favor of higher priority jobs). The value 100 is 1287 defined to indicate the highest possible priority. 1288 Priority is expected to be evenly or 'normally' distributed 1289 1290 across this range. The mapping of vendor-defined priority 1291 over this range is implementation-specific. -2 indicates 1292 unknown. 1293

1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306	jobProcessAfterDateAndTime(51), DateAndTime (SNMPv2-TC) OCTETS: 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 jmJobState object to pendingHeld and add the jobProcessAfterSpecified bit value to the job's jmJobStateReasons1 object. When the specified date and time arrives, the server SHALL remove the jobProcessAfterSpecified bit value from the job's jmJobStateReasons1 object and, if no other reasons remain, SHALL change the job's jmJobState object to pending.
1307	jobHold(52), JmBooleanTC
1308	INTEGER: If the value is 'true(4)', a client has
1309	explicitly specified that the job is to be held until
1310	explicitly released. Until the job is explicitly released
1311	by a client, the job SHALL be in the pendingHeld state with
1312	the jobHoldSpecified value in the jmJobStateReasons1
1313	attribute.
1314	
1315	jobHoldUntil(53), JmJobStringTC (SIZE(063))
1316	OCTETS: The named time period during which the job SHALL
1317	become a candidate for processing, such as 'evening',
1318	'night', 'weekend', 'second-shift', 'third-shift', etc.,
1319	(supported values configured by the system administrator).
1320	See IPP [ipp-model] for the standard keyword values. Until
1321	that time period arrives, the job SHALL be in the
1322	pendingHeld state with the jobHoldUntilSpecified value in
1323	the jmJobStateReasons1 object. The value 'no-hold' SHALL
1324	indicate explicitly that no time period has been specified;
1325	the absence of this attribute SHALL indicate implicitly
1326	that no time period has been specified.
1327	chat no time period has been specified.
1328	outputBin(54), Integer32 (02147483647)
1329	
	AND/OR
1330	JmJobStringTC (SIZE(063))
1331	INTEGER: MULTI-ROW: The output subunit index in the
1332	Printer MIB[print-mib]
1333	
1334	AND/OR
1335	
1336	OCTETS: MULTI-ROW: the name or number (represented as
1337	ASCII digits) of the output bin to which all or part of the
1338	job is placed in.
1339	

	INTERNET-DRAFT	Job Monito	oring MIB,	V1. <u>3</u>	November 8	, 1998
1340 1341 1342 1343		MULTI-ROW: ment in this	The numb	Integer32 (- er of sides, res/used.		, that
1344 1345 1346 1347			Type of	JmFinishingT finishing th		ment
1348 1349 1350 1351	++++++++++++++++++++++++++++++++++++++					++
1352 1353 1354	+ For device:	-++++++++++++	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++		++
1355 1356 1357 1358 1359		MULTI-ROW: cument in the	The prin	JmPrintQuali t quality se printers tha	lection req	
1360 1361 1362 1363 1364		MULTI-ROW:	The prin n the job	JmPrintQuali t quality se for printers	lection act	
1365 1366 1367 1368 1369	printerResolu OCTETS: document selection	MULTI-ROW: in the job i	The print	JmPrinterRes er resolutio rs that supp	n requested	
1370 1371 1372 1373 1374	by a doci	MULTI-ROW:	The print job for p	JmPrinterRes er resolutio rinters that	n actually	used
1375 1376 1377 1378 1379	for docur	MULTI-ROW:	The tone job for p	JmTonerEcono r economy se rinters that	lection req	
1380 1381 1382 1383 1384	used by a	MULTI-ROW:	The tone the job f	JmTonerEcono r economy se or printers	lection act	
1385 1386 1387 1388 1389 1390 1391	document density 100 is th range, SP	MULTI-ROW: in this job evels. Leve he highest de	The tone for devic el 1 is th ensity lev	Integer32 (- r density re es that can e lowest den el. Devices ge evenly on	quested for vary toner sity and le with a sma	vel

Bergman, Hastings, Isaacson, Lewis Informational [Page 35]

1392	
1393	tonerDensityUsed(77), Integer32 (-2100)
1394	INTEGER: MULTI-ROW: The toner density used by documents
1395	in this job for devices that can vary toner density levels.
1396	Level 1 is the lowest density and level 100 is the highest
1397	density level. Devices with a smaller range, SHALL map the
1398	1-100 range evenly onto the implemented range.
1399	1-100 fange evenity onto the impremented fange.
1400	
1401	+ Job Progress attributes (requested and consumed)
1402	
1403	+ Pairs of these attributes can be used by monitoring
1404	+ applications to show an indication of relative progress
1405	+ to users. See section 3.4, entitled 'Monitoring Job
1406	Progress'.
1407	+++++++++++++++++++++++++++++++++++++++
1408	
1409	jobCopiesRequested(90), Integer32 (-22147483647)
1410	INTEGER: The number of copies of the entire job that are
1411	to be produced.
1412	<b>-</b>
1413	jobCopiesCompleted(91), Integer32 (-22147483647)
1414	INTEGER: The number of copies of the entire job that have
1415	been completed so far.
1416	been completed bo ful.
1417	documentCopiesRequested(92), Integer32 (-22147483647)
1418	INTEGER: The total count of the number of document copies
1419	requested for the job as a whole. If there are documents
1420	A, B, and C, and document B is specified to produce 4
1421	
1422	copies, the number of document copies requested is 6 for
	the job.
1423	
1424	This attribute SHALL be used only when a job has multiple
1425	documents. The jobCopiesRequested attribute SHALL be used
1426	when the job has only one document.
1427	
1428	documentCopiesCompleted(93), Integer32 (-22147483647)
1429	INTEGER: The total count of the number of document copies
1430	completed so far for the job as a whole. If there are
1431	documents A, B, and C, and document B is specified to
1432	produce 4 copies, the number of document copies starts a 0
1433	and runs up to 6 for the job as the job processes.
1434	
1435	This attribute SHALL be used only when a job has multiple
1436	documents. The jobCopiesCompleted attribute SHALL be used
1437	when the job has only one document.
1438	

Job Monitoring MIB, V1.3 November 8, 1998

Integer32 (-2..2147483647) 1439 jobKOctetsTransferred(94), 1440 INTEGER: The number of K (1024) octets transferred to the 1441 server or device to which the agent is providing access. This count is independent of the number of copies of the 1442 job or documents that will be produced, but it is only a 1443 1444 measure of the number of bytes transferred to the server or 1445 device. 1446 1447 The agent SHALL round the actual number of octets 1448 transferred up to the next higher K. Thus 0 octets SHALL 1449 be represented as '0', 1-1024 octets SHALL BE represented as '1', 1025-2048 SHALL be '2', etc. When the job 1450 1451 completes, the values of the jmJobKOctetsPerCopyRequested 1452 object and the jobKOctetsTransferred attribute SHALL be 1453 equal. 1454 1455 NOTE - The jobKOctetsTransferred can be used with the 1456 jmJobKOctetsPerCopyRequested object in order to produce a 1457 relative indication of the progress of the job for agents 1458 that do not implement the jmJobKOctetsProcessed object. 1459 1460 sheetCompletedCopyNumber(95), Integer32 (-2..2147483647) 1461 INTEGER: The number of the copy being stacked for the 1462 current document. This number starts at 0, is set to 1 1463 when the first sheet of the first copy for each document is 1464 being stacked and is equal to n where n is the nth sheet 1465 stacked in the current document copy. See section 3.4, 1466 entitled 'Monitoring Job Progress'. 1467 1468 sheetCompletedDocumentNumber(96), Integer32 (-2..2147483647) 1469 INTEGER: The ordinal number of the document in the job 1470 that is currently being stacked. This number starts at 0, 1471 increments to 1 when the first sheet of the first document in the job is being stacked, and is equal to n where n is 1472 1473 the nth document in the job, starting with 1. 1474 1475 Implementations that only support one document jobs SHOULD 1476 NOT implement this attribute. 1477 JmJobCollationTypeTC 1478 jobCollationType(97), INTEGER: The type of job collation. See also Section 3.4, 1479 1480 entitled 'Monitoring Job Progress'. 1481

1482 1483 1484	++++++++++++++++++++++++++++++++++++++
1485 1486 1487	<pre>+ See the definition of the terms 'impression', 'sheet', + and 'page' in Section 2. +</pre>
1488 1489 1490 1491	+ See also jmJobImpressionsPerCopyRequested and + jmJobImpressionsCompleted objects in the jmJobTable. ++++++++++++++++++++++++++++++++++++
1492 1493 1494 1495	<pre>impressionsSpooled(110), Integer32 (-22147483647) INTEGER: The number of impressions spooled to the server or device for the job so far.</pre>
1496 1497 1498 1499	<pre>impressionsSentToDevice(111), Integer32 (-22147483647) INTEGER: The number of impressions sent to the device for the job so far.</pre>
1500 1501 1502 1503	<pre>impressionsInterpreted(112), Integer32 (-22147483647) INTEGER: The number of impressions interpreted for the job so far.</pre>
1503 1504 1505 1506 1507 1508 1509 1510 1511 1512	<pre>impressionsCompletedCurrentCopy(113),</pre>
1513 1514 1515	This value SHALL be reset to 0 for each document in the job and for each document copy.
1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529	<pre>fullColorImpressionsCompleted(114), Integer32 (-22147483647) INTEGER: The number of full color impressions completed by the device for this job so far. For printing, the impressions completed includes interpreting, marking, and stacking the output. For other types of job services, the number of impressions completed includes the number of impressions processed. Full color impressions are typically defined as those requiring 3 or more colorants, but this MAY vary by implementation. In any case, the value of this attribute counts by 1 for each side that has full color, not by the number of colors per side (and the other impression counters are incremented, except highlightColorImpressionsCompleted(115)).</pre>

1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543	<pre>highlightColorImpressionsCompleted(115),</pre>
1544 1545 1546	++++++++++++++++++++++++++++++++++++++
1547 1548 1549	+ + See the definition of 'impression', 'sheet', and 'page' + in Section 2.
1550 1551	***************************************
1552 1553 1554	pagesRequested(130), Integer32 (-22147483647) INTEGER: The number of logical pages requested by the job to be processed.
1555 1556 1557 1558	pagesCompleted(131), Integer32 (-22147483647) INTEGER: The number of logical pages completed for this job so far.
1559 1560 1561 1562 1563 1564 1565 1566	For implementations where multiple copies are produced by the interpreter with only a single pass over the data, the final value SHALL be equal to the value of the pagesRequested object. For implementations where multiple copies are produced by the interpreter by processing the data for each copy, the final value SHALL be a multiple of the value of the pagesRequested object.
1567 1568 1569 1570	NOTE - See the impressionsCompletedCurrentCopy and pagesCompletedCurrentCopy attributes for attributes that are reset on each document copy.
1571 1572 1573 1574 1575 1576 1577	NOTE - The pagesCompleted object can be used with the pagesRequested object to provide an indication of the relative progress of the job, provided that the multiplicative factor is taken into account for some implementations of multiple copies.

# INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

1578 1579 1580 1581 1582	pagesCompletedCurrentCopy(132), Integer32 (-22147483647) INTEGER: The number of logical pages completed for the current copy of the document so far. This value SHALL be reset to 0 for each document in the job and for each document copy.
1583	document copy.
1584	*****
1585	+ Sheet attributes
1586	+
1587	+ See the definition of 'impression', 'sheet', and 'page'
1588	+ in Section 2.
1589	+++++++++++++++++++++++++++++++++++++++
1590	
1591	sheetsRequested(150), Integer32 (-22147483647)
1592	INTEGER: The total number of medium sheets requested to be
1593	produced for this job.
1594	1 5
1595	Unlike the jmJobKOctetsPerCopyRequested and
1596	jmJobImpressionsPerCopyRequested attributes, the
1597	sheetsRequested(150) attribute SHALL include the
1598	multiplicative factor contributed by the number of copies
1599	and so is the total number of sheets to be produced by the
1600	job, as opposed to the size of the document(s) submitted.
1601	
1602	sheetsCompleted(151), Integer32 (-22147483647)
1603	INTEGER: The total number of medium sheets that have
1604	completed marking and stacking for the entire job so far
1605	whether those sheets have been processed on one side or on
1606	both.
1607	
1608	<pre>sheetsCompletedCurrentCopy(152), Integer32 (-22147483647)</pre>
1609	INTEGER: The number of medium sheets that have completed
1610	marking and stacking for the current copy of a document in
1611	the job so far whether those sheets have been processed on
1612	one side or on both.
1613	
1614	The value of this attribute SHALL be 0 before the job
1615	starts processing and SHALL be reset to 1 after the first
1616	sheet of each document and document copy in the job is
1617	processed and stacked.
1618	

Job Monitoring MIB, V1.3 November 8, 1998

1619 1620 + Resources attributes (requested and consumed) 1621 + 1622 + Pairs of these attributes can be used by monitoring 1623 + applications to show an indication of relative usage to 1624 + users, i.e., a 'thermometer'. 1625 1626 1627 mediumRequested(170), JmMediumTypeTC 1628 AND/OR 1629 JmJobStringTC (SIZE(0..63)) 1630 INTEGER: MULTI-ROW: The type 1631 AND/OR 1632 OCTETS: MULTI-ROW: the name of the medium that is 1633 required by the job. 1634 1635 NOTE - The name (JmJobStringTC) values correspond to the 1636 name values of the prtInputMediaName object in the Printer MIB [print-mib] and the name, size, and input tray values 1637 1638 of the IPP 'media' attribute [ipp-model]. 1639 1640 mediumConsumed(171), Integer32 (-2..2147483647) 1641 AND 1642 JmJobStringTC (SIZE(0..63)) 1643 INTEGER: MULTI-ROW: The number of sheets 1644 AND 1645 OCTETS: MULTI-ROW: the name of the medium that has been 1646 consumed so far whether those sheets have been processed on 1647 one side or on both. 1648 1649 This attribute SHALL have both Integer32 and OCTET STRING 1650 (represented as JmJobStringTC) values. 1651 1652 NOTE - The name (JmJobStringTC) values correspond to the name values of the prtInputMediaName object in the Printer 1653 1654 MIB [print-mib] and the name, size, and input tray values 1655 of the IPP 'media' attribute [ipp-model]. 1656 1657 Integer32 (-2..2147483647) colorantRequested(172), 1658 AND/OR 1659 JmJobStringTC (SIZE(0..63)) 1660 INTEGER: MULTI-ROW: The index (prtMarkerColorantIndex) in 1661 the Printer MIB[print-mib] 1662 AND/OR 1663 OCTETS: MULTI-ROW: the name of the colorant requested. 1664 1665 NOTE - The name (JmJobStringTC) values correspond to the 1666 name values of the prtMarkerColorantValue object in the 1667 Printer MIB. Examples are: red, blue.

1.000	
1668	Tr =
1669 1670	colorantConsumed(173), Integer32 (-22147483647)
1671	AND/OR JmJobStringTC (SIZE(063))
1672	
1673	the Printer MIB[print-mib]
1674	AND/OR
1675	OCTETS: MULTI-ROW: the name of the colorant consumed.
1676	NOTE The news (Intelectoring TO) and we according to the
1677	NOTE - The name (JmJobStringTC) values correspond to the
1678	name values of the prtMarkerColorantValue object in the
1679	Printer MIB. Examples are: red, blue
1680	$T_{\rm m} = \frac{1}{2} \left( \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \right)$
1681	mediumTypeConsumed(174), Integer32 (-22147483647)
1682	AND
1683	JmJobStringTC (SIZE(063))
1684	INTEGER: MULTI-ROW: The number of sheets of the indicated
1685	medium type that has been consumed so far whether those
1686	sheets have been processed on one side or on both
1687	AND
1688	OCTETS: MULTI-ROW: the name of that medium type.
1689	
1690	This attribute SHALL have both Integer32 and OCTET STRING
1691	(represented as JmJobStringTC) values.
1692	
1693	NOTE - The type name (JmJobStringTC) values correspond to
1694	the type name values of the prtInputMediaType object in the
1695	Printer MIB [print-mib]. Values are: 'stationery',
1696	'transparency', 'envelope', etc. These medium type names
1697	correspond to the enum values of JmMediumTypeTC used in the
1698	mediumRequested attribute.
1699	
1700	mediumSizeConsumed(175), Integer32 (-22147483647)
1701	AND
1702	JmJobStringTC (SIZE(063))
1703	INTEGER: MULTI-ROW: The number of sheets of the indicated
1704	medium size that has been consumed so far whether those
1705	sheets have been processed on one side or on both
1706	AND
1707	OCTETS: MULTI-ROW: the name of that medium size.
1708	
1709	This attribute SHALL have both Integer32 and OCTET STRING
1710	(represented as JmJobStringTC) values.
1711	
1712	NOTE - The size name (JmJobStringTC) values correspond to
1713	the size name values in the Printer MIB [print-mib]
1714	Appendix B. These size name values are also a subset of
1715	the keyword values defined by [ipp-model] for the 'media'
1716	Job Template attribute. Values are: 'letter', 'a', 'iso-
1717	a4', 'jis-b4', etc.
1718	

1719	+++++++++++++++++++++++++++++++++++++++
1720	+ Time attributes (set by server or device)
1721	+
1722	+ This section of attributes are ones that are set by the
1723	+ server or device that accepts jobs. Two forms of time are
1724	+ provided. Each form is represented in a separate attribute.
1725	+ See section 3.1.2 and section 3.1.3 for the
1726	+ conformance requirements for time attribute for agents and
1727	+ monitoring applications, respectively. The two forms are:
1728	+
1729	+ 'DateAndTime' is an 8 or 11 octet binary encoded year,
1730	+ month, day, hour, minute, second, deci-second with
1731	+ optional offset from UTC. See SNMPv2-TC [SMIv2-TC].
1732	+
1733	+ NOTE: 'DateAndTime' is not printable characters; it is
	+ binary.
1734	-
1735	+
1736	+ 'JmTimeStampTC' is the time of day measured in the number of
1737	+ seconds since the system was booted.
1738	***************************************
1739	
1740	jobSubmissionToServerTime(190), JmTimeStampTC
1741	AND/OR
1742	DateAndTime
1743	INTEGER: Configuration 3 only: The time
1744	AND/OR
1745	OCTETS: the date and time that the job was submitted to
1746	the server (as distinguished from the device which uses
1747	jobSubmissionTime).
1748	
1749	jobSubmissionTime(191), JmTimeStampTC
1750	AND/OR
1751	DateAndTime
1752	INTEGER: Configurations 1, 2, and 3: The time
1753	AND/OR
1754	OCTETS: the date and time that the job was submitted to
1755	the server or device to which the agent is providing
1756	access.
1757	
1758	jobStartedBeingHeldTime(192), JmTimeStampTC
1759	AND/OR
1760	DateAndTime
1761	INTEGER: The time
1762	AND/OR
1763	
	OCTETS: the date and time that the job last entered the
1764	pendingHeld state. If the job has never entered the
1765	pendingHeld state, then the value SHALL be '0' or the
1766	attribute SHALL not be present in the table.

1767	
1768	jobStartedProcessingTime(193), JmTimeStampTC
1769	AND/OR
1770	DateAndTime
1771	INTEGER: The time
1772	AND/OR
1773	OCTETS: the date and time that the job started processing.
1774	
1775	jobCompletionTime(194), JmTimeStampTC
1776	AND/OR
1777	DateAndTime
1778	INTEGER: The time
1779	AND/OR
1780	OCTETS: the date and time that the job entered the
1781	completed, canceled, or aborted state.
1782	
1783	jobProcessingCPUTime(195) Integer32 (-22147483647)
1784	UNITS 'seconds'
1785	INTEGER: The amount of CPU time in seconds that the job
1786	has been in the processing state. If the job enters the
1787	processingStopped state, that elapsed time SHALL not be
1788	included. In other words, the jobProcessingCPUTime value
1789	SHOULD be relatively repeatable when the same job is
1790	processed again on the same device.
1791	3.3.9 Job State Reason bit definitions

1792 The JmJobStateReasonsNTC (N=1..4) textual-conventions are used with the jmJobStateReasons1 object and jobStateReasonsN (N=2..4), respectively, 1793 1794 to provide additional information regarding the current jmJobState 1795 object value. These values MAY be used with any job state or states 1796 for which the reason makes sense.

NOTE - While values cannot be added to the jmJobState object without 1797 1798 impacting deployed clients that take actions upon receiving jmJobState values, it is the intent that additional JmJobStateReasonsNTC enums can 1799 be defined and registered without impacting such deployed clients. In 1800 1801 other words, the jmJobStateReasons1 object and jobStateReasonsN 1802 attributes are intended to be extensible.

1803 NOTE - The Job Monitoring MIB contains a superset of the IPP values[ipp-model] for the IPP 'job-state-reasons' attribute, since the 1804 Job Monitoring MIB is intended to cover other job submission protocols 1805 as well. Also some of the names of the reasons have been changed from 1806 'printer' to 'device', since the Job Monitoring MIB is intended to 1807 1808 cover additional types of devices, including input devices, such as 1809 scanners.

Bergman, Hastings, Isaacson, Lewis Informational [Page 44]

## 1810 **3.3.9.1 JmJobStateReasons1TC specification**

	e following standard values are defined (in hexadecimal) as powers of
12 tw	o, since multiple values MAY be used at the same time. For ease of
13 un	derstanding, the JmJobStateReasons1TC reasons are presented in the
14 or	der in which the reasons are likely to occur (if implemented),
	arting with the 'jobIncoming' value and ending with the
16 'j	obCompletedWithErrors' value.
17	
18	other 0x1
19	The job state reason is not one of the standardized or
20	registered reasons.
21	
2	unknown 0x2
}	The job state reason is not known to the agent or is
	indeterminent.
	jobIncoming 0x4
	The job has been accepted by the server or device, but the
	server or device is expecting (1) additional operations
	from the client to finish creating the job and/or (2) is
	accessing/accepting document data.
	accessing/accepting document data.
	submissionInterrupted 0x8
	The job was not completely submitted for some unforeseen
	reason, such as: (1) the server has crashed before the job
	was closed by the client, (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, (3) the client crashed or failed to close the job
	before the time-out period.
	jobOutgoing 0x10
	Configuration 2 only: The server is transmitting the job
	to the device.
	jobHoldSpecified 0x20
	The value of the job's jobHold(52) attribute is TRUE. The
	job SHALL NOT be a candidate for processing until this
	reason is removed and there are no other reasons to hold
	the job.
	jobHoldUntilSpecified 0x40
	The value of the job's jobHoldUntil(53) attribute specifies
	a time period that is still in the future. The job SHALL
	NOT be a candidate for processing until this reason is
	removed and there are no other reasons to hold the job.
	jobProcessAfterSpecified 0x80
	The value of the job's jobProcessAfterDateAndTime(51)
	attribute specifies a time that is still in the future.
Б.	reman Usetinge Tesseen Lowie Informational [Dago 45]

Bergman, Hastings, Isaacson, Lewis Informational [Page 45]

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

1860	The job SHALL NOT be a candidate for processing until this
1861	reason is removed and there are no other reasons to hold
1862	the job.

1863

1864	resourcesAreNotReady 0x100
1865	At least one of the resources needed by the job, such as
1866	media, fonts, resource objects, etc., is not ready on any
1867	of the physical devices for which the job is a candidate.
1868	This condition MAY be detected when the job is accepted, or
1869	subsequently while the job is pending or processing,
1870	depending on implementation.
1871	
1872	deviceStoppedPartly 0x200
1873	One or more, but not all, of the devices to which the job
1874	is assigned are stopped. If all of the devices are stopped
1875	(or the only device is stopped), the deviceStopped reason
1876	SHALL be used.
1877	
1878	deviceStopped 0x400
1879	The device(s) to which the job is assigned is (are all)
1880	stopped.
1881	
1882	jobInterpreting 0x800
1883	The device to which the job is assigned is interpreting the
1884	document data.
1885	
1886	jobPrinting 0x1000
1887	The output device to which the job is assigned is marking
1888	media. This value is useful for servers and output devices
1889	which spend a great deal of time processing (1) when no
1890	marking is happening and then want to show that marking is
1891	now happening or (2) when the job is in the process of
1892	being canceled or aborted while the job remains in the
1893	processing state, but the marking has not yet stopped so
1894	that impression or sheet counts are still increasing for
1895	the job.
1896	
1897	jobCanceledByUser 0x2000
1898	The job was canceled by the owner of the job, i.e., by a
1899	user whose name is the same as the value of the job's
1900	jmJobOwner object, or by some other authorized end-user,
1901	such as a member of the job owner's security group.
1902	
1903	jobCanceledByOperator 0x4000
1904	The job was canceled by the operator, i.e., by a user who
1905	has been authenticated as having operator privileges
1906	(whether local or remote).
1907	· · · · · · · · · · · · · · · · · · ·
1908	jobCanceledAtDevice 0x8000
1909	The job was canceled by an unidentified local user, i.e., a
1910	user at a console at the device.
1911	

1912 1913 1914 1915 1916 1917 1918	abortedBySystem 0x10000 The job (1) is in the process of being aborted, (2) has been aborted by the system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the 'pendingHeld' state, so that a user or operator can manually try the job again.
1919 1920 1921 1922 1923 1924	processingToStopPoint 0x20000 The requester has issued an operation to cancel or interrupt the job or the server/device has aborted the job, but the server/device is still performing some actions on the job until a specified stop point occurs or job termination/cleanup is completed.
1925 1926 1927 1928 1929	This reason is recommended to be used in conjunction with the processing job state to indicate that the server/device is still performing some actions on the job while the job remains in the processing state. After all the job's
1930 1931 1932 1933 1934	resources consumed counters have stopped incrementing, the server/device moves the job from the processing state to the canceled or aborted job states. serviceOffLine 0x40000
1935 1936 1937 1938 1939	The service or document transform is off-line and accepting no jobs. All pending jobs are put into the pendingHeld state. This situation could be true if the service's or document transform's input is impaired or broken.
1939 1940 1941 1942 1943	jobCompletedSuccessfully 0x80000 The job completed successfully. jobCompletedWithWarnings 0x100000
1944 1945 1946 1947	The job completed with warnings.jobCompletedWithErrors0x200000The job completed with errors (and possibly warnings too).
1948 1949 1950 1951 1952	The following additional job state reasons have been added to represent job states that are in ISO DPA[iso-dpa] and other job submission protocols:
1952 1953 1954 1955 1956 1957 1958 1959 1960 1961	jobPaused 0x40000 The job has been indefinitely suspended by a client issuing an operation to suspend the job so that other jobs may proceed using the same devices. The client MAY issue an operation to resume the paused job at any time, in which case the agent SHALL remove the jobPaused values from the job's jmJobStateReasons1 object and the job is eventually resumed at or near the point where the job was paused.

Bergman, Hastings, Isaacson, Lewis Informational [Page 48]

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

1962	jobInterrupted 0x800000
1963	The job has been interrupted while processing by a client
1964	issuing an operation that specifies another job to be run
1965	instead of the current job. The server or device will
1966	automatically resume the interrupted job when the
1967	interrupting job completes.
1968	
1969	jobRetained 0x1000000
1970	The job is being retained by the server or device with all
1971	of the job's document data (and submitted resources, such
1972	as fonts, logos, and forms, if any). Thus a client could
1973	issue an operation to the server or device to either (1)
1974	re-do the job (or a copy of the job) on the same server or
1975	device or (2) resubmit the job to another server or device.
1976	When a client could no longer re-do/resubmit the job, such
1977	as after the document data has been discarded, the agent
1978	SHALL remove the jobRetained value from the
1979	jmJobStateReasons1 object.
1980	
1981	These bit definitions are the equivalent of a type 2 enum except that
1982	combinations of bits may be used together. See section 3.7.1.2. The
1983	remaining bits are reserved for future standardization and/or
1984	registration.
TJOI	
1005	2.2.0.0 To Tabdhaha Daga son s0000 son sifi sabian
1985	3.3.9.2 JmJobStateReasons2TC specification
1986	The following standard values are defined (in hexadecimal) as powers of
1986 1987	
1986 1987 1988	The following standard values are defined (in hexadecimal) as <i>powers of</i> two, since multiple values MAY be used at the same time.
1986 1987 1988 1989	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time.
1986 1987 1988 1989 1990	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded 0x1</u> An outbound gateway has transmitted all of the job's job
1986 1987 1988 1989	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded 0x1</u> An outbound gateway has transmitted all of the job's job
1986 1987 1988 1989 1990 1991	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. Cascaded 0x1 An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling
1986 1987 1988 1989 1990 1991 1992	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded 0x1</u> An outbound gateway has transmitted all of the job's job
1986 1987 1988 1989 1990 1991 1992 1993	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded 0x1</u> <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator 0x2</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded 0x1</u> <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator 0x2</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> 0x2 <u>The administrator has deleted the job.</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> 0x2 <u>The administrator has deleted the job.</u> <u>discardTimeArrived</u> 0x4
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> <u>0x1</u> <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> <u>0x2</u> <u>The administrator has deleted the job.</u> <u>discardTimeArrived</u> <u>0x4</u> <u>The job has been deleted due to the fact that the time    </u>
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> 0x2 <u>The administrator has deleted the job.</u> <u>discardTimeArrived</u> 0x4 <u>The job has been deleted due to the fact that the time</u> <u>specified by the job's job-discard-time attribute has</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> <u>0x1</u> <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> <u>0x2</u> <u>The administrator has deleted the job.</u> <u>discardTimeArrived</u> <u>0x4</u> <u>The job has been deleted due to the fact that the time    </u>
1986 1987 1988 1999 1990 1991 1993 1994 1995 1996 1997 1998 1999 2000 2001	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> 0x2 <u>The administrator has deleted the job.</u> <u>discardTimeArrived</u> 0x4 <u>The job has been deleted due to the fact that the time</u> <u>specified by the job's job-discard-time attribute has</u>
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> 0x1 <u>An outbound gateway has transmitted all of the job's job</u> <u>and document attributes and data to another spooling</u> <u>system.</u> <u>deletedByAdministrator</u> 0x2 <u>The administrator has deleted the job.</u> <u>discardTimeArrived</u> 0x4 <u>The job has been deleted due to the fact that the time</u> <u>specified by the job's job-discard-time attribute has</u>
1986 1987 1988 1999 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time.         cascaded       0x1         An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system.         deletedByAdministrator       0x2         The administrator has deleted the job.         discardTimeArrived       0x4         The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived.         postProcessingFailed       0x8
1986 1987 1988 1990 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time.         cascaded       0x1         An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system.         deletedByAdministrator       0x2         The administrator has deleted the job.         discardTimeArrived       0x4         The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived.         postProcessingFailed       0x8         The post-processing agent failed while trying to log
1986 1987 1988 1999 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time.         cascaded       0x1         An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system.         deletedByAdministrator       0x2         The administrator has deleted the job.         discardTimeArrived       0x4         The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived.         postProcessingFailed       0x8         The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has
1986 1987 1988 1999 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time.         cascaded       0x1         An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system.         deletedByAdministrator       0x2         The administrator has deleted the job.         discardTimeArrived       0x4         The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived.         postProcessingFailed       0x8         The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has been placed into the completed state with the jobRetained
1986 1987 1988 1999 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time.         cascaded       0x1         An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system.         deletedByAdministrator       0x2         The administrator has deleted the job.         discardTimeArrived       0x4         The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived.         postProcessingFailed       0x8         The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has
1986 1987 1988 1999 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system. <u>deletedByAdministrator</u> The administrator has deleted the job. <u>discardTimeArrived</u> The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived. <u>postProcessingFailed</u> <u>Dx8</u> The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has been placed into the completed state with the jobRetained jmJobStateReasons1 object value for a system-defined period
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2001 2002 2003 2004 2005 2006 2007	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. Cascaded 0x1 An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system. deletedByAdministrator 0x2 The administrator has deleted the job. discardTimeArrived 0x4 The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived. postProcessingFailed 0x8 The post-processing agent failed while trying to log accounting attributes for the job' therefore the job has been placed into the completed state with the jobRetained jmJobStateReasons1 object value for a system-defined period of time, so the administrator can examine it, resubmit it,
1986 1987 1988 1999 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. <u>cascaded</u> An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system. <u>deletedByAdministrator</u> The administrator has deleted the job. <u>discardTimeArrived</u> The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived. <u>postProcessingFailed</u> <u>Dx8</u> The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has been placed into the completed state with the jobRetained jmJobStateReasons1 object value for a system-defined period
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2001 2002 2003 2004 2005 2006 2007	The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. Cascaded 0x1 An outbound gateway has transmitted all of the job's job and document attributes and data to another spooling system. deletedByAdministrator 0x2 The administrator has deleted the job. discardTimeArrived 0x4 The job has been deleted due to the fact that the time specified by the job's job-discard-time attribute has arrived. postProcessingFailed 0x8 The post-processing agent failed while trying to log accounting attributes for the job' therefore the job has been placed into the completed state with the jobRetained jmJobStateReasons1 object value for a system-defined period of time, so the administrator can examine it, resubmit it,

Bergman, Hastings, Isaacson, Lewis Informational [Page 49]

0.01.0	
2010	jobTransforming 0x10
2011	The server/device is interpreting document data and
2012	producing another electronic representation.
2013	
2014	maxJobFaultCountExceeded 0x20
2015	The job has faulted several times and has exceeded the
2016	administratively defined fault count limit.
2017	
2018	devicesNeedAttentionTimeOut 0x40
2019	One or more document transforms that the job is using needs
2020	
	human intervention in order for the job to make progress,
2021	but the human intervention did not occur within the site-
2022	settable time-out value.
2023	
2024	needsKeyOperatorTimeOut 0x80
2025	One or more devices or document transforms that the job is
2026	using need a specially trained operator (who may need a key
2027	to unlock the device and gain access) in order for the job
2028	to make progress, but the key operator intervention did not
2029	occur within the site-settable time-out value.
2030	
2031	jobStartWaitTimeOut 0x100
2032	The server/device has stopped the job at the beginning of
2033	processing to await human action, such as installing a
2034	special cartridge or special non-standard media, but the
2035	job was not resumed within the site-settable time-out value
2035	and the server/device has transitioned the job to the
2030	
	pendingHeld state.
2038	
2039	jobEndWaitTimeOut 0x200
2040	The server/device has stopped the job at the end of
2041	processing to await human action, such as removing a
2042	special cartridge or restoring standard media, but the job
2043	was not resumed within the site-settable time-out value and
2044	the server/device has transitioned the job to the completed
2045	state.
2046	
2047	jobPasswordWaitTimeOut 0x400
2048	The server/device has stopped the job at the beginning of
2049	processing to await input of the job's password, but the
2050	password was not received within the site-settable time-out
2051	value.
2052	
2053	deviceTimedOut 0x800
2054	A device that the job was using has not responded in a
2055	period specified by the device's site-settable attribute.
2055	Period presided by the device p pice-periodic attitude.
2057	connectingToDeviceTimeOut 0x1000
2057	
	The server is attempting to connect to one or more devices
2059	which may be dial-up, polled, or queued, and so may be busy
2060	with traffic from other systems, but server was unable to

Bergman, Hastings, Isaacson, Lewis Informational [Page 50]

2061	connect to the device within the site-settable time-out
2062	value.
2063	
2064	transferring 0x2000
2065	The job is being transferred to a down stream server or
2066	downstream device.
2067	
2068	queuedInDevice 0x4000
2069	The server/device has queued the job in a down stream
2070	server or downstream device.
2071	
2072	jobQueued 0x8000
2073	The server/device has queued the document data.
2074	
2075	jobCleanup 0x10000
2076	The server/device is performing cleanup activity as part of
2077	ending normal processing.
2078	
2079	jobPasswordWait 0x20000
2080	The server/device has selected the job to be next to
2081	process, but instead of assigning resources and starting
2082	the job processing, the server/device has transitioned the
2083	job to the pendingHeld state to await entry of a password
2084	(and dispatched another job, if there is one).
2085	
2086	validating 0x40000
2087	The server/device is validating the job after accepting the
2088	job.
2089	
2090	queueHeld 0x80000
2091	The operator has held the entire job set or queue.
2092	
2093	jobProofWait 0x100000
2094	The job has produced a single proof copy and is in the
2095	pendingHeld state waiting for the requester to issue an
2096 2097	operation to release the job to print normally, obeying any
	job and document copy attributes that were originally
2098 2099	submitted.
2099 2100	heldForDiagnostics 0x200000
2100	heldForDiagnostics 0x200000 The system is running intrusive diagnostics, so that all
2101 2102	
ZIUZ	jobs are being held.

2103	noSpaceOnServer 0x800000
2104	There is no room on the server to store all of the job.
2105	
2106	pinRequired 0x1000000
2107	The System Administrator settable device policy is (1) to
2108	require PINs, and (2) to hold jobs that do not have a pin
2109	supplied as an input parameter when the job was created.
2110	
2111	exceededAccountLimit 0x2000000
2112	The account for which this job is drawn has exceeded its
2113	limit. This condition SHOULD be detected before the job is
2114	scheduled so that the user does not wait until his/her job
2115	is scheduled only to find that the account is overdrawn.
2116	This condition MAY also occur while the job is processing
2117	either as processing begins or part way through processing.
2118	
2119	heldForRetry 0x4000000
2120	The job encountered some errors that the server/device
2121	could not recover from with its normal retry procedures,
2122	but the error might not be encountered if the job is
2123	processed again in the future. Example cases are phone
2124	number busy or remote file system in-accessible. For such
2125	a situation, the server/device SHALL transition the job
2126	from the processing to the pendingHeld, rather than to the
2127	aborted state.
2128	
2129	The following values are from the X/Open PSIS draft standard:
2130	
2131	canceledByShutdown 0x8000000
2132	The job was canceled because the server or device was
2133	shutdown before completing the job.
2134	
2135	deviceUnavailable 0x1000000
2136	This job was aborted by the system because the device is
2137	currently unable to accept jobs.
2138	
2139	wrongDevice 0x2000000
2140	This job was aborted by the system because the device is
2141	unable to handle this particular job; the spooler SHOULD
2142	try another device or the user should submit the job to
2143 2144	another device.
	bed Teb 04000000
2145	badJob 0x4000000
2146	This job was aborted by the system because this job has a
2147	major problem, such as an ill-formed PDL; the spooler
2148	SHOULD not even try another device.
2149	
2150	These bit definitions are the equivalent of a type 2 enum except that
2150 2151	These bit definitions are the equivalent of a type 2 enum except that combinations of them may be used together. See section 3.7.1.2.
TCTP	Comprime tons of them may be used together. See Section 3.7.1.2.

### 2152 3.3.9.3 JmJobStateReasons3TC specification

2153 2154 2155 2156 2157	This textual-convention is used with the jobStateReasons3 attribute to provides additional information regarding the jmJobState object. The following standard values are defined (in hexadecimal) as powers of two, since multiple values may be used at the same time:
2157 2158 2159 2160 2161 2162	jobInterruptedByDeviceFailure 0x1 A device or the print system software that the job was using has failed while the job was processing. The server or device is keeping the job in the pendingHeld state until an operator can determine what to do with the job.
2163 2164 2165 2166	These bit definitions are the equivalent of a type 2 enum except that combinations of them may be used together. See section 3.7.1.2. The remaining bits are reserved for future standardization and/or registration.
2167	3.3.9.4 JmJobStateReasons4TC specification
2168 2169 2170 2171 2172 2173	This textual-convention is used with the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. The following standard values are defined (in hexadecimal) as powers of two, since multiple values MAY be used at the same time. None defined at this time.
2174	These bit definitions are the equivalent of a type 2 enum except that

2175 combinations of them may be used together. See section 3.7.1.2. The 2176 remaining bits are reserved for future standardization and/or 2177 registration.

### 2178 3.4 Monitoring Job Progress

2179 There are a number of objects and attributes for monitoring the progress of a job. These objects and attributes count the number of K 2180 2181 octets, impressions, sheets, and pages requested or completed. For impressions and sheets, "completed" means stacked, unless the 2182 implementation is unable to detect when each sheet is stacked, in which 2183 2184 case stacked is approximated when processing of each sheet completes. There are objects and attributes for the overall job and for the 2185 current copy of the document currently being stacked. For the latter, 2186 2187 the rate at which the various objects and attributes count depends on 2188 the sheet and document collation of the job.

2189 Job Collation included sheet collation and document collation. Sheet collation is defined to be the ordering of sheets within a document 2190 2191 copy. Document collation is defined to be ordering of document copies 2192 within a multi-document job. There are three types of job collation (see terminology definitions in Section 2): 2193

Bergman, Hastings, Isaacson, Lewis Informational [Page 53] Job Monitoring MIB, V1.3 November 8, 1998

2194 1. uncollatedSheets(3) - No collation of the sheets within each document copy, i.e., each sheet of a document that is to 2195 2196 produce multiple copies is replicated before the next sheet in 2197 the document is processed and stacked. If the device has an output bin collator, the uncollatedSheets(3) value may actually 2198 2199 produce collated sheets as far as the user is concerned (in the output bins). However, when the job collation is the 2200 'uncollatedSheets(3)' value, job progress is indistinguishable 2201 to a monitoring application between a device that has an output 2202 2203 bin collator and one that does not. 2. collatedDocuments(4) - Collation of the sheets within each 2204 document copy is performed within the printing device by making 2205 2206 multiple passes over either the source or an intermediate representation of the document. In addition, when there are multiple documents per job, the i'th copy of each document is 2207 2208 stacked before the j'th copy of each document, i.e., the 2209 documents are collated within each job copy. For example, if a 2210 job is submitted with documents, A and B, the job is made 2211 2212 available to the end user as: A, B, A, B, .... The 2213 'collatedDocuments(4)' value corresponds to the IPP [ipp-model] 2214 'separate-documents-collated-copies' value of the "multiple-2215 document-handling" attribute. 2216 2217 If jobCopiesRequested or documentCopiesRequested = 1, then 2218 jobCollationType is defined as 4. 2219 3. uncollatedDocuments(5) - Collation of the sheets within each document copy is performed within the printing device by making 2220 multiple passes over either the source or an intermediate 2221 2222 representation of the document. In addition, when there are multiple documents per job, all copies of the first document in 2223 2224 the job are stacked before the any copied of the next document 2225 in the job, i.e., the documents are uncollated within the job. 2226 For example, if a job is submitted with documents, A and B, the 2227 job is mad available to the end user as: A, A, ..., B, B, .... 2228 The 'uncollatedDocuments(5)' value corresponds to the IPP [ippmodel] 'separate-documents-uncollated-copies' value of the 2229 2230 "multiple-document-handling" attribute. Consider the following four variables that are used to monitor the 2231 2232 progress of a job's impressions: 2233 1. jmJobImpressionsCompleted - counts the total number of 2234 impressions stacked for the job 2235 2. impressionsCompletedCurrentCopy - counts the number of 2236 impressions stacked for the current document copy 2237 3. sheetCompletedCopyNumber - identifies the number of the copy for the current document being stacked where the first copy is 2238 2239 1. Bergman, Hastings, Isaacson, Lewis Informational [Page 54]

4. sheetCompletedDocumentNumber - identifies the current document 2240 2241 within the job that is being stacked where the first document 2242 in a job is 1. NOTE: this attribute SHOULD NOT be implemented for implementations that only support one document per job. 2243

2244 For each of the three types of job collation, a job with three copies of two documents (1, 2), where each document consists of 3 impressions, 2245 the four variables have the following values as each sheet is stacked 2246 2247 for one-sided printing:

### 2248

Job Collation Type = uncollatedSheets(3)

2	2	4	9	
~	~	-	-	

jmJobImpressions Completed	Impressions CompletedCurrent Copy	sheetCompleted CopyNumber	sheetCompleted DocumentNumber
0	0	0	0
1 2	⊥ 1	1 2	⊥ 1
3	1	3	1
4	2	1	1
5	2	2	1
б	2	3	1
7	3	1	1
8	3	2	1
9	3	3	1
10	1	1	2
11	1	2	2
12	1	3	2
13	2	1	2
14	2	2	2
15	2	3	2
16	3	1	2
17	3	2	2
18	3	3	2

2250

Bergman, Hastings, Isaacson, Lewis Informational [Page 56]

### 2251

Job Collation Type = collatedDocuments(4)

2252

JmJobImpressions Completed	Impressions CompletedCurrent Copy	sheetCompleted CopyNumber	sheetCompleted DocumentNumber
0	0	0	0
1	1	1	1
2	2	1	1
3	3	1	1
4	1	1	2
5	2	1	2
6	3	1	2
7	1	2	1
8	2	2	1
9	3	2	1
10	1	2	2
11	2	2	2
12	3	2	2
13	1	3	1
14	2	3	1
15	3	3	1
16	1	- 3	2
17	2	3	2
18	3	3	2

2253

Bergman, Hastings, Isaacson, Lewis Informational [Page 57]

2254	
2255	

Job Collation Type = uncollatedDocuments(5)

 jmJobImpressions Completed	Impressions CompletedCurrent Copy	sheetCompleted CopyNumber	sheetCompleted DocumentNumber
0	0	0	0
1	1	1	1
2	2	1	1
3	3	1	1
4	1	2	1
5	2	2	1
б	3	2	1
7	1	3	1
8	2	3	1
9	3	3	1
10	1	1	2
11	2	1	2
12	3	1	2
13	1	2	2
14	2	2	2
15	3	2	2
16	1	3	2
17	2	3	2
18	3	3	2

### 2256

#### 3.5 Job Identification 2257

2258 There are a number of attributes that permit a user, operator or system 2259 administrator to identify jobs of interest, such as jobURI, jobName, jobOriginatingHost, etc. In addition, there is a jmJobSubmissionID 2260 2261 object that is a text string table index. Being a table index allows a 2262 monitoring application to quickly locate and identify a particular job 2263 of interest that was submitted from a particular client by the user 2264 invoking the monitoring application without having to scan the entire job table. The Job Monitoring MIB needs to provide for identification 2265 2266 of the job at both sides of the job submission process. The primary 2267 identification point is the client side. The jmJobSubmissionID allows the monitoring application to identify the job of interest from all the 2268 2269 jobs currently "known" by the server or device. The value of 2270 jmJobSubmissionID can be assigned by either the client's local system 2271 or a downstream server or device. The point of assignment depends on 2272 the job submission protocol in use.

2273 The server/device-side identifier, called the jmJobIndex object, SHALL 2274 be assigned by the SNMP Job Monitoring MIB agent when the server or device accepts the jobs from submitting clients. The jmJobIndex object 2275 2276 allows the interested party to obtain all objects desired that relate 2277 to a particular job. See Section 3.2, entitled 'The Job Tables and the

Bergman, Hastings, Isaacson, Lewis Informational [Page 58]

2278 Oldest Active and Newest Active Indexes' for the specification of how 2279 the agent SHALL assign the jmJobIndex values.

The MIB provides a mapping table that maps each jmJobSubmissionID value 2280 to a corresponding jmJobIndex value generated by the agent, so that an 2281 2282 application can determine the correct value for the jmJobIndex value for the job of interest in a single Get operation, given the Job 2283 2284 Submission ID. See the jmJobIDGroup.

2285 In some configurations there may be more than one application program 2286 that monitors the same job when the job passes from one network entity to another when it is submitted. See configuration 3. When there are 2287 multiple job submission IDs, each entity MAY supply an appropriate 2288 2289 jmJobSubmissionID value. In this case there would be a separate entry in the jmJobSubmissionID table, one for each jmJobSubmissionID. All 2290 2291 entries would map to the same jmJobIndex that contains the job data. 2292 When the job is deleted, it is up to the agent to remove all entries that point to the job from the jmJobSubmissionID table as well. 2293

2294 The jobName attribute provides a name that the user supplies as a job 2295 attribute with the job. The jobName attribute is not necessarily 2296 unique, even for one user, let alone across users.

2297 3.5.1 The Job Submission ID specifications

This section specifies the formats for each of the registered Job 2298 2299 Submission Ids. This format is used by the JmJobSubmissionIDTypeTC. Each job submission ID is a fixed-length, 48-octet printable US-ASCII 2300 [US-ASCII] coded character string containing no control characters, 2301 2302 consisting of the following fields:

2303

2304	octet 1: The format letter identifying the format. The US-
2305	ASCII characters '0-9', 'A-Z', and 'a-z' are assigned in
2306	order giving 62 possible formats.
2307	octets 2-40: A 39-character, US-ASCII trailing SPACE filled
2308	field specified by the format letter, if the data is less
2309	than 39 ASCII characters.
2310	octets 41-48: A sequential or random US-ASCII number to make
2311	the ID quasi-unique.
2312	

2313 If the client does not supply a job submission ID in the job submission protocol, then the agent SHALL assign a job submission ID using any of 2314 2315 the standard formats that are reserved for the agent. Clients SHALL 2316 not use formats that are reserved for agents and agents SHALL NOT use formats that are reserved for clients, in order to reduce conflicts in 2317 ID generation. See the description for which formats are reserved for 2318 2319 clients or for agents.

2320	Registration of additional formats may be done following the procedures
2321	described in Section 3.7.3.
2322	The format values defined at the time of completion of this
2323	specification are:
2324	
2325	Format
2326	Letter Description
2327	
2328	'0' Job Owner generated by the server/device
2329	octets 2-40: The last 39 bytes of the jmJobOwner object. octets 41-48: The US-ASCII 8-decimal-digit sequential number
2330 2331	assigned by the agent.
2331 2332	
2332 2333	This format is reserved for agents.
2333 2334	NOTE - Clients wishing to use a job submission ID that
2334	incorporates the job owner, SHALL use format '8', not
2336	format '0'.
2330	
2338	'1' Job Name
2339	octets 2-40: The last 39 bytes of the jobName attribute.
2340	octets 41-48: The US-ASCII 8-decimal-digit random number
2341	assigned by the client.
2342	This format is reserved for clients.
2343	THIS TOTMAL IS TESETVED TOT CITETIES.
2344	'2' Client MAC address
2345	octets 2-40: The client MAC address: in hexadecimal with each
2346	nibble of the 6 octet address being '0'-'9' or 'A' - 'F'
2347	(uppercase only). Most significant octet first.
2348	octets 41-48: The US-ASCII 8-decimal-digit sequential number
2349	assigned by the client.
2350	This format is reserved for clients.
2351	
2352	'3' Client URL
2353	octets 2-40: The last 39 bytes of the client URL [URI-spec].
2354	octets 41-48: The US-ASCII 8-decimal-digit sequential number
2355	assigned by the client.
2356	This format is reserved for clients.
2357	
2358	'4' Job URI
2359	octets 2-40: The last 39 bytes of the URI [URI-spec] assigned
2360	by the server or device to the job when the job was
2361	submitted for processing.
2362	octets 41-48: The US-ASCII 8-decimal-digit sequential number
2363	assigned by the agent.
2364	This format is reserved for agents.
2365	
2366	<u>'5' POSIX User Number</u>
2367	octets 2-40: The last 39 bytes of a user number, such as POSIX
2368	user number.
2369	octets 41-48: The US-ASCII 8-decimal-digit sequential number
2370	assigned by the client.

Bergman, Hastings, Isaacson, Lewis Informational [Page 60]

2371 <u>This format is reserved for clients.</u> 2372	
2373 '6' User Account Number	
2374 Octets 2-40: The last 39 bytes of the user account number	er.
2375 octets 41-48: The US-ASCII 8-decimal-digit sequential nu	
2376 assigned by the client.	
2377 This format is reserved for clients.	
2378	
2379 '7' DTMF Incoming FAX routing number	
2380 Octets 2-40: The last 39 bytes of the DTMF incoming FAX	
2381 routing number.	
2382 octets 41-48: The US-ASCII 8-decimal-digit sequential nu	mber
2383 assigned by the client.	
2384 This format is reserved for clients.	
2385	
2386 '8' Job Owner supplied by the client	
2387 <u>octets 2-40</u> : The last 39 bytes of the job owner name (th	hat the
2388 agent returns in the jmJobOwner object).	
2389 octets 41-48: The US-ASCII 8-decimal-digit sequential nu	mber
2390 assigned by the client.	
2391 This format is reserved for clients. See format '0' which	-hie
2392 reserved for agents.	
2393	
2394 '9' Host Name	
2395 octets 2-40: The last 39 bytes of the host name with tra	ailing
2396 SPACES that submitted the job to this server/device u	
2397 protocol, such as LPD [RFC-1179] which includes the h	
2398 name in the job submission protocol.	
2399 octets 41-48: The US-ASCII 8-decimal-digit leading zero	
2400 representation of the job id generated by the submitt	ting
2401 server (configuration 3) or the client (configuration	
2402 2), such as in the LPD protocol.	<u> </u>
2403 This format is reserved for clients.	
2405 'A' AppleTalk Protocol	
2406 octets 2-40: Contains the AppleTalk printer name, with t	the
2407 first character of the name in octet 2. AppleTalk p	
2408 names are a maximum of 31 characters. Any unused por	
2409 of this field shall be filled with spaces.	
2410 octets 41-48: '00000XXX', where 'XXX' is the 3-digit US-	-ASCII
2411 decimal representation of the Connection Id.	
2412 This format is reserved for agents.	
2413	

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

2414	'B' NetWare PServer
2415	octets 2-40: Contains the Directory Path Name as recorded by
2416	the Novell File Server in the queue directory. If the
2417	string is less than 40 octets, the left-most character in
2418	the string shall appear in octet position 2. Otherwise,
2419	only the last 39 bytes shall be included. Any unused
2420	portion of this field shall be filled with spaces.
2421	octets 41-48: '000XXXXX' The US-ASCII representation of the
2422	Job Number as per the NetWare File Server Queue Management
2423	Services.
2424	This format is reserved for agents.
2425	
2426	'C' Server Message Block protocol (SMB)
2427	octets 2-40: Contains a decimal (US-ASCII coded)
2428	representation of the 16 bit SMB Tree Id field, which
2429	uniquely identifies the connection that submitted the job
2430	to the printer. The most significant digit of the numeric
2430	
	string shall be placed in octet position 2. All unused
2432	portions of this field shall be filled with spaces. The
2433	SMB Tree Id has a maximum value of 65,535.
2434	octets 41-48: The US-ASCII 8-decimal-digit leading zero
2435	representation of the File Handle returned from the device
2436	to the client in response to a Create Print File command.
2437	This format is reserved for agents.
2438	
2439	'D' Transport Independent Printer/System Interface (TIP/SI)
2440	octets 2-40: Contains the Job Name from the Job Control-Start
2441	Job (JC-SJ) command. If the Job Name portion is less than
2442	40 octets, the left-most character in the string shall
2443	appear in octet position 2. Any unused portion of this
2444	field shall be filled with spaces. Otherwise, only the
2445	last 39 bytes shall be included.
2446	octets 41-48: The US-ASCII 8-decimal-digit leading zero
2447	representation of the jmJobIndex assigned by the agent.
2448	This format is reserved for agents, since the agent supplies
2449	octets 41-48, though the client supplies the job name. See
2450	format '1' reserved to clients to submit job name ids in
2451	which they supply octets 41-48.
2452	
2453	'E' IPDS on the MVS or VSE platform
2454	
2455	octets 2-40: Contains bytes 2-27 of the XOH Define Group
2456	Boundary Group ID triplet. Octet position 2 MUST carry the
2457	value x'01'. Bytes 28-40 MUST be filled with spaces.
2458	octets 41-48: The US-ASCII 8-decimal-digit leading zero
2459	representation of the jmJobIndex assigned by the agent.
2460	This format is reserved for agents, since the agent supplies
2461	octets 41-48, though the client supplies the job name.
2462	

Job Monitoring MIB, V1.3 November 8, 1998

2463	'F' IPDS on the VM platform
2464	octets 2-40: Contains bytes 2-31 of the XOH Define Group
2465	Boundary Group ID triplet. Octet position 2 MUST carry the
2466	value x'02'. Bytes 32-40 MUST be filled with spaces.
2467	octets 41-48: The US-ASCII 8-decimal-digit leading zero
2468	representation of the jmJobIndex assigned by the agent.
2469	This format is reserved for agents, since the agent supplies
2470	octets 41-48, though the client supplies the file name.
2471	
2472	'G' IPDS on the OS/400 platform
2473	octets 2-40: Contains bytes 2-36 of the XOH Define Group
2474	Boundary Group ID triplet. Octet position 2 MUST carry the
2475	value x'03'. Bytes 37-40 MUST be filled with spaces.
2476	octets 41-48: The US-ASCII 8-decimal-digit leading zero
2477	representation of the jmJobIndex assigned by the agent.
2478	This format is reserved for agents, since the agent supplies
2479	octets 41-48, though the client supplies the job name.
2480	
2481	NOTE - the job submission id is only intended to be unique between a
2482	limited set of clients for a limited duration of time, namely, for the
2483	life time of the job in the context of the server or device that is
2484	processing the job. Some of the formats include something that is
2485	unique per client and a random number so that the same job submitted by
2486	the same client will have a different job submission id. For other
2487	formats, where part of the id is guaranteed to be unique for each
2488	client, such as the MAC address or URL, a sequential number SHOULD
2489	suffice for each client (and may be easier for each client to manage).
2490	Therefore, the length of the job submission id has been selected to
2491	reduce the probability of collision to an extremely low number, but is
2492	not intended to be an absolute guarantee of uniqueness. None-the-less,
2493	collisions are remotely possible, but without bad consequences, since
2494	this MIB is intended to be used only for monitoring jobs, not for

- 2495 controlling and managing them.
- 2496

2497

### 3.6 Internationalization Considerations 2498

- 2499 This section describes the internationalization considerations included in this MIB. 2500
- 2501 3.6.1 Text generated by the server or device

2502 There are a few objects and attributes generated by the server or device that SHALL be represented using the Universal Multiple-Octet 2503 2504 Coded Character Set (UCS) [ISO-10646]. These objects and attributes 2505 are always supplied (if implemented) by the agent, not by the job 2506 submitting client:

Bergman, Hastings, Isaacson, Lewis Informational [Page 63]

- 2507 1. jmGeneralJobSetName object
- 2508 2. processingMessage(6) attribute
- 2509 3. physicalDevice(32) (name value) attribute

The character encoding scheme for representing these objects and 2510 2511 attributes SHALL be UTF-8 as recommended by RFC 2130 [RFC 2130] and the "IETF Policy on Character Sets and Language" [char-set policy]. The 2512 2513 'JmUTF8StringTC' textual convention is used to indicate UTF-8 text 2514 strings.

2515 NOTE - For strings in 7-bit US-ASCII, there is no impact since the UTF-2516 8 representation of 7-bit ASCII is identical to the US-ASCII [US-ASCII] 2517 encoding.

2518 The text contained in the processingMessage(6) attribute is generated by the server/device. The natural language for the 2519 2520 processingMessage(6) attribute is identified by the 2521 processingMessageNaturalLangTag(7) attribute. The processingMessageNaturalLangTag(7) attribute uses the 2522 2523 JmNaturalLanguageTagTC textual convention which SHALL conform to the 2524 language tag mechanism specified in RFC 1766 [RFC-1766]. The 2525 JmNaturalLanguageTagTC value is the same as the IPP [IPP-model] 'naturalLanguage' attribute syntax. RFC 1766 specifies that a US-ASCII 2526 string consisting of the natural language followed by an optional 2527 2528 country field. Both fields use the same two-character codes from ISO 2529 639 [ISO-639] and ISO 3166 [ISO-3166], respectively, that are used in 2530 the Printer MIB for identifying language and country.

2531 Examples of the values of the processingMessageNaturalLangTag(7) 2532 attribute include:

- 2533 1. 'en' for English
- 2534 2. 'en-us' for US English
- 3. 'fr' 2535 for French
- 4. 'de' for German 2536
- 2537 3.6.2 Text supplied by the job submitter

2538 All of the objects and attributes represented by the 'JmJobStringTC' textual-convention are either (1) supplied in the job submission 2539 2540 protocol by the client that submits the job to the server or device or 2541 (2) are defaulted by the server or device if the job submitting client 2542 does not supply values. The agent SHALL represent these objects and 2543 attributes in the MIB either (1) in the coded character set as they were submitted or (2) MAY convert the coded character set to another 2544 2545 coded character set or encoding scheme. In any case, the resulting 2546 coded character set representation SHOULD be UTF-8 [UTF-8], but SHALL 2547 be one in which the code positions from 0 to 31 is not used, 32 to 127 2548 is US-ASCII [US-ASCII], 127 is not unused, and the remaining code 2549 positions 128 to 255 represent single-byte or multi-byte graphic characters structured according to ISO 2022 [ISO 2022] or are unused. 2550

Bergman, Hastings, Isaacson, Lewis Informational [Page 64]

Job Monitoring MIB, V1.3 November 8, 1998

2551 The coded character set SHALL be one of the ones registered with IANA 2552 [IANA] and SHALL be identified by the jobCodedCharSet attribute in the 2553 jmJobAttributeTable for the job. If the agent does not know what coded character set was used by the job submitting client, the agent SHALL 2554 2555 either (1) return the 'unknown(2)' value for the jobCodedCharSet 2556 attribute or (2) not return the jobCodedCharSet attribute for the job.

2557 Examples of coded character sets which meet this criteria for use as 2558 the value of the jobCodedCharSet job attribute are: US-ASCII [US-ASCII], ISO 8859-1 (Latin-1) [ISO 8859-1], any ISO 8859-n, HP Roman8, 2559 2560 IBM Code Page 850, Windows Default 8-bit set, UTF-8 [UTF-8], US-ASCII plus JIS X0208-1990 Japanese [JIS X0208], US-ASCII plus GB2312-1980 PRC 2561 2562 Chinese [GB2312]. See the IANA registry of coded character sets [IANA 2563 charsets].

2564 Examples of coded character sets which do not meet this criteria are: 2565 national 7-bit sets conforming to ISO 646 (except US-ASCII), EBCDIC, and ISO 10646 (Unicode) [ISO-10646]. In order to represent Unicode 2566 2567 characters, the UTF-8 [UTF-8] encoding scheme SHALL be used which has 2568 been assigned the MIBenum value of '106' by IANA.

2569 The jobCodedCharSet attribute uses the imported 'CodedCharSet' textual-2570 convention from the Printer MIB [printmib].

2571 The natural language for attributes represented by the textualconvention JmJobStringTC is identified either (1) by the 2572 2573 jobNaturalLanguageTag(9) attribute or is keywords in US-English (as in 2574 IPP). A monitoring application SHOULD attempt to localize keywords into the language of the user by means of some lookup mechanism. 2575 Ιf the keyword value is not known to the monitoring application, the 2576 2577 monitoring application SHOULD assume that the value is in the natural 2578 language specified by the job's jobNaturalLanguageTag(9) attribute and 2579 SHOULD present the value to its user as is. The jobNaturalLanguageTag(9) attribute value SHALL have the same syntax and 2580 2581 semantics as the processingMessageNaturalLangTag(7) attribute, except 2582 that the jobNaturalLanguageTag(9) attribute identifies the natural language of attributes supplied by the job submitter instead of the 2583 2584 natural language of the processingMessage(6) attribute. See Section 2585 3.6.1.

3.6.3 'DateAndTime' for representing the date and time 2586

2587 This MIB also contains objects that are represented using the DateAndTime textual convention from SMIv2 [SMIv2-TC]. 2588 The iob 2589 management application SHALL display such objects in the locale of the 2590 user running the monitoring application.

#### 2591 3.7 IANA and PWG Registration Considerations

2592 This MIB does not require any additional registration schemes for IANA, 2593 but does depend on registration schemes that other Internet standards

Bergman, Hastings, Isaacson, Lewis Informational [Page 65]

- 2594 track specifications have set up. The names of these IANA registration 2595 assignments under the /in-notes/iana/assignments/ path:
- 2596 1. printer-language-numbers - used as enums in the documentFormat(38) 2597 attribute
- 2598 2. media-types - uses as keywords in the documentFormat(38) attribute
- 3. character-sets used as enums in the jobCodedCharSet(8) attribute 2599

2600 The Printer Working Group (PWG) will handle registration of additional enums after approving this standard, according to the procedures 2601 described in this section: 2602

2603

2604 3.7.1 PWG Registration of enums

2605 This specification uses textual conventions to define enumerated values 2606 (enums) and bit values. Enumerations (enums) and bit values are sets 2607 of symbolic values defined for use with one or more objects or 2608 attributes. All enumeration sets and bit value sets are assigned a 2609 symbolic data type name (textual convention). As a convention the 2610 symbolic name ends in "TC" for textual convention. These enumerations 2611 are defined at the beginning of the MIB module specification.

The PWG has defined several type of enumerations for use in the Job 2612 Monitoring MIB and the Printer MIB[print-mib]. These types differ in 2613 the method employed to control the addition of new enumerations. 2614 2615 Throughout this document, references to "type n enum", where n can be 2616 1, 2 or 3 can be found in the various tables. The definitions of these 2617 types of enumerations are:

2618 3.7.1.1 Type 1 enumerations

Type 1 enumeration: All the values are defined in the Job Monitoring 2619 2620 MIB specification (RFC for the Job Monitoring MIB). Additional 2621 enumerated values require a new RFC.

2622 There are no type 1 enums in the current draft.

2623 3.7.1.2 Type 2 enumerations

2624 Type 2 enumeration: An initial set of values are defined in the Job 2625 Monitoring MIB specification. Additional enumerated values are 2626 registered with the PWG.

2627 The following type 2 enums are contained in the current draft : 2628 1. JmUTF8StringTC

Bergman, Hastings, Isaacson, Lewis Informational [Page 66]

- 2629 2. JmJobStringTC
- 2630 3. JmNaturalLanguageTagTC
- 2631 4. JmTimeStampTC

2632

- 5. JmFinishingTC [same enum values as IPP "finishing" attribute]
- 2633 6. JmPrintQualityTC [same enum values as IPP "print-quality" 2634 attributel
- 2635 7. JmTonerEconomyTC
- 8. JmMediumTypeTC 2636
- 2637 9. JmJobSubmissionIDTypeTC
- 2638 10.JmJobCollationTypeTC
- 2639 11.JmJobStateTC [same enum values as IPP "job-state" attribute]
- 2640 12.JmAttributeTypeTC

2641 For those textual conventions that have the same enum values as the 2642 indicated IPP Job attribute are simultaneously registered by the PWG 2643 for use with IPP [ipp-model] and the Job Monitoring MIB.

2644 3.7.1.3 Type 3 enumeration

2645 Type 3 enumeration: An initial set of values are defined in the Job 2646 Monitoring MIB specification. Additional enumerated values are 2647 registered through the PWG without PWG review.

- 2648 There are no type 3 enums in the current draft.
- 2649 3.7.2 PWG Registration of type 2 bit values

#### This draft contains the following type 2 bit value textual-conventions: 2650

- 2651 1. JmJobServiceTypesTC
- 2652 2. JmJobStateReasons1TC
- 2653 3. JmJobStateReasons2TC
- 2654 4. JmJobStateReasons3TC
- 2655 5. JmJobStateReasons4TC

These textual-conventions are defined as bits in an Integer so that 2656 they can be used with SNMPv1 SMI. The jobStateReasonsN (N=1..4) 2657 attributes are defined as bit values using the corresponding 2658 2659 JmJobStateReasonsNTC textual-conventions.

2660 The registration of JmJobServiceTypesTC and JmJobStateReasonsNTC bit 2661 values follow the procedures for a type 2 enum as specified in Section 2662 3.7.1.2.

3.7.3 PWG Registration of Job Submission Id Formats 2663

2664 In addition to enums and bit values, this specification assigns a 2665 single ASCII digit or letter to various job submission ID formats. See 2666 the JmJobSubmissionIDTypeTC textual-convention and the object. The

Bergman, Hastings, Isaacson, Lewis Informational [Page 67]

2667 registration of JobSubmissionID format numbers follows the procedures for a type 2 enum as specified in Section 3.7.1.2. 2668

2669 3.7.4 PWG Registration of MIME types/sub-types for document-formats

2670 The documentFormat(38) attribute has MIME type/sub-type values for indicating document formats which IANA registers as "media type" names. 2671 The values of the documentFormat(38) attribute are the same as the 2672 2673 corresponding Internet Printing Protocol (IPP) "document-format" Job 2674 attribute values [ipp-model].

2675 3.8 Security Considerations

2676 3.8.1 Read-Write objects

2677 All objects are read-only, greatly simplifying the security 2678 considerations. If another MIB augments this MIB, that MIB might 2679 accept SNMP Write operations to objects in that MIB whose effect is to 2680 modify the values of read-only objects in this MIB. However, that MIB 2681 SHALL have to support the required access control in order to achieve 2682 security, not this MIB.

2683 3.8.2 Read-Only Objects In Other User's Jobs

2684 The security policy of some sites MAY be that unprivileged users can 2685 only get the objects from jobs that they submitted, plus a few minimal objects from other jobs, such as the jmJobKOctetsPerCopyRequested and 2686 2687 jmJobKOctetsProcessed objects, so that a user can tell how busy a printer is. Other sites MAY allow all unprivileged users to see all 2688 2689 objects of all jobs. This MIB does not require, nor does it specify 2690 how, such restrictions would be implemented. A monitoring application SHOULD enforce the site security policy with respect to returning 2691 2692 information to an unprivileged end user that is using the monitoring application to monitor jobs that do not belong to that user, i.e., the 2693 2694 jmJobOwner object in the jmJobTable does not match the user's user 2695 name.

2696 An operator is a privileged user that would be able to see all objects 2697 of all jobs, independent of the policy for unprivileged users.

#### 3.9 Notifications 2698

2699 This MIB does not specify any notifications. For simplicity, 2700 management applications are expected to poll for status. The jmGeneralJobPersistence and jmGeneralAttributePersistence objects 2701 assist an application to determine the polling rate. The resulting 2702 2703 network traffic is not expected to be significant.

- 2704 4 MIB specification
- 2705 The following pages constitute the actual Job Monitoring MIB.

Bergman, Hastings, Isaacson, Lewis Informational [Page 69]

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 2706 Job-Monitoring-MIB DEFINITIONS ::= BEGIN 2707 2708 TMPORTS MODULE-IDENTITY, OBJECT-TYPE, enterprises, Integer32 FROM SNMPv2-SMI TEXTUAL-CONVENTION FROM SNMPv2-TC MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF; -- The following textual-conventions are needed to implement -- certain attributes, but are not needed to compile this MIB. -- They are provided here for convenience: -- hrDeviceIndex FROM HOST-RESOURCES-MIB -- DateAndTime FROM SNMPv2-TC -- PrtInterpreterLangFamilyTC, -- CodedCharSet FROM Printer-MIB 2709 2710 -- Use the enterprises arc assigned to the PWG which is pwg(2699). 2711 -- Group all PWG mibs under mibs(1). 2712 2713 jobmonMIB MODULE-IDENTITY 2714 LAST-UPDATED "981108<del>1002</del>0000Z" 2715 ORGANIZATION "Printer Working Group (PWG)" 2716 CONTACT-INFO 2717 "Tom Hastings 2718 Postal: Xerox Corp. 2719 Mail stop ESAE-231 2720 701 S. Aviation Blvd. 2721 El Segundo, CA 90245 2722 2723 Tel: (301)333-6413 2724 Fax: (301)333 - 55142725 E-mail: hastings@cp10.es.xerox.com 2726 2727 Send questions and comments to the Printer Working Group (PWG) 2728 using the Job Monitoring Project (JMP) Mailing List: 2729 jmp@pwq.org 2730 2731 For further information, including how to subscribe to the jmp mailing list, access the PWG web page under 'JMP': 2732 2733 2734 http://www.pwg.org/ 2735 2736 Implementers of this specification are encouraged to join the 2737 jmp mailing list in order to participate in discussions on any clarifications needed and registration proposals being reviewed 2738 2739 in order to achieve consensus." 2740 DESCRIPTION 2741 "The MIB module for monitoring job in servers, printers, and other devices. 2742 2743 2744 Version: 1.32" 2745 ::= { enterprises pwg(2699) mibs(1) jobmonMIB(1) }

Bergman, Hastings, Isaacson, Lewis Informational [Page 70]

Job Monitoring MIB, V1.3 November 8, 1998

2746 2747 -- Textual conventions for this MIB module 2748 2749 JmUTF8StringTC ::= TEXTUAL-CONVENTION 2750 DISPLAY-HINT "255a" 2751 current STATUS 2752 DESCRIPTION "To facilitate internationalization, this TC represents 2753 2754 information taken from the ISO/IEC IS 10646-1 character set, 2755 encoded as an octet string using the UTF-8 character encoding 2756 scheme. 2757 2758 See section 3.6.1, entitled: 'Text generated by the server or 2759 device'." 2760 SYNTAX OCTET STRING (SIZE (0..63)) 2761 2762 2763 2764 2765 JmJobStringTC ::= TEXTUAL-CONVENTION 2766 STATUS current 2767 DESCRIPTION "To facilitate internationalization, this TC represents 2768 2769 information using any coded character set registered by IANA as 2770 specified in section 3.7. While it is recommended that the coded character set be UTF-8 [UTF-8], the actual coded 2771 character set SHALL be indicated by the value of the 2772 2773 jobCodedCharSet(8) attribute for the job. 2774 2775 See section 3.6.2, entitled: 'Text supplied by the job 2776 submitter'." 2777 SYNTAX OCTET STRING (SIZE (0..63)) 2778 2779 2780 2781 2782 JmNaturalLanguageTagTC ::= TEXTUAL-CONVENTION 2783 STATUS current 2784 DESCRIPTION 2785 "An IETF RFC 1766-compliant 'language tag', with zero or more sub-tags that identify a natural language. While RFC 1766 2786 specifies that the US-ASCII values are case-insensitive, this 2787 MIB specification requires that all characters SHALL be lower 2788 case in order to simplify comparing by management applications. 2789 2790 2791 See section 3.6.1, entitled: 'Text generated by the server or 2792 device' and section 3.6.2, entitled: 'Text supplied by the job 2793 submitter'." 2794 SYNTAX OCTET STRING (SIZE (0..63)) 2795 2796 2797 JmTimeStampTC ::= TEXTUAL-CONVENTION Bergman, Hastings, Isaacson, Lewis Informational [Page 71]

2798	STATUS current
2799	DESCRIPTION
2800	"The simple time at which an event took place. The units are
2801	in seconds since the system was booted.
2802	2
2803	NOTE - JmTimeStampTC is defined in units of seconds, rather
2804	than 100ths of seconds, so as to be simpler for agents to
2805	implement (even if they have to implement the 100ths of a
2806	second to comply with implementing sysUpTime in MIB-II[mib-
2807	II].)
2808	±±]·)
2809	NOTE - JmTimeStampTC is defined as an Integer32 so that it can
2810	be used as a value of an attribute, i.e., as a value of the
2811	jmAttributeValueAsInteger object. The TimeStamp textual-
2812	convention defined in SNMPv2-TC [SMIv2-TC] is defined as an
2812	APPLICATION 3 IMPLICIT INTEGER tag, not an Integer32 which is
2813	defined in SNMPv2-SMI [SMIv2-TC] as UNIVERSAL 2 IMPLICIT
2814	
	INTEGER, so cannot be used in this MIB as one of the values of
2816	jmAttributeValueAsInteger."
2817	SYNTAX INTEGER (02147483647)
2818	
2819	
2820	
2821	
2822	JmJobSourcePlatformTypeTC ::= TEXTUAL-CONVENTION
2823	STATUS current
2824	DESCRIPTION
2825	"The source platform type that can submit jobs to servers or
2826	devices in any of the 3 configurations.
2827	
2828	This is a type 2 enumeration. See Section 3.7.1.2. See also
2829	IANA operating-system-names registry."
2830	SYNTAX INTEGER {
	other(1),
	unknown(2),
	sptUNIX(3), UNIX
	sptOS2(4), $$ OS/2
	sptPCDOS(5), DOS
	sptNT(6), NT
	sptMVS(7), MVS
	sptVM(8), VM
	sptOS400(9), OS/400
	sptVMS(10), VMS
	sptWindows(11), Windows
	sptNetWare(12) NetWare
2831	}
2832	

```
2833
2834
      JmFinishingTC ::= TEXTUAL-CONVENTION
          STATUS current
2835
2836
          DESCRIPTION
              "The type of finishing operation.
2837
2838
2839
              These values are the same as the enum values of the IPP
              'finishings' attribute. See Section 3.7.1.2.
2840
2841
2842
              other(1),
2843
                  Some other finishing operation besides one of the specified
2844
                  or registered values.
2845
2846
              unknown(2),
2847
                  The finishing is unknown.
2848
2849
              none(3),
2850
                  Perform no finishing.
2851
2852
              staple(4),
2853
                  Bind the document(s) with one or more staples. The exact
2854
                  number and placement of the staples is site-defined.
2855
2856
              punch(5),
2857
                  This value indicates that holes are required in the
2858
                  finished document. The exact number and placement of the
2859
                  holes is site-defined The punch specification MAY be
                  satisfied (in a site- and implementation-specific manner)
2860
                  either by drilling/punching, or by substituting pre-drilled
2861
2862
                  media.
2863
2864
              cover(6),
                  This value is specified when it is desired to select a non-
2865
2866
                  printed (or pre-printed) cover for the document. This does
2867
                  not supplant the specification of a printed cover (on cover
2868
                  stock medium) by the document itself.
2869
             bind(7)
2870
2871
                  This value indicates that a binding is to be applied to the
                  document; the type and placement of the binding is product-
2872
2873
                  specific.
2874
2875
              This is a type 2 enumeration. See Section 3.7.1.2."
2876
          SYNTAX
                      INTEGER {
2877
              other(1),
2878
              unknown(2),
              none(3),
2879
2880
              staple(4),
2881
              punch(5),
2882
              cover(6),
2883
              bind(7)
          }
2884
```

2885 2886 JmPrintQualityTC ::= TEXTUAL-CONVENTION 2887 2888 STATUS current 2889 DESCRIPTION 2890 "Print quality settings. 2891 2892 These values are the same as the enum values of the IPP 'print-2893 quality' attribute. See Section 3.7.1.2. 2894 2895 This is a type 2 enumeration. See Section 3.7.1.2." 2896 INTEGER { SYNTAX -- Not one of the specified or registered other(1), -- values. -- The actual value is unknown. unknown(2), draft(3), -- Lowest quality available on the printer. normal(4), -- Normal or intermediate quality on the -- printer. -- Highest quality available on the printer. hiqh(5) 2897 } 2898 2899 2900 2901 JmPrinterResolutionTC ::= TEXTUAL-CONVENTION 2902 2903 STATUS current 2904 DESCRIPTION 2905 "Printer resolutions. 2906 2907 Nine octets consisting of two 4-octet SIGNED-INTEGERs followed 2908 by a SIGNED-BYTE. The values are the same as those specified 2909 in the Printer MIB [printmib]. The first SIGNED-INTEGER 2910 contains the value of prtMarkerAddressabilityXFeedDir. The 2911 second SIGNED-INTEGER contains the value of 2912 prtMarkerAddressabilityFeedDir. The SIGNED-BYTE contains the 2913 value of prtMarkerAddressabilityUnit. 2914 2915 Note: the latter value is either 3 (tenThousandsOfInches) or 4 (micrometers) and the addressability is in 10,000 units of 2916 2917 measure. Thus the SIGNED-INTEGERs represent integral values in either dots-per-inch or dots-per-centimeter. 2918 2919 2920 The syntax is the same as the IPP 'printer-resolution' 2921 attribute. See Section 3.7.1.2." 2922 SYNTAX OCTET STRING (SIZE(9)) 2923

```
INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998
2924
2925 JmTonerEconomyTC ::= TEXTUAL-CONVENTION
2926
         STATUS current
2927
         DESCRIPTION
2928
              "Toner economy settings.
2929
2930
             This is a type 2 enumeration. See Section 3.7.1.2."
         SYNTAX INTEGER {
2931
             unknown(2), -- unknown.
              off(3),
                            -- Off. Normal. Use full toner.
              on(4)
                           -- On. Use less toner than normal.
          }
2932
2933
2934
2935
2936
    JmBooleanTC ::= TEXTUAL-CONVENTION
2937
         STATUS current
2938
         DESCRIPTION
             "Boolean true or false value.
2939
2940
2941
             This is a type 2 enumeration. See Section 3.7.1.2."
         SYNTAX INTEGER {
2942
              unknown(2), -- unknown.
              false(3),
                            -- FALSE.
              true(4)
                           -- TRUE.
2943
          }
2944
2945
2946
2947
      JmMediumTypeTC ::= TEXTUAL-CONVENTION
2948
         STATUS current
2949
         DESCRIPTION
2950
              "Identifies the type of medium.
2951
2952
             other(1),
2953
                 The type is neither one of the values listed in this
2954
                 specification nor a registered value.
2955
2956
             unknown(2),
2957
                 The type is not known.
2958
2959
             stationery(3),
2960
                 Separately cut sheets of an opaque material.
2961
2962
             transparency(4),
2963
                 Separately cut sheets of a transparent material.
2964
2965
             envelope(5),
2966
                 Envelopes that can be used for conventional mailing
2967
                 purposes.
```

2968 2969	envelopePlain(6),
2970 2971	Envelopes that are not preprinted and have no windows.
2972 2973 2974	envelopeWindow(7), Envelopes that have windows for addressing purposes.
2975 2976 2977	continuousLong(8), Continuously connected sheets of an opaque material
2978	connected along the long edge.
2979 2980 2981 2982	continuousShort(9), Continuously connected sheets of an opaque material connected along the short edge.
2982 2983 2984 2985	tabStock(10), Media with tabs.
2985 2986 2987 2988	multiPartForm(11), Form medium composed of multiple layers not pre-attached to one another; each sheet MAY be drawn separately from an
2989 2990 2991	<pre>input source. labels(12),</pre>
2992 2993	Label-stock.
2994 2995 2996 2997	multiLayer(13) Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers.
2998 2999 3000	This is a type 2 enumeration. See Section 3.7.1.2. These enum values correspond to the keyword name strings of the prtInputMediaType object in the Printer MIB [print-mib]. There
3001 3002 3003	is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER {
3004 3005 3006	other(1), unknown(2), stationery(3),
3007 3008 3009	<pre>transparency(4), envelope(5), envelopePlain(6),</pre>
3010 3011 3012	envelopeWindow(7), continuousLong(8), continuousShort(9),
3013 3014 3015 3016	<pre>tabStock(10), multiPartForm(11), labels(12), multiLayer(13)</pre>
3017 3018 3019	<pre> } imatclbayer(15) }</pre>

```
INTERNET-DRAFT
```

```
3020
      JmJobCollationTypeTC ::= TEXTUAL-CONVENTION
3021
          STATUS
                      current
3022
          DESCRIPTION
              "This value is the type of job collation. Implementations that
3023
              don't support multiple documents or don't support multiple
3024
3025
              copies SHALL NOT support the uncollatedDocuments(5) value.
3026
              This is a type 2 enumeration. See Section 3.7.1.2. See also
3027
3028
              Section 3.4, entitled 'Monitoring Job Progress'."
3029
          SYNTAX
                      INTEGER {
3030
              other(1),
3031
              unknown(2),
3032
              uncollatedSheets(3),
                                       -- sheets within each document copy
3033
                                       -- are not collated: 1 1 ..., 2 2 ...,
                                       -- No corresponding value of IPP
3034
3035
                                       -- "multiple-document-handling"
                                       -- internal collated sheets,
3036
              collatedDocuments(4),
3037
                                       -- documents: A, B, A, B, ...
                                       -- Corresponds to IPP "multiple-
3038
3039
                                       -- document-handling"='separate-
3040
                                       -- documents-collated-copies'
                                       -- internal collated sheets,
3041
              uncollatedDocuments(5)
3042
                                       -- documents: A, A, ..., B, B, ...
                                       -- Corresponds to IPP "multiple-
3043
3044
                                       -- document-handling"='separate-
3045
                                       -- documents-uncollated-copies'
          }
3046
3047
3048
3049
      JmJobSubmissionIDTypeTC ::= TEXTUAL-CONVENTION
3050
          STATUS
                      current
3051
          DESCRIPTION
3052
               "Identifies the format type of a job submission ID.
3053
3054
              Each job submission ID is a fixed-length, 48-octet printable
3055
              US-ASCII [US-ASCII] coded character string containing no
              control characters, consisting of the fields defined in section
3056
              3.5.1. following fields:
3057
3058
3059
              - octet 1: The format letter identifying the format. The US-
                  ASCII characters '0 9', 'A Z', and 'a z' are assigned in
3060
3061
                  order giving 62 possible formats.
3062
              - octets 2 40: A 39 character, US ASCII trailing SPACE filled
                  field specified by the format letter, if the data is less
3063
3064
                  than 39 ASCII characters.
3065
              - octets 41 48: A sequential or random US ASCII number to make
3066
                  the ID quasi unique.
3067
3068
              If the client does not supply a job submission ID in the job
3069
              submission protocol, then the agent SHALL assign a job
3070
              submission ID using any of the standard formats that are
3071
              reserved for the agent. Clients SHALL not use formats that are
```

Bergman, Hastings, Isaacson, Lewis Informational [Page 77]

	INTERNET-DRAFT Job Monitoring MIB, V1. <u>3</u> <u>November 8</u> , 1998	8
3072 3073 3074 3075	reserved for agents and agents SHALL NOT use formats that are reserved for clients, in order to reduce conflicts in ID generation. See the description for which formats are reserved for clients or for agents.	<del>d</del>
3076 3077 3078 3079	Registration of additional formats may be done following the procedures described in Section 3.7.3.	
3080 3081 3082	The format values defined at the time of completion of this specification are:	
3083 3084 3085 3086	Format Letter Description  '0' Job Owner generated by the server/device	
3087 3088 3089	octets 2 40: The last 39 bytes of the jmJobOwner object. octets 41 48: The US ASCII 8 decimal digit sequential number assigned by the agent.	
3090 3091 3092 3093	This format is reserved for agents. NOTE Clients wishing to use a job submission ID that incorporates the job owner, SHALL use format '8', not	
3094 3095 3096 3097	format '0'. <del>'1' Job Name</del> <del>octets 2 40: The last 39 bytes of the jobName attribute.</del>	
3098 3099 3100	octets 41 48: The US ASCII 8 decimal digit random number assigned by the client. This format is reserved for clients.	
3101 3102 3103 3104	<del>'2' Client MAC address</del> octets 2 40: The client MAC address: in hexadecimal with each nibble of the 6 octet address being '0' '9' or 'A' 'F'	
3105 3106 3107	(uppercase only). Most significant octet first. octets 41 48: The US ASCII 8 decimal digit sequential number assigned by the client.	
3108 3109 3110 3111	This format is reserved for clients. '3' Client URL octets 2 40: The last 39 bytes of the client URL [URI spec].	
3112 3113 3114	octets 41 48: The US ASCII 8 decimal digit sequential number assigned by the client. This format is reserved for clients.	
3115 3116 3117 3118	<del>'4' Job URI</del> octets 2 40: The last 39 bytes of the URI [URI spec] assigned by the server or device to the job when the job was	
3119 3120 3121	submitted for processing. octets 41 48: The US ASCII 8 decimal digit sequential number assigned by the agent.	
3122 3123	This format is reserved for agents.	

3124	<u>'5' POSIX User Number</u>
3125	
	octets 2 40: The last 39 bytes of a user number, such as POSIX
3126	user number.
3127	octets 41 48: The US ASCII 8 decimal digit sequential number
3128	assigned by the client.
3129	This format is reserved for clients.
3130	
3131	<del>'6' User Account Number</del>
3132	octets 2 40: The last 39 bytes of the user account number.
3133	octets 41 48: The US ASCII 8 decimal digit sequential number
3134	assigned by the client.
3135	This format is reserved for clients.
3136	
3137	<u>'7' DTMF Incoming FAX routing number</u>
3138	octets 2 40: The last 39 bytes of the DTMF incoming FAX
3139	-
3140	routing number. octets 41 48: The US ASCII 8 decimal digit sequential number
3141	assigned by the client.
3142	This format is reserved for clients.
3143	THIS TORMAC IS RESERVED FOR CITCHES.
3144	'8' Job Owner supplied by the client
3145	octets 2 40: The last 39 bytes of the job owner name (that the
3146	
	agent returns in the jmJobOwner object).
3147	octets 41 48: The US ASCII 8 decimal digit sequential number
3148	assigned by the client.
3149	This format is reserved for clients. See format '0' which is
3150	<del>reserved for agents.</del>
3151	
3152	<del>'9' Host Name</del>
3153	octets 2-40: The last 39 bytes of the host name with trailing
3154	SPACES that submitted the job to this server/device using a
3155	protocol, such as LPD [RFC 1179] which includes the host
3156	name in the job submission protocol.
3157	octets 41 48: The US ASCII 8 decimal digit leading zero
3158	representation of the job id generated by the submitting
3159	server (configuration 3) or the client (configuration 1 and
3160	2), such as in the LPD protocol.
3161	This format is reserved for clients.
3162	THE FORMAGE IN REDCEVED FOR CITCHEN.
3163	(A) AppleTalk Drotogal
	'A' AppleTalk Protocol
3164	octets 2 40: Contains the AppleTalk printer name, with the
3165	first character of the name in octet 2. AppleTalk printer
3166	names are a maximum of 31 characters. Any unused portion
3167	of this field shall be filled with spaces.
3168	octets 41-48: '00000XXX', where 'XXX' is the 3 digit US-ASCII
3169	decimal representation of the Connection Id.
3170	This format is reserved for agents.
3171	

3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182	<pre>'B' NetWare PServer octets 2 40: Contains the Directory Path Name as recorded by the Novell File Server in the queue directory. If the string is less than 40 octets, the left most character in the string shall appear in octet position 2. Otherwise, only the last 39 bytes shall be included. Any unused portion of this field shall be filled with spaces. octets 41 48: '000XXXXY' The US ASCII representation of the Job Number as per the NetWare File Server Queue Management Services.</pre>
3183 3184 3185 3186	<del>'C' Server Message Block protocol (SMB)</del> <del>octets 2 40: Contains a decimal (US ASCII coded)</del> <del>representation of the 16 bit SMB Tree Id field, which</del>
3187 3188 3189 3190	uniquely identifies the connection that submitted the job to the printer. The most significant digit of the numeric string shall be placed in octet position 2. All unused portions of this field shall be filled with spaces. The
3191 3192 3193	SMB Tree Id has a maximum value of 65,535. octets 41 48: The US ASCII 8 decimal digit leading zero representation of the File Handle returned from the device
3194 3195 3196	to the client in response to a Create Print File command. This format is reserved for agents.
3197 3198 3199 3200 3201 3202 3203	'D' Transport Independent Printer/System Interface (TIP/SI) octets 2 40: Contains the Job Name from the Job Control Start Job (JC SJ) command. If the Job Name portion is less than 40 octets, the left most character in the string shall appear in octet position 2. Any unused portion of this field shall be filled with spaces. Otherwise, only the last 39 bytes shall be included.
3204 3205 3206 3207	octets 41 48: The US ASCII 8 decimal digit leading zero representation of the jmJobIndex assigned by the agent. This format is reserved for agents, since the agent supplies octets 41 48, though the client supplies the job name. See
3208 3209 3210 3211	format '1' reserved to clients to submit job name ids in which they supply octets 41 48. <u>'E' IPDS on the MVS or VSE platform</u>
3212 3213 3214	octets 2 40: Contains bytes 2 27 of the XOH Define Group Boundary Group ID triplet. Octet position 2 MUST carry the
3215 3216 3217 3218 3219 3220	value x'01'. Bytes 28 40 MUST be filled with spaces. octets 41 48: The US ASCII 8 decimal digit leading zero representation of the jmJobIndex assigned by the agent. This format is reserved for agents, since the agent supplies octets 41 48, though the client supplies the job name.

3221	<del>'F' IPDS on the VM platform</del>
3222	octets 2 40: Contains bytes 2 31 of the XOH Define Group
3223	Boundary Group ID triplet. Octet position 2 MUST carry the
3224	value x'02'. Bytes 32-40 MUST be filled with spaces.
3225	octets 41 48: The US ASCII 8 decimal digit leading zero
3226	representation of the jmJobIndex assigned by the agent.
3227	This format is reserved for agents, since the agent supplies
3228	octets 41 48, though the client supplies the file name.
3229	
3230	<del>'G' IPDS on the OS/400 platform</del>
3231	octets 2 40: Contains bytes 2 36 of the XOH Define Group
3232	Boundary Group ID triplet. Octet position 2 MUST carry the
3233	value x'03'. Bytes 37 40 MUST be filled with spaces.
3234	octets 41 48: The US ASCII 8 decimal digit leading zero
3235	representation of the jmJobIndex assigned by the agent.
3236	This format is reserved for agents, since the agent supplies
3237	octets 41-48, though the client supplies the job name.
3238	
3239	NOTE the job submission id is only intended to be unique
3240	between a limited set of clients for a limited duration of
3241	time, namely, for the life time of the job in the context of
3242	the server or device that is processing the job. Some of the
3243	formats include something that is unique per client and a
3244	random number so that the same job submitted by the same client
3245	will have a different job submission id. For other formats,
3246	where part of the id is guaranteed to be unique for each
3247	<del>client, such as the MAC address or URL, a sequential number</del>
3248	SHOULD suffice for each client (and may be easier for each
3249	client to manage). Therefore, the length of the job submission
3250	id has been selected to reduce the probability of collision to
3251	an extremely low number, but is not intended to be an absolute
3252	guarantee of uniqueness. None the less, collisions are
3253	remotely possible, but without bad consequences, since this MIB
3254	is intended to be used only for monitoring jobs, not for
3255	controlling and managing them.
3256	
3257	This is like a type 2 enumeration. See section 3.7.3."
3258	SYNTAX OCTET STRING(SIZE(1)) ASCII '0'-'9', 'A'-'Z', 'a'-'z'

Bergman, Hastings, Isaacson, Lewis Informational [Page 81]

3259 3260 JmJobStateTC ::= TEXTUAL-CONVENTION 3261 STATUS current 3262 DESCRIPTION "The current state of the job (pending, processing, completed, 3263 3264 etc.). 3265 The following figure shows the normal job state transitions: 3266 3267 3268 +---> canceled(7)
/
+---> completed(9)
/
--->+ | | | +----> aborted(8)
/
+---> aborted(8) +---> canceled(7) 3269 3270 3271 3272 3273 3274 +---> pendingHeld(4) processingStopped(6) ---+ 3275 3276 Figure 4 - Normal Job State Transitions 3277 3278 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden. Not shown are 3279 the transitions to the canceled state from the pending, 3280 pendingHeld, and processingStopped states. 3281 3282 Jobs in the pending, processing, and processingStopped states 3283 are called 'active', while jobs in the pendingHeld, canceled, 3284 aborted, and completed states are called 'inactive'. Jobs 3285 reach one of the three terminal states: completed, canceled, or 3286 aborted, after the jobs have completed all activity, and all 3287 3288 MIB objects and attributes have reached their final values for 3289 the job. 3290 3291 These values are the same as the enum values of the IPP 'job-3292 state' job attribute. See Section 3.7.1.2. 3293 3294 unknown(2), The job state is not known, or its state is indeterminate. 3295 3296 3297 pending(3), The job is a candidate to start processing, but is not yet 3298 3299 processing. 3300 3301 pendingHeld(4), 3302 The job is not a candidate for processing for any number of 3303 reasons but will return to the pending state as soon as the 3304 reasons are no longer present. The job's jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4) 3305 attributes SHALL indicate why the job is no longer a 3306 3307 candidate for processing. The reasons are represented as 3308 bits in the jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4) attributes. See the 3309

Bergman, Hastings, Isaacson, Lewis Informational [Page 82]

3310

3311

3312 3313

3314

3315 3316

3317 3318

3319 3320

3321 3322

3323

3324 3325

3326

3327

3328 3329

3330

3331

3332 3333

3334

3335

3336

3337

3338 3339

3340

3341 3342

3343

3344

3345

3346 3347

3348

3349

3350

3351

3352 3353

3354 3355

3356

3357

JmJobStateReasonsNTC (N=1..4) textual convention for the specification of each reason.

processing(5),

One or more of:

1. the job is using, or is attempting to use, one or more purely software processes that are analyzing, creating, or interpreting a PDL, etc.,

the job is using, or is attempting to use, one or more 2. hardware devices that are interpreting a PDL, making marks on a medium, and/or performing finishing, such as stapling, etc.,

OR

3. (configuration 2) the server 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.

When the job is in the processing state, the entire job state includes the detailed status represented in the device MIB indicated by the hrDeviceIndex value of the job's physicalDevice attribute, if the agent implements such a device MIB.

Implementations MAY, though they NEED NOT, include additional values in the job's jmJobStateReasons1 object to indicate the progress of the job, such as adding the jobPrinting value to indicate when the device is actually making marks on a medium and/or the processingToStopPoint value to indicate that the server or device is in the process of canceling or aborting the job.

processingStopped(6),

The job has stopped while processing for any number of reasons and will return to the processing state as soon as the reasons are no longer present.

The job's jmJobStateReasons1 object and/or the job's jobStateReasonsN (N=2..4) attributes MAY indicate why the job has stopped processing. For example, if the output device is stopped, the deviceStopped value MAY be included in the job's jmJobStateReasons1 object.

3358 NOTE - When an output device is stopped, the device usually indicates its condition in human readable form at the 3359 3360 device. The management application can obtain more complete device status remotely by querying the appropriate 3361

Bergman, Hastings, Isaacson, Lewis Informational [Page 83]

3362 3363 3364	device MIB using the job's deviceIndex attribute(s), if the agent implements such a device MIB
3365	canceled(7),
3366	A client has canceled the job and the server or device has
3367	
	completed canceling the job AND all MIB objects and
3368	attributes have reached their final values for the job.
3369	While the server or device is canceling the job, the job's
3370	jmJobStateReasons1 object SHOULD contain the
3371	processingToStopPoint value and one of the canceledByUser,
3372	canceledByOperator, or canceledAtDevice values. The
3373	canceledByUser, canceledByOperator, or canceledAtDevice
3374	values remain while the job is in the canceled state.
3375	
3376	aborted(8),
3377	The job has been aborted by the system, usually while the
3378	job was in the processing or processingStopped state and
3379	the server or device has completed aborting the job AND all
3380	MIB objects and attributes have reached their final values
3381	for the job. While the server or device is aborting the
3382	job, the job's jmJobStateReasons1 object MAY contain the
3383	processingToStopPoint and abortedBySystem values. If
3384	implemented, the abortedBySystem value SHALL remain while
3385	the job is in the aborted state.
3386	
3387	completed(9)
3388	The job has completed successfully or with warnings or
3389	errors after processing and all of the media have been
3390	successfully stacked in the appropriate output bin(s) AND
3391	all MIB objects and attributes have reached their final
3392	values for the job. The job's jmJobStateReasons1 object
3393	SHOULD contain one of: completedSuccessfully,
3394	completedWithWarnings, or completedWithErrors values.
3395	
3396	This is a type 2 enumeration. See Section 3.7.1.2."
3397	SYNTAX INTEGER {
3398	unknown(2),
3399	pending(3),
3400	pendingHeld(4),
3401	processing(5),
3402	processingStopped(6),
3403	canceled(7),
3404	aborted(8),
3405	completed(9)
3406	}

3407					
3408	JmAttributeTypeTC ::= TEXTUAL-CONVENTION				
3409	STATUS current				
3410	DESCRIPTION				
3411	"The type of the attribute wh	ich identifies the attribute.			
3412					
3413		re grouped logically with values			
3414	assigned in groups of 20, so t				
3415 3416	their logical grouping.	assigned a value that is part of			
3410 3417	their logical grouping.				
3417	Values in the range 2**20 to (	2**31-1 are reserved for private			
3410	or experimental usage. This is				
3420		nenters are warned that use of			
3421	such values may conflict with				
3422		o request registration of enum			
3423	values following the procedure				
3424	varues for owing the procedure				
3425	See Section 3.2 entitled 'The	Attribute Mechanism' for a			
3426	description of this textual-co				
3427		n 3.3.8 for the specification of			
3428	each attribute. The comment(s	s) after each enum assignment			
3429	specifies the data type(s) of				
3430					
3431	This is a type 2 enumeration.	See Section 3.7.1.2."			
3432					
3433	SYNTAX INTEGER {				
3434	other(1),	Integer32 (-22147483647)			
3435		AND/OR			
3436		OCTET STRING(SIZE(063))			
3437					
3438	Job State attributes:				
3439	jobStateReasons2(3),	JmJobStateReasons2TC			
3440	jobStateReasons3(4),	JmJobStateReasons3TC			
3441	jobStateReasons4(5),	JmJobStateReasons4TC			
3442	<pre>processingMessage(6),</pre>	JmUTF8StringTC (SIZE(063))			
3443	processingMessageNaturalLangTa				
3444		OCTET STRING(SIZE(063))			
3445	jobCodedCharSet(8),	CodedCharSet			
3446	jobNaturalLanguageTag(9),	OCTET STRING(SIZE(063))			
3447					

3448 3449 3450 3451 3452 3453 3454 3455 3456 3457 3458 3459 3460 3461 3462 3463 3464 3465 3464 3465 3466 3467 3468 3469 3470 3471	<pre> Job Identification attribute jobURI(20), jobAccountName(21), serverAssignedJobName(22), jobName(23), jobServiceTypes(24), jobSourceChannelIndex(25), jobSourcePlatformType(26), submittingServerName(27), submittingApplicationName(28), jobOriginatingHost(29), deviceNameRequested(30), queueNameRequested(30), queueNameRequested(31), physicalDevice(32),</pre> numberOfDocuments(33), fileName(34), documentName(35), jobComment(36), documentFormatIndex(37), documentFormat(38),	<pre>s:  OCTET STRING(SIZE(063))  JmJobStringTC (SIZE(063))  JmJobStringTC (SIZE(063))  JmJobServiceTypesTC  Integer32 (02147483647)  JmJobSourcePlatformTypeTC  JmJobStringTC (SIZE(063))  JmJobStringTC (SIZE(063))  JmJobStringTC (SIZE(063))  JmJobStringTC (SIZE(063))  JmJobStringTC (SIZE(063))  JmJobStringTC (SIZE(063))  Integer32 (-22147483647)  JmJobStringTC (SIZE(063))  Integer32 (02147483647)  PrtInterpreterLangFamilyTC  AND/OR  OCTET STRING(SIZE(063))</pre>
3472 3473 3474 3475 3476 3477 3478 3479 3480 3481 3482	<pre> Job Parameter attributes: jobPriority(50), jobProcessAfterDateAndTime(51), jobHold(52), jobHoldUntil(53), outputBin(54), sides(55), finishing(56),</pre>	<pre> Integer32 (-2100)  DateAndTime (SNMPv2-TC)  JmBooleanTC  JmJobStringTC (SIZE(063))  Integer32 (02147483647)  AND/OR  JmJobStringTC (SIZE(063))  Integer32 (-22)  JmFinishingTC</pre>
3483 3484 3485 3486 3487 3488 3489 3490 3491 3492 3493	Image Quality attributes: printQualityRequested(70), printQualityUsed(71), printerResolutionRequested(72), printerResolutionUsed(73), tonerEcomonyRequested(74), tonerEcomonyUsed(75), tonerDensityRequested(76), tonerDensityUsed(77),	<ul> <li>JmPrintQualityTC</li> <li>JmPrintQualityTC</li> <li>JmPrinterResolutionTC</li> <li>JmPrinterResolutionTC</li> <li>JmTonerEconomyTC</li> <li>JmTonerEconomyTC</li> <li>Integer32 (-2100)</li> <li>Integer32 (-2100)</li> </ul>

3494 3495 3496 3497 3498 3499 3500 3501 3502 3503 3504	Job Progress attributes: jobCopiesRequested(90), jobCopiesCompleted(91), documentCopiesRequested(92), documentCopiesCompleted(93), jobKOctetsTransferred(94), sheetCompletedCopyNumber(95), sheetCompletedDocumentNumber(96 jobCollationType(97),	Integer32 (-22147483647) Integer32 (-22147483647) Integer32 (-22147483647) Integer32 (-22147483647) Integer32 (-22147483647) Integer32 (-22147483647) ), Integer32 (-22147483647) JmJobCollationTypeTC
3505 3506 3507 3508 3509 3510 3511	Impression attributes: impressionsSpooled(110), impressionsSentToDevice(111), impressionsInterpreted(112), impressionsCompletedCurrentCopy fullColorImpressionsCompleted(1)	Integer32 (-22147483647) 14),
3512 3513 3514 3515 3516 3517	<pre>highlightColorImpressionsComple  Page attributes: pagesRequested(130), pagesCompleted(131),</pre>	Integer32 (-22147483647)
3518 3519 3520 3521 3522	pagesCompletedCurrentCopy(132),	
3523 3524 3525 3526 3527 3528	<pre>sheetsCompleted(151), sheetsCompletedCurrentCopy(152) Resource attributes: mediumRequested(170),</pre>	
3529 3530 3531 3532 3533	<pre>mediumConsumed(171), colorantReguested(172),</pre>	JmJobStringTC (SIZE(063)) Integer32 (-22147483647) AND JmJobStringTC (SIZE(063)) Integer32 (-22147483647)
3534 3535 3536 3537 3538	colorantConsumed(173),	AND/OR JmJobStringTC (SIZE(063)) Integer32 (-22147483647) AND/OR JmJobStringTC (SIZE(063))
3539 3540 3541 3542 3543	<pre>mediumTypeConsumed(174), mediumSizeConsumed(175),</pre>	Integer32 (-22147483647) AND JmJobStringTC (SIZE(063)) Integer32 (-22147483647) AND
3544 3545		JmJobStringTC (SIZE(063))

3546		Time attributes:	
3547		jobSubmissionToServerTime(190),	JmTimeStampTC
3548			AND/OR
3549			DateAndTime
3550		jobSubmissionTime(191),	 JmTimeStampTC
3551			 AND/OR
3552			 DateAndTime
3553		jobStartedBeingHeldTime(192),	 JmTimeStampTC
3554			 AND/OR
3555			 DateAndTime
3556		<pre>jobStartedProcessingTime(193),</pre>	 JmTimeStampTC
3557			 AND/OR
3558			 DateAndTime
3559		jobCompletionTime(194),	 JmTimeStampTC
3560			 AND/OR
3561			 DateAndTime
3562		jobProcessingCPUTime(195)	 Integer32 (-22147483647)
3563	}		
3564	-		

3565 3566 3567	JmJobServiceTypesTC ::= TEXTUAL-CONVENTION STATUS current DESCRIPTION
3567 3568 3569 3570 3571 3572 3573 3574 3575 3576 3577 3578	"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. For example, a job service might scan, faxOut, and print a single job. In this case, three bits would be set in the jobServiceTypes attribute, corresponding to the hexadecimal values: 0x8 + 0x20 + 0x4, respectively, yielding: 0x2C.
3579 3580 3581 3582 3583 3583	Whether this attribute is set from a job attribute supplied by the job submission client or is set by the recipient job submission server or device depends on the job submission protocol. With either implementation, the agent SHALL return a non-zero value for this attribute indicating the type of the job.
3585 3586 3587 3588 3589 3590	One of the purposes of this 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 attribute is in the job identification category.
3591 3592 3593 3594 3595	The following service component types are defined (in hexadecimal) and are assigned a separate bit value for use with the jobServiceTypes attribute:
3596 3597 3598 3599	0x1 The job contains some instructions that are not one of the identified types.
3600 3601 3602 3603	unknown 0x2 The job contains some instructions whose type is unknown to the agent.
3604 3605 3606 3607	0x4 The job contains some instructions that specify printing scan 0x8
3608 3609 3610	The job contains some instructions that specify scanning faxIn 0x10
3611 3612 3613 3614 3615	The job contains some instructions that specify receive fax faxOut 0x20 The job contains some instructions that specify sending fax

	INTERNET-DRAFT	Job Monitoring MIB, V1. <mark>3</mark>	November 8, 1998
3616 3617 3618 3619		0x40 job contains some instructions that s or documents	specify accessing
3620 3621 3622 3623		0x80 job contains some instructions that s or documents	specify storing
3624 3625 3626 3627		0x100 job contains some instructions that ribution of documents using an elect	
3628 3629 3630 3631	except th section 3	definitions are the equivalent of nat combinations of them MAY be used 3.7.1.2." INTEGER (02147483647) 31 bits	l together. See
3632 3633 3634 3635	JmJobStateReasons	1TC ::= TEXTUAL-CONVENTION	
3636 3637		current	
3638 3639 3640 3641 3642 3643 3643 3644 3645	"The JmJo with the ( <i>N</i> =24), regarding MAY be us makes ser	bStateReasonsNTC (N=14) textual-c jmJobStateReasons1 object and jobSt respectively, to provide additiona the current jmJobState object valu sed with any job state or states for use. <u>See section</u> 3.3.9.1 for the sp defined for use with the JmJobStat	ateReasonsN l information le. These values which the reason pecification of each
3646 3647 3648 3649 3650 3651 3652	without i receiving JmJobStat without i jmJobStat	tile values cannot be added to the j mpacting deployed clients that take j jmJobState values, it is the inten eReasonsATC enums can be defined an mpacting such deployed clients. In eReasons1 object and jobStateReason to be extensible.	e actions upon t that additional d registered t other words, the
3653 3654 3655 3656 3657 3658 3659 3660	<del>values[ig since the submissic reasons h Job Monit</del>	ne Job Monitoring MIB contains a sup op model] for the IPP 'job state rea by Monitoring MIB is intended to on protocols as well. Also some of have been changed from 'printer' to coring MIB is intended to cover addi including input devices, such as sc	sons' attribute, cover other job the names of the 'device', since the tional types of
3661 3662 3663 3664 3665 3666	<del>powers of</del> time. Fo reasons a	wing standard values are defined (i <del>two, since multiple values MAY be</del> be ease of understanding, the JmJobS are presented in the order in which booccur (if implemented), starting w	used at the same StateReasons1TC the reasons are

3667	'jobIncoming' value and ending with the
3668 3669	<u>'jobCompletedWithErrors' value.</u>
3670	01
3670	<del>other 0x1</del> The job state reason is not one of the standardized or
3672	registered reasons.
3673	
3674	unknown 0x2
3675	The job state reason is not known to the agent or is
3676	indeterminent.
3677	
3678	jobIncoming 0x4
3679	The job has been accepted by the server or device, but the
3680	server or device is expecting (1) additional operations
3681	from the client to finish creating the job and/or (2) is
3682	accessing/accepting document data.
3683	
3684	submissionInterrupted 0x8
3685	The job was not completely submitted for some unforeseen
3686	reason, such as: (1) the server has crashed before the job
3687	was closed by the client, (2) the server or the document
3688	transfer method has crashed in some non recoverable way
3689	before the document data was entirely transferred to the
3690	<del>server, (3) the client crashed or failed to close the job</del>
3691	before the time out period.
3692	
3693	-jobOutgoing 0x10
3694	Configuration 2 only: The server is transmitting the job
3695	to the device.
3696	
3697	<del>jobHoldSpecified 0x20</del>
3698	The value of the job's jobHold(52) attribute is TRUE. The
3699	job SHALL NOT be a candidate for processing until this
3700	reason is removed and there are no other reasons to hold
3701	the job.
3702	
3703	jobHoldUntilSpecified 0x40
3704	The value of the job's jobHoldUntil(53) attribute specifies
3705	a time period that is still in the future. The job SHALL
3706	NOT be a candidate for processing until this reason is
3707	<del>removed and there are no other reasons to hold the job.</del>
3708	
3709	<del>jobProcessAfterSpecified 0x80</del>
3710	The value of the job's jobProcessAfterDateAndTime(51)
3711	attribute specifies a time that is still in the future.
3712	The job SHALL NOT be a candidate for processing until this
3713	reason is removed and there are no other reasons to hold
3714	<del>the job.</del>
3715	-

3716	resourcesAreNotReady 0x100
3717	At least one of the resources needed by the job, such as
3718	media, fonts, resource objects, etc., is not ready on any
3719	of the physical devices for which the job is a candidate.
3720	This condition MAY be detected when the job is accepted, or
3721	subsequently while the job is pending or processing,
3722	depending on implementation.
3723	
3723	
-	deviceStoppedPartly 0x200
3725	One or more, but not all, of the devices to which the job
3726	is assigned are stopped. If all of the devices are stopped
3727	(or the only device is stopped), the deviceStopped reason
3728	SHALL be used.
3729	
3730	deviceStopped 0x400
3731	The device(s) to which the job is assigned is (are all)
3732	<del>stopped.</del>
3733	
3734	jobInterpreting 0x800
3735	The device to which the job is assigned is interpreting the
3736	document data.
3737	
3738	jobPrinting 0x1000
3739	The output device to which the job is assigned is marking
3740	media. This value is useful for servers and output devices
3741	which spend a great deal of time processing (1) when no
3742	marking is happening and then want to show that marking is
3743	now happening or (2) when the job is in the process of
3744	being canceled or aborted while the job remains in the
3745	processing state, but the marking has not yet stopped so
3745	that impression or sheet counts are still increasing for
3747	the job.
3748	
3749	jobCanceledByUser 0x2000
3750	The job was canceled by the owner of the job, i.e., by a
3751	user whose name is the same as the value of the job's
3752	jmJobOwner object, or by some other authorized end user,
3753	<del>such as a member of the job owner's security group.</del>
3754	
3755	<del>jobCanceledByOperator 0x4000</del>
3756	<del>The job was canceled by the operator, i.e., by a user who</del>
3757	has been authenticated as having operator privileges
3758	<del>(whether local or remote).</del>
3759	
3760	<del>jobCanceledAtDevice 0x8000</del>
3761	The job was canceled by an unidentified local user, i.e., a
3762	user at a console at the device.
3763	

3764	abortedBySystem 0x10000
3765	The job (1) is in the process of being aborted, (2) has
3766	been aborted by the system and placed in the 'aborted'
3767	state, or (3) has been aborted by the system and placed in
3768	the 'pendingHeld' state, so that a user or operator can
3769	manually try the job again.
3770	
3771	processingToStopPoint 0x20000
3772	The requester has issued an operation to cancel or
3773	interrupt the job or the server/device has aborted the job,
3774	but the server/device is still performing some actions on
-	
3775	the job until a specified stop point occurs or job
3776	termination/cleanup is completed.
3777	
3778	This reason is recommended to be used in conjunction with
3779	the processing job state to indicate that the server/device
3780	is still performing some actions on the job while the job
3781	remains in the processing state. After all the job's
3782	resources consumed counters have stopped incrementing, the
3783	server/device moves the job from the processing state to
3784	the canceled or aborted job states.
3785	
3786	serviceOffLine 0x40000
3787	The service or document transform is off line and accepting
3788	no jobs. All pending jobs are put into the pendingHeld
3789	state. This situation could be true if the service's or
3790	document transform's input is impaired or broken.
3791	
3792	jobCompletedSuccessfully 0x80000
3793	The job completed successfully.
3794	
3795	<del>jobCompletedWithWarnings 0x100000</del>
3796	The job completed with warnings.
3797	
3798	jobCompletedWithErrors 0x200000
3799	The job completed with errors (and possibly warnings too).
3800	
3801	
3802	The following additional job state reasons have been added to
3803	represent job states that are in ISO DPA[iso dpa] and other job
3804	submission protocols:
3805	
3806	-iobPaused 0x400000
3807	
3808	an operation to suspend the job so that other jobs may
3809	proceed using the same devices. The client MAY issue an
3810	operation to resume the paused job at any time, in which
3811	case the agent SHALL remove the jobPaused values from the
3812	job's jmJobStateReasons1 object and the job is eventually
3813	resumed at or near the point where the job was paused.
3814	

3815	jobInterrupted 0x800000
3816	The job has been interrupted while processing by a client
3817	issuing an operation that specifies another job to be run
3818	instead of the current job. The server or device will
3819	automatically resume the interrupted job when the
3820	interrupting job completes.
3821	
3822	jobRetained 0x1000000
3823	The job is being retained by the server or device with all
3824	of the job's document data (and submitted resources, such
3825	
	as fonts, logos, and forms, if any). Thus a client could
3826	issue an operation to the server or device to either (1)
3827	re do the job (or a copy of the job) on the same server or
3828	device or (2) resubmit the job to another server or device.
3829	When a client could no longer re do/resubmit the job, such
3830	<del>as after the document data has been discarded, the agent</del>
3831	SHALL remove the jobRetained value from the
3832	<del>jmJobStateReasons1 object.</del>
3833	
3834	These bit definitions are the equivalent of a type 2 enum
3835	except that combinations of bits may be used together. See
3836	section 3.7.1.2. The remaining bits are reserved for future
3837	standardization and/or registration."
3838	SYNTAX INTEGER (02147483647) 31 bits, all but sign bit
3839	
3840	
3841	
3842	JmJobStateReasons2TC ::= TEXTUAL-CONVENTION
3843	
	STATUS current
3844	DESCRIPTION
3845	"This textual-convention is used with the jobStateReasons2
3846	attribute to provides additional information regarding the
3847	jmJobState object. See <u>section</u> 3.3.9.2 for the specification
3848	of JmJobStateReasons2TC. See section 3.3.9.1 for the
3849	description under JmJobStateReasons1TC for additional
3850	information that applies to all reasons.
3851	
3852	The following standard values are defined (in hexadecimal) as
3853	powers of two, since multiple values may be used at the same
3854	time:
3855	
3856	<del>cascaded 0x1</del>
3857	An outbound gateway has transmitted all of the job's job
3858	and document attributes and data to another spooling
3859	system.
3860	
3861	deletedByAdministrator 0x2
3862	The administrator has deleted the job.
3863	THE administrator has dereted the job.
3864	discardTimeArrived 0x4
3864 3865	discardTimeArrived 0x4 The job has been deleted due to the fact that the time
2002	The job has been dereced due to the ract that the tille

3866	specified by the job's job discard time attribute has
3867	arrived.
3868	
3869	postProcessingFailed 0x8
3870	The post processing agent failed while trying to log
3871	accounting attributes for the job; therefore the job has
3872	been placed into the completed state with the jobRetained
3873	jmJobStateReasons1 object value for a system defined period
3874	of time, so the administrator can examine it, resubmit it,
3875	etc.
3876	
3877	jobTransforming 0x10
3878	The server/device is interpreting document data and
3879	producing another electronic representation.
3880	
3881	maxJobFaultCountExceeded 0x20
3882	The job has faulted several times and has exceeded the
3883	administratively defined fault count limit.
3884	administratively actined fault count fimit.
3885	devicesNeedAttentionTimeOut 0x40
3886	One or more document transforms that the job is using needs
3887	human intervention in order for the job to make progress,
3888	but the human intervention did not occur within the site-
3889	<del>settable time out value.</del>
3890	
3891	needsKeyOperatorTimeOut 0x80
3892	One or more devices or document transforms that the job is
3893	
	using need a specially trained operator (who may need a key
3894	to unlock the device and gain access) in order for the job
3895	to make progress, but the key operator intervention did not
3896	occur within the site settable time out value.
3897	
3898	<del>jobStartWaitTimeOut 0x100</del>
3899	The server/device has stopped the job at the beginning of
3900	processing to await human action, such as installing a
3901	special cartridge or special non standard media, but the
3902	iob was not resumed within the site settable time out value
	and the server/device has transitioned the job to the
3903	
3904	<del>pendingHeld state.</del>
3905	
3906	<del>jobEndWaitTimeOut 0x200</del>
3907	The server/device has stopped the job at the end of
3908	processing to await human action, such as removing a
3909	special cartridge or restoring standard media, but the job
3910	was not resumed within the site settable time out value and
3911	the server/device has transitioned the job to the completed
3912	
	<del>state.</del>
3913	
3914	jobPasswordWaitTimeOut 0x400
3915	The server/device has stopped the job at the beginning of
3916	processing to await input of the job's password, but the

3917	password was not received within the site settable time out
3918	
3919	
3920	deviceTimedOut 0x800
3921	A device that the job was using has not responded in a
3922	period specified by the device's site settable attribute.
	period specified by the device's site settable attribute.
3923	
3924	connectingToDeviceTimeOut 0x1000
3925	The server is attempting to connect to one or more devices
3926	which may be dial up, polled, or queued, and so may be busy
3927	with traffic from other systems, but server was unable to
3928	<del>connect to the device within the site settable time out</del>
3929	<del>value.</del>
3930	
3931	transferring 0x2000
3932	The job is being transferred to a down stream server or
3933	downstream device.
3934	
3935	queuedInDevice 0x4000
3936	The server/device has queued the job in a down stream
3937	<del>server or downstream device.</del>
3938	
3939	iobQueued 0x8000
3940	The server/device has queued the document data.
3941	
3942	<del>jobCleanup 0x10000</del>
3943	The server/device is performing cleanup activity as part of
3944	ending normal processing.
3945	ending normal processing.
	jobPasswordWait 0x20000
3946	
3947	The server/device has selected the job to be next to
3948	<del>process, but instead of assigning resources and starting</del>
3949	the job processing, the server/device has transitioned the
3950	job to the pendingHeld state to await entry of a password
3951	(and dispatched another job, if there is one).
3952	
3953	validating 0x40000
3954	The server/device is validating the job after accepting the
3955	job.
3956	J0D.
3957	queueHeld 0x80000
3958	The operator has held the entire job set or queue.
3959	
3960	<del>jobProofWait 0x100000</del>
3961	The job has produced a single proof copy and is in the
3962	<del>pendingHeld state waiting for the requester to issue an</del>
3963	operation to release the job to print normally, obeying any
3964	job and document copy attributes that were originally
3965	submitted.
3966	

3967	heldForDiagnostics	<del></del>		
3968	The system is running	intrusive diagnostics,	<del>so that al</del>	ŀ

jobs are being held. 3969

Bergman, Hastings, Isaacson, Lewis Informational [Page 97]

3970	noSpaceOnServer 0x800000
3971	There is no room on the server to store all of the job.
3972	
3973	pinRequired 0x1000000
3974	The System Administrator settable device policy is (1) to
3975	require PINs, and (2) to hold jobs that do not have a pin
3976	supplied as an input parameter when the job was created.
	supplied as an input parameter when the job was created.
3977	
3978	exceededAccountLimit 0x2000000
3979	<del>The account for which this job is drawn has exceeded its</del>
3980	limit. This condition SHOULD be detected before the job is
3981	<del>scheduled so that the user does not wait until his/her job</del>
3982	is scheduled only to find that the account is overdrawn.
3983	This condition MAY also occur while the job is processing
3984	either as processing begins or part way through processing.
	ercher as processing begins or part way chrough processing.
3985	
3986	heldForRetry 0x4000000
3987	The job encountered some errors that the server/device
3988	<del>could not recover from with its normal retry procedures,</del>
3989	but the error might not be encountered if the job is
3990	processed again in the future. Example cases are phone
3991	number busy or remote file system in accessible. For such
3992	a situation, the server/device SHALL transition the job
3993	
	from the processing to the pendingHeld, rather than to the
3994	aborted state.
3995	
3996	The following values are from the X/Open PSIS draft standard:
3997	
3998	canceledByShutdown 0x8000000
3999	The job was canceled because the server or device was
4000	shutdown before completing the job.
4001	bildedown before compreering ene job.
4002	deviceUnavailable 0x1000000
4003	This job was aborted by the system because the device is
4004	<del>currently unable to accept jobs.</del>
4005	
4006	wrongDevice 0x2000000
4007	This job was aborted by the system because the device is
4008	unable to handle this particular job; the spooler SHOULD
4009	try another device or the user should submit the job to
4010	another device.
	another device.
4011	
4012	badJob 0x4000000
4013	<del>This job was aborted by the system because this job has a</del>
4014	major problem, such as an ill formed PDL; the spooler
4015	<del>SHOULD not even try another device.</del>
4016	
4017	These bit definitions are the equivalent of a type 2 enum
4018	except that combinations of them may be used together. See
4019	section 3.7.1.2. See the description under
4020	JmJobStateReasons1TC and the jobStateReasons2 attribute."
4021	SYNTAX INTEGER (02147483647) 31 bits, all but sign bit

Bergman, Hastings, Isaacson, Lewis Informational [Page 98]

<pre>JmJobStateReasons3TC ::= TEXTUAL-CONVENTION STATUS current DESCRIPTION "This textual-convention is used with the jobStateReasons3 attribute to provides additional information regarding the jmJobState object. See section 3.3.9.3 for the specification of JmJobStateReasons3TC. See section 3.3.9.1 for the description under JmJobStateReasons1TC for additional information that applies to all reasons.</pre>
The following standard values are defined (in hexadecimal) as powers of two, since multiple values may be used at the same time:
jobInterruptedByDeviceFailure 0x1 A device or the print system software that the job was using has failed while the job was processing. The server or device is keeping the job in the pendingHeld state until an operator can determine what to do with the job.
These bit definitions are the equivalent of a type 2 enum except that combinations of them may be used together. See section 3.7.1.2. The remaining bits are reserved for future standardization and/or registration. See the description under JmJobStateReasons1TC and the jobStateReasons3 attribute." SYNTAX INTEGER (02147483647) 31 bits, all but sign bit
SINIAX INIEGER (U214/40304/) SI DIUS, AII DUU SIGH DIU
JmJobStateReasons4TC ::= TEXTUAL-CONVENTION STATUS current DESCRIPTION
"This textual-convention is used in the jobStateReasons4
attribute to provides additional information regarding the
jmJobState object. See section 3.3.9.4 for the specification
of JmJobStateReasons4TC. See section 3.3.9.1 for the
description under JmJobStateReasons1TC for additional
information that applies to all reasons.
The following standard values are defined (in hexadecimal) as
powers of two, since multiple values may be used at the same
time:
none yet defined. These bits are reserved for future
standardization and/or registration.
Standardrigation and, or registration.
These bit definitions are the equivalent of a type 2 enum except that combinations of them may be used together. See

4073	section	n 3.7.1.2.	See	the des		<del>n ur</del>	<del>ıder</del>				
4074	<del>JmJobS</del>	tateReason	<del>s1TC</del>	and the	jobStat	eRea	asons4	att	<del>ribut</del>	e."	
4075	SYNTAX	INTEGER	(02	14748364		31	bits,	all	but	sign	bit

```
4076
      jobmonMIBObjects OBJECT IDENTIFIER ::= { jobmonMIB 1 }
4077
4078
4079
      -- The General Group (MANDATORY)
4080
4081
      -- The jmGeneralGroup consists entirely of the jmGeneralTable.
4082
      jmGeneral OBJECT IDENTIFIER ::= { jobmonMIBObjects 1 }
4083
4084
4085
      jmGeneralTable OBJECT-TYPE
4086
                      SEQUENCE OF JmGeneralEntry
          SYNTAX
4087
          MAX-ACCESS not-accessible
4088
                     current
          STATUS
4089
          DESCRIPTION
4090
              "The jmGeneralTable consists of information of a general nature
4091
              that are per-job-set, but are not per-job. See Section 2
              entitled 'Terminology and Job Model' for the definition of a
4092
4093
              job set.
4094
4095
              The MANDATORY-GROUP macro specifies that this group is
4096
              MANDATORY."
4097
          ::= { jmGeneral 1 }
4098
4099
4100
      jmGeneralEntry OBJECT-TYPE
4101
          SYNTAX
                      JmGeneralEntry
4102
          MAX-ACCESS not-accessible
4103
          STATUS
                      current
4104
          DESCRIPTION
4105
              "Information about a job set (queue).
4106
4107
              An entry SHALL exist in this table for each job set."
4108
          INDEX { jmGeneralJobSetIndex }
4109
          ::= { jmGeneralTable 1 }
4110
4111
4112
      JmGeneralEntry ::= SEQUENCE {
4113
          jmGeneralJobSetIndex
                                                Integer32 (1...32767),
                                                Integer32 (0..2147483647),
4114
          imGeneralNumberOfActiveJobs
4115
          jmGeneralOldestActiveJobIndex
                                                Integer32 (0..2147483647),
                                                Integer32 (0..2147483647),
4116
          jmGeneralNewestActiveJobIndex
4117
          jmGeneralJobPersistence
                                                Integer32 (15..2147483647),
4118
          jmGeneralAttributePersistence
                                                Integer32 (15..2147483647),
4119
                                                JmUTF8StringTC (SIZE(0..63))
          jmGeneralJobSetName
4120
      }
4121
```

```
INTERNET-DRAFT
                             Job Monitoring MIB, V1.3 November 8, 1998
4122
      jmGeneralJobSetIndex OBJECT-TYPE
4123
          SYNTAX Integer32 (1...32767)
          MAX-ACCESS not-accessible
4124
4125
          STATUS
                      current
4126
          DESCRIPTION
4127
              "A unique value for each job set in this MIB. The jmJobTable
4128
              and jmAttributeTable tables have this same index as their
4129
              primary index.
4130
4131
              The value(s) of the jmGeneralJobSetIndex SHALL be persistent
4132
              across power cycles, so that clients that have retained
4133
              jmGeneralJobSetIndex values will access the same job sets upon
4134
              subsequent power-up.
4135
4136
              An implementation that has only one job set, such as a printer
4137
              with a single queue, SHALL hard code this object with the value
4138
              1.
4139
4140
              See Section 2 entitled 'Terminology and Job Model' for the
4141
              definition of a job set.
4142
              Corresponds to the first index in jmJobTable and
4143
              jmAttributeTable."
4144
          ::= { jmGeneralEntry 1 }
4145
4146
4147
      jmGeneralNumberOfActiveJobs OBJECT-TYPE
4148
                      Integer32 (0..2147483647)
          SYNTAX
4149
          MAX-ACCESS
                     read-only
4150
          STATUS
                      current
4151
          DESCRIPTION
4152
              "The current number of 'active' jobs in the jmJobIDTable,
4153
              jmJobTable, and jmAttributeTable, i.e., the total number of
4154
              jobs that are in the pending, processing, or processingStopped
4155
              states. See the JmJobStateTC textual-convention for the exact
4156
              specification of the semantics of the job states."
4157
          DEFVAL
                      { 0 } -- no jobs
          ::= { jmGeneralEntry 2 }
4158
4159
```

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 4160 jmGeneralOldestActiveJobIndex OBJECT-TYPE 4161 SYNTAX Integer32 (0..2147483647) MAX-ACCESS read-only 4162 4163 STATUS current 4164 DESCRIPTION 4165 "The jmJobIndex of the oldest job that is still in one of the 4166 'active' states (pending, processing, or processingStopped). In other words, the index of the 'active' job that has been in 4167 4168 the job tables the longest. 4169 4170 If there are no active jobs, the agent SHALL set the value of 4171 this object to 0. 4172 4173 See Section 3.2 entitled 'The Job Tables and the Oldest Active 4174 and Newest Active Indexes' for a description of the usage of 4175 this object." 4176 DEFVAL { 0 } -- no active jobs 4177 ::= { jmGeneralEntry 3 } 4178 4179 4180 jmGeneralNewestActiveJobIndex OBJECT-TYPE 4181 4182 SYNTAX Integer32 (0..2147483647) 4183 MAX-ACCESS read-only 4184 STATUS current 4185 DESCRIPTION 4186 "The jmJobIndex of the newest job that is in one of the 'active' states (pending, processing, or processingStopped). 4187 In other words, the index of the 'active' job that has been 4188 4189 most recently added to the job tables. 4190 4191 When all jobs become 'inactive', i.e., enter the pendingHeld, completed, canceled, or aborted states, the agent SHALL set the 4192 4193 value of this object to 0. 4194 4195 See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes' for a description of the usage of 4196 4197 this object." DEFVAL { 0 } -- no active jobs 4198 4199 ::= { jmGeneralEntry 4 } 4200

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 4201 jmGeneralJobPersistence OBJECT-TYPE 4202 Integer32 (15..2147483647) SYNTAX 4203 UNTTS "seconds" 4204 MAX-ACCESS read-only current 4205 STATUS 4206 DESCRIPTION 4207 "The minimum time in seconds for this instance of the Job Set that an entry SHALL remain in the jmJobIDTable and jmJobTable 4208 4209 after processing has completed, i.e., the minimum time in 4210 seconds starting when the job enters the completed, canceled, 4211 or aborted state. 4212 4213 Configuring this object is implementation-dependent. 4214 4215 This value SHALL be equal to or greater than the value of 4216 jmGeneralAttributePersistence. This value SHOULD be at least 4217 60 which gives a monitoring or accounting application one minute in which to poll for job data." 4218 { 60 } 4219 DEFVAL -- one minute 4220 ::= { jmGeneralEntry 5 } 4221 4222 4223 4224 jmGeneralAttributePersistence OBJECT-TYPE Integer32 (15..2147483647) 4225 SYNTAX 4226 UNITS "seconds" 4227 MAX-ACCESS read-only 4228 STATUS current 4229 DESCRIPTION 4230 "The minimum time in seconds for this instance of the Job Set 4231 that an entry SHALL remain in the jmAttributeTable after 4232 processing has *completed*, i.e., the time in seconds starting 4233 when the job enters the completed, canceled, or aborted state. 4234 4235 Configuring this object is implementation-dependent. 4236 4237 This value SHOULD be at least 60 which gives a monitoring or accounting application one minute in which to poll for job 4238 4239 data." DEFVAL 4240 { 60 } -- one minute ::= { jmGeneralEntry 6 } 4241 4242

4243 jmGeneralJobSetName OBJECT-TYPE 4244 SYNTAX JmUTF8StringTC (SIZE(0..63)) 4245 MAX-ACCESS read-only 4246 STATUS current 4247 DESCRIPTION 4248 "The human readable name of this job set assigned by the system administrator (by means outside of this MIB). Typically, this 4249 name SHOULD be the name of the job queue. If a server or 4250 device has only a single job set, this object can be the 4251 administratively assigned name of the server or device itself. 4252 4253 This name does not need to be unique, though each job set in a 4254 single Job Monitoring MIB SHOULD have distinct names. 4255 4256 NOTE - If the job set corresponds to a single printer and the Printer MIB is implemented, this value SHOULD be the same as 4257 4258 the prtGeneralPrinterName object in the draft Printer MIB 4259 [print-mib-draft]. If the job set corresponds to an IPP Printer, this value SHOULD be the same as the IPP 'printer-4260 name' Printer attribute. 4261 4262 4263 NOTE - The purpose of this object is to help the user of the 4264 job monitoring application distinguish between several job sets 4265 in implementations that support more than one job set. 4266 4267 See the OBJECT compliance macro for the minimum maximum length 4268 required for conformance." DEFVAL { ''H } -- empty string 4269 ::= { jmGeneralEntry 7 } 4270 4271 4272 4273 4274 4275

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 4276 -- The Job ID Group (MANDATORY) 4277 4278 -- The jmJobIDGroup consists entirely of the jmJobIDTable. 4279 4280 jmJobID OBJECT IDENTIFIER ::= { jobmonMIBObjects 2 } 4281 4282 jmJobIDTable OBJECT-TYPE 4283 SYNTAX SEQUENCE OF JmJobIDEntry 4284 MAX-ACCESS not-accessible 4285 STATUS current 4286 DESCRIPTION 4287 "The jmJobIDTable provides a correspondence map (1) between the 4288 job submission ID that a client uses to refer to a job and (2) 4289 the jmGeneralJobSetIndex and jmJobIndex that the Job Monitoring MIB agent assigned to the job and that are used to access the 4290 job in all of the other tables in the MIB. If a monitoring 4291 4292 application already knows the jmGeneralJobSetIndex and the 4293 jmJobIndex of the job it is querying, that application NEED NOT 4294 use the jmJobIDTable. 4295 4296 The MANDATORY-GROUP macro specifies that this group is 4297 MANDATORY." 4298 ::= { jmJobID 1 } 4299 4300 4301 4302 jmJobIDEntry OBJECT-TYPE 4303 SYNTAX JmJobIDEntry 4304 MAX-ACCESS not-accessible 4305 STATUS current 4306 DESCRIPTION 4307 "The map from (1) the jmJobSubmissionID to (2) the 4308 jmGeneralJobSetIndex and jmJobIndex. 4309 4310 An entry SHALL exist in this table for each job currently known 4311 to the agent for all job sets and job states. There MAY be 4312 more than one jmJobIDEntry that maps to a single job. This many to one mapping can occur when more than one network entity 4313 along the job submission path supplies a job submission ID. 4314 See Section 3.5. However, each job SHALL appear once and in 4315 one and only one job set." 4316 4317 INDEX { jmJobSubmissionID } 4318 ::= { jmJobIDTable 1 } 4319 4320 JmJobIDEntry ::= SEQUENCE { 4321 jmJobSubmissionID OCTET STRING(SIZE(48)), 4322 jmJobIDJobSetIndex Integer32 (0..32767), 4323 Integer32 (0..2147483647) jmJobIDJobIndex 4324 } 4325

4326	5	omissionID OBJECT-TYPE
4327		TAX OCTET STRING(SIZE(48))
4328		-ACCESS not-accessible
4329	STA	
4330	DES	CRIPTION
4331		"A quasi-unique 48-octet fixed-length string ID which
4332		identifies the job within a particular client-server
4333		environment. There are multiple formats for the
4334		jmJobSubmissionID. Each format SHALL be uniquely identified.
4335		See the JmJobSubmissionIDTypeTC textual convention. Each
4336		format SHALL be registered using the procedures of a type 2
4337		enum. See section 3.7.3 entitled: 'PWG Registration of Job
4338		Submission Id Formats'.
4339		
4340		If the requester (client or server) does not supply a job
4341		submission ID in the job submission protocol, then the
4342		recipient (server or device) SHALL assign a job submission ID
4343		using any of the standard formats that have been reserved for
4344		agents and adding the final 8 octets to distinguish the ID from
4345		others submitted from the same requester.
4346		-
4347		The monitoring application, whether in the client or running
4348		separately, MAY use the job submission ID to help identify
4349		which jmJobIndex was assigned by the agent, i.e., in which row
4350		the job information is in the other tables.
4351		
4352		NOTE - fixed-length is used so that a management application
4353		can use a shortened GetNext varbind (in SNMPv1 and SNMPv2) in
4354		order to get the next submission ID, disregarding the remainder
4355		of the ID in order to access jobs independent of the trailing
4356		identifier part, e.g., to get all jobs submitted by a
4357		particular jmJobOwner or submitted from a particular MAC
4358		address.
4359		
4360		See the JmJobSubmissionIDTypeTC textual convention.
4361		See APPENDIX B - Support of Job Submission Protocols."
4362	::=	{ jmJobIDEntry 1 }
4363		

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 4364 jmJobIDJobSetIndex OBJECT-TYPE 4365 SYNTAX Integer32 (0...32767) 4366 MAX-ACCESS read-only 4367 STATUS current 4368 DESCRIPTION 4369 "This object contains the value of the jmGeneralJobSetIndex for 4370 the job with the jmJobSubmissionID value, i.e., the job set index of the job set in which the job was placed when that 4371 4372 server or device accepted the job. This 16-bit value in 4373 combination with the jmJobIDJobIndex value permits the 4374 management application to access the other tables to obtain the 4375 job-specific objects for this job. 4376 4377 See jmGeneralJobSetIndex in the jmGeneralTable." 4378 DEFVAL { 0 } -- 0 indicates no job set index 4379 ::= { jmJobIDEntry 2 } 4380 4381 4382 4383 jmJobIDJobIndex OBJECT-TYPE 4384 Integer32 (0..2147483647) SYNTAX MAX-ACCESS read-only 4385 4386 STATUS current 4387 DESCRIPTION 4388 "This object contains the value of the jmJobIndex for the job with the jmJobSubmissionID value, i.e., the job index for the 4389 job when the server or device accepted the job. This value, in 4390 combination with the jmJobIDJobSetIndex value, permits the 4391 4392 management application to access the other tables to obtain the 4393 job-specific objects for this job. 4394 4395 See jmJobIndex in the jmJobTable." 4396 DEFVAL { 0 } -- 0 indicates no jmJobIndex value. 4397 ::= { jmJobIDEntry 3 } 4398 4399 4400 4401

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

4402 -- The Job Group (MANDATORY) 4403 4404 -- The jmJobGroup consists entirely of the jmJobTable. 4405 jmJob OBJECT IDENTIFIER ::= { jobmonMIBObjects 3 } 4406 4407 jmJobTable OBJECT-TYPE 4408 4409 SYNTAX SEQUENCE OF JmJobEntry 4410 MAX-ACCESS not-accessible 4411 STATUS current 4412 DESCRIPTION 4413 "The jmJobTable consists of basic job state and status 4414 information for each job in a job set that (1) monitoring 4415 applications need to be able to access in a single SNMP Get 4416 operation, (2) that have a single value per job, and (3) that 4417 SHALL always be implemented. 4418 4419 The MANDATORY-GROUP macro specifies that this group is 4420 MANDATORY." 4421 ::= { jmJob 1 } 4422 4423 4424 4425 jmJobEntry OBJECT-TYPE 4426 SYNTAX JmJobEntry 4427 MAX-ACCESS not-accessible current 4428 STATUS 4429 DESCRIPTION 4430 "Basic per-job state and status information. 4431 4432 An entry SHALL exist in this table for each job, no matter what 4433 the state of the job is. Each job SHALL appear in one and only 4434 one job set. 4435 4436 See Section 3.2 entitled 'The Job Tables'." 4437 INDEX { jmGeneralJobSetIndex, jmJobIndex } 4438 ::= { jmJobTable 1 } 4439 4440 JmJobEntry ::= SEQUENCE { 4441 jmJobIndex Integer32 (1..2147483647), 4442 imJobState JmJobStateTC, 4443 jmJobStateReasons1 JmJobStateReasons1TC, Integer32 (-2..2147483647), 4444 jmNumberOfInterveningJobs jmJobKOctetsPerCopyRequested Integer32 (-2..2147483647), 4445 4446 jmJobKOctetsProcessed Integer32 (-2..2147483647), 4447 jmJobImpressionsPerCopyRequested Integer32 (-2..2147483647), 4448 jmJobImpressionsCompleted Integer32 (-2..2147483647), 4449 JmJobStringTC (SIZE(0..63)) jmJobOwner 4450 } 4451

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

4452	jmJobIndex OBJECT-TYPE
4453	SYNTAX Integer32 (12147483647)
4454	MAX-ACCESS not-accessible
4455	STATUS current
4456	DESCRIPTION
4457	"The sequential, monatonically increasing identifier index for
4458	the job generated by the server or device when that server or
4459	device accepted the job. This index value permits the
4460	management application to access the other tables to obtain the
4461	job-specific row entries.
4462	
4463	See Section 3.2 entitled 'The Job Tables and the Oldest Active
4464	and Newest Active Indexes'.
4465	See Section 3.5 entitled 'Job Identification'.
4466	See also
4467	
4468	jmGeneralNewestActiveJobIndex for the largest value of
4469	jmJobIndex.
4470	See JmJobSubmissionIDTypeTC for a limit on the size of this
4471	index if the agent represents it as an 8-digit decimal number."
4472	::= { jmJobEntry 1 }
4473	
4474	
4475	
4476	jmJobState OBJECT-TYPE
4477	SYNTAX JmJobStateTC
4478	MAX-ACCESS read-only
4479	STATUS current
4480	DESCRIPTION
4481	"The current state of the job (pending, processing, completed,
4482	etc.). Agents SHALL implement only those states which are
4483	appropriate for the particular implementation. However,
4484	management applications SHALL be prepared to receive all the
4485	standard job states.
4486	
4487	The final value for this object SHALL be one of: completed,
4488	canceled, or aborted. The minimum length of time that the
4489	agent SHALL maintain MIB data for a job in the completed,
4490	canceled, or aborted state before removing the job data from
4491	the jmJobIDTable and jmJobTable is specified by the value of
4492	the jmGeneralJobPersistence object."
4493	DEFVAL { unknown } default is unknown
4494	::= { jmJobEntry 2 }
4495	

4496 jmJobStateReasons1 OBJECT-TYPE 4497 SYNTAX JmJobStateReasons1TC 4498 MAX-ACCESS read-only 4499 STATUS current 4500 DESCRIPTION 4501 "Additional information about the job's current state, i.e., 4502 information that augments the value of the job's jmJobState 4503 object. 4504 4505 Implementation of any reason values is OPTIONAL, but an agent 4506 SHOULD return any reason information available. These values 4507 MAY be used with any job state or states for which the reason makes sense. Since the Job State Reasons will be more dynamic 4508 than the Job State, it is recommended that a job monitoring 4509 application read this object every time jmJobState is read. 4510 4511 When the agent cannot provide a reason for the current state of 4512 the job, the value of the jmJobStateReasons1 object and 4513 jobStateReasonsN attributes SHALL be 0. 4514 4515 The jobStateReasonsN (N=2..4) attributes provide further 4516 additional information about the job's current state." 4517 { 0 } -- no reasons DEFVAL 4518 ::= { jmJobEntry 3 } 4519 4520 4521 4522 jmNumberOfInterveningJobs OBJECT-TYPE 4523 SYNTAX Integer32 (-2..2147483647) 4524 MAX-ACCESS read-only 4525 STATUS current 4526 DESCRIPTION 4527 "The number of jobs that are expected to complete processing before this job has completed processing according to the 4528 implementation's queuing algorithm, if no other jobs were to be 4529 4530 submitted. In other words, this value is the job's queue 4531 position. The agent SHALL return a value of 0 for this 4532 attribute when the job is the next job to complete processing 4533 (or has completed processing)." -- default is no intervening jobs. 4534 DEFVAL { 0 } ::= { jmJobEntry 4 } 4535 4536

4537 jmJobKOctetsPerCopyRequested OBJECT-TYPE 4538 SYNTAX Integer32 (-2..2147483647) MAX-ACCESS read-only 4539 4540 STATUS current 4541 DESCRIPTION 4542 "The total size in K (1024) octets of the document(s) being requested to be processed in the job. The agent SHALL round the actual number of octets up to the next highest K. Thus 0 4543 4544 octets is represented as '0', 1-1024 octets is represented as 4545 4546 '1', 1025-2048 is represented as '2', etc. 4547 4548 In computing this value, the server/device SHALL NOT include 4549 the multiplicative factors contributed by (1) the number of 4550 document copies, and (2) the number of job copies, independent 4551 of whether the device can process multiple copies of the job or 4552 document without making multiple passes over the job or 4553 document data and independent of whether the output is collated or not. Thus the server/device computation is independent of 4554 the implementation and indicates the size of the document(s) 4555 4556 measured in K octets independent of the number of copies." 4557 DEFVAL { -2 } -- the default is unknown(-2) 4558 ::= { jmJobEntry 5 } 4559 4560 4561 4562 jmJobKOctetsProcessed OBJECT-TYPE 4563 SYNTAX Integer32 (-2..2147483647) 4564 MAX-ACCESS read-only 4565 STATUS current 4566 DESCRIPTION 4567 "The total number of octets processed by the server or device measured in units of K (1024) octets so far. The agent SHALL 4568 4569 round the actual number of octets processed up to the next 4570 higher K. Thus 0 octets is represented as '0', 1-1024 octets is represented as '1', 1025-2048 octets is '2', etc. For 4571 4572 printing devices, this value is the number interpreted by the 4573 page description language interpreter rather than what has been 4574 marked on media. 4575 4576 For implementations where multiple copies are produced by the 4577 interpreter with only a single pass over the data, the final 4578 value SHALL be equal to the value of the jmJobKOctetsPerCopyRequested object. For implementations where 4579 multiple copies are produced by the interpreter by processing 4580 4581 the data for each copy, the final value SHALL be a multiple of 4582 the value of the jmJobKOctetsPerCopyRequested object. 4583 4584 NOTE - See the impressionsCompletedCurrentCopy and pagesCompletedCurrentCopy attributes for attributes that are 4585 4586 reset on each document copy. 4587

```
INTERNET-DRAFT
                             Job Monitoring MIB, V1.3 November 8, 1998
4588
              NOTE - The jmJobKOctetsProcessed object can be used with the
4589
              jmJobKOctetsPerCopyRequested object to provide an indication of
4590
              the relative progress of the job, provided that the
4591
              multiplicative factor is taken into account for some
4592
              implementations of multiple copies."
4593
                                -- default is no octets processed.
          DEFVAL
                      { 0 }
          ::= { jmJobEntry 6 }
4594
4595
4596
4597
      jmJobImpressionsPerCopyRequested OBJECT-TYPE
4598
          SYNTAX
                      Integer32 (-2..2147483647)
4599
          MAX-ACCESS read-only
4600
          STATUS
                      current
4601
          DESCRIPTION
              "The total size in number of impressions of the document(s)
4602
4603
              submitted.
4604
4605
              In computing this value, the server/device SHALL NOT include
              the multiplicative factors contributed by (1) the number of
4606
4607
              document copies, and (2) the number of job copies, independent
4608
              of whether the device can process multiple copies of the job or
4609
              document without making multiple passes over the job or
              document data and independent of whether the output is collated
4610
4611
              or not. Thus the server/device computation is independent of
4612
              the implementation and reflects the size of the document(s)
4613
              measured in impressions independent of the number of copies.
4614
4615
              See the definition of the term 'impression' in Section 2."
                      \{-2\} -- default is unknown(-2)
          DEFVAL
4616
          ::= { jmJobEntry 7 }
4617
4618
4619
4620
      jmJobImpressionsCompleted OBJECT-TYPE
4621
          SYNTAX
                      Integer32 (-2..2147483647)
4622
          MAX-ACCESS read-only
4623
          STATUS
                      current
4624
          DESCRIPTION
               "The total number of impressions completed for this job so far.
4625
4626
              For printing devices, the impressions completed includes
4627
              interpreting, marking, and stacking the output. For other
              types of job services, the number of impressions completed
4628
4629
              includes the number of impressions processed.
4630
4631
              NOTE - See the impressionsCompletedCurrentCopy and
4632
              pagesCompletedCurrentCopy attributes for attributes that are
4633
              reset on each document copy.
4634
4635
              NOTE - The jmJobImpressionsCompleted object can be used with
              the jmJobImpressionsPerCopyRequested object to provide an
4636
4637
              indication of the relative progress of the job, provided that
4638
              the multiplicative factor is taken into account for some
4639
              implementations of multiple copies.
```

Bergman, Hastings, Isaacson, Lewis Informational

[Page 113]

```
4640
4641
              See the definition of the term 'impression' in Section 2 and
4642
              the counting example in Section 3.4 entitled 'Monitoring Job
              Progress'."
4643
          DEFVAL \{0\}
4644
                             -- default is no octets
4645
          ::= { jmJobEntry 8 }
4646
4647
4648
4649
      jmJobOwner OBJECT-TYPE
4650
          SYNTAX JmJobStringTC (SIZE(0..63))
4651
          MAX-ACCESS read-only
4652
          STATUS
                    current
4653
          DESCRIPTION
              "The coded character set name of the user that submitted the
4654
4655
              job.
                    The method of assigning this user name will be system
4656
              and/or site specific but the method MUST ensure that the name
4657
              is unique to the network that is visible to the client and
4658
              target device.
4659
4660
              This value SHOULD be the most authenticated name of the user
              submitting the job.
4661
4662
4663
              See the OBJECT compliance macro for the minimum maximum length
4664
              required for conformance."
          DEFVAL { ''H } -- default is empty string
4665
          ::= { jmJobEntry 9 }
4666
4667
4668
4669
4670
```

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 4671 -- The Attribute Group (MANDATORY) 4672 4673 -- The jmAttributeGroup consists entirely of the jmAttributeTable. 4674 \_\_\_ -- Implementation of the objects in this group is MANDATORY. 4675 4676 -- See Section 3.1 entitled 'Conformance Considerations'. 4677 -- An agent SHALL implement any attribute if (1) the server or device -- supports the functionality represented by the attribute and (2) the 4678 4679 -- information is available to the agent. 4680 4681 jmAttribute OBJECT IDENTIFIER ::= { jobmonMIBObjects 4 } 4682 4683 4684 4685 jmAttributeTable OBJECT-TYPE 4686 SYNTAX SEQUENCE OF JmAttributeEntry 4687 MAX-ACCESS not-accessible 4688 STATUS current 4689 DESCRIPTION 4690 "The jmAttributeTable SHALL contain attributes of the job and 4691 document(s) for each job in a job set. Instead of allocating 4692 distinct objects for each attribute, each attribute is 4693 represented as a separate row in the jmAttributeTable. 4694 4695 The MANDATORY-GROUP macro specifies that this group is 4696 MANDATORY. An agent SHALL implement any attribute if (1) the server or device supports the functionality represented by the 4697 attribute and (2) the information is available to the agent. " 4698 4699 ::= { jmAttribute 1 } 4700 4701 4702 4703 jmAttributeEntry OBJECT-TYPE 4704 SYNTAX JmAttributeEntry 4705 MAX-ACCESS not-accessible 4706 STATUS current 4707 DESCRIPTION 4708 "Attributes representing information about the job and 4709 document(s) or resources required and/or consumed. 4710 4711 Each entry in the jmAttributeTable is a per-job entry with an 4712 extra index for each type of attribute (jmAttributeTypeIndex) 4713 that a job can have and an additional index 4714 (jmAttributeInstanceIndex) for those attributes that can have 4715 multiple instances per job. The jmAttributeTypeIndex object 4716 SHALL contain an enum type that indicates the type of attribute 4717 (see the JmAttributeTypeTC textual-convention). The value of 4718 the attribute SHALL be represented in either the jmAttributeValueAsInteger or jmAttributeValueAsOctets objects, 4719 4720 and/or both, as specified in the JmAttributeTypeTC textual-4721 convention. 4722

Bergman, Hastings, Isaacson, Lewis Informational [Page 115]

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733	server or the job s As the do additiona this tabl actually jmAttribu	device is able to o ubmission protocol is cuments are interpre- l attributes and so e. As the attribute consumed, the usage teValueAsInteger obj	In the jmAttributeTable as the discover the attributes either from tself or from the document PDL. eted, the interpreter MAY discover the agent adds additional rows to es that represent resources are counter contained in the ject is incremented according to escription of the JmAttributeTypeTC
4734 4735 4736	at least		n row in the jmAttributeTable for er a job completes as specified by ence object.
4737 4738 4739 4740	Zero or m a job set		ist in this table for each job in
4740 4741 4742 4743 4744 4745	descripti	on of the jmAttribut neralJobSetIndex, jm stanceIndex }	e Attribute Mechanism' for a eTable." JobIndex, jmAttributeTypeIndex,
4746 4747 4748 4749 4750 4751 4752 4753	JmAttributeEntry jmAttributeTy jmAttributeIn jmAttributeVa jmAttributeVa }	peIndex stanceIndex lueAsInteger	JmAttributeTypeTC, Integer32 (132767), Integer32 (-22147483647), OCTET STRING(SIZE(063))

4754 jmAttributeTypeIndex OBJECT-TYPE 4755 SYNTAX JmAttributeTypeTC 4756 MAX-ACCESS not-accessible 4757 STATUS current 4758 DESCRIPTION 4759 "The type of attribute that this row entry represents. 4760 The type MAY identify information about the job or document(s) 4761 4762 or MAY identify a resource required to process the job before 4763 the job start processing and/or consumed by the job as the job 4764 is processed. 4765 Examples of job attributes (i.e., apply to the job as a whole) 4766 4767 that have only one instance per job include: jobCopiesRequested(90), documentCopiesRequested(92), 4768 4769 jobCopiesCompleted(91), documentCopiesCompleted(93), while 4770 examples of job attributes that may have more than one instance 4771 per job include: documentFormatIndex(37), and documentFormat(38). 4772 4773 4774 Examples of document attributes (one instance per document) 4775 include: fileName(34), and documentName(35). 4776 4777 Examples of required and consumed resource attributes include: 4778 pagesRequested(130), mediumRequested(170), pagesCompleted(131), 4779 and mediumConsumed(171), respectively." ::= { jmAttributeEntry 1 } 4780 4781 4782 4783 4784 jmAttributeInstanceIndex OBJECT-TYPE 4785 SYNTAX Integer32 (1..32767) 4786 MAX-ACCESS not-accessible 4787 STATUS current 4788 DESCRIPTION 4789 "A running 16-bit index of the attributes of the same type for 4790 each job. For those attributes with only a single instance per job, this index value SHALL be 1. For those attributes that 4791 4792 are a single value per document, the index value SHALL be the document number, starting with 1 for the first document in the 4793 job. Jobs with only a single document SHALL use the index 4794 4795 value of 1. For those attributes that can have multiple values 4796 per job or per document, such as documentFormatIndex(37) or 4797 documentFormat(38), the index SHALL be a running index for the 4798 job as a whole, starting at 1." 4799 ::= { jmAttributeEntry 2 } 4800

Job Monitoring MIB, V1.3 November 8, 1998

INTERNET-DRAFT

4801 jmAttributeValueAsInteger OBJECT-TYPE 4802 SYNTAX Integer32 (-2..2147483647) 4803 MAX-ACCESS read-only 4804 STATUS current 4805 DESCRIPTION 4806 "The integer value of the attribute. The value of the attribute SHALL be represented as an integer if the enum 4807 description in the JmAttributeTypeTC textual-convention 4808 4809 definition has the tag: 'INTEGER:'. 4810 4811 Depending on the enum definition, this object value MAY be an 4812 integer, a counter, an index, or an enum, depending on the jmAttributeTypeIndex value. The units of this value are 4813 4814 specified in the enum description. 4815 4816 For those attributes that are accumulating job consumption as the job is processed as specified in the JmAttributeTypeTC 4817 textual-convention, SHALL contain the final value after the job 4818 completes processing, i.e., this value SHALL indicate the total 4819 4820 usage of this resource made by the job. 4821 4822 A monitoring application is able to copy this value to a 4823 suitable longer term storage for later processing as part of an 4824 accounting system. 4825 4826 Since the agent MAY add attributes representing resources to this table while the job is waiting to be processed or being 4827 processed, which can be a long time before any of the resources 4828 are actually used, the agent SHALL set the value of the 4829 4830 jmAttributeValueAsInteger object to 0 for resources that the 4831 job has not yet consumed. 4832 4833 Attributes for which the concept of an integer value is 4834 meaningless, such as fileName(34), jobName, and 4835 processingMessage, do not have the 'INTEGER:' tag in the 4836 JmAttributeTypeTC definition and so an agent SHALL always 4837 return a value of '-1' to indicate 'other' for the value of the jmAttributeValueAsInteger object for these attributes. 4838 4839 4840 For attributes which do have the 'INTEGER:' tag in the JmAttributeTypeTC definition, if the integer value is not (yet) 4841 4842 known, the agent either (1) SHALL not materialize the row in the jmAttributeTable until the value is known or (2) SHALL 4843 4844 return a '-2' to represent an 'unknown' counting integer value, a '0' to represent an 'unknown' index value, and a '2' to 4845 4846 represent an 'unknown(2)' enum value." 4847 DEFVAL { -2 } -- default value is unknown(-2) ::= { jmAttributeEntry 3 } 4848 4849

4850 jmAttributeValueAsOctets OBJECT-TYPE 4851 SYNTAX OCTET STRING(SIZE(0..63)) 4852 MAX-ACCESS read-only 4853 STATUS current 4854 DESCRIPTION 4855 "The octet string value of the attribute. The value of the attribute SHALL be represented as an OCTET STRING if the enum 4856 description in the JmAttributeTypeTC textual-convention 4857 4858 definition has the tag: 'OCTETS:'. 4859 4860 Depending on the enum definition, this object value MAY be a 4861 coded character set string (text), such as 'JmUTF8StringTC', or 4862 a binary octet string, such as 'DateAndTime'. 4863 Attributes for which the concept of an octet string value is 4864 4865 meaningless, such as pagesCompleted, do not have the tag 'OCTETS:' in the JmAttributeTypeTC definition and so the agent 4866 4867 SHALL always return a zero length string for the value of the 4868 jmAttributeValueAsOctets object. 4869 4870 For attributes which do have the 'OCTETS:' tag in the JmAttributeTypeTC definition, if the OCTET STRING value is not 4871 (yet) known, the agent either SHALL NOT materialize the row in 4872 4873 the jmAttributeTable until the value is known or SHALL return a 4874 zero-length string." 4875 DEFVAL { ''H } -- empty string ::= { jmAttributeEntry 4 } 4876 4877

4878	
4879	The Mirror Attribute Group (OPTIONAL)
4880	
4881	The jmMirrorAttrGroup consists entirely of the jmMirrorAttrTable.
4882	
4883	Implementation of the objects in this group is OPTIONAL.
4884	Con Soution 2 1 ontitled (Conformance Congiderations)
	See Section 3.1 entitled 'Conformance Considerations'.
4885	The jmMirrorAttrTable complements the MANDATORY jmAttributeTable.
4886	 The jmMirrorAttrTable provides access to all of the attributes that an implementation supports, sorted by attribute type (traditional
4887	The jmMirrorAttrTable provides access to all of the attributes that
4888	
4889	SNMP MIB access), rather than being sorted by job set and job index
4890	(modern object-oriented access) as in the analogous
4891	jmAttributeTable.
4892	
4893	jmMirrorAttr OBJECT IDENTIFIER ::= { jobmonMIBObjects 5 }
4894	
4895	jmMirrorAttrTable OBJECT-TYPE
4896	SYNTAX SEQUENCE OF JmAttributeEntry
4897	MAX-ACCESS not-accessible
4898	STATUS current
4899	DESCRIPTION
4900	"The jmMirrorAttrTable is an OPTIONAL table which provides
4900 4901	identical attributes to the jmAttributeTable but with a
4902	different index structure. See jmAttributeTable for further
4903	details.
4904	
4905	See Section 3.3 entitled 'The Attribute Mechanism' for a
4906	description of the jmMirrorAttrTable."
4907	::= { jmMirror 1 }
4908	
4909	
4910	
4911	jmMirrorAttrEntry OBJECT-TYPE
4912	SYNTAX JmMirrorAttrEntry
4913	MAX-ACCESS not-accessible
4914	STATUS current
4915	DESCRIPTION
4916	"The attributes that represent information about each job and
4917	documents or resources required and/or consumed.
4918	
4919	Each entry in jmMirrorAttrTable is a per-attribute entry with a
4920	primary index for each type of attribute jmMirrorAttrTypeIndex)
4921	that a job can have and secondary indices which specify job set
4922	(jmJobSetIndex), job instance (jmJobIndex), and attribute
4923	instance (jmMirrorAttrInstanceIndex).
4924	
4925	An agent which implements the jmMirrorAttrTable SHALL create
4926	and maintain a row in the jmMirrorAttrTable for each
4927	corresponding row in the jmAttributeTable."
4928	<pre>INDEX { jmMirrorAttrTypeIndex, jmGeneralJobSetIndex, jmJobIndex,</pre>
4929	jmMirrorAttrInstanceIndex }

Bergman, Hastings, Isaacson, Lewis Informational [Page 120]

Job Monitoring MIB, V1.3 November 8, 1998

[Page 121]

4930 ::= { jmMirrorAttrTable 1 } 4931 4932 JmMirrorAttrEntry ::= SEQUENCE { jmMirrorAttrTypeIndex 4933 JmAttributeTypeTC, Integer32 (1..32767), 4934 jmMirrorAttrInstanceIndex Integer32 (-2..2147483647), 4935 imMirrorAttrValueAsInteger OCTET STRING(SIZE(0..63)) 4936 jmMirrorAttrValueAsOctets } 4937 4938 4939 jmMirrorAttrTypeIndex OBJECT-TYPE 4940 JmAttributeTypeTC SYNTAX 4941 MAX-ACCESS not-accessible 4942 STATUS current 4943 DESCRIPTION 4944 "The type of attribute that this row entry represents. 4945 See jmAttributeTypeIndex in jmAttributeTable for complete 4946 4947 description." 4948 ::= { jmMirrorAttrEntry 1 } 4949 4950 jmMirrorAttrInstanceIndex OBJECT-TYPE Integer32 (1..32767) 4951 SYNTAX not-accessible 4952 MAX-ACCESS 4953 STATUS current 4954 DESCRIPTION 4955 "The instance of attribute that this row entry represents. 4956 4957 See jmAttributeInstanceIndex in jmAttributeTable for complete 4958 description." ::= { jmMirrorAttrEntry 2 } 4959 4960 jmMirrorAttrValueAsInteger OBJECT-TYPE 4961 Integer32 (-2..2147483647) 4962 SYNTAX 4963 MAX-ACCESS read-only 4964 STATUS current 4965 DESCRIPTION 4966 "The integer value of the attribute. 4967 See jmAttributeValueAsInteger in jmAttributeTable for complete 4968 4969 description." -2 } -- default value is unknown(-2) 4970 DEFVAL 4971 ::= { jmMirrorAttrEntry 3 } 4972 4973 jmMirrorAttrValueAsOctets OBJECT-TYPE 4974 SYNTAX OCTET STRING(SIZE(0..63)) 4975 MAX-ACCESS read-only 4976 STATUS current 4977 DESCRIPTION 4978 "The octet string value of the attribute. 4979 See jmAttributeValueAsOctets in jmAttributeTable for complete 4980 description." 4981

Bergman, Hastings, Isaacson, Lewis Informational

 4982
 DEFVAL { ' 'H }
 -- empty string

 4983
 ::= { jmMirrorAttrEntry 4 }

Bergman, Hastings, Isaacson, Lewis Informational [Page 122]

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 4984 -- Notifications and Trapping 4985 -- Reserved for the future 4986 4987 jobmonMIBNotifications OBJECT IDENTIFIER ::= { jobmonMIB 2 } 4988 4989 4990 -- Conformance Information 4991 4992 jmMIBConformance OBJECT IDENTIFIER ::= { jobmonMIB 3 } 4993 4994 4995 4996 4997 -- compliance statements 4998 jmMIBCompliance MODULE-COMPLIANCE 4999 STATUS current 5000 DESCRIPTION 5001 "The compliance statement for agents that implement the job monitoring MIB." 5002 5003 MODULE -- this module 5004 MANDATORY-GROUPS { jmGeneralGroup, jmJobIDGroup, jmJobGroup, jmAttributeGroup } 5005 5006 5007 GROUP jmMirrorAttrGroup 5008 DESCRIPTION 5009 "The mirror attribute group (sorted by attribute type). 5010 Implementation of this group is OPTIONAL. 5011 5012 An agent that implements the jmMirrorAttrTable SHALL create and 5013 maintain for the same time a row in the jmMirrorAttrTable for 5014 each corresponding row in the jmAttributeTable." 5015 5016 OBJECT jmGeneralJobSetName 5017 JmUTF8StringTC (SIZE(0..8)) SYNTAX 5018 DESCRIPTION 5019 "Only 8 octets maximum string length NEED be supported by the agent." 5020 5021 5022 OBJECT imJobOwner 5023 SYNTAX JmJobStringTC (SIZE(0..16)) 5024 DESCRIPTION 5025 "Only 16 octets maximum string length NEED be supported by the 5026 agent." 5027 -- There are no CONDITIONALLY MANDATORY or OPTIONAL groups. 5028 5029 5030 ::= { jmMIBConformance 1 } 5031

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998 5032 jmMIBGroups OBJECT IDENTIFIER ::= { jmMIBConformance 2 } 5033 5034 jmGeneralGroup OBJECT-GROUP OBJECTS { 5035 5036 jmGeneralNumberOfActiveJobs, jmGeneralOldestActiveJobIndex, jmGeneralNewestActiveJobIndex, jmGeneralJobPersistence, 5037 jmGeneralAttributePersistence, jmGeneralJobSetName} 5038 5039 STATUS current 5040 DESCRIPTION 5041 "The general group." 5042 ::= { jmMIBGroups 1 } 5043 5044 5045 5046 jmJobIDGroup OBJECT-GROUP 5047 OBJECTS { 5048 jmJobIDJobSetIndex, jmJobIDJobIndex } STATUS current 5049 5050 DESCRIPTION 5051 "The job ID group." 5052 ::= { jmMIBGroups 2 } 5053 5054 5055 5056 jmJobGroup OBJECT-GROUP OBJECTS { 5057 jmJobState, jmJobStateReasons1, jmNumberOfInterveningJobs, 5058 5059 jmJobKOctetsPerCopyRequested, jmJobKOctetsProcessed, 5060 jmJobImpressionsPerCopyRequested, jmJobImpressionsCompleted, 5061 jmJobOwner } 5062 STATUS current 5063 DESCRIPTION "The job group." 5064 5065 ::= { jmMIBGroups 3 } 5066 5067 5068 5069 jmAttributeGroup OBJECT-GROUP 5070 OBJECTS { jmAttributeValueAsInteger, jmAttributeValueAsOctets } 5071 5072 STATUS current 5073 DESCRIPTION 5074 "The attribute group." 5075 ::= { jmMIBGroups 4 } 5076 5077 5078 jmMirrorAttrGroup OBJECT-GROUP OBJECTS { 5079 jmMirrorAttrValueAsInteger, jmMirrorAttrValueAsOctets } 5080 5081 STATUS current 5082 DESCRIPTION

5083		"The mirror attribute group (sorted by attribute type).
5084		Implementation of this group is OPTIONAL.
5085		
5086		An agent which implements the jmMirrorAttrTable SHALL create
5087		and maintain for the same time a row in the jmMirrorAttrTable
5088		for each corresponding row in the jmAttributeTable."
5089		::= { jmMIBGroups 5 }
5090		
5091		
5092	END	

Bergman, Hastings, Isaacson, Lewis Informational [Page 125]

## 5093 5 Appendix A - Implementing the Job Life Cycle

5094 The job object has well-defined states and client operations that 5095 affect the transition between the job states. Internal server and 5096 device actions also affect the transitions of the job between the job 5097 states. These states and transitions are referred to as the job's life 5098 cycle.

Not all implementations of job submission protocols have all of the 5099 5100 states of the job model specified here. The job model specified here 5101 is intended to be a superset of most implementations. It is the 5102 purpose of the agent to map the particular implementation's job life 5103 cycle onto the one specified here. The agent MAY omit any states not 5104 implemented. Only the processing and completed states are required to be implemented by an agent. However, a conforming management 5105 5106 application SHALL be prepared to accept any of the states in the job 5107 life cycle specified here, so that the management application can 5108 interoperate with any conforming agent.

5109 The job states are intended to be user visible. The agent SHALL make these states visible in the MIB, but only for the subset of job states 5110 5111 that the implementation has. Some implementations MAY need to have sub-states of these user-visible states. The jmJobStateReasons1 object 5112 and the jobStateReasonsN (N=2..4) attributes can be used to represent 5113 5114 the sub-states of the jobs.

5115 Job states are intended to last a user-visible length of time in most 5116 implementations. However, some jobs may pass through some states in 5117 zero time in some situations and/or in some implementations.

5118 The job model does not specify how accounting and auditing is 5119 implemented, except to assume that accounting and auditing logs are separate from the job life cycle and last longer than job entries in 5120 the MIB. Jobs in the completed, aborted, or canceled states are not 5121 5122 logs, since jobs in these states are accessible via SNMP protocol 5123 operations and SHALL be removed from the Job Monitoring MIB tables 5124 after a site-settable or implementation-defined period of time. An 5125 accounting application MAY copy accounting information incrementally to 5126 an accounting log as a job processes, or MAY be copied while the job is in the canceled, aborted, or completed states, depending on 5127 5128 implementation. The same is true for auditing logs.

5129 The jmJobState object specifies the standard job states. The normal 5130 job state transitions are shown in the state transition diagram 5131 presented in Table 1.

5132 6 APPENDIX B - Support of Job Submission Protocols

5133 A companion PWG document, entitled "Job Submission Protocol Mapping Recommendations for the Job Monitoring MIB" [protomap] contains the 5134 5135 recommended usage of each of the objects and attributes in this MIB 5136 with a number of job submission protocols. In particular, which job 5137 submission ID format should be used is indicated for each job 5138 submission protocol.

5139 Some job submission protocols have support for the client to specify a 5140 job submission ID. A second approach is to enhance the document format to embed the job submission ID in the document data. This second 5141 5142 approach is independent of the job submission protocol. This appendix 5143 lists some examples of these approaches.

5144 Some PJL implementations wrap a banner page as a PJL job around a job submitted by a client. If this results in multiple job submission IDs, 5145 the agent SHALL create multiple jmJobIDEntry rows in the jmJobIDTable that each point to the same job entry in the job tables. See the 5146 5147 5148 specification of the jmJobIDEntry.

5149 7 References

[char-set-policy] Harald Avelstrand, "IETF Policy on Character Sets and Language", June 1997. Latest draft: <draft-avelstrand-charset-5150 5151 5152 policy-00.txt>

- 5153 [GB2312] GB 2312-1980, "Chinese People's Republic of China (PRC) mixed one byte and two byte coded character set" 5154
- 5155 [hr-mib] P. Grillo, S. Waldbusser, "Host Resources MIB", RFC 1514, 5156 September 1993
- 5157 [iana] J. Reynolds, and J. Postel, "Assigned Numbers", STD 2, RFC 1700, 5158 ISI, October 1994.

[IANA-charsets] Coded Character Sets registered by IANA and assigned an 5159 5160 enum value for use in the CodedCharSet textual convention imported from 5161 the Printer MIB. See ftp://ftp.isi.edu/innotes/iana/assignments/character-sets 5162

5163 [iana-media-types] IANA Registration of MIME media types (MIME content 5164 types/subtypes). See ftp://ftp.isi.edu/in-notes/iana/assignments/

[ipp-model] Internet Printing Protocol/1.0: Model and Semantics, work 5165 5166 in progress on the IETF standards track. See draft-ietf-ipp-model-5167 09.txt. See also http://www.pwg.org/ipp/index.html

5168 [ISO-639] ISO 639:1988 (E/F) - Code for Representation of names of 5169 languages - The International Organization for Standardization, 1st 5170 edition, 1988.

Bergman, Hastings, Isaacson, Lewis Informational [Page 127]

- [ISO 646] ISO/IEC 646:1991, "Information technology -- ISO 7-bit coded 5171 5172 character set for information interchange", JTC1/SC2.
- [ISO 8859] ISO/IEC 8859-1:1987, "Information technology -- 8-bit single 5173 5174 byte coded graphic character sets - Part 1: Latin alphabet No. 1, 5175 JTC1/SC2."
- [ISO 2022] ISO/IEC 2022:1994 "Information technology -- Character 5176 5177 code structure and extension techniques", JTC1/SC2.
- 5178 [ISO-3166] ISO 3166:1988 (E/F) - Codes for representation of names of countries - The International Organization for Standardization, 3rd 5179 5180 edition, 1988-08-15."
- [ISO-10646] ISO/IEC 10646-1:1993, "Information technology -- Universal 5181 5182 Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and 5183 Basic Multilingual Plane, JTC1/SC2.
- 5184 [iso-dpa] ISO/IEC 10175 Document Printing Application (DPA). See 5185 ftp://ftp.pwg.org/pub/pwg/dpa/
- [JIS X0208] JIS X0208-1990, "Japanese two byte coded character set." 5186
- 5187 [mib-II] MIB-II, RFC 1213.
- 5188 [print-mib] Smith, R., Wright, F., Hastings, T., Zilles, S. and 5189 Gyllenskog, J., "Printer MIB", RFC 1759, proposed IETF standard, March 5190 1995. See also [print-mib-draft].
- [print-mib-draft] Turner, R., "Printer MIB", work in progress, on the 5191 5192 standards track as a draft standard: <draft-ietf-printmib-mib-info-5193 02.txt>, October 15, 1997.
- 5194 [protomap] Bergman, R., "Job Submission Protocol Mapping Recommendations for the Job Monitoring MIB, "work in progress as an 5195 5196 informational RFC. See <draft-bergman-printmib-job-protomap-01.txt>, 5197 January 12, 1998.
- [pwq] The Printer Working Group is a printer industry consortium open 5198 to any individuals. For more information, access the PWG web page: 5199 http://www.pwg.org 5200
- 5201 [req-words] S. Bradner, "Keywords for use in RFCs to Indicate 5202 Requirement Levels", RFC 2119, March 1997.
- [rfc 1738] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform 5203 Resource Locators (URL)", RFC 1738, December 1994. 5204
- [RFC-1766] Avelstrand, H., "Tags for the Identification of Languages", 5205 5206 RFC 1766, March 1995.

5207 [rfc 2130] C. Weider, C. Preston, K. Simonsen, H. Alvestrand, R. 5208 Atkinson, M. Crispin, and P. Svanberg, "The Report of the IAB Character Set Workshop held 29 Feb-1 March, 1997", April 1997, RFC 2130. 5209

5210 [SMIv2-SMI] J. Case, et al. "Structure of Management Information for 5211 Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 5212 1902, January 1996.

5213 [SMIv2-TC] J. Case, et al. "Textual Conventions for Version 2 of the 5214 Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.

5215 [tipsi] IEEE 1284.1, Transport-independent Printer System Interface 5216 (TIPSI).

[URI-spec] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform 5217 5218 Resource Locators (URL)", RFC 1738, December, 1994.

5219 [US-ASCII] Coded Character Set - 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986. 5220

5221 [UTF-8] F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 5222 10646", RFC 2044, October 1996.

5223 5224 5225 5226 5227 5228	8	Author's Addresses Ron Bergman Dataproducts Corp. 1757 Tapo Canyon Road Simi Valley, CA 93063-3394
5229		Phone: 805-578-4421
5230		Fax: 805-578-4001
5231		Email: rbergman@dpc.com
5232		
5233		
5234		Tom Hastings
5235		Xerox Corporation, ESAE-231
5236		701 S. Aviation Blvd.
5237		El Segundo, CA 90245
5238		-1 010 000 6440
5239		Phone: 310-333-6413
5240		Fax: 310-333-5514
5241		EMail: hastings@cp10.es.xerox.com
5242		
5243 5244		Saott à Tappagon
5244		Scott A. Isaacson Novell, Inc.
5245		122 E 1700 S
5247		Provo, UT 84606
5248		11000, 01 01000
5249		Phone: 801-861-7366
5250		Fax: 801-861-4025

Bergman, Hastings, Isaacson, Lewis Informational

INTERNET-DRAFT Job Monitoring MIB, V1.3 November 8, 1998

5251 5252	EMail: scott_isaacson@novell.com
5253	
5254	Harry Lewis
5255	IBM Corporation
5256	6300 Diagonal Hwy
5257	Boulder, CO 80301
5258	
5259	Phone: (303) 924-5337
5260	Fax:
5261	Email: harryl@us.ibm.com
5262	
5263	
5264	Send questions and comments to the Printer Working Group (PWG)
5265	using the Job Monitoring Project (JMP) Mailing List: jmp@pwg.org
5266	
5267	To learn how to subscribe, send email to: jmp-request@pwg.org
5268	
5269	Implementers of this specification are encouraged to join the jmp
5270	mailing list in order to participate in discussions on any
5271	clarifications needed and registration proposals for additional
5272	attributes and values being reviewed in order to achieve consensus.
5273	
5274	For further information, access the PWG web page under "JMP":
5275	
5276	http://www.pwg.org/
5277	
5278	Other Participants:
5279	Chuck Adams - Tektronix
5280	Jeff Barnett - IBM
5281	Keith Carter, IBM Corporation
5282	Jeff Copeland - QMS
5283	Andy Davidson - Tektronix
5284	Roger deBry - IBM
5285	Mabry Dozier - QMS
5286	Lee Farrell - Canon
5287	Steve Gebert - IBM
5288	Robert Herriot - Sun Microsystems Inc.
5289	Shige Kanemitsu - Kyocera
5290	David Kellerman - Northlake Software
5291	Rick Landau - Digital
5292	Pete Loya - HP
5293	Ray Lutz - Cognisys
5294	Jay Martin - Underscore
5295	Mike MacKay, Novell, Inc.
5296	Stan McConnell - Xerox
5290	Carl-Uno Manros, Xerox, Corp.
5298	Pat Nogay - IBM
5299	Bob Pentecost - HP
5300	Rob Rhoads - Intel

Bergman, Hastings, Isaacson, Lewis Informational [Page 130]

5301	David Roach - Unisys
5302	Stuart Rowley - Kyocera
5303	Hiroyuki Sato - Canon
5304	Bob Setterbo - Adobe
5305	Gail Songer, EFI
5306	Mike Timperman - Lexmark
5307	Randy Turner - Sharp
5308	William Wagner - Digital Products
5309	Jim Walker - Dazel
5310	Chris Wellens - Interworking Labs
5311	Rob Whittle - Novell
5312	Don Wright - Lexmark
5313	Lloyd Young - Lexmark
5314	Atsushi Yuki - Kyocera
E 2 1 E	Deter Tehler Yerey Corp

- 5315 Peter Zehler, Xerox, Corp.
- 5316 9 Change History

5317 This section summarizes the changes in each version after version 1.0 5318 in reverse chronological order.

- 5319 9.1 Changes to produce version 1.3, dated November 8, 1998
- 5320 The following changes were made to version 1.2, dated October 2, 1998 5321 to make version 1.3, dated November 8, 1998:
- 5322 1. Added the Mirror table.
- 5323 2. Moved the JmJobSubmissionIDTypeTC, JmJobStateReasons1TC,
   5324 JmJobStateReasons2TC, JmJobStateReasons3TC, and JmJobStateReasons4TC
   5325 assignments out of the MIB and into the Introduction.
- 5326
- 5327 9.2 Changes to produce version 1.2, dated October 2, 1998
- 5328 The following changes were made to version 1.1, dated October 1, 1998 5329 to make version 1.2, dated October 2, 1998:
- 5330 1. Removed all REFERENCE clauses since they referred to sections in the 5331 specification that were not in the MIB.
- 5332 2. Moved the definitions of the attributes from the TC to a new section 3.3.8.
- 5334 3. Removed the attributes from the Table of Contents
- 5335 4. Added the data types as ASN.1 comments after each attribute enum.
- 5336 5. Changed a number of occurrences of "SHALL" to "is" when they were 5337 just definitions, rather than conformance requirements.

Bergman, Hastings, Isaacson, Lewis Informational [Page 131]

- 5338
- 5339 9.3 Changes to produce version 1.1, dated October 1, 1998
- The following changes were made to version 1.0, dated February 3, 1998 5340 5341 to make version 1.1, dated October 1, 1998:
- 1. Clarified sections 3.3.3 and 3.3.7 so that the DEFVAL of 0 for index 5342 5343 attributes is different from the DEFVAL for 5344 jmAttributeValueAsInteger which is -2.
- 5345 2. Clarified the relationships of the values of the 5346 JmJobCollationTypeTC with the IPP "multiple-document-handling" 5347 attribute.
- 5348 3. Clarified that the values of the mediumRequested(170) and mediumConsumed(171) attributes may be any of the IPP 'media' values 5349 which are media names, media size names, and input tray names. 5350
- 5351 4. Added the two attributes approved by the PWG for registration in 5352 April 1998: mediumTypeConsumed(174) and mediumSizeConsumed(175).
- 5353 5. Changed "insure" to "ensure'.
- 5354 6. Correct an incorrect reference in the jmAttributeEntry DESCRIPTION 5355 from jmJobTable to jmAttributeTable.

Job Monitoring MIB, V1.3 November 8, 1998

5356 10 INDEX

5357 This index includes the textual conventions, the objects, and the attributes. Textual conventions all start with the prefix: "JM" and 5358 end with the suffix: "TC". Objects all starts with the prefix: "jm" 5359 5360 followed by the group name. Attributes are identified with enums, and 5361 so start with any lower case letter and have no special prefix. 5362 5363 colorantConsumed, 42 5364 colorantRequested, 41 5365 deviceNameRequested, 31 documentCopiesCompleted, 36 5366 5367 documentCopiesRequested, 36 5368 documentFormat, 33 5369 documentFormatIndex, 32 5370 documentName, 32 5371 fileName, 32 5372 finishing, 35 5373 fullColorImpressionsCompleted, 38 5374 highlightColorImpressionsCompleted, 39 5375 impressionsCompletedCurrentCopy, 38 5376 impressionsInterpreted, 38 5377 impressionsSentToDevice, 38 5378 impressionsSpooled, 38 5379 jmAttributeInstanceIndex, 117 5380 jmAttributeTypeIndex, 117 5381 JmAttributeTypeTC, 85 5382 jmAttributeValueAsInteger, 118 5383 jmAttributeValueAsOctets, 119 5384 JmBooleanTC, 75 5385 JmFinishingTC, 73 5386 jmGeneralAttributePersistence, 104 jmGeneralJobPersistence, 104 5387 5388 jmGeneralJobSetIndex, 102 5389 jmGeneralJobSetName, 105 5390 jmGeneralNewestActiveJobIndex, 103 5391 jmGeneralNumberOfActiveJobs, 102 5392 jmGeneralOldestActiveJobIndex, 103 5393 JmJobCollationTypeTC, 77 5394 jmJobIDJobIndex, 108 5395 jmJobIDJobSetIndex, 108 5396 jmJobImpressionsCompleted, 113 5397 jmJobImpressionsPerCopyRequested, 113 5398 jmJobIndex, 110 5399 jmJobKOctetsPerCopyRequested, 112 5400 jmJobKOctetsProcessed, 112 5401 jmJobOwner, 114 5402 JmJobServiceTypesTC, 89 5403 JmJobSourcePlatformTypeTC, 72 5404 jmJobState, 110 5405 jmJobStateReasons1, 111 Bergman, Hastings, Isaacson, Lewis Informational [Page 133]

Bergman, Hastings, Isaacson, Lewis

Job Monitoring MIB, V1.3 November 8, 1998

5484