1 INTERNET-DRAFT R. Bergman 2 Dataproducts Corp. 3 T. Hastings 4 Xerox Corporation 5 S. Isaacson 6 Novell, Inc. 7 H. Lewis 8 IBM Corp. 9 January 13, 1998 10 Job Monitoring MIB - V1 11 <draft-ietf-printmib-job-monitor-07.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 progress." 22 23 To learn the current status of any Internet-Draft, please check 24 the "lid-abstracts.txt" listing contained in the Internet-Drafts 25 Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net 2.6 (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 July 13, 1998. 29 30 Abstract 31 This document has been developed and approved by the Printer 32 Working Group (PWG) as a PWG standard. It is intended to be 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 accounting data after the completion of a job. 38 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 42 printing is outside the scope of this Job Monitoring MIB. Future 43 extensions to this MIB may include, but are not limited to, fax 44 machines and scanners.

	INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1	998
45 46	TABLE OF CONTENTS	
47	1. INTRODUCTION	8
48	1.1 Types of Information in the MIB	8
49	1.2 Types of Job Monitoring Applications	10
50	2. TERMINOLOGY AND JOB MODEL	11
51 52 53 54 55	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 2.1.3 Configuration 3 - client-server-printer - client monitor	
56 57	<pre>printer agent and server 3. MANAGED OBJECT USAGE</pre>	16 18
58 59 60 61 62 63 64	 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 	18 18 19 19 19 19
65	3.2 The Job Tables and the Oldest Active and Newest Active Indexes	19
66 67 68 69	3.3 The Attribute Mechanism 3.3.1 Conformance of Attribute Implementation 3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes	21 22 22
70 71 72 73 74 75	 3.3.3 Data Sub-types and Attribute Naming Conventions 3.3.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes 3.3.5 Requested Objects and Attributes 3.3.6 Consumption Attributes 3.3.7 Index Value Attributes 	23 24 24 25 25
76	3.4 Monitoring Job Progress	25
77	3.5 Job Identification	29
78 79 80 81	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	30 30 31 32

	INTERNET-DRAFT Job Mc	onitoring MIB,	V1.0	January	1998
82 83 84 85 86 87 88 89 90	3.7 IANA and PWG Registration 3.7.1 PWG Registration 3.7.1.1 Type 1 enumer 3.7.1.2 Type 2 enumer 3.7.1.3 Type 3 enumer 3.7.2 PWG Registration 3.7.3 PWG Registration 3.7.4 PWG Registration formats 34	of enums cations cations cation of type 2 bit of Job Submis	values sion Id Formats	ocument-	32 33 33 34 34 34
91 92 93	3.8 Security Considerations 3.8.1 Read-Write object 3.8.2 Read-Only Objects	S	r's Jobs		34 34 35
94	3.9 Notifications				35
95	4. MIB SPECIFICATION				35
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118	Textual Conventions for thi JmUTF8StringTC JmJobStringTC JmNaturalLanguageTagTC JmTimeStampTC JmJobSourcePlatformTypeTC JmFinishingTC JmPrintQualityTC JmPrinterResolutionTC JmTonerEconomyTC JmBooleanTC JmMediumTypeTC JmJobStateTC JmJobStateTC JmAttributeTypeTC other (Int32(-2) Job State attributes jobStateReasons2 (3 jobStateReasons3 (3 jobStateReasons4 (3 processingMessageNatu jobCodedCharSet (Composite the state of the state o	and/or Octets JmJobStateReas JmJobStateReas JmJobStateReas (UTF8String63) uralLangTag odedCharSet)	ons2TC) ons3TC)		37 37 37 37 38 39 40 41 41 41 41 41 41 50 51 51 51 51 51 52 52
119 120 121 122 123 124 125 126 127 128	jobNaturalLanguageTag Job Identification at jobURI (Octets(06 jobAccountName (Oct serverAssignedJobName jobName (JobString6	g (Octets63) tributes 53)) tets63) e (JobString 53) mJobServiceTyp c (Int32(0 e (JmJobSour	esTC))) cePlatformTypeTC	2)	53 53 53 54 54 55 55 55 55

[Page 4]

	INTERNET-DRAFT Job	Monitoring MIB,	v1.0	January 1998
129	submittingApplicat			55
130	jobOriginatingHost			56
131	deviceNameRequeste			56
132	queueNameRequested			56
133	physicalDevice (nd/or UTF8String6	
134	numberOfDocuments			56
135	fileName (JobStr	U		57
136	documentName (Jo			57
137	jobComment (JobS			57
138	documentFormatInde			57
139	documentFormat (angramily re and/o	
140	Job Parameter attr			58 58
141 142	jobPriority (Int			58 59
142 143	jobProcessAfterDat jobHold (JmBoole		eandiime)	59 59
143 144	jobHoldUntil (Jumbolie			59
144 145	outputBin (Int32		S + $ring (3)$	59
146	sides (Int32(-2.		Jaci IIIgos /	60
140	finishing (JmFir			60 60
148	Image Quality attrik		and used)	60 60
149	printQualityReques			60 60
150	printQualityUsed			60 60
151	printerResolutionF			
152	printerResolution			c, 60
153	tonerEcomonyReques			60
154	tonerEcomonyUsed			60
155	tonerDensityReques			60
156	tonerDensityUsed			61
157	Job Progress attribu			61
158	jobCopiesRequested			61
159	jobCopiesCompleted			61
160	documentCopiesRequ			61
161	documentCopiesComp			61
162	jobKOctetsTransfer	red (Int32(-2.))	62
163	sheetCompletedCopy	Number (Int32)	(-2))4	62
164	sheetCompletedDocu			62
165	jobCollationType			62
166	Impression attribu			63
167	impressionsSpooled			63
168	impressionsSentToL			63
169	impressionsInterpr			63
170	impressionsComplet			63
171	fullColorImpressio			63
172	highlightColorImpr			64
173 174	Page attributes (rec		amea)	64 64
174 175	pagesRequested (pagesCompleted (64 64
175 176	pagesCompleted		2(-2)	65
177	Sheet attributes (65
178	sheetsRequested			65
179	sheetsCompleted			65
180	sheetsCompletedCur		32(-2))	65
-	Denimina Mastina Tarada		· · · · · · ·	

181	Resource attributes (requested and consumed)	65
182	mediumRequested (JmMediumTypeTC and/or JobString63)	66
183	mediumConsumed (Int32(-2) and/or JobString63)	66
184	colorantRequested (Int32(-2) and/or JobString63)	66
185	colorantConsumed (Int32(-2) and/or JobString63)	67
186	Time attributes (set by server or device)	67
187	jobSubmissionToServerTime (JmTimeStampTC and/or DateAnd	
188	jobSubmissionTime (JmTimeStampTC and/or DateAndTime)	67
189	jobStartedBeingHeldTime (JmTimeStampTC)	68
190	jobStartedProcessingTime (JmTimeStampTC and/or DateAnd	[ime)68
191	jobCompletionTime (JmTimeStampTC and/or DateAndTime)	68
192	jobProcessingCPUTime (Int32(-2))	68
193	JmJobServiceTypesTC	70
194	JmJobStateReasons1TC	72
195	JmJobStateReasons2TC	76
196	JmJobStateReasons3TC	80
197	JmJobStateReasons4TC	80
198	The General Group (MANDATORY)	81
199	jmGeneralJobSetIndex (Int32(132767))	82
200	jmGeneralNumberOfActiveJobs (Int32(0))	82
201	jmGeneralOldestActiveJobIndex (Int32(0))	83
202	jmGeneralNewestActiveJobIndex (Int32(0))	83
	jmGeneralJobPersistence (Int32(15))	
203		84
204	jmGeneralAttributePersistence (Int32(15))	84
205	jmGeneralJobSetName (UTF8String63)	85
206	The Job ID Group (MANDATORY)	85
207	jmJobSubmissionID (OCTET STRING(SIZE(48)))	87
208	jmJobIDJobSetIndex (Int32(032767))	88
209	jmJobIDJobIndex (Int32(0))	88
210	The Job Group (MANDATORY)	88
211	jmJobIndex (Int32(1))	90
212	jmJobState (JmJobStateTC)	90
212		91
	jmJobStateReasons1 (JmJobStateReasons1TC)	
214	jmNumberOfInterveningJobs (Int32(-2))	91
215	jmJobKOctetsPerCopyRequested (Int32(-2))	92
216	jmJobKOctetsProcessed (Int32(-2))	92
217	jmJobImpressionsPerCopyRequested (Int32(-2))	93
218	jmJobImpressionsCompleted (Int32(-2))	93
219	jmJobOwner (JobString63)	94
220	The Attribute Group (MANDATORY)	94
221	jmAttributeTypeIndex (JmAttributeTypeTC)	97
222	jmAttributeInstanceIndex (Int32(132767))	97
223	jmAttributeValueAsInteger (Int32(-2))	98
	jmAttributeValueAsOctets (Octets63)	99
224	JMALLI IDULEVATUEASOCLEUS (OCLEUS63)	99
225	5. APPENDIX A - IMPLEMENTING THE JOB LIFE CYCLE	102
226	6. APPENDIX B - SUPPORT OF JOB SUBMISSION PROTOCOLS	103
227	7. REFERENCES	103
۲ ۲ ۲		TOD
	Bergman, Hastings, Isaacson, LewisInformational [H	Page 6]

	INT	ERNET-DRAFT	Job	Monitoring	MIB,	V1.0	January 2	1998
228	8.	AUTHOR'S ADDRESSES						105
229 230	9.	INDEX						108

231

Job Monitoring MIB

1. Introduction 232

233 This specification defines an official Printer Working Group (PWG) [PWG] standard SNMP MIB for the monitoring of jobs on network printers. 234 235 This specification is being published as an IETF Information Document 236 for the convenience of the Internet community. In consultation with 237 the IETF Application Area Directors, it was concluded properly belongs 238 as an Information document, because this MIB monitors a service node on 239 the network, rather than a network node proper.

240 The Job Monitoring MIB is intended to be implemented by an agent within 241 a printer or the first server closest to the printer, where the printer 242 is either directly connected to the server only or the printer does not 243 contain the job monitoring MIB agent. It is recommended that implementations place the SNMP agent as close as possible to the 244 processing of the print job. This MIB applies to printers with and 245 without spooling capabilities. This MIB is designed to be compatible 246 247 with most current commonly-used job submission protocols. In most 248 environments that support high function job submission/job control 249 protocols, like ISO DPA[iso-dpa], those protocols would be used to 250 monitor and manage print jobs rather than using the Job Monitoring MIB.

251 The Job Monitoring MIB consists of a General Group, a Job Submission ID 252 Group, a Job Group, and an Attribute Group. Each group is a table. 253 All accessible objects are read-only. The General Group contains 254 general information that applies to all jobs in a job set. The Job 255 Submission ID table maps the job submission ID that the client uses to 256 identify a job to the jmJobIndex that the Job Monitoring Agent uses to 257 identify jobs in the Job and Attribute tables. The Job table contains the MANDATORY integer job state and status objects. The Attribute table consists of multiple entries per job that specify (1) job and 258 259 document identification and parameters, (2) requested resources, and (3) consumed resources during and after job processing/printing. A 260 261 262 larger number of job attributes are defined as textual conventions that an agent SHALL return if the server or device implements the 263 264 functionality so represented and the agent has access to the 265 information.

266 1.1 Types of Information in the MIB

267 The job MIB is intended to provide the following information for the indicated Role Models in the Printer MIB[print-mib] (Appendix D - Roles 268 269 of Users).

- 270 User:
- 271 Provide the ability to identify the least busy printer. The user 272 will be able to determine the number and size of jobs waiting for 273 each printer. No attempt is made to actually predict the length 274 of time that jobs will take.
- 275 Provide the ability to identify the current status of the user's 276 job (user queries).
- 277 Provide a timely indication that the job has completed and where 278 it can be found.
- 279 Provide error and diagnostic information for jobs that did not 280 successfully complete.
- 281 Operator:
- Provide a presentation of the state of all the jobs in the print system.
- Provide the ability to identify the user that submitted the print job.
- Provide the ability to identify the resources required by each job.
- 288 Provide the ability to define which physical printers are 289 candidates for the print job.
- Provide some idea of how long each job will take. However, exact
 estimates of time to process a job is not being attempted.
 Instead, objects are included that allow the operator to be able
 to make gross estimates.
- 294 Capacity Planner:
- 295 Provide the ability to determine printer utilization as a 296 function of time.
- 297 Provide the ability to determine how long jobs wait before 298 starting to print.
- 299 Accountant:
- 300 Provide information to allow the creation of a record of 301 resources consumed and printer usage data for charging users or 302 groups for resources consumed.
- 303 Provide information to allow the prediction of consumable usage 304 and resource need.

January 1998

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

311 1.2 Types of Job Monitoring Applications

312 The Job Monitoring MIB is designed for the following types of 313 monitoring applications:

- Monitor a single job starting when the job is submitted and
 ending a defined period after the job completes. The Job
 Submission ID table provides the map to find the specific job
 to be monitored.
- 2. Monitor all 'active' jobs in a queue, which this specification 318 319 generalizes to a "job set". End users may use such a program 320 when selecting a least busy printer, so the MIB is designed for 321 such a program to start up quickly and find the information 322 needed quickly without having to read all (completed) jobs in 323 order to find the active jobs. System operators may also use such a program, in which case it would be running for a long 324 325 period of time and may also be interested in the jobs that have 326 completed. Finally such a program may be used to provide an 327 enhanced console and logging capability.
- 328 3. Collect resource usage for accounting or system utilization 329 purposes that copy the completed job statistics to an accounting system. It is recognized that depending on 330 331 accounting programs to copy MIB data during the job-retention 332 period is somewhat unreliable, since the accounting program may 333 not be running (or may have crashed). Such a program is also expected to keep a shadow copy of the entire Job Attribute 334 335 table including completed, canceled, and aborted jobs which the 336 program updates on each polling cycle. Such a program polls at 337 the rate of the persistence of the Attribute table. The design 338 is not optimized to help such an application determine which 339 jobs are completed, canceled, or aborted. Instead, the 340 application SHALL query each job that the application's shadow 341 copy shows was not complete, canceled, or aborted at the previous poll cycle to see if it is now complete or canceled, 342 343 plus any new jobs that have been submitted.

344 The MIB provides a set of objects that represent a compatible subset of 345 job and document attributes of the ISO DPA standard[iso-dpa] and the 346 Internet Printing Protocol (IPP)[ipp-model], so that coherence is 347 maintained between these two protocols and the information presented to end users and system operators by monitoring applications. However, 348 349 the job monitoring MIB is intended to be used with printers that 350 implement other job submitting and management protocols, such as IEEE 1284.1 (TIPSI)[tipsi], as well as with ones that do implement ISO DPA. 351

352 Thus the job monitoring MIB does not require implementation of either 353 the ISO DPA or IPP protocols.

The MIB is designed so that an additional MIB(s) can be specified in 354 355 the future for monitoring multi-function (scan, FAX, copy) jobs as an 356 augmentation to this MIB.

2. Terminology and Job Model 357

358 This section defines the terms that are used in this specification and the general model for jobs in alphabetical order. 359

360 NOTE - Existing systems use conflicting terms, so these terms are drawn from the ISO 10175 Document Printing Application (DPA) 361 362 standard[iso-dpa]. For example, PostScript systems use the term session for what is called a *job* in this specification and the term 363 364 job to mean what is called a *document* in this specification.

365 Accounting Application: The SNMP management application that copies 366 job information to some more permanent medium so that another 367 application can perform accounting on the data for Accountants, Asset 368 Managers, and Capacity Planners use.

The network entity that accepts SNMP requests from a monitor or 369 Agent: 370 accounting application and provides access to the instrumentation for 371 managing jobs modeled by the management objects defined in the Job 372 Monitoring MIB module for a server or a device.

373 Attribute: A name, value-pair that specifies a job or document 374 instruction, a status, or a condition of a job or a document that has 375 been submitted to a server or device. A particular attribute NEED NOT 376 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 377 378 value, either because (1) the client supplied a value in the job 379 submission protocol, (2) the document data contained an embedded 380 attribute, or (3) the server or device supplied a default value. An agent SHALL represent an attribute as an entry (row) in the Attribute 381 table in this MIB in which entries are present only when necessary. 382 383 Attributes are identified in this MIB by an enum.

384 Client: The network entity that end users use to submit jobs to 385 spoolers, servers, or printers and other devices, depending on the configuration, using any job submission protocol over a serial or 386 387 parallel port to a directly-connected device or over the network to a 388 networked-connected device.

Device: A hardware entity that (1) interfaces to humans, such as a 389 390 device that produces marks on paper or scans marks on paper to produce 391 an electronic representation, (2) accesses digital media, such as CD-392 ROMs, or (3) interfaces electronically to another device, such as sends 393 FAX data to another FAX device.

394 Document: A sub-section within a job that contains print data and 395 *document instructions* that apply to just the document.

396 Document Instruction: An instruction specifying how to process the 397 document. Document instructions MAY be passed in the job submission 398 protocol separate from the actual document data, or MAY be embedded in 399 the document data or a combination, depending on the job submission 400 protocol and implementation.

401 End User: A user that uses a client to submit a print job. See 402 "user".

403 Impression: For a print job, an impression is the passage of the 404 entire side of a sheet by the marker, whether or not any marks are made 405 and independent of the number of passes that the side makes past the 406 marker. Thus a four pass color process counts as a single impression, 407 as does highlight color. Impression counters count all kinds: monochrome, highlight color, and full process color, while full color 408 counters only count full color impressions, and high light color 409 410 counters only count high light color impressions.

411 One-sided processing involves one impression per sheet. Two-sided 412 processing involves two impressions per sheet. If a two-sided document 413 has an odd number of pages, the last sheet still counts as two 414 impressions, if that sheet makes two passes through the marker or the 415 marker marks on both sides of a sheet in a single pass. Two-up 416 printing is the placement of two logical pages on one side of a sheet 417 and so is still a single impression. See "page" and "sheet".

418 NOTE - Since impressions include blank sides, it is suggested that 419 accounting application implementers consider charging for sheets, 420 rather than impressions, possibly using the value of the sides 421 attribute to select different charges for one-sided versus two-sided 422 printing, since some users may think that impressions don't include 423 blank sides.

424 Internal Collation: The production of the sheets for each document copy 425 performed within the printing device by making multiple passes over 426 either the source or an intermediate representation of the document.

427 Job: A unit of work whose results are expected together without 428 interjection of unrelated results. A job contains one or more 429 documents.

430 Job Accounting: The activity of a management application of accessing 431 the MIB and recording what happens to the job during and after the 432 processing of the job.

433 Job Instruction: An instruction specifying how, when, or where the job 434 is to be processed. Job instructions MAY be passed in the job 435 submission protocol or MAY be embedded in the document data or a 436 combination depending on the job submission protocol and

437 implementation.

438 Job Monitoring (using SNMP): The activity of a management application of accessing the MIB and (1) identifying jobs in the job tables being 439 processed by the server, printer or other devices, and (2) displaying 440 441 information to the user about the processing of the job.

442 Job Monitoring Application: The SNMP management application that End 443 Users, and System Operators use to monitor jobs using SNMP. A monitor 444 MAY be either a separate application or MAY be part of the client that 445 also submits jobs. See "monitor".

446 Job Set: A group of jobs that are queued and scheduled together according to a specified scheduling algorithm for a specified device or 447 448 set of devices. For implementations that embed the SNMP agent in the 449 device, the MIB job set normally represents all the jobs known to the 450 device, so that the implementation only implements a single job set. 451 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 452 453 device or (2) set of devices, if the server uses a single queue to load 454 balance between several devices. Each job set is disjoint; no job SHALL be represented in more than one MIB job set. 455

456 Monitor: Short for Job Monitoring Application.

457 Page: A page is a logical division of the original source document. 458 Number up is the imposition of more than one page on a single side of a sheet. See "impression" and "sheet" and "two-up". 459

460 Proxy: An agent that acts as a concentrator for one or more other agents by accepting SNMP operations on the behalf of one or more other 461 462 agents, forwarding them on to those other agents, gathering responses 463 from those other agents and returning them to the original requesting 464 monitor.

465 Queuing: The act of a *device* or *server* of ordering (queuing) the jobs 466 for the purposes of scheduling the jobs to be processed.

467 Printer: A device that puts marks on media.

468 Server: A network entity that accepts jobs from clients and in turn 469 submits the jobs to *printers* and other *devices* that may be directly 470 connected to the server via a serial or parallel port or may be on the 471 network. A server MAY be a printer supervisor control program, or a 472 print spooler.

473 Sheet: A sheet is a single instance of a medium, whether printing on 474 one or both sides of the medium. See "impression" and "page".

475 SNMP Information Object: A name, value-pair that specifies an action, a status, or a condition in an SNMP MIB. Objects are identified in 476 477 SNMP by an OBJECT IDENTIFIER.

Spooler: A server that accepts jobs, spools the data, and decides when 478 479 and on which printer to print the job. A spooler is a client to a printer or a printer supervisor, depending on implementation. 480

481 Spooling: The act of a *device* or *server* of (1) accepting jobs and (2) writing the job's attributes and document data on to secondary storage. 482

483 Stacked: When a media sheet is placed in an output bin of a device.

484 Supervisor: A server that contains a control program that controls a 485 printer or other device. A supervisor is a client to the printer or 486 other device.

487 System Operator: A user that uses a monitor to monitor the system and 488 carries out tasks to keep the system running.

489 System Administrator: A user that specifies policy for the system.

490 Two-up: The placement of two pages on one side of a sheet so that each 491 side or impressions counts as two pages. See "page" and "sheet".

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

493 2.1 System Configurations for the Job Monitoring MIB

494 This section enumerates the three configurations in which the Job 495 Monitoring MIB is intended to be used. To simplify the pictures, the devices are shown as printers. See section 1.1 entitled "Types of 496 497 Information in the MIB".

498 The diagram in the Printer MIB[print-mib] entitled: "One Printer's View 499 of the Network" is assumed for this MIB as well. Please refer to that diagram to aid in understanding the following system configurations. 500

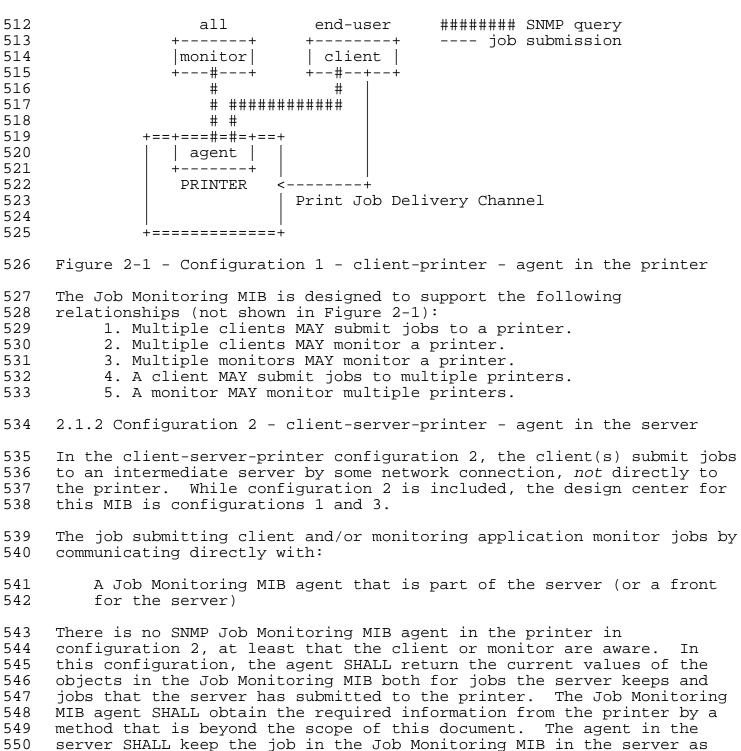
501 2.1.1 Configuration 1 - client-printer

In the client-printer configuration 1, the client(s) submit jobs 502 503 directly to the printer, either by some direct connect, or by network 504 connection.

505 The job submitting client and/or monitoring application monitor jobs by 506 communicating directly with an agent that is part of the printer. The 507 agent in the printer SHALL keep the job in the Job Monitoring MIB as long as the job is in the printer, plus a defined time period after the 508 job enters the completed state in which accounting programs can copy 509 510 out the accounting data from the Job Monitoring MIB.

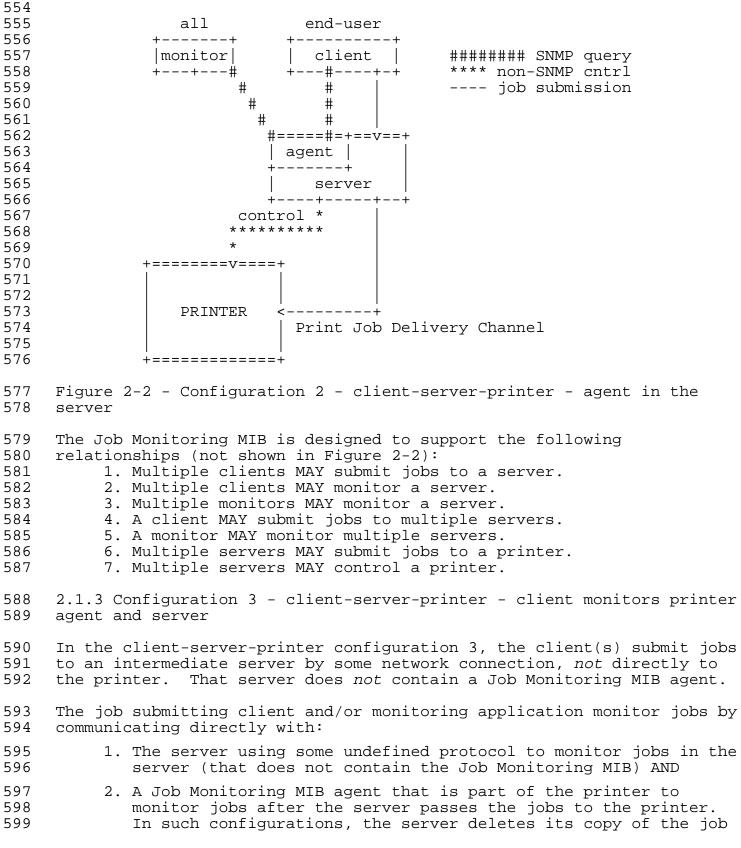
511

551



552 job enters the completed state in which accounting programs can copy out the accounting data from the Job Monitoring MIB. 553

long as the job is in the printer, plus a defined time period after the



January 1998

600 601 602

614

from the server after submitting the job to the printer usually almost immediately (before the job does much processing, if any).

603 In configuration 3, the agent (in the printer) SHALL keep the values of 604 the objects in the Job Monitoring MIB that the agent implements updated 605 for a job that the server has submitted to the printer. The agent SHALL obtain information about the jobs submitted to the printer from 606 607 the server (either in the job submission protocol, in the document 608 data, or by direct query of the server), in order to populate some of 609 the objects the Job Monitoring MIB in the printer. The agent in the 610 printer SHALL keep the job in the Job Monitoring MIB as long as the job 611 is in the Printer, and longer in order to implement the completed state 612 in which monitoring programs can copy out the accounting data from the 613 Job Monitoring MIB.

0		
615	all end-user	
616	++ ++	
617	monitor client ######## SNMP query	
618	++ +* ++-+ **** non-SNMP query	
619	# * * job submission	
620	# * *	
621	# * *	
622	# *====v====v===+	
623	#	
624	# server	
625	#	
626	# ++	
627	# optional#	
628	# #########	
629	# #	
630	+==+=v=+==+	
631	agent	
632	++	
633	PRINTER <+	
634	Print Job Delivery Channel	
635		
636	+========+	
637	Figure 2-3 - Configuration 3 - client-server-printer - client	m

Figure 2-3 - Configuration 3 - client-server-printer - client monitors 637 638 printer agent and server

639 The Job Monitoring MIB is designed to support the following 640 relationships (not shown in Figure 2-3): 641

- 1. Multiple clients MAY submit jobs to a server. 642
 - 2. Multiple clients MAY monitor a server.
- 643 3. Multiple monitors MAY monitor a server. 644
 - 4. A client MAY submit jobs to multiple servers.
- 5. A monitor MAY monitor multiple servers. 645
- 646 6. Multiple servers MAY submit jobs to a printer.
- 647 7. Multiple servers MAY control a printer.

- 648 3. Managed Object Usage
- 649 This section describes the usage of the objects in the MIB.

3.1 Conformance Considerations 650

651 In order to achieve interoperability between job monitoring 652 applications and job monitoring agents, this specification includes the

- conformance requirements for both monitoring applications and agents. 653
- 654 3.1.1 Conformance Terminology
- This specification uses the verbs: "SHALL", "SHOULD", "MAY", and "NEED 655 656 NOT" to specify conformance requirements according to RFC 2119 [req-657 words] as follows:
- 658 "SHALL": indicates an action that the subject of the sentence must 659 implement in order to claim conformance to this specification
- 660 "MAY": indicates an action that the subject of the sentence does not 661 have to implement in order to claim conformance to this 662 specification, in other words that action is an implementation option
- 663 "NEED NOT": indicates an action that the subject of the sentence 664 does not have to implement in order to claim conformance to this 665 specification. The verb "NEED NOT" is used instead of "may not", since "may not" sounds like a prohibition. 666
- 667 "SHOULD": indicates an action that is recommended for the subject of the sentence to implement, but is not required, in order to claim 668 669 conformance to this specification.
- 670 3.1.2 Agent Conformance Requirements
- 671 A conforming agent:
- 672 1. SHALL implement all MANDATORY groups in this specification.
- 2. SHALL implement any attributes if (1) the server or device 673 674 supports the functionality represented by the attribute and (2) 675 the information is available to the agent.
- 676 3. SHOULD implement both forms of an attribute if it implements an attribute that permits a choice of INTEGER and OCTET STRING 677 forms, since implementing both forms may help management 678 679 applications by giving them a choice of representations, since 680 the representation are equivalent. See the JmAttributeTypeTC textual-convention. 681
- 682 NOTE - This MIB, like the Printer MIB, is written following the subset 683 of SMIv2 that can be supported by SMIv1 and SNMPv1 implementations.

684 3.1.2.1 MIB II System Group objects

685 The Job Monitoring MIB agent SHALL implement all objects in the System Group of MIB-II[mib-II], whether the Printer MIB[print-mib] is 686 687 implemented or not.

3.1.2.2 MIB II Interface Group objects 688

689 The Job Monitoring MIB agent SHALL implement all objects in the Interfaces Group of MIB-II[mib-II], whether the Printer MIB[print-mib] 690 691 is implemented or not.

692 3.1.2.3 Printer MIB objects

693 If the agent is providing access to a device that is a printer, the 694 agent SHALL implement all of the MANDATORY objects in the Printer 695 MIB[print-mib] and all the objects in other MIBs that conformance to 696 the Printer MIB requires, such as the Host Resources MIB[hr-mib]. Τf the agent is providing access to a server that controls one or more 697 698 direct-connect or networked printers, the agent NEED NOT implement the 699 Printer MIB and NEED NOT implement the Host Resources MIB.

- 700 3.1.3 Job Monitoring Application Conformance Requirements
- 701 A conforming job monitoring application:
- 702 1. SHALL accept the full syntactic range for all objects in all 703 MANDATORY groups and all MANDATORY attributes that are required 704 to be implemented by an agent according to Section 3.1.2 and 705 SHALL either present them to the user or ignore them.
- 706 2. SHALL accept the full syntactic range for all attributes, 707 including enum and bit values specified in this specification 708 and additional ones that may be registered with the PWG and 709 SHALL either present them to the user or ignore them. Τn particular, a conforming job monitoring application SHALL not 710 711 malfunction when receiving any standard or registered enum or bit values. See Section 3.7 entitled "IANA and PWG 712 Registration Considerations". 713
- 714 3. SHALL NOT fail when operating with agents that materialize 715 attributes after the job has been submitted, as opposed to when 716 the job is submitted.
- 717 4. SHALL, if it supports a time attribute, accept either form of 718 the time attribute, since agents are free to implement either 719 time form.
- 720 3.2 The Job Tables and the Oldest Active and Newest Active Indexes

721 The jmJobTable and jmAttributeTable contain objects and attributes, 722 respectively, for each job in a job set. These first two indexes are: 723 1. jmGeneralJobSetIndex - which job set 724 2. jmJobIndex - which job in the job set

725 In order for a monitoring application to quickly find that active jobs 726 (jobs in the pending, processing, or processingStopped states), the MIB 727 contains two indexes:

- 728 1. jmGeneralOldestActiveJobIndex the index of the active job 729 that has been in the tables the longest.
- 7302. jmGeneralNewestActiveJobIndex the index of the active job731that has been most recently added to the tables.

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 agent is providing access. If the incremented value of jmJobIndex would exceed the implementation-defined maximum value for jmJobIndex, the agent SHALL 'wrap' back to 1. An agent uses the resulting value of jmJobIndex for storing information in the jmJobTable and the jmAttributeTable about the job.

739 It is recommended that the largest value for jmJobIndex be much larger 740 than the maximum number of jobs that the implementation can contain at 741 a single time, so as to minimize the premature re-use of a jmJobIndex 742 value for a newer job while clients retain the same 'stale' value for 743 an older job.

744 It is recommended that agents that are providing access to 745 servers/devices that already allocate job-identifiers for jobs as 746 integers use the same integer value for the jmJobIndex. Then 747 management applications using this MIB and applications using other protocols will see the same job identifiers for the same jobs. 748 Agents providing access to systems that contain jobs with a job identifier of 749 0 SHALL map the job identifier value 0 to a jmJobIndex value that is 750 751 one higher than the highest job identifier value that any job can have 752 on that system. Then only job 0 will have a different job-identifier 753 value than the job's jmJobIndex value.

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

Fach time a new job is accepted by the server or device that the agent is providing access to AND that job is to be 'active' (pending, processing, or processingStopped, but not pendingHeld), the agent SHALL copy the value of the job's jmJobIndex to the jmGeneralNewestActiveJobIndex object. If the new job is to be 'inactive' (pendingHeld state), the agent SHALL not change the value of jmGeneralNewestActiveJobIndex object (though the agent SHALL assign the

767 next incremental jmJobIndex value to the job).

January 1998

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

Whenever a job transitions from one of the 'inactive' job states to one of the 'active' job states (from pendingHeld to pending or processing), the agent SHALL update the value of either the jmGeneralOldestActiveJobIndex or the jmGeneralNewestActiveJobIndex objects, or both, if the job's jmJobIndex value is outside the range between jmGeneralOldestActiveJobIndex and

- 781 jmGeneralNewestActiveJobIndex.
- 782 When all jobs become 'inactive', i.e., enter the pendingHeld,
- 783 completed, canceled, or aborted states, the agent SHALL set the value
- 784 of both the jmGeneralOldestActiveJobIndex and

785 jmGeneralNewestActiveJobIndex objects to 0.

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

791 If an application detects that the jmGeneralNewestActiveJobIndex is 792 smaller than jmGeneralOldestActiveJobIndex, the job index has wrapped. 793 In this case, the application SHALL reset the index to 1 when the end 794 of the table is reached and continue the GetNext operations to find the 795 rest of the active jobs.

796 NOTE - Applications detect the end of the jmAttributeTable table when 797 the OID returned by the GetNext operation is an OID in a different MIB. 798 There is no object in this MIB that specifies the maximum value for the 799 jmJobIndex supported by the implementation.

800 When the server or device is power-cycled, the agent SHALL remember the 801 next jmJobIndex value to be assigned, so that new jobs are not assigned 802 the same jmJobIndex as recent jobs before the power cycle.

803 3.3 The Attribute Mechanism

Attributes are similar to information objects, except that attributes 804 805 are identified by an enum, instead of an OID, so that attributes may be 806 registered without requiring a new MIB. Also an implementation that does not have the functionality represented by the attribute can omit 807 the attribute entirely, rather than having to return a distinguished 808 809 value. The agent is free to materialize an attribute in the 810 jmAttributeTable as soon as the agent is aware of the value of the 811 attribute.

- 812 The agent materializes job attributes in a four-indexed 813 jmAttributeTable:
- 814 1. jmGeneralJobSetIndex - which job set
- 2. jmJobIndex which job in the job set 815
- 816 3. jmAttributeTypeIndex - which attribute
- 817 4. jmAttributeInstanceIndex - which attribute instance for those 818 attributes that can have multiple values per job.

819 Some attributes represent information about a job, such as a file-name, 820 a document-name, a submission-time or a completion time. Other 821 attributes represent resources required, e.g., a medium or a colorant, 822 etc. to process the job before the job starts processing OR to indicate 823 the amount of the resource consumed during and after processing, e.g., 824 pages completed or impressions completed. If both a required and a 825 consumed value of a resource is needed, this specification assigns two separate attribute enums in the textual convention. 826

827 NOTE - The table of contents lists all the attributes in order. This 828 order is the order of enum assignments which is the order that the SNMP 829 GetNext operation returns attributes. Most attributes apply to all three configurations covered by this MIB specification (see section 2.1 830 entitled "System Configurations for the Job Monitoring MIB"). Those 831 832 attributes that apply to a particular configuration are indicated as 'Configuration n:' and SHALL NOT be used with other configurations. 833

834 3.3.1 Conformance of Attribute Implementation

An agent SHALL implement any attribute if (1) the server or device 835 836 supports the functionality represented by the attribute and (2) the 837 information is available to the agent. The agent MAY create the attribute row in the jmAttributeTable when the information is available 838 839 or MAY create the row earlier with the designated 'unknown' value 840 appropriate for that attribute. See next section.

If the server or device does not implement or does not provide access 841 842 to the information about an attribute, the agent SHOULD NOT create the 843 corresponding row in the jmAttributeTable.

3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes 844

845 Some attributes have a 'useful' Integer32 value, some have a 'useful' OCTET STRING value, some MAY have either or both depending on 846 implementation, and some MUST have both. See the JmAttributeTypeTC 847 848 textual convention for the specification of each attribute.

849 SNMP requires that if an object cannot be implemented because its 850 values cannot be accessed, then a compliant agent SHALL return an SNMP 851 error in SNMPv1 or an exception value in SNMPv2. However, this MIB has 852 been designed so that 'all' objects can and SHALL be implemented by an 853 agent, so that neither the SNMPv1 error nor the SNMPv2 exception value

854 SHALL be generated by the agent. This MIB has also been designed so that when an agent materializes an attribute, the agent SHALL 855 856 materialize a row consisting of both the jmAttributeValueAsInteger and 857 jmAttributeValueAsOctets objects.

858 In general, values for objects and attributes have been chosen so that 859 a management application will be able to determine whether a 'useful', 860 'unknown', or 'other' value is available. When a useful value is not 861 available for an object that agent SHALL return a zero-length string 862 for octet strings, the value 'unknown(2)' for enums, a '0' value for an 863 object that represents an index in another table, and a value '-2' for 864 counting integers.

865 Since each attribute is represented by a row consisting of both the 866 jmAttributeValueAsInteger and jmAttributeValueAsOctets MANDATORY objects, SNMP requires that the agent SHALL always create an attribute 867 868 row with both objects specified. However, for most attributes the 869 agent SHALL return a "useful" value for one of the objects and SHALL return the 'other' value for the other object. For integer only 870 871 attributes, the agent SHALL always return a zero-length string value 872 for the jmAttributeValueAsOctets object. For octet string only 873 attributes, the agent SHALL always return a '-1' value for the 874 jmAttributeValueAsInteger object.

875 3.3.3 Data Sub-types and Attribute Naming Conventions

876 Many attributes are sub-typed to give a more specific data type than 877 Integer32 or OCTET STRING. The data sub-type of each attribute is indicated on the first line(s) of the description. Some attributes 878 879 have several different data sub-type representations. When an 880 attribute has both an Integer32 data sub-type and an OCTET STRING data 881 sub-type, the attribute can be represented in a single row in the 882 jmAttributeTable. In this case, the data sub-type name is not included as the last part of the name of the attribute, e.g., documentFormat(38) 883 884 which is both an enum and/or a name. When the data sub-types cannot be 885 represented by a single row in the jmAttributeTable, each such representation is considered a separate attribute and is assigned a 886 887 separate name and enum value. For these attributes, the name of the 888 data sub-type is the last part of the name of the attribute: Name, 889 Index, DateAndTime, TimeStamp, etc. For example,

documentFormatIndex(37) is an index. 890

891 NOTE: The Table of Contents also lists the data sub-type and/or data sub-types of each attribute, using the textual-convention name when 892 893 such is defined. The following abbreviations are used in the Table of 894 Contents as shown:

895

'Int32(-2)'	Integer32(-22147483647)
'Int32(0)'	Integer32(02147483647)
'Int32(1)'	Integer32(12147483647)
'Int32(mn)'	For all other Integer ranges, the lower
	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.

896 3.3.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes

897 Most attributes SHALL have only one row per job. However, a few 898 attributes can have multiple values per job or even per document, where 899 each value is a separate row in the jmAttributeTable. Unless indicated with 'MULTI-ROW:' in the JmAttributeTypeTC description, an agent SHALL 900 901 ensure that each attribute occurs only once in the jmAttributeTable for 902 a job. Most of the 'MULTI-ROW' attributes do not allow duplicate 903 values, i.e., the agent SHALL ensure that each value occurs only once 904 for a job. Only if the specification of the 'MULTI-ROW' attribute also 905 says "the values NEED NOT be unique" can the agent allow duplicate values to occur for the job. 906

907 NOTE - Duplicates are allowed for 'extensive' 'MULTI-ROW' attributes, 908 such as fileName(34) or documentName(35) which are specified to be 'per-document' attributes, but are not allowed for 'intensive' 'MULTI-909 ROW' attributes, such as mediumConsumed(171) and documentFormat(38) 910 911 which are specified to be 'per-job' attributes.

912 3.3.5 Requested Objects and Attributes

A number of objects and attributes record requirements for the job. 913 Such object and attribute names end with the word 'Requested'. In the 914 interests of brevity, the phrase 'requested' SHALL mean: (1) requested 915 by the client (or intervening server) in the job submission protocol 916 917 and MAY also mean (2) embedded in the submitted document data, and/or 918 (3) defaulted by the recipient device or server with the same semantics as if the requester had supplied, depending on implementation. Also if 919 920 a value is supplied by the job submission client, and the server/device 921 determines a better value, through processing or other means, the agent 922 MAY return that better value for such object and attribute.

923 3.3.6 Consumption Attributes

924 A number of objects and attributes record consumption. Such attribute names end with the word 'Completed' or 'Consumed'. If the job has not 925 926 yet consumed what that resource is metering, the agent either: (1) 927 SHALL return the value 0 or (2) SHALL not add this attribute to the jmAttributeTable until the consumption begins. In the interests of 928 brevity, the semantics for 0 is specified once here and is not repeated 929 for each consumption attribute specification and a DEFVAL of 0 is 930 931 indicated.

932 3.3.7 Index Value Attributes

A number of attributes are indexes in other tables. Such attribute 933 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 934 935 SHALL either: (1) return the value 0 or (2) not add this attribute to 936 the jmAttributeTable until the index value is assigned. In the 937 interests of brevity, the semantics for 0 is specified once here and is 938 939 not repeated for each index attribute specification and a DEFVAL of 0 940 is indicated.

941 3.4 Monitoring Job Progress

942 There are a number of objects and attributes for monitoring the progress of a job. These objects and attributes count the number of K octets, impressions, sheets, and pages requested or completed. For 943 944 945 impressions and sheets, "completed" SHALL mean stacked, unless the implementation is unable to detect when each sheet is stacked, in which 946 947 case stacked is approximated when processing of each sheet completes. 948 There are objects and attributes for the overall job and for the 949 current copy of the document currently being stacked. For the latter, 950 the rate at which the various objects and attributes count depends on 951 the sheet and document collation of the job.

952 Job Collation included sheet collation and document collation. Sheet 953 collation is defined to be the ordering of sheets within a document 954 copy. Document collation is defined to be ordering of document copies 955 within a multi-document job. There are three types of job collation 956 (see terminology definitions in Section 2):

957 1. Uncollated Sheets - No collation of the sheets within each 958 document copy, i.e., each sheet of a document that is to 959 produce multiple copies is replicated before the next sheet in 960 the document is processed and stacked. If the device has an 961 output bin collator, uncollated sheets may actually produce 962 collated sheets as far as the user is concerned (in the output bins). However, when the job collation is 'uncollated sheets', job progress is indistinguishable to 963 job progress is indistinguishable to a monitoring application 964 965 between a device that has an output bin collator and one that 966 does not.

Job Monitoring MIB, V1.0 January 1998

- 967 2. Collated Documents - Collation of the sheets within each 968 document copy is performed within the printing device by making 969 multiple passes over either the source or an intermediate 970 representation of the document. In addition, when there are 971 multiple documents per job, the i'th copy of each document is 972 stacked before the j'th copy of each document, i.e., the documents are collated within each job copy. For example, if a 973 job is submitted with documents, A and B, the job is made 974 975 available to the end user as: A, B, A, B, Collated Document correspond to the IPP [ipp-model] 'separate-documents-collated-976 977 copies' value of the "multiple-document-handling" attribute. 978
- 979 If jobCopiesRequested or documentCopiesRequested = 1, then 980 jobCollationType is defined as 4.
- 981 3. Uncollated Documents - Collation of the sheets within each document copy is performed within the printing device by making 982 multiple passes over either the source or an intermediate 983 representation of the document. In addition, when there are 984 985 multiple documents per job, all copies of the first document in 986 the job are stacked before the any copied of the next document 987 in the job, i.e., the documents are uncollated within the job. For example, if a job is submitted with documents, A and B, the 988 989 job is mad available to the end user as: A, A, ..., B, B, 990 Uncollated Documents correspond to the IPP [ipp-model] 991 'separate-documents-uncollated-copies' value of the "multipledocument-handling" attribute. 992
- 993 Consider the following four variables that are used to monitor the 994 progress of a job's impressions:
- 995 1. jmJobImpressionsCompleted - counts the total number of 996 impressions stacked for the job
- 997 2. impressionsCompletedCurrentCopy - counts the number of 998 impressions stacked for the current document copy
- 3. sheetCompletedCopyNumber identifies the number of the copy 999 1000 for the current document being stacked where the first copy is 1001 1.
- 1002 4. sheetCompletedDocumentNumber - identifies the current document 1003 within the job that is being stacked where the first document in a job is 1. NOTE: this attribute SHOULD NOT be implemented 1004 1005 for implementations that only support one document per job.

1006 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, 1007 the four variables have the following values as each sheet is stacked 1008 1009 for one-sided printing:

1010

Job Collation Type = Uncollated Sheets

1	\cap	1	1
ㅗ	υ	ㅗ	т.

jmJobImpressions Completed	Impressions CompletedCurrent Copy	sheetCompleted CopyNumber	sheetCompleted DocumentNumber
0	0	0	0
1	1	1	1
2	1	2	1
3	1	3	1
4	2	1	1
5	2	2	1
6	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

1012

1013

Job Collation Type	= Collated Documents
--------------------	----------------------

1014

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
б	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

1015

Job Collation Type = Uncollated Documents

1	0	1	6

1017					
TOT 1	jmJobImpressions Completed	Impressions CompletedCurrent Copy	sheetCompleted CopyNumber	sheetCompleted DocumentNumber	
	0 1 2 3 4 5 6 7 8 9 10 11	0 1 2 3 1 2 3 1 2 3 1 2 3 1 2	0 1 1 2 2 2 3 3 3 3 1 1	0 1 1 1 1 1 1 1 2 2	
	12 13	3 1	1 2	2 2	
	14 15	2 3	2 2	2 2	
	16 17	1 2	3	2	
	18	3	3	2	

1018

1019 3.5 Job Identification

1020 There are a number of attributes that permit a user, operator or system administrator to identify jobs of interest, such as jobURI, jobName, jobOriginatingHost, etc. In addition, there is a jmJobSubmissionID 1021 1022 1023 object that is a text string table index. Being a table index allows a 1024 monitoring application to quickly locate and identify a particular job 1025 of interest that was submitted from a particular client by the user invoking the monitoring application without having to scan the entire 1026 job table. The Job Monitoring MIB needs to provide for identification 1027 1028 of the job at both sides of the job submission process. The primary 1029 identification point is the client side. The jmJobSubmissionID allows the monitoring application to identify the job of interest from all the 1030 1031 jobs currently "known" by the server or device. The value of 1032 jmJobSubmissionID can be assigned by either the client's local system or a downstream server or device. The point of assignment depends on 1033 1034 the job submission protocol in use.

1035 The server/device-side identifier, called the jmJobIndex object, SHALL be assigned by the SNMP Job Monitoring MIB agent when the server or 1036 device accepts the jobs from submitting clients. The jmJobIndex object 1037 1038 allows the interested party to obtain all objects desired that relate 1039 to a particular job. See Section 3.2, entitled 'The Job Tables and the

Bergman, Hastings, Isaacson, LewisInformational

[Page 29]

1040 Oldest Active and Newest Active Indexes' for the specification of how 1041 the agent SHALL assign the jmJobIndex values.

1042 The MIB provides a mapping table that maps each jmJobSubmissionID value 1043 to a corresponding jmJobIndex value generated by the agent, so that an 1044 application can determine the correct value for the jmJobIndex value 1045 for the job of interest in a single Get operation, given the Job 1046 Submission ID. See the jmJobIDGroup.

1047 In some configurations there may be more than one application program 1048 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 1049 multiple job submission IDs, each entity MAY supply an appropriate 1050 1051 jmJobSubmissionID value. In this case there would be a separate entry in the jmJobSubmissionID table, one for each jmJobSubmissionID. All 1052 1053 entries would map to the same jmJobIndex that contains the job data. 1054 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. 1055

1056 The jobName attribute provides a name that the user supplies as a job 1057 attribute with the job. The jobName attribute is not necessarily 1058 unique, even for one user, let alone across users.

- 1059 3.6 Internationalization Considerations
- 1060 This section describes the internationalization considerations included 1061 in this MIB.

1062 3.6.1 Text generated by the server or device

1063 There are a few objects and attributes generated by the server or 1064 device that SHALL be represented using the Universal Multiple-Octet 1065 Coded Character Set (UCS) [ISO-10646]. These objects and attributes 1066 are always supplied (if implemented) by the agent, not by the job 1067 submitting client:

1068

1069

- 1. jmGeneralJobSetName object
- processingMessage(6) attribute
- 1070 3. physicalDevice(32) (name value) attribute

1071 The character encoding scheme for representing these objects and 1072 attributes SHALL be UTF-8 as recommended by RFC 2130 [RFC 2130] and the 1073 "IETF Policy on Character Sets and Language" [char-set policy]. The 1074 'JmUTF8StringTC' textual convention is used to indicate UTF-8 text 1075 strings.

1076 NOTE - For strings in 7-bit US-ASCII, there is no impact since the UTF-1077 8 representation of 7-bit ASCII is identical to the US-ASCII [US-ASCII] 1078 encoding.

1079 The text contained in the processingMessage(6) attribute is generated 1080 by the server/device. The natural language for the 1081 processingMessage(6) attribute is identified by the

1082 processingMessageNaturalLangTag(7) attribute. The 1083

processingMessageNaturalLangTag(7) attribute uses the 1084

JmNaturalLanguageTagTC textual convention which SHALL conform to the language tag mechanism specified in RFC 1766 [RFC-1766]. The 1085

1086 JmNaturalLanguageTagTC value is the same as the IPP [IPP-model]

1087 'naturalLanguage' attribute syntax. RFC 1766 specifies that a US-ASCII 1088 string consisting of the natural language followed by an optional

country field. Both fields use the same two-character codes from ISO 1089

1090 639 [ISO-639] and ISO 3166 [ISO-3166], respectively, that are used in 1091 the Printer MIB for identifying language and country.

1092 Examples of the values of the processingMessageNaturalLangTag(7) 1093 attribute include: 1. 'en'

- 1094 for English
- 2. 'en-us' for US English 1095
- 1096 3. 'fr' for French
- 1097 4. 'de' for German

3.6.2 Text supplied by the job submitter 1098

1099 All of the objects and attributes represented by the 'JmJobStringTC' textual-convention are either (1) supplied in the job submission 1100 1101 protocol by the client that submits the job to the server or device or 1102 (2) are defaulted by the server or device if the job submitting client 1103 does not supply values. The agent SHALL represent these objects and 1104 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 1105 1106 coded character set or encoding scheme. In any case, the resulting 1107 coded character set representation SHOULD be UTF-8 [UTF-8], but SHALL be one in which the code positions from 0 to 31 SHALL not be used, 321108 1109 to 127 SHALL be US-ASCII [US-ASCII], 127 SHALL be unused, and the 1110 remaining code positions 128 to 255 SHALL represent single-byte or 1111 multi-byte graphic characters structured according to ISO 2022 [ISO 1112 2022] or SHALL be unused.

1113 The coded character set SHALL be one of the ones registered with IANA [IANA] and SHALL be identified by the jobCodedCharSet attribute in the 1114 jmJobAttributeTable for the job. If the agent does not know what coded 1115 1116 character set was used by the job submitting client, the agent SHALL 1117 either (1) return the 'unknown(2)' value for the jobCodedCharSet 1118 attribute or (2) not return the jobCodedCharSet attribute for the job.

1119 Examples of coded character sets which meet this criteria for use as 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, 1120 1121 1122 IBM Code Page 850, Windows Default 8-bit set, UTF-8 [UTF-8], US-ASCII 1123 plus JIS X0208-1990 Japanese [JIS X0208], US-ASCII plus GB2312-1980 PRC 1124 Chinese [GB2312]. See the IANA registry of coded character sets [IANA 1125 charsets].

1126 Examples of coded character sets which do not meet this criteria are: 1127 national 7-bit sets conforming to ISO 646 (except US-ASCII), EBCDIC,

1128 and ISO 10646 (Unicode) [ISO-10646]. In order to represent Unicode characters, the UTF-8 [UTF-8] encoding scheme SHALL be used which has 1129 1130 been assigned the MIBenum value of '106' by IANA.

1131 The jobCodedCharSet attribute uses the imported 'CodedCharSet' textual-1132 convention from the Printer MIB [printmib].

1133 The natural language for attributes represented by the textualconvention JmJobStringTC SHALL be identified either (1) by the 1134 jobNaturalLanguageTag(9) attribute or SHALL be keywords in US-English 1135 1136 (as in IPP). A monitoring application SHOULD attempt to localize keywords into the language of the user by means of some lookup 1137 1138 mechanism. If the keyword value is not known to the monitoring 1139 application, the monitoring application SHOULD assume that the value is 1140 in the natural language specified by the job's jobNaturalLanguageTag(9) 1141 attribute and SHOULD present the value to its user as is. The 1142 jobNaturalLanguageTag(9) attribute value SHALL have the same syntax and 1143 semantics as the processingMessageNaturalLangTag(7) attribute, except that the jobNaturalLanguageTag(9) attribute identifies the natural 1144 1145 language of attributes supplied by the job submitter instead of the 1146 natural language of the processingMessage(6) attribute. See Section 1147 3.6.1.

3.6.3 'DateAndTime' for representing the date and time 1148

1149 This MIB also contains objects that are represented using the 1150 DateAndTime textual convention from SMIv2 [SMIv2-TC]. The job 1151 management application SHALL display such objects in the locale of the 1152 user running the monitoring application.

1153 3.7 IANA and PWG Registration Considerations

1154 This MIB does not require any additional registration schemes for IANA, 1155 but does depend on registration schemes that other Internet standards 1156 track specifications have set up. The names of these IANA registration 1157 assignments under the /in-notes/iana/assignments/ path:

- 1158 1. printer-language-numbers - used as enums in the documentFormat(38) 1159 attribute
- 1160 2. media-types - uses as keywords in the documentFormat(38) attribute
- 1161 3. character-sets - used as enums in the jobCodedCharSet(8) attribute

1162 During the development of this standard, the Printer Working Group 1163 (PWG) will register additional enums while the standard is in the 1164 proposed and draft states according to the procedures described in this The PWG will handle registration of additional enums after 1165 section. 1166 approving this standard, according to the procedures described in this 1167 section:

1168 3.7.1 PWG Registration of enums

1169 This specification uses textual conventions to define enumerated values 1170 (enums) and bit values. Enumerations (enums) and bit values are sets 1171 of symbolic values defined for use with one or more objects or 1172 attributes. All enumeration sets and bit value sets are assigned a symbolic data type name (textual convention). As a convention the 1173 symbolic name ends in "TC" for textual convention. These enumerations 1174 1175 are defined at the beginning of the MIB module specification.

1176 The PWG has defined several type of enumerations for use in the Job Monitoring MIB and the Printer MIB[print-mib]. These types differ in 1177 the method employed to control the addition of new enumerations. 1178 1179 Throughout this document, references to "type n enum", where n can be 1180 1, 2 or 3 can be found in the various tables. The definitions of these 1181 types of enumerations are:

1182 3.7.1.1 Type 1 enumerations

1183 Type 1 enumeration: All the values are defined in the Job Monitoring 1184 MIB specification (RFC for the Job Monitoring MIB). Additional 1185 enumerated values require a new RFC.

- 1186 There are no type 1 enums in the current draft.
- 1187 3.7.1.2 Type 2 enumerations

Type 2 enumeration: An initial set of values are defined in the Job 1188 Monitoring MIB specification. Additional enumerated values are 1189 1190 registered with the PWG.

- 1191 The following type 2 enums are contained in the current draft :
- 1192 1. JmUTF8StringTC
- 1193 2. JmJobStringTC

1194

1196

1199

- 3. JmNaturalLanguageTagTC
- 1195 4. JmTimeStampTC
 - 5. JmFinishingTC [same enum values as IPP "finishing" attribute]
- 6. JmPrintQualityTC [same enum values as IPP "print-quality" 1197 1198 attributel
 - 7. JmTonerEconomyTC
- 8. JmMediumTypeTC 1200
- 9. JmJobSubmissionIDTypeTC 1201
- 1202 10.JmJobCollationTypeTC
- 11.JmJobStateTC [same enum values as IPP "job-state" attribute] 1203
- 1204 12.JmAttributeTypeTC

1205 For those textual conventions that have the same enum values as the 1206 indicated IPP Job attribute SHALL be simultaneously registered by the PWG for use with IPP [ipp-model] and the Job Monitoring MIB. 1207

1208 3.7.1.3 Type 3 enumeration

1209 Type 3 enumeration: An initial set of values are defined in the Job 1210 Monitoring MIB specification. Additional enumerated values are 1211 registered through the PWG without PWG review.

- 1212 There are no type 3 enums in the current draft.
- 1213 3.7.2 PWG Registration of type 2 bit values

1214 This draft contains the following type 2 bit value textual-conventions: 1215 1. JmJobServiceTypesTC

- 2. JmJobStateReasons1TC 1216
- 1217 3. JmJobStateReasons2TC
- 4. JmJobStateReasons3TC 1218
- 1219 5. JmJobStateReasons4TC

1220 These textual-conventions are defined as bits in an Integer so that they can be used with SNMPv1 SMI. The jobStateReasonsN (N=1..4) 1221 1222 attributes are defined as bit values using the corresponding 1223 JmJobStateReasonsNTC textual-conventions.

The registration of JmJobServiceTypesTC and JmJobStateReasonsNTC bit 1224 1225 values SHALL follow the procedures for a type 2 enum as specified in 1226 Section 3.7.1.2.

1227 3.7.3 PWG Registration of Job Submission Id Formats

1228 In addition to enums and bit values, this specification assigns a single ASCII digit or letter to various job submission ID formats. 1229 See 1230 the JmJobSubmissionIDTypeTC textual-convention and the object. The 1231 registration of JobSubmissionID format numbers SHALL follow the 1232 procedures for a type 2 enum as specified in Section 3.7.1.2.

1233 3.7.4 PWG Registration of MIME types/sub-types for document-formats

1234 The documentFormat(38) attribute has MIME type/sub-type values for indicating document formats which IANA registers as "media type" names. 1235 1236 The values of the documentFormat(38) attribute are the same as the 1237 corresponding Internet Printing Protocol (IPP) "document-format" Job 1238 attribute values [ipp-model].

- 1239 3.8 Security Considerations
- 1240 3.8.1 Read-Write objects

1241 All objects are read-only, greatly simplifying the security 1242 considerations. If another MIB augments this MIB, that MIB might accept SNMP Write operations to objects in that MIB whose effect is to 1243 1244 modify the values of read-only objects in this MIB. However, that MIB 1245 SHALL have to support the required access control in order to achieve 1246 security, not this MIB.

Bergman, Hastings, Isaacson, LewisInformational

[Page 34]

1247 3.8.2 Read-Only Objects In Other User's Jobs

1248 The security policy of some sites MAY be that unprivileged users can 1249 only get the objects from jobs that they submitted, plus a few minimal objects from other jobs, such as the jmJobKOctetsPerCopyRequested and 1250 1251 jmJobKOctetsProcessed objects, so that a user can tell how busy a printer is. Other sites MAY allow all unprivileged users to see all 1252 1253 objects of all jobs. This MIB does not require, nor does it specify 1254 how, such restrictions would be implemented. A monitoring application 1255 SHOULD enforce the site security policy with respect to returning 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 1256 1257 1258 jmJobOwner object in the jmJobTable does not match the user's user 1259 name.

1260 An operator is a privileged user that would be able to see all objects 1261 of all jobs, independent of the policy for unprivileged users.

1262 3.9 Notifications

1263 This MIB does not specify any notifications. For simplicity,

management applications are expected to poll for status. The 1264 jmGeneralJobPersistence and jmGeneralAttributePersistence objects 1265 1266 assist an application to determine the polling rate. The resulting 1267 network traffic is not expected to be significant.

1268 4. MIB specification

1269 The following pages constitute the actual Job Monitoring MIB.

	INTERNET-DRAFT	Job Monitoring MIB,	V1.0	January 1998	
1270 1271	Job-Monitoring-MIB DEFINITIONS ::= BEGIN				
1272	<pre>IMPORTS 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</pre>				
1273 1274 1275	Use the enterprises arc assigned to the PWG which is pwg(2699). Assign the first value: jobmonMIB(1) immediately under pwg(2669).				
1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1292 1292 1292 1292 1293 1294 1295 1294 1295 1296 1297 1298 1299 1300 1301 1302 1304 1305 1307 1308	<pre>jobmonMIB MODULE-IDENTITY LAST-UPDATED "9801130000Z" ORGANIZATION "Printer Working Group (PWG)" CONTACT-INFO "Tom Hastings Postal: Xerox Corp. Mail stop ESAE-231 701 S. Aviation Blvd. El Segundo, CA 90245 Tel: (301)333-6413 Fax: (301)333-5514 E-mail: hastings@cpl0.es.xerox.com Send questions and comments to the Printer Working Group (PWG) using the Job Monitoring Project (JMP) Mailing List: jmp@pwg.org</pre>				
	<pre>For further information, including how to subscribe to the jmp mailing list, access the PWG web page under 'JMP': <u>http://www.pwg.org/</u> Implementers of this specification are encouraged to join the jmp mailing list in order to participate in discussions on any clarifications needed and registration proposals being reviewed in order to achieve consensus." DESCRIPTION "The MIB module for monitoring job in servers, printers, and other devices. Version: 1.0" ::= { enterprises pwg(2699) jobmonMIB(1) }</pre>				

Job Monitoring MIB, V1.0

```
1310
1311
      -- Textual conventions for this MIB module
1312
1313
      JmUTF8StringTC ::= TEXTUAL-CONVENTION
1314
          DISPLAY-HINT "255a"
1315
          STATUS
                      current
1316
          DESCRIPTION
1317
              "To facilitate internationalization, this TC represents
1318
              information taken from the ISO/IEC IS 10646-1 character set,
1319
              encoded as an octet string using the UTF-8 character encoding
1320
              scheme."
1321
          REFERENCE
              "See section 3.6.1, entitled: 'Text generated by the server or
1322
1323
              device'."
1324
          SYNTAX
                     OCTET STRING (SIZE (0..63))
1325
1326
1327
1328
1329
      JmJobStringTC ::= TEXTUAL-CONVENTION
1330
          STATUS
                   current
1331
          DESCRIPTION
              "To facilitate internationalization, this TC represents
1332
1333
              information using any coded character set registered by IANA as
1334
              specified in section 3.7. While it is recommended that the
              coded character set be UTF-8 [UTF-8], the actual coded
1335
              character set SHALL be indicated by the value of the
1336
1337
              jobCodedCharSet(8) attribute for the job."
1338
          REFERENCE
1339
              "See section 3.6.2, entitled: 'Text supplied by the job
1340
              submitter'."
1341
          SYNTAX OCTET STRING (SIZE (0..63))
1342
1343
1344
1345
1346
      JmNaturalLanguageTagTC ::= TEXTUAL-CONVENTION
1347
          STATUS
                  current
1348
          DESCRIPTION
1349
              "An IETF RFC 1766-compliant 'language tag', with zero or more
1350
              sub-tags that identify a natural language. While RFC 1766
1351
              specifies that the US-ASCII values are case-insensitive, this
1352
              MIB specification requires that all characters SHALL be lower
              case in order to simplify comparing by management
1353
1354
              applications."
1355
          REFERENCE
1356
              "See section 3.6.1, entitled: 'Text generated by the server or
              device' and section 3.6.2, entitled: 'Text supplied by the job
1357
              submitter'."
1358
1359
          SYNTAX
                    OCTET STRING (SIZE (0..63))
1360
1361
```

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 1362 JmTimeStampTC ::= TEXTUAL-CONVENTION 1363 STATUS current 1364 DESCRIPTION 1365 "The simple time at which an event took place. The units SHALL be in seconds since the system was booted. 1366 1367 1368 NOTE - JmTimeStampTC is defined in units of seconds, rather than 100ths of seconds, so as to be simpler for agents to 1369 1370 implement (even if they have to implement the 100ths of a 1371 second to comply with implementing sysUpTime in MIB-II[mib-1372 II].) 1373 NOTE - JmTimeStampTC is defined as an Integer32 so that it can be used as a value of an attribute, i.e., as a value of the 1374 1375 jmAttributeValueAsInteger object. The TimeStamp textual-1376 1377 convention defined in SNMPv2-TC [SMIv2-TC] is defined as an 1378 APPLICATION 3 IMPLICIT INTEGER tag, not an Integer 32 which is 1379 defined in SNMPv2-SMI [SMIv2-TC] as UNIVERSAL 2 IMPLICIT 1380 INTEGER, so cannot be used in this MIB as one of the values of 1381 jmAttributeValueAsInteger." 1382 SYNTAX INTEGER(0..2147483647) 1383 1384 1385 1386 1387 JmJobSourcePlatformTypeTC ::= TEXTUAL-CONVENTION 1388 STATUS current 1389 DESCRIPTION 1390 "The source platform type that can submit jobs to servers or 1391 devices in any of the 3 configurations." 1392 REFERENCE 1393 "This is a type 2 enumeration. See Section 3.7.1.2. See also 1394 IANA operating-system-names registry." 1395 INTEGER { SYNTAX other(1), unknown(2), sptUNIX(3),--UNIXsptOS2(4),--OS/2sptPCDOS(5),--DOSsptNT(6),--NT sptMVS(7), -- MVS -- VM sptVM(8), spt0S400(9),-- OS/400sptVMS(10),-- VMSsptWindows(11),-- WindowssptNetWare(12)-- NetWare } 1396 1397

```
1398
1399
      JmFinishingTC ::= TEXTUAL-CONVENTION
1400
          STATUS current
1401
          DESCRIPTION
              "The type of finishing operation.
1402
1403
1404
              These values are the same as the enum values of the IPP
              'finishings' attribute. See Section 3.7.1.2.
1405
1406
1407
              other(1),
1408
                  Some other finishing operation besides one of the specified
1409
                  or registered values.
1410
1411
              unknown(2),
1412
                  The finishing is unknown.
1413
1414
             none(3),
1415
                  Perform no finishing.
1416
1417
             staple(4),
1418
                  Bind the document(s) with one or more staples. The exact
                  number and placement of the staples is site-defined.
1419
1420
1421
             punch(5),
1422
                  This value indicates that holes are required in the
1423
                  finished document. The exact number and placement of the
                  holes is site-defined The punch specification MAY be
1424
                  satisfied (in a site- and implementation-specific manner)
1425
1426
                  either by drilling/punching, or by substituting pre-drilled
1427
                  media.
1428
1429
             cover(6),
1430
                  This value is specified when it is desired to select a non-
1431
                  printed (or pre-printed) cover for the document. This does
1432
                  not supplant the specification of a printed cover (on cover
1433
                  stock medium) by the document itself.
1434
              bind(7)
1435
1436
                  This value indicates that a binding is to be applied to the
1437
                  document; the type and placement of the binding is product-
                  specific."
1438
1439
          REFERENCE
1440
              "This is a type 2 enumeration. See Section 3.7.1.2."
1441
                      INTEGER {
          SYNTAX
1442
              other(1),
1443
              unknown(2),
1444
             none(3),
            staple(4),
punch(5),
1445
1446
1447
             cover(6),
1448
             bind(7)
1449 }
```

```
Bergman, Hastings, Isaacson, LewisInformational
```

1450

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

1451 JmPrintQualityTC ::= TEXTUAL-CONVENTION 1452 1453 STATUS current 1454 DESCRIPTION 1455 "Print quality settings. 1456 1457 These values are the same as the enum values of the IPP 'print-1458 quality' attribute. See Section 3.7.1.2." 1459 REFERENCE 1460 "This is a type 2 enumeration. See Section 3.7.1.2." SYNTAX INTEGER { 1461 other(1), -- Not one of the specified or registered -- values. unknown(2), -- The actual value is unknown. draft(3), -- Lowest quality available on the printer. normal(4), -- Normal or intermediate quality on the -- printer. -- Highest quality available on the printer. high(5) 1462 } 1463 1464 1465 1466 1467 JmPrinterResolutionTC ::= TEXTUAL-CONVENTION 1468 STATUS current 1469 DESCRIPTION 1470 "Printer resolutions. 1471 1472 Nine octets consisting of two 4-octet SIGNED-INTEGERs followed 1473 by a SIGNED-BYTE. The values are the same as those specified 1474 in the Printer MIB [printmib]. The first SIGNED-INTEGER contains the value of prtMarkerAddressabilityXFeedDir. 1475 The 1476 second SIGNED-INTEGER contains the value of 1477 prtMarkerAddressabilityFeedDir. The SIGNED-BYTE contains the 1478 value of prtMarkerAddressabilityUnit. 1479 Note: the latter value is either 3 (tenThousandsOfInches) or 4 1480 1481 (micrometers) and the addressability is in 10,000 units of 1482 measure. Thus the SIGNED-INTEGERs represent integral values in 1483 either dots-per-inch or dots-per-centimeter. 1484 1485 The syntax is the same as the IPP 'printer-resolution' attribute. See Section 3.7.1.2." 1486 1487 SYNTAX OCTET STRING (SIZE(9)) 1488

```
1489
1490 JmTonerEconomyTC ::= TEXTUAL-CONVENTION
1491
         STATUS current
1492
         DESCRIPTION
1493
             "Toner economy settings."
1494
         REFERENCE
1495
             "This is a type 2 enumeration. See Section 3.7.1.2."
1496 SYNTAX INTEGER {
             unknown(2), -- unknown.
              off(3),
                           -- Off. Normal. Use full toner.
                          -- On. Use less toner than normal.
              on(4)
          }
1497
1498
1499
1500
1501
    JmBooleanTC ::= TEXTUAL-CONVENTION
1502
      STATUS current
1503
         DESCRIPTION
             "Boolean true or false value."
1504
1505
         REFERENCE
1506
             "This is a type 2 enumeration. See Section 3.7.1.2."
       SYNTAX INTEGER {
1507
              unknown(2), -- unknown.
              false(3), -- TRUE.
                           -- FALSE.
1508
          }
1509
1510
1511
1512 JmMediumTypeTC ::= TEXTUAL-CONVENTION
1513
         STATUS current
1514
         DESCRIPTION
1515
             "Identifies the type of medium.
1516
1517
             other(1),
1518
                 The type is neither one of the values listed in this
1519
                 specification nor a registered value.
1520
1521
             unknown(2),
1522
                 The type is not known.
1523
1524
             stationery(3),
1525
                 Separately cut sheets of an opaque material.
1526
1527
            transparency(4),
1528
                 Separately cut sheets of a transparent material.
1529
1530
            envelope(5),
                 Envelopes that can be used for conventional mailing
1531
1532
                 purposes.
```

1533	
1534	envelopePlain(6),
1535	Envelopes that are not preprinted and have no windows.
1536	
1537	envelopeWindow(7),
1538	Envelopes that have windows for addressing purposes.
1539	Enveropes that have windows for addressing purposes.
	rout invourt our (0)
1540	continuousLong(8),
1541	Continuously connected sheets of an opaque material
1542	connected along the long edge.
1543	
1544	continuousShort(9),
1545	Continuously connected sheets of an opaque material
1546	connected along the short edge.
1547	
1548	<pre>tabStock(10),</pre>
1549	Media with tabs.
1550	
1551	multiPartForm(11),
1552	Form medium composed of multiple layers not pre-attached to
1553	one another; each sheet MAY be drawn separately from an
1554	
	input source.
1555	
1556	labels(12),
1557	Label-stock.
1558	
1559	multiLayer(13)
1560	Form medium composed of multiple layers which are pre-
1560	Form medium composed of multiple layers which are pre-
1560 1561 1562	Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers."
1560 1561 1562 1563	Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE
1560 1561 1562 1563 1564	Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "This is a type 2 enumeration. See Section 3.7.1.2. These
1560 1561 1562 1563 1564 1565	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "This is a type 2 enumeration. See Section 3.7.1.2. These enum values correspond to the keyword name strings of the</pre>
1560 1561 1562 1563 1564 1565 1566	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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</pre>
1560 1561 1562 1563 1564 1565 1566 1567	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values."</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER {</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1571	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope(5),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope(5),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1572 1573 1574 1575	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope(5), envelopePlain(6),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1572 1573 1574 1575 1576	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope(5), envelopePlain(6), envelopeWindow(7),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1570 1571 1572 1572 1573 1574 1575 1576 1577 1577	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope[5), envelopePlain(6), envelopeWindow(7), continuousLong(8), continuousShort(9),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1572 1573 1574 1575 1576 1577 1578 1579	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope[5), envelopePlain(6), envelopeWindow(7), continuousLong(8), continuousShort(9), tabStock(10),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1572 1573 1574 1575 1576 1577 1578 1579 1580	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelope[5), envelopePlain(6), envelopeWindow(7), continuousLong(8), continuousShort(9), tabStock(10), multiPartForm(11),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelopePlain(6), envelopeWindow(7), continuousCong(8), continuousShort(9), tabStock(10), multiPartForm(11), labels(12),</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelopePlain(6), envelopeWindow(7), continuousLong(8), continuousShort(9), tabStock(10), multiPartForm(11), labels(12), multiLayer(13)</pre>
1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581	<pre>Form medium composed of multiple layers which are pre- attached to one another, e.g. for use with impact printers." REFERENCE "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 is no printer description attribute in IPP/1.0 that represents these values." SYNTAX INTEGER { other(1), unknown(2), stationery(3), transparency(4), envelopePlain(6), envelopeWindow(7), continuousCong(8), continuousShort(9), tabStock(10), multiPartForm(11), labels(12),</pre>

Bergman, Hastings, Isaacson, LewisInformational

```
1585
1586
      JmJobCollationTypeTC ::= TEXTUAL-CONVENTION
1587
          STATUS current
1588
          DESCRIPTION
              "This value is the type of job collation. Implementations that
1589
1590
              don't support multiple documents or don't support multiple
              copies SHALL NOT support the uncollatedDocuments(5) value."
1591
1592
          REFERENCE
1593
              "This is a type 2 enumeration. See Section 3.7.1.2. See also
1594
              Section 3.4, entitled 'Monitoring Job Progress'."
1595
                     INTEGER {
          SYNTAX
1596
              other(1),
1597
              unknown(2),
1598
              uncollatedSheets(3), -- sheets within each document copy
                                      -- are not collated: 1 1 ..., 2 2 ...,
1599
1600
             collatedDocuments(4),
                                     -- internal collated sheets,
                                      -- documents: A, B, A, B, ...
1601
1602
             uncollatedDocuments(5) -- internal collated sheets,
                                      -- documents: A, A, ..., B, B, ...
1603
          }
1604
1605
1606
      JmJobSubmissionIDTypeTC ::= TEXTUAL-CONVENTION
1607
          STATUS
                      current
1608
          DESCRIPTION
1609
              "Identifies the format type of a job submission ID.
1610
              Each job submission ID is a fixed-length, 48-octet printable
1611
1612
              US-ASCII [US-ASCII] coded character string containing no
1613
              control characters, consisting of the following fields:
1614
1615
                octet 1: The format letter identifying the format. The US-
                  ASCII characters '0-9', 'A-Z', and 'a-z' are assigned in
1616
                  order giving 62 possible formats.
1617
1618
                octets 2-40: A 39-character, US-ASCII trailing SPACE filled
1619
                  field specified by the format letter, if the data is less
1620
                  than 39 ASCII characters.
                octets 41-48: A sequential or random US-ASCII number to make
1621
1622
                  the ID quasi-unique.
1623
1624
              If the client does not supply a job submission ID in the job
              submission protocol, then the agent SHALL assign a job
1625
1626
              submission ID using any of the standard formats that are
1627
              reserved for the agent. Clients SHALL not use formats that are
              reserved for agents and agents SHALL NOT use formats that are
1628
1629
              reserved for clients, in order to reduce conflicts in ID
1630
             generation. See the description for which formats are reserved
1631
             for clients or for agents.
1632
1633
             Registration of additional formats may be done following the
1634
              procedures described in Section 3.7.3.
1635
```

1636 The format values defined at the time of completion of this 1637 specification are: 1638 1639 Format 1640 Letter Description 1641 _____ 1642 '0' Job Owner generated by the server/device octets 2-40: The last 39 bytes of the jmJobOwner object. 1643 octets 41-48: The US-ASCII 8-decimal-digit sequential number 1644 1645 assigned by the agent. 1646 This format is reserved for agents. 1647 NOTE - Clients wishing to use a job submission ID that 1648 1649 incorporates the job owner, SHALL use format '8', not format '0'. 1650 1651 1652 '1' Job Name octets 2-40: The last 39 bytes of the jobName attribute. octets 41-48: The US-ASCII 8-decimal-digit random number 1653 1654 assigned by the client. 1655 1656 This format is reserved for clients. 1657 1658 '2' Client MAC address octets 2-40: The client MAC address: in hexadecimal with each 1659 1660 nibble of the 6 octet address being '0'-'9' or 'A' - 'F' (uppercase only). Most significant octet first. 1661 octets 41-48: The US-ASCII 8-decimal-digit sequential number 1662 1663 assigned by the client. 1664 This format is reserved for clients. 1665 1666 '3' Client URL 1667 octets 2-40: The last 39 bytes of the client URL [URI-spec]. octets 41-48: The US-ASCII 8-decimal-digit sequential number 1668 1669 assigned by the client. 1670 This format is reserved for clients. 1671 '4' Job URI 1672 octets 2-40: The last 39 bytes of the URI [URI-spec] assigned 1673 1674 by the server or device to the job when the job was 1675 submitted for processing. octets 41-48: The US-ASCII 8-decimal-digit sequential number 1676 1677 assigned by the agent. 1678 This format is reserved for agents. 1679 1680 '5' POSIX User Number octets 2-40: The last 39 bytes of a user number, such as POSIX 1681 1682 user number. user number. octets 41-48: The US-ASCII 8-decimal-digit sequential number 1683 1684 assigned by the client. 1685 This format is reserved for clients. 1686

1687	'6' User Account Number
1688	octets 2-40: The last 39 bytes of the user account number.
1689	octets 41-48: The US-ASCII 8-decimal-digit sequential number
1690	assigned by the client.
1691	This format is reserved for clients.
1692	
1693	'7' DTMF Incoming FAX routing number
1694	octets 2-40: The last 39 bytes of the DTMF incoming FAX
1695	routing number.
1696	octets 41-48: The US-ASCII 8-decimal-digit sequential number
1697	assigned by the client.
1698	This format is reserved for clients.
1699	THIS TOTMAL IS TESETVED TOT CITENES.
	101 Job Orman groupling by the glient
1700	'8' Job Owner supplied by the client
1701	octets 2-40: The last 39 bytes of the job owner name (that the
1702	agent returns in the jmJobOwner object).
1703	octets 41-48: The US-ASCII 8-decimal-digit sequential number
1704	assigned by the client.
1705	This format is reserved for clients. See format '0' which is
1706	reserved for agents.
1707	
1708	'9' Host Name
1709	octets 2-40: The last 39 bytes of the host name with trailing
1710	SPACES that submitted the job to this server/device using a
1711	protocol, such as LPD [RFC-1179] which includes the host
1712	name in the job submission protocol.
1713	octets 41-48: The US-ASCII 8-decimal-digit leading zero
1714	representation of the job id generated by the submitting
1715	server (configuration 3) or the client (configuration 1 and
1716	2), such as in the LPD protocol.
1717	This format is reserved for clients.
1718	
1719	'A' AppleTalk Protocol
1720	octets 2-40: Contains the AppleTalk printer name, with the
1721	first character of the name in octet 2. AppleTalk printer
1722	names are a maximum of 31 characters. Any unused portion
1723	of this field shall be filled with spaces.
1724	octets 41-48: '00000XXX', where 'XXX' is the 3-digit US-ASCII
1725	decimal representation of the Connection Id.
1726	This format is reserved for agents.
1727	THIS TOTMAL IS TESCIVED TOT AGENES.
1728	'B' NetWare PServer
1729	octets 2-40: Contains the Directory Path Name as recorded by
1730	
1731	string is less than 40 octets, the left-most character in
1732	the string shall appear in octet position 2. Otherwise,
1733	only the last 39 bytes shall be included. Any unused
1734	portion of this field shall be filled with spaces.
1735	octets 41-48: '000XXXXX' The US-ASCII representation of the
1736	Job Number as per the NetWare File Server Queue Management
1737	Services.
1738	This format is reserved for agents.

Bergman, Hastings, Isaacson, LewisInformational

[Page 45]

1739	
1740	'C' Server Message Block protocol (SMB)
1741	octets 2-40: Contains a decimal (US-ASCII coded)
1742	representation of the 16 bit SMB Tree Id field, which
1743	uniquely identifies the connection that submitted the job
1744	to the printer. The most significant digit of the numeric
1745	string shall be placed in octet position 2. All unused
1746	portions of this field shall be filled with spaces. The
1747	SMB Tree Id has a maximum value of 65,535.
1748	octets 41-48: The US-ASCII 8-decimal-digit leading zero
1749	representation of the File Handle returned from the device
1750	to the client in response to a Create Print File command.
1751	This format is reserved for agents.
1752	mib format ib feberved for ageneb.
1753	'D' Transport Independent Printer/System Interface (TIP/SI)
1754	octets 2-40: Contains the Job Name from the Job Control-Start
1755	Job (JC-SJ) command. If the Job Name portion is less than
1756	40 octets, the left-most character in the string shall
1757	appear in octet position 2. Any unused portion of this
1758	field shall be filled with spaces. Otherwise, only the
1759	last 39 bytes shall be included.
1760	octets 41-48: The US-ASCII 8-decimal-digit leading zero
1761	
	representation of the jmJobIndex assigned by the agent.
1762	This format is reserved for agents, since the agent supplies
1763	octets 41-48, though the client supplies the job name. See
1764	format '1' reserved to clients to submit job name ids in
1765	which they supply octets 41-48.
1766	
1767	NOTE - the job submission id is only intended to be unique
1768	between a limited set of clients for a limited duration of
1769	time, namely, for the life time of the job in the context of
1770	the server or device that is processing the job. Some of the
1771	formats include something that is unique per client and a
1772	random number so that the same job submitted by the same client
1773	will have a different job submission id. For other formats,
1774	where part of the id is guaranteed to be unique for each
1775	client, such as the MAC address or URL, a sequential number
1776	SHOULD suffice for each client (and may be easier for each
1777	client to manage). Therefore, the length of the job submission
1778	id has been selected to reduce the probability of collision to
1779	an extremely low number, but is not intended to be an absolute
1780	guarantee of uniqueness. None-the-less, collisions are
1781	remotely possible, but without bad consequences, since this MIB
1782	is intended to be used only for monitoring jobs, not for
1783	controlling and managing them."
1784	REFERENCE
1785	"This is like a type 2 enumeration. See section 3.7.3."
1786	SYNTAX OCTET STRING(SIZE(1)) ASCII '0'-'9', 'A'-'Z', 'a'-'z'

1787 1788 JmJobStateTC ::= TEXTUAL-CONVENTION 1789 STATUS current 1790 DESCRIPTION 1791 "The current state of the job (pending, processing, completed, 1792 etc.). 1793 1794 The following figure shows the normal job state transitions: 1795 1796 +---> canceled(7) +---> pending(3) -----> processing(5) -----> completed(9) 1797 1798 / processing(s) / compresed(s)
/
/
/
/
aborted(8)
/
/ --->+ | | v 1799 1800 v
v
+---> pendingHeld(4) processingStopped(6) ---+ 1801 1802 1803 1804 Figure 4 - Normal Job State Transitions 1805 1806 Normally a job progresses from left to right. Other state 1807 transitions are unlikely, but are not forbidden. Not shown are 1808 the transitions to the canceled state from the pending, pendingHeld, and processingStopped states. 1809 1810 Jobs in the pending, processing, and processingStopped states 1811 1812 are called 'active', while jobs in the pendingHeld, canceled, aborted, and completed states are called 'inactive'. Jobs 1813 reach one of the three terminal states: completed, canceled, or 1814 1815 aborted, after the jobs have completed all activity, and all 1816 MIB objects and attributes have reached their final values for 1817 the job. 1818 1819 These values are the same as the enum values of the IPP 'job-1820 state' job attribute. See Section 3.7.1.2. 1821 1822 unknown(2), The job state is not known, or its state is indeterminate. 1823 1824 1825 pending(3), The job is a candidate to start processing, but is not yet 1826 1827 processing. 1828 1829 pendingHeld(4), 1830 The job is not a candidate for processing for any number of 1831 reasons but will return to the pending state as soon as the 1832 reasons are no longer present. The job's jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4) 1833 attributes SHALL indicate why the job is no longer a 1834 1835 candidate for processing. The reasons are represented as 1836 bits in the jmJobStateReasons1 object and/or 1837 jobStateReasonsN (N=2..4) attributes. See the

Bergman, Hastings, Isaacson, LewisInformational

[Page 47]

	INTERNET-DRAFT	Job Monitoring	MIB, V1.0	January 1998
1838 1839 1840		ceReasons <i>N</i> TC (<i>N</i> =1. ation of each reas		nvention for the
1841 1842 1843	processing(5 One or mo			
1843 1844 1845 1846 1847	purely s			o use, one or more zing, creating, or
1848 1849 1850 1851 1852	hardware	devices that are	interpreting a	o use, one or more PDL, making marks , such as stapling,
1853 1854	OR			
1855 1856 1857 1858 1859 1860	printing either b because	, but the output d	levice is not y sn't reached th in the output	e output device or device or some
1861 1862 1863 1864 1865 1866	state in device M job's ph	job is in the pro cludes the detaile IB indicated by th ysicalDevice attri evice MIB.	ed status repre ne hrDeviceInde	sented in the x value of the
1867 1868 1869 1870 1871 1872 1873	addition indicate jobPrint making ma value to	the progress of t	job's jmJobStat the job, such a cate when the d and/or the proc e server or dev	eReasons1 object to s adding the evice is actually essingToStopPoint ice is in the
1874 1875 1876 1877 1878 1879	reasons	has stopped while	the processin	any number of g state as soon as
1879 1880 1881 1882 1883 1884 1885	jobState job has device i	s jmJobStateReasor ReasonsN (N=24) stopped processing s stopped, the dev ob's jmJobStateRea	attributes MAY g. For example viceStopped val	indicate why the
1885 1886 1887 1888 1889	indicate: device.	s its condition ir The management ap	n human readabl pplication can	

Bergman, Hastings, Isaacson, LewisInformational

[Page 48]

1890 1891	device MIB using the job's deviceIndex attribute(s), if the agent implements such a device MIB
1892	
1893	canceled(7),
1894	A client has canceled the job and the server or device has
1895	completed canceling the job AND all MIB objects and
1896	attributes have reached their final values for the job.
1897	While the server or device is canceling the job, the job's
1898	jmJobStateReasons1 object SHOULD contain the
1899	processingToStopPoint value and one of the canceledByUser,
1900	canceledByOperator, or canceledAtDevice values. The
1901	canceledByUser, canceledByOperator, or canceledAtDevice
1902	values remain while the job is in the canceled state.
1903	
1904	aborted(8),
1905	The job has been aborted by the system, usually while the
1906	job was in the processing or processingStopped state and
1907	the server or device has completed aborting the job AND all
1908	MIB objects and attributes have reached their final values
1909	for the job. While the server or device is aborting the
1910	job, the job's jmJobStateReasons1 object MAY contain the
1911	processingToStopPoint and abortedBySystem values. If
1912	implemented, the abortedBySystem value SHALL remain while
1913	the job is in the aborted state.
1914	
1915	completed(9)
1916	The job has completed successfully or with warnings or
1917	errors after processing and all of the media have been
1918	successfully stacked in the appropriate output bin(s) AND
1919	all MIB objects and attributes have reached their final
1920	values for the job. The job's jmJobStateReasons1 object
1921	SHOULD contain one of: completedSuccessfully,
1922	completedWithWarnings, or completedWithErrors values."
1923	REFERENCE
1924	"This is a type 2 enumeration. See Section 3.7.1.2."
1925 1926	SYNTAX INTEGER {
1926 1927	unknown(2),
	pending(3),
1928	pendingHeld(4),
1929	processing(5),
1930 1931	processingStopped(6),
1931	canceled(7),
1932	aborted(8),
1933	completed(9)
1904	\$

1935		
1936	JmAttributeTypeTC ::= TEXTUAL-CONVENTION	
1937	STATUS current	
1938	DESCRIPTION	
1939	"The type of the attribute which identifies the attribute.	
1940		
1941	In the following definitions of the enums, each description	
1942	indicates whether the useful value of the attribute SHALL be	
1943	represented using the jmAttributeValueAsInteger or the	
1944	jmAttributeValueAsOctets objects by the initial tag: 'INTEGE	ER:′
1945	or 'OCTETS:', respectively.	
1946		
1947	Some attributes allow the agent implementer a choice of usef	Eul
1948	values of either an integer, an octets representation, or bo	
1949	depending on implementation. These attributes are indicated	
1950	with 'INTEGER:' AND/OR 'OCTETS:' tags.	
1951		
1952	A very few attributes require both objects at the same time	to
1953	represent a pair of useful values (see mediumConsumed(171)).	
1954	These attributes are indicated with 'INTEGER:' AND 'OCTETS:	
1955	tags. See the jmAttributeGroup for the descriptions of the	зе
1956	two MANDATORY objects.	
1957	Ş	
1958	NOTE - The enum assignments are grouped logically with value	2S
1959	assigned in groups of 20, so that additional values may be	
1960	registered in the future and assigned a value that is part of	сf
1961	their logical grouping.	
1962	5 5 1 5	
1963	Values in the range 2**30 to 2**31-1 are reserved for privat	ce
1964	or experimental usage. This range corresponds to the same	
1965	range reserved in IPP. Implementers are warned that use of	
1966	such values may conflict with other implementations.	
1967	Implementers are encouraged to request registration of enum	
1968	values following the procedures in Section 3.7.1.	
1969		
1970	NOTE: No attribute name exceeds 31 characters.	
1971		
1972	The standard attribute types defined at the time of completi	ion
1973	of the specification are:	
1974		
1975	jmAttributeTypeIndex Datatype	
1976		
1977		
1978	other(1), Integer32(-22147483647)	
1979	AND/OR	
1980	OCTET STRING(SIZE(063))	
1981	INTEGER: and/or OCTETS: An attribute that is not in t	the
1982	list and/or that has not been approved and registered wi	ith
1983	the PWG.	

1984	*****
1985	+ Job State attributes
1986	+ , mba fallessing attributer menific the state of a ish
1987	+ The following attributes specify the state of a job.
1988	+++++++++++++++++++++++++++++++++++++++
1989	
1990	jobStateReasons2(3), JmJobStateReasons2TC
1991	INTEGER: Additional information about the job's current
1992	state that augments the jmJobState object. See the
1993	description under the JmJobStateReasons1TC textual-
1994	convention.
1995	
1996	jobStateReasons3(4), JmJobStateReasons3TC
1997	INTEGER: Additional information about the job's current
1998	state that augments the jmJobState object. See the
1999	description under JmJobStateReasons1TC textual-convention.
2000	description ander onoobstatekeasonsrit textuar convention,
2000	jobStateReasons4(5), JmJobStateReasons4TC
2002	INTEGER: Additional information about the job's current
2003	state that augments the jmJobState object. See the
2004	description under JmJobStateReasons1TC textual-convention.
2005	
2006	processingMessage(6), JmUTF8StringTC(SIZE(063))
2007	OCTETS: MULTI-ROW: A coded character set message that is
2008	generated by the server or device during the processing of
2009	the job as a simple form of processing log to show progress
2010	and any problems. The natural language of each value is
2011	specified by the corresponding
2012	processingMessageNaturalLangTag(7) value.
2013	
2014	NOTE - This attribute is intended for such conditions as
2015	interpreter messages, rather than being the printable form
2016	of the jmJobState and jmJobStateReasons1 objects and
2017	jobStateReasons2, jobStateReasons3, and jobStateReasons4
2018	attributes. In order to produce a localized printable form
2019	of these job state objects/attribute, a management
2020	application SHOULD produce a message from their enum and
2020	bit values.
2021	Dit Values.
	NOTE These is no ish description attailed in $TDD/1$ (
2023	NOTE - There is no job description attribute in IPP/1.0
2024	that corresponds to this attribute and this attribute does
2025	not correspond to the IPP/1.0 'job-state-message' job
2026	description attribute, which is just a printable form of
2027	the IPP 'job-state' and 'job-state-reasons' job attributes.
2028	
2029	There is no restriction for the same message occurring in
2030	multiple rows.
2031	

2032 2033 2034 2035 2036 2037	<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>
2037 2038 2039 2040 2041 2042 2043 2044	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.
2045 2046 2047 2048 2049	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.
2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064	<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>
2065 2066 2067 2068 2069 2070 2071 2072	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. If the agent does not know what coded character set was
2073 2074 2075 2076 2077	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 Monitoring MIB	3, V1.0	January 1998
2078 2079 2080 2081 2082 2083 2083 2084 2085	OCTETS: by the for the by the mediumR	anguageTag(9), The natural language job submitter or defau job, i.e., all object 'JmJobStringTC' textua equested, etc. See Se d by the job submitter	of the job attri lted by the serv s and attributes l-convention, su ection 3.6.2, ent	butes supplied er or device represented ach as jobName,
2083 2086 2087 2088 2089 2090 2091 2092	by the return jobNatu	agent does not know wh job submitting client, a zero length string v ralLanguageTag(9) attr ralLanguageTag(9) att	the agent SHALI value for the vibute or (2) not	either (1) return
2093 2094 2095	+ Job Ident	++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++
2095 2096 2097 2098 2099 2100 2101	+ operator,	wing attributes help a or an accounting prog ++++++++++++++++++++++++++++++++++++	ram identify a j	job.
2102 2103 2104 2105 2106	jobURI(20), OCTETS: Identif example	ier (URI) [RFC-1738].		ource
2107 2108 2109 2110	SNMP Ge	The agent may be able t operation from small e the entire URI.		
2111 2112 2113		URI exceeds 63 octets, with the next 63 octe		
2114 2115 2116 2117	a URI,	IPP [ipp-model] has a though the URI standar mum length.		
2118 2119 2120 2121 2122 2123 2124 2125 2126 2127	submitt or cate custome	. , ,	ed data supplied counting service vices provided, per.	IAY be coded l by the es to allocate such as a

2128 serverAssignedJobName(22), JmJobStringTC(SIZE(0..63)) OCTETS: Configuration 3 only: The human readable string 2129 2130 name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is 2131 providing access to with this MIB. 2132 2133 2134 NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when 2135 2136 either the client does not support the jmJobSubmissionID or 2137 the server does not pass the jmJobSubmissionID through to 2138 the device. 2139 2140 jobName(23), JmJobStringTC(SIZE(0..63)) 2141 OCTETS: The human readable string name of the job as assigned by the submitting user to help the user 2142 2143 distinguish between his/her various jobs. This name does not need to be unique. 2144 2145 This attribute is intended for enabling a user or the 2146 2147 user's application to convey a job name that MAY be printed 2148 on a start sheet, returned in a query result, or used in notification or logging messages. 2149 2150 2151 In order to assist users to find their jobs for job 2152 submission protocols that don't supply a jmJobSubmissionID, the agent SHOULD maintain the jobName attribute for the 2153 time specified by the jmGeneralJobPersistence object, 2154 2155 rather than the (shorter) jmGeneralAttributePersistence 2156 object. 2157 2158 If this attribute is not specified when the job is 2159 submitted, no job name is assumed, but implementation specific defaults are allowed, such as the value of the 2160 documentName attribute of the first document in the job or 2161 2162 the fileName attribute of the first document in the job. 2163 The jobName attribute is distinguished from the jobComment 2164 attribute, in that the jobName attribute is intended to 2165 2166 permit the submitting user to distinguish between different 2167 jobs that he/she has submitted. The jobComment attribute is intended to be free form additional information that a 2168 2169 user might wish to use to communicate with himself/herself, 2170 such as a reminder of what to do with the results or to 2171 indicate a different set of input parameters were tried in 2172 several different job submissions. 2173

2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184	<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>
2185 2186 2187 2188 2189 2190 2191	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.
2192 2193 2194 2195 2196 2197	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.
2198 2199 2200 2201 2202	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.
2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213	<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>
2214 2215 2216 2217	<pre>submittingServerName(27), JmJobStringTC(SIZE(063)) OCTETS: For configuration 3 only: The administrative name of the server that submitted the job to the device.</pre>
2217 2218 2219 2220 2221 2222	submittingApplicationName(28), JmJobStringTC(SIZE(063)) OCTETS: The name of the client application (not the server in configuration 3) that submitted the job to the server or device.

2223 2224 2225 2226 2227	<pre>jobOriginatingHost(29), JmJobStringTC(SIZE(063)) OCTETS: The name of the client host (not the server host name in configuration 3) that submitted the job to the server or device.</pre>
2228 2229 2230 2231 2232 2233 2234 2235 2236 2237	<pre>deviceNameRequested(30), JmJobStringTC(SIZE(063)) OCTETS: The administratively defined coded character set name of the target device requested by the submitting user. For configuration 1, its value corresponds to the Printer MIB[print-mib]: prtGeneralPrinterName object. For configuration 2 and 3, its value is the name of the logical or physical device that the user supplied to indicate to the server on which device(s) they wanted the job to be processed.</pre>
2238 2239 2240 2241 2242 2243 2243 2244 2245	<pre>queueNameRequested(31), JmJobStringTC(SIZE(063)) OCTETS: The administratively defined coded character set name of the target queue requested by the submitting user. For configuration 1, its value corresponds to the queue in the device for which the agent is providing access. For configuration 2 and 3, its value is the name of the queue that the user supplied to indicate to the server on which device(s) they wanted the job to be processed.</pre>
2246 2247 2248 2249 2250	NOTE - typically an implementation SHOULD support either the deviceNameRequested or queueNameRequested attribute, but not both.
2250 2251 2252 2253 2254 2255 2256 2257 2258 2259	<pre>physicalDevice(32), hrDeviceIndex AND/OR JmUTF8StringTC(SIZE(063)) INTEGER: MULTI-ROW: The index of the physical device MIB instance requested/used, such as the Printer MIB[print- mib]. This value is an hrDeviceIndex value. See the Host Resources MIB[hr-mib]. AND/OR</pre>
2260 2261 2262	OCTETS: MULTI-ROW: The name of the physical device to which the job is assigned.
2263 2264 2265 2266	numberOfDocuments(33), Integer32(-22147483647) INTEGER: The number of documents in this job.
2267 2268 2269	The agent SHOULD return this attribute if the job has more than one document.

2270	fileName(34), JmJobStringTC(SIZE(063))
2271	OCTETS: MULTI-ROW: The coded character set file name or
2272	URI[URI-spec] of the document.
2273	
2274	There is no restriction on the same file name occurring in
2275	multiple rows.
2276	marcipie rowb.
2277	documentName(35), JmJobStringTC(SIZE(063))
2278	OCTETS: MULTI-ROW: The coded character set name of the
2279	document.
2280	document.
2281	There is no restriction on the same document name occurring
2282	in multiple rows.
2283	
2284	jobComment(36), JmJobStringTC(SIZE(063))
2285	OCTETS: An arbitrary human-readable coded character text
2286	string supplied by the submitting user or the job
2287	submitting application program for any purpose. For
2288	example, a user might indicate what he/she is going to do
2289	with the printed output or the job submitting application
2290	program might indicate how the document was produced.
2291	
2292	The jobComment attribute is not intended to be a name; see
2293	the jobName attribute.
2294	
2295	documentFormatIndex(37), Integer32(02147483647)
2296	INTEGER: MULTI-ROW: The index in the prtInterpreterTable
2297	in the Printer MIB[print-mib] of the page description
2298	language (PDL) or control language interpreter that this
2299	job requires/uses. A document or a job MAY use more than
2300	one PDL or control language.
2301	one PDL of concrot tanguage.
	NOWE De with all intensive attaikutes where multiple were
2302	NOTE - As with all intensive attributes where multiple rows
2303	are allowed, there SHALL be only one distinct row for each
2304	distinct interpreter; there SHALL be no duplicates.
2305	
2306	NOTE - This attribute type is intended to be used with an
2307	agent that implements the Printer MIB and SHALL not be used
2308	if the agent does not implement the Printer MIB. Such an
2309	agent SHALL use the documentFormat attribute instead.
2310	

2311 documentFormat(38), PrtInterpreterLangFamilyTC 2312 AND/OR 2313 OCTET STRING(SIZE(0..63)) 2314 INTEGER: MULTI-ROW: The interpreter language family corresponding to the Printer MIB[print-mib] 2315 2316 prtInterpreterLangFamily object, that this job requires/uses. A document or a job MAY use more than one 2317 PDL or control language. 2318 2319 2320 AND/OR 2321 2322 OCTETS: MULTI-ROW: The document format registered as a media type[iana-media-types], i.e., the name of the MIME 2323 2324 content-type/subtype. Examples: 'application/postscript', 'application/vnd.hp-PCL', 'application/pdf', 'text/plain' 2325 2326 (US-ASCII SHALL be assumed), 'text/plain; charset=iso-8859-1', and 'application/octet-stream'. The IPP 'document-2327 format' job attribute uses these same values with the same 2328 2329 semantics. See the IPP [ipp-model] 'mimeMediaType' 2330 attribute syntax and the document-format attribute for 2331 further examples and explanation. 2332 2333 2334 2335 + Job Parameter attributes 2336 + + The following attributes represent input parameters 2337 + supplied by the submitting client in the job submission 2338 2339 + protocol. 2340 2341 2342 jobPriority(50), Integer32(-2..100) INTEGER: The priority for scheduling the job. It is used 2343 2344 by servers and devices that employ a priority-based 2345 scheduling algorithm. 2346 2347 A higher value specifies a higher priority. The value 1 is defined to indicate the lowest possible priority (a job 2348 2349 which a priority-based scheduling algorithm SHALL pass over in favor of higher priority jobs). The value 100 is 2350 2351 defined to indicate the highest possible priority. 2352 Priority is expected to be evenly or 'normally' distributed across this range. The mapping of vendor-defined priority 2353 over this range is implementation-specific. -2 indicates 2354 2355 unknown. 2356

2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2365 2366 2367 2368 2369	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.
2370	jobHold(52), JmBooleanTC
2371	INTEGER: If the value is 'true(4)', a client has
2372	explicitly specified that the job is to be held until
2373	explicitly released. Until the job is explicitly released
2374	by a client, the job SHALL be in the pendingHeld state with
2375	the jobHoldSpecified value in the jmJobStateReasons1
2376	attribute.
2377	
2378	jobHoldUntil(53), JmJobStringTC(SIZE(063))
2379	OCTETS: The named time period during which the job SHALL
2380	become a candidate for processing, such as 'evening',
2381	'night', 'weekend', 'second-shift', 'third-shift', etc., as
2382	defined by the system administrator. See IPP [ipp-model]
2383	for the standard keyword values. Until that time period
2384	arrives, the job SHALL be in the pendingHeld state with the
2385	jobHoldUntilSpecified value in the jmJobStateReasons1
2386	object. The value 'no-hold' SHALL indicate explicitly that
2387	no time period has been specified; the absence of this
2388	attribute SHALL indicate implicitly that no time period has
2389	been specified.
2390	
2391	outputBin(54), Integer32(02147483647)
2392	AND/OR
2393	JmJobStringTC(SIZE(063))
2394	INTEGER: MULTI-ROW: The output subunit index in the
2395	Printer MIB[print-mib]
2396	
2397	AND/OR
2398	
2399	OCTETS: MULTI-ROW: the name or number (represented as
2400	ASCII digits) of the output bin to which all or part of the
2401 2402	job is placed in.
ムヨリム	

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 2403 sides(55), Integer32(-2..2)2404 INTEGER: MULTI-ROW: The number of sides, '1' or '2', that 2405 any document in this job requires/used. 2406 JmFinishingTC 2407 finishing(56), INTEGER: MULTI-ROW: Type of finishing that any document 2408 in this job requires/used. 2409 2410 2411 2412 2413 + Image Quality attributes (requested and consumed) 2414 + 2415 + For devices that can vary the image quality. 2416 2417 2418 printQualityRequested(70), JmPrintQualityTC 2419 INTEGER: MULTI-ROW: The print quality selection requested for a document in the job for printers that allow quality 2420 differentiation. 2421 2422 printQualityUsed(71), 2423 JmPrintQualityTC INTEGER: MULTI-ROW: The print quality selection actually 2424 used by a document in the job for printers that allow 2425 2426 quality differentiation. 2427 2428 printerResolutionRequested(72), JmPrinterResolutionTC OCTETS: MULTI-ROW: The printer resolution requested for a 2429 2430 document in the job for printers that support resolution 2431 selection. 2432 printerResolutionUsed(73), JmPrinterResolutionTC 2433 2434 OCTETS: MULTI-ROW: The printer resolution actually used by a document in the job for printers that support 2435 2436 resolution selection. 2437 tonerEcomonyRequested(74), JmTonerEconomyTC 2438 INTEGER: MULTI-ROW: The toner economy selection requested 2439 for documents in the job for printers that allow toner 2440 2441 economy differentiation. 2442 2443 tonerEcomonyUsed(75), JmTonerEconomyTC INTEGER: MULTI-ROW: The toner economy selection actually 2444 used by documents in the job for printers that allow toner 2445 2446 economy differentiation. 2447 tonerDensityRequested(76), Integer32(-2..100) 2448 INTEGER: MULTI-ROW: The toner density requested for a 2449 document in this job for devices that can vary toner 2450 density levels. Level 1 is the lowest density and level 2451 100 is the highest density level. Devices with a smaller 2452 2453 range, SHALL map the 1-100 range evenly onto the implemented range. 2454

Bergman, Hastings, Isaacson, LewisInformational

[Page 60]

2457INTEGER:MULTI-ROW:The toner density used by documents2458in this job for devices that can vary toner density levels.2459Level 1 is the lowest density and level 100 is the highest2460density level.24611-100 range evenly onto the implemented range.246224632463*********************************	2455	
2458in this job for devices that can vary toner density levels.2459Level 1 is the lowest density and level 100 is the highest2460density level. Devices with a smaller range, SHALL map the24611-100 range evenly onto the implemented range.246224632463*********************************	2456	
2459Level 1 is the lowest density and level 100 is the highest2460density level. Devices with a smaller range, SHALL map the2461l-100 range evenly onto the implemented range.2463*********************************	2457	INTEGER: MULTI-ROW: The toner density used by documents
2460density level. Devices with a smaller range, SHALL map the 1-100 range evenly onto the implemented range.24611-100 range evenly onto the implemented range.2462246324642465+ Job Progress attributes (requested and consumed) +2466+ Pairs of these attributes can be used by monitoring a applications to show an indication of relative progress24692470+ You users. See section 3.4, entitled + 'Monitoring Job Progress'.247124722473247424752475247624762477jobCopiesCompleted(91),INTEGER: The number of copies of the entire job that are to be produced.24762477247824782481documentCopiesRequested(92),INTEGER: The total count of the number of document copies requested for the job as a whole. If there are documents A, B, and C, and document Dis specified to produce 4 copies, the number of document.2481248224832484A, B, and C, and document copies requested is 6 for the job.2485248624872488248824892490249124912492249324942494249524952496249624962497249824982498 <td>2458</td> <td>in this job for devices that can vary toner density levels.</td>	2458	in this job for devices that can vary toner density levels.
24611-100 range evenly onto the implemented range.2462246324642465+ Job Progress attributes (requested and consumed)2466+2467+ Pairs of these attributes can be used by monitoring2468+ applications to show an indication of relative progress2469+ to users. See section 3.4, entitled2470471***********************************	2459	Level 1 is the lowest density and level 100 is the highest
2462	2460	density level. Devices with a smaller range, SHALL map the
246324642465+ Job Progress attributes (requested and consumed)2466+2467+ Pairs of these attributes can be used by monitoring2468+ applications to show an indication of relative progress2469+ to users. See section 3.4, entitled24702471***********************************	2461	1-100 range evenly onto the implemented range.
2464+++++++++++++++++++++++++++++++++	2462	
2465+ Job Progress attributes (requested and consumed)2466+2467+ Pairs of these attributes can be used by monitoring2468+ applications to show an indication of relative progress2469+ to users. See section 3.4, entitled2470+ 'Monitoring Job Progress'.2471+++++++++++++++++++++++++++++++++	2463	
2466+2467+2468+2469+2469+2470+247124712472247324742474247524752476247624772478247824792479247424742475247524762477247824782479247924802481documentCopiesRequested(92),2482248324842484248524852486248624862487248824892480248024812481248224832484248424852486248624872488248924892480248124812482248324842484248524852486248624872488248924892480248124812482248224832484248424852485248624862487<	2464	+++++++++++++++++++++++++++++++++++++++
2467+ Pairs of these attributes can be used by monitoring2468+ applications to show an indication of relative progress2469+ to users. See section 3.4, entitled2470+ 'Monitoring Job Progress'.2471+++++++++++++++++++++++++++++++++	2465	+ Job Progress attributes (requested and consumed)
2468+ applications to show an indication of relative progress2469+ to users. See section 3.4, entitled2470+ 'Monitoring Job Progress'.2471+************************************	2466	
2468+ applications to show an indication of relative progress2469+ to users. See section 3.4, entitled2470+ 'Monitoring Job Progress'.2471+************************************	2467	+ Pairs of these attributes can be used by monitoring
2469+ to users. See section 3.4, entitled2470+ 'Monitoring Job Progress'.2471+ 'Monitoring Job Progress'.2472jobCopiesRequested(90), Integer32(-22147483647)2473jobCopiesCompleted(91), Integer32(-22147483647)2476iNTEGER: The number of copies of the entire job that are to be produced.2476jobCopiesCompleted(91), Integer32(-22147483647)2477jobCopiesCompleted(91), Integer32(-22147483647)2478INTEGER: The number of copies of the entire job that have been completed so far.2480documentCopiesRequested(92), Integer32(-22147483647)2481documentCopiesRequested(92), Integer32(-22147483647)2482INTEGER: The total count of the number of document copies requested for the job as a whole. If there are documents A, B, and C, and document B is specified to produce 4 copies, the number of document.2489This attribute SHALL be used only when a job has multiple documents. The jobCopiesRequested attribute SHALL be used when the job has only one document.2491iNTEGER: The total count of the number of document copies completed so far for the job as a whole. If there are documents A, B, and C, and document B is specified to produce 4 copies, the number of document copies starts a 0 and runs up to 6 for the job as the job processes.2496This attribute SHALL be used only when a job has multiple documents. The jobCopiesCompleted attribute SHALL be used2499This attribute SHALL be used only when a job has multiple documents. The jobCopiesCompleted attribute SHALL be used	2468	
 2470 + 'Monitoring Job Progress'. 2471 2472 2473 jobCopiesRequested(90), Integer32(-22147483647) 2474 INTEGER: The number of copies of the entire job that are to be produced. 2476 2477 jobCopiesCompleted(91), Integer32(-22147483647) 2478 INTEGER: The number of copies of the entire job that have been completed so far. 2480 2481 documentCopiesRequested(92), Integer32(-22147483647) 2482 INTEGER: The total count of the number of document copies requested for the job as a whole. If there are documents A, B, and C, and document copies requested is 6 for the job. 2487 This attribute SHALL be used only when a job has multiple documentCopiesCompleted(93), Integer32(-22147483647) 2492 documentCopiesCompleted(93), Integer32(-22147483647) 2493 INTEGER: The total count of the number of document copies completed so far for the job as a whole. If there are documents and and count of the number of document copies completed (93), Integer32(-22147483647) 2494 INTEGER: The total count of the number of document copies completed so far for the job as a whole. If there are documents A, B, and C, and document B is specified to produce 4 copies, the number of document copies starts a 0 and runs up to 6 for the job as the job processes. 2499 This attribute SHALL be used only when a job has multiple documents A. The jobCopiesCompleted attribute SHALL be used 	2469	
2471*********************************	2470	
24722473jobCopiesRequested(90),Integer32(-22147483647)2474INTEGER: The number of copies of the entire job that are2475to be produced.2476jobCopiesCompleted(91),Integer32(-22147483647)2477jobCopiesCompleted(91),Integer32(-22147483647)2478INTEGER: The number of copies of the entire job that have2479been completed so far.2480documentCopiesRequested(92),Integer32(-22147483647)2481documentCopiesRequested(92),Integer32(-22147483647)2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2488documentS. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491INTEGER: The total count of the number of document copies2492documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494copies, the number of document copies2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple <td>2471</td> <td></td>	2471	
2473jobCopiesRequested(90),Integer32(-22147483647)2474INTEGER: The number of copies of the entire job that are2475to be produced.2476jobCopiesCompleted(91),Integer32(-22147483647)2477jobCopiesCompleted(91),Integer32(-22147483647)2478INTEGER: The number of copies of the entire job that have2480been completed so far.2481documentCopiesRequested(92),Integer32(-22147483647)2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2490when the job has only one document.2491INTEGER: The total count of the number of document copies2493completed so far for the job as a whole. If there are2494copiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499	2472	
2474INTEGER: The number of copies of the entire job that are to be produced.2475to be produced.2476jobCopiesCompleted(91), Integer32(-22147483647)2478INTEGER: The number of copies of the entire job that have been completed so far.2480documentCopiesRequested(92), Integer32(-22147483647)2481documentCopiesRequested(92), Integer32(-22147483647)2482INTEGER: The total count of the number of document copies requested for the job as a whole. If there are documents2485copies, the number of document copies requested is 6 for the job.2486This attribute SHALL be used only when a job has multiple documents. The jobCopiesRequested attribute SHALL be used when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647) INTEGER: The total count of the number of document copies completed so far for the job as a whole. If there are document SA, B, and C, and document B is specified to produce 4 copies, the number of document copies completed so far for the job as a whole. If there are documents A, B, and C, and document B is specified to produce 4 copies, the number of document copies starts a 0 and runs up to 6 for the job as the job processes.2499This attribute SHALL be used only when a job has multiple documents. The jobCopiesCompleted attribute SHALL be used2499This attribute SHALL be used only when a job has multiple documents. The jobCopiesCompleted attribute SHALL be used	2473	iobCopiesRequested(90), Integer32(-22147483647)
2475to be produced.24762477247724782479247924802481documentCopiesRequested(92),248224832484248424842485248624862486248724882488248924892489248024812484248524862486248624872488248824892489249024912492249224932493249424952495249524962497249824982499249924992499249024912500260271272273274274275275276277278278279279279270271271272273274274274275275276277278278279279270271271	2474	
24762477jobCopiesCompleted(91),Integer32(-22147483647)2478INTEGER: The number of copies of the entire job that have2479been completed so far.2480documentCopiesRequested(92),Integer32(-22147483647)2481documentCopiesRequested(92),Integer32(-22147483647)2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2490when the job has only one document.2491documentCopiesCompleted(93),Integer32(-22147483647)2492documentCopiesCompleted(93),Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494copiesCompleted(93),Integer32(-22147483647)2495document CopiesCompleted(93),Integer32(-22147483647)2496produce 4 copies, the number of document copies2497and runs up to 6 for the job as a whole. If there are2498documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2490		1 2
2477jobCopiesCompleted(91),Integer32(-22147483647)2478INTEGER: The number of copies of the entire job that have2479been completed so far.2480documentCopiesRequested(92),Integer32(-22147483647)2481documentCopiesRequested(92),Integer32(-22147483647)2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2490when the job has only one document.2491documentCopiesCompleted(93),2492documentCopiesCompleted(93),2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499		
2478INTEGER: The number of copies of the entire job that have been completed so far.2480248124822481248224832484248424842485248524862486248724882489248924892490249124922492249324942494249524952496249724982498249924912492249524952496249624972498249824992499249924902491250024912492249324942495249524962496249724982498249924992499249924992499249924992499249024912491249224932494249524952496249624972498249824992499249924992499249924992		iobCopiesCompleted(91). Integer32(-22147483647)
2479been completed so far.2480248124812482248324842484248424852485248624862487248824892489248024812482248324842484248524862486248724882489249024902491249124922492249324952496249624972498249824982499249624962497249824982499<		
24802481documentCopiesRequested(92), Integer32(-22147483647)2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2491documents. The jobCopiesCompleted attribute SHALL be used		
2481documentCopiesRequested(92),Integer32(-22147483647)2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2492INTEGER: The total count of the number of document copies2494copies, the number of document B is specified to2495produce 4 copies, the number of document copies starts a 02496and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2491documents A, B, and C, and document copies starts a 02496and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2499This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2482INTEGER: The total count of the number of document copies2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2491when the job has only one document.2492documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494copies, the number of document copies2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2499documents. The jobCopiesCompleted attribute SHALL be used		documentCopiesRequested(92), Integer32(-22147483647)
2483requested for the job as a whole. If there are documents2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2492documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2484A, B, and C, and document B is specified to produce 42485copies, the number of document copies requested is 6 for2486the job.2487This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2492documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2485copies, the number of document copies requested is 6 for2486the job.248724882488This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2486the job.2487248824882489248924902490249124912492249224932494249424952495249624962497249824982499249924912500249124922493249424952495249624972498249824992499249924992499250024912500249224932494249524952496249724982499249924992499250024912500 </td <td></td> <td></td>		
24872488This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2492documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2488This attribute SHALL be used only when a job has multiple2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2489documents. The jobCopiesRequested attribute SHALL be used2490when the job has only one document.2491documentCopiesCompleted(93), Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02498and runs up to 6 for the job as the job processes.2499This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used	2488	This attribute SHALL be used only when a job has multiple
2490when the job has only one document.24912492249324932494249424952495249624972498249824992499249924992499249924992499249024912492249324942495249524962497249824992500249425002495249624992499249925002500		
24912492documentCopiesCompleted(93),Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2492documentCopiesCompleted(93),Integer32(-22147483647)2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2493INTEGER: The total count of the number of document copies2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		documentCopiesCompleted(93). Integer32(-2.,2147483647)
2494completed so far for the job as a whole. If there are2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2495documents A, B, and C, and document B is specified to2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2496produce 4 copies, the number of document copies starts a 02497and runs up to 6 for the job as the job processes.2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2497and runs up to 6 for the job as the job processes.249824982499This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2498This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2499This attribute SHALL be used only when a job has multiple2500documents. The jobCopiesCompleted attribute SHALL be used		
2500 documents. The jobCopiesCompleted attribute SHALL be used		This attribute SHALL be used only when a job has multiple
	2502	

January 1998

2503 jobKOctetsTransferred(94), Integer32(-2..2147483647) 2504 INTEGER: The number of K (1024) octets transferred to the 2505 server or device to which the agent is providing access. This count is independent of the number of copies of the 2506 job or documents that will be produced, but it is only a 2507 2508 measure of the number of bytes transferred to the server or 2509 device. 2510 2511 The agent SHALL round the actual number of octets 2512 transferred up to the next higher K. Thus 0 octets SHALL 2513 be represented as '0', 1-1024 octets SHALL BE represented as '1', 1025-2048 SHALL be '2', etc. When the job 2514 2515 completes, the values of the jmJobKOctetsPerCopyRequested 2516 object and the jobKOctetsTransferred attribute SHALL be 2517 equal. 2518 2519 NOTE - The jobKOctetsTransferred can be used with the 2520 jmJobKOctetsPerCopyRequested object in order to produce a relative indication of the progress of the job for agents 2521 2522 that do not implement the jmJobKOctetsProcessed object. 2523 2524 sheetCompletedCopyNumber(95), Integer32(-2..2147483647) 2525 The number of the copy being stacked for the INTEGER: 2526 current document. This number starts at 0, is set to 1 2527 when the first sheet of the first copy for each document is 2528 being stacked and is equal to n where n is the nth sheet stacked in the current document copy. See section 3.4, 2529 2530 entitled 'Monitoring Job Progress'. 2531 2532 sheetCompletedDocumentNumber(96), Integer32(-2..2147483647) 2533 INTEGER: The ordinal number of the document in the job 2534 that is currently being stacked. This number starts at 0, 2535 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 2536 2537 the nth document in the job, starting with 1. 2538 Implementations that only support one document jobs SHOULD 2539 2540 NOT implement this attribute. 2541 JmJobCollationTypeTC 2542 jobCollationType(97), 2543 INTEGER: The type of job collation. See also Section 3.4, 2544 entitled 'Monitoring Job Progress'. 2545

2546 2547 2548 + Impression attributes 2549 + + See the definition of the terms 'impression', 'sheet', 2550 2551 + and 'page' in Section 2. 2552 2553 + See also jmJobImpressionsPerCopyRequested and 2554 + jmJobImpressionsCompleted objects in the jmJobTable. 2555 2556 impressionsSpooled(110), Integer32(-2..2147483647) 2557 INTEGER: The number of impressions spooled to the server 2558 2559 or device for the job so far. 2560 2561 impressionsSentToDevice(111), Integer32(-2..2147483647) INTEGER: The number of impressions sent to the device for 2562 2563 the job so far. 2564 2565 impressionsInterpreted(112), Integer32(-2..2147483647) 2566 INTEGER: The number of impressions interpreted for the job so far. 2567 2568 2569 impressionsCompletedCurrentCopy(113), Integer32(-2..2147483647) 2570 INTEGER: The number of impressions completed by the device 2571 for the current copy of the current document so far. For printing, the impressions completed includes interpreting, 2572 2573 marking, and stacking the output. For other types of job 2574 services, the number of impressions completed includes the number of impressions processed. 2575 2576 2577 This value SHALL be reset to 0 for each document in the job 2578 and for each document copy. 2579 2580 fullColorImpressionsCompleted(114), Integer32(-2..2147483647) 2581 INTEGER: The number of full color impressions completed by the device for this job so far. For printing, the 2582 impressions completed includes interpreting, marking, and 2583 2584 stacking the output. For other types of job services, the number of impressions completed includes the number of 2585 impressions processed. Full color impressions are typically 2586 2587 defined as those requiring 3 or more colorants, but this 2588 MAY vary by implementation. In any case, the value of this attribute counts by 1 for each side that has full color, 2589 not by the number of colors per side (and the other 2590 2591 impression counters are incremented, except 2592 highlightColorImpressionsCompleted(115)). 2593

Bergman, Hastings, Isaacson, LewisInformational

2595 Integer32(-221474	
	483647)
2596 INTEGER: The number of highlight color impression	ons
2597 completed by the device for this job so far. For	r printing,
2598 the impressions completed includes interpreting,	marking,
and stacking the output. For other types of job	services,
2600 the number of impressions completed includes the	number of
2601 impressions processed. Highlight color impression	
2602 typically defined as those requiring black plus of	
	In any
2604 case, the value of this attribute counts by 1 for	
2605 that has highlight color (and the other impression)	
2606 are incremented, except	on ocaneers
2607 fullColorImpressionsCompleted(114)).	
2608	
2609	
2610 ++++++++++++++++++++++++++++++++++++	
2610 + Page attributes	****
2612 +	
2612 + See the definition of 'impression', 'sheet', and 'r	nago /
2613 + See the definition of impression, sheet, and $r2614 + in Section 2.$	page
2615 ++++++++++++++++++++++++++++++++++++	****
2615	****
2617 pagesRequested(130), Integer32(-221474	102617)
2618 INTEGER: The number of logical pages requested k	
	by the job
2619 to be processed. 2620	
	102617)
2621pagesCompleted(131),Integer32(-221474)2622INTEGER: The number of logical pages completed f	
2622 INTEGER: The humber of fogical pages completed i 2623 job so far.	LOI CHIS
2623 JOD SO TAL. 2624	
2624 2625 For implementations where multiple copies are pro	odugod by
2625 For imprementations where multiple copies are pro- 2626 the interpreter with only a single pass over the	
	uala, lile
-	
2628 pagesRequested object. For implementations where	
2629 copies are produced by the interpreter by process	
2630 data for each copy, the final value SHALL be a mu	uitiple or
2631 the value of the pagesRequested object.	
2633 NOTE - See the impressionsCompletedCurrentCopy ar	
2634 pagesCompletedCurrentCopy attributes for attribut	tes that
are reset on each document copy.	
	I I
2637 NOTE - The pagesCompleted object can be used with	
2638 pagesRequested object to provide an indication of	t the
2639 relative progress of the job, provided that the	
2640 multiplicative factor is taken into account for s	some
2641 implementations of multiple copies.	
2642	

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 2643 pagesCompletedCurrentCopy(132), Integer32(-2..2147483647) INTEGER: The number of logical pages completed for the 2644 2645 current copy of the document so far. This value SHALL be reset to 0 for each document in the job and for each 2646 2647 document copy. 2648 2649 2650 2651 + Sheet attributes 2652 + + See the definition of 'impression', 'sheet', and 'page' 2653 2654 + in Section 2. 2655 2656 sheetsRequested(150), 2657 Integer32(-2..2147483647) 2658 INTEGER: The total number of medium sheets requested to be 2659 produced for this job. 2660 2661 Unlike the jmJobKOctetsPerCopyRequested and 2662 jmJobImpressionsPerCopyRequested attributes, the 2663 sheetsRequested(150) attribute SHALL include the multiplicative factor contributed by the number of copies 2664 and so is the total number of sheets to be produced by the 2665 2666 job, as opposed to the size of the document(s) submitted. 2667 2668 sheetsCompleted(151), Integer32(-2..2147483647) INTEGER: The total number of medium sheets that have 2669 2670 completed marking and stacking for the entire job so far 2671 whether those sheets have been processed on one side or on 2672 both. 2673 2674 sheetsCompletedCurrentCopy(152), Integer32(-2..2147483647) INTEGER: The number of medium sheets that have completed 2675 2676 marking and stacking for the current copy of a document in 2677 the job so far whether those sheets have been processed on 2678 one side or on both. 2679 The value of this attribute SHALL be 0 before the job 2680 2681 starts processing and SHALL be reset to 1 after the first 2682 sheet of each document and document copy in the job is processed and stacked. 2683 2684 2685

Job Monitoring MIB, V1.0

2686 2687 + Resources attributes (requested and consumed) 2688 + 2689 + Pairs of these attributes can be used by monitoring + applications to show an indication of relative usage to 2690 2691 + users. 2692 2693 2694 mediumRequested(170), JmMediumTypeTC 2695 AND/OR 2696 JmJobStringTC(SIZE(0..63)) INTEGER: MULTI-ROW: The type 2697 2698 AND/OR 2699 OCTETS: MULTI-ROW: the name of the medium that is 2700 required by the job. 2701 2702 NOTE - The name (JmJobStringTC) values correspond to the prtInputMediaName object in the Printer MIB [print-mib] and 2703 2704 the values of the IPP 'media' attribute. 2705 2706 mediumConsumed(171), Integer32(-2..2147483647) 2707 AND 2708 JmJobStringTC(SIZE(0..63)) INTEGER: MULTI-ROW: The number of sheets 2709 2710 AND 2711 OCTETS: MULTI-ROW: the name of the medium that has been 2712 consumed so far whether those sheets have been processed on 2713 one side or on both. 2714 2715 This attribute SHALL have both Integer32 and OCTET STRING 2716 (represented as JmJobStringTC) values. 2717 2718 NOTE - The name (JmJobStringTC) values correspond to the 2719 name values of the prtInputMediaName object in the Printer 2720 MIB [print-mib]. 2721 2722 colorantRequested(172), Integer32(-2..2147483647) 2723 AND/OR 2724 JmJobStringTC(SIZE(0..63)) INTEGER: MULTI-ROW: The index (prtMarkerColorantIndex) in 2725 the Printer MIB[print-mib] 2726 2727 AND/OR 2728 OCTETS: MULTI-ROW: the name of the colorant requested. 2729 2730 NOTE - The name (JmJobStringTC) values correspond to the 2731 name values of the prtMarkerColorantValue object in the 2732 Printer MIB. Examples are: red, blue.

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 2733 colorantConsumed(173), Integer32(-2..2147483647) 2734 AND/OR 2735 JmJobStringTC(SIZE(0..63)) 2736 INTEGER: MULTI-ROW: The index (prtMarkerColorantIndex) in 2737 the Printer MIB[print-mib] 2738 AND/OR 2739 OCTETS: MULTI-ROW: the name of the colorant consumed. 2740 2741 NOTE - The name (JmJobStringTC) values correspond to the 2742 name values of the prtMarkerColorantValue object in the 2743 Printer MIB. Examples are: red, blue 2744 2745 2746 2747 + Time attributes (set by server or device) 2748 + 2749 + This section of attributes are ones that are set by the 2750 + server or device that accepts jobs. Two forms of time are + provided. Each form is represented in a separate attribute. 2751 2752 + See section 3.1.2 and section 3.1.3 for the 2753 + conformance requirements for time attribute for agents and 2754 + monitoring applications, respectively. The two forms are: 2755 + 2756 + 'DateAndTime' is an 8 or 11 octet binary encoded year, 2757 + month, day, hour, minute, second, deci-second with + optional offset from UTC. See SNMPv2-TC [SMIv2-TC]. 2758 2759 + 2760 + NOTE: 'DateAndTime' is not printable characters; it is 2761 + binary. 2762 + 2763 + 'JmTimeStampTC' is the time of day measured in the number of 2764 + seconds since the system was booted. 2765 2766 2767 jobSubmissionToServerTime(190), JmTimeStampTC 2768 AND/OR 2769 DateAndTime 2770 INTEGER: Configuration 3 only: The time 2771 AND/OR 2772 OCTETS: the date and time that the job was submitted to 2773 the server (as distinguished from the device which uses 2774 jobSubmissionTime). 2775 2776 jobSubmissionTime(191), JmTimeStampTC 2777 AND/OR 2778 DateAndTime INTEGER: Configurations 1, 2, and 3: The time 2779 2780 AND/OR 2781 OCTETS: the date and time that the job was submitted to 2782 the server or device to which the agent is providing 2783 access. 2784

Bergman, Hastings, Isaacson, LewisInformational

[Page 67]

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 2785 jobStartedBeingHeldTime(192), JmTimeStampTC 2786 AND/OR 2787 DateAndTime 2788 INTEGER: The time 2789 AND/OR 2790 OCTETS: the date and time that the job last entered the pendingHeld state. If the job has never entered the 2791 pendingHeld state, then the value SHALL be '0' or the 2792 attribute SHALL not be present in the table. 2793 2794 2795 jobStartedProcessingTime(193), JmTimeStampTC 2796 AND/OR 2797 DateAndTime 2798 INTEGER: The time 2799 AND/OR 2800 OCTETS: the date and time that the job started processing. 2801 jobCompletionTime(194), 2802 JmTimeStampTC 2803 AND/OR 2804 DateAndTime 2805 INTEGER: The time 2806 AND/OR OCTETS: the date and time that the job entered the 2807 2808 completed, canceled, or aborted state. 2809 2810 jobProcessingCPUTime(195) Integer32(-2..2147483647) 2811 UNITS 'seconds' 2812 INTEGER: The amount of CPU time in seconds that the job 2813 has been in the processing state. If the job enters the processingStopped state, that elapsed time SHALL not be 2814 2815 included. In other words, the jobProcessingCPUTime value 2816 SHOULD be relatively repeatable when the same job is 2817 processed again on the same device." 2818 2819 REFERENCE 2820 "See Section 3.2 entitled 'The Attribute Mechanism' for a description of this textual-convention and its use in the 2821 jmAttributeTable. 2822 2823 This is a type 2 enumeration. See Section 3.7.1.2." 2824 SYNTAX INTEGER { 2825 2826 other(1), 2827 -- Job State attributes: 2828 jobStateReasons2(3), 2829 2830 jobStateReasons3(4), 2831 jobStateReasons4(5), processingMessage(6),
processingMessageNaturalLangTag(7),
ishgeheldte 2832 2833 jobCodedCharSet(8), 2834 2835 jobNaturalLanguageTag(9), 2836

2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873	<pre> Job Identification attributes: jobURI(20), jobAccountName(21), serverAssignedJobName(22), jobServiceTypes(24), jobSourceChannelIndex(25), jobSourcePlatformType(26), submittingServerName(27), submittingApplicationName(28), jobOriginatingHost(29), deviceNameRequested(30), queueNameRequested(31), physicalDevice(32), numberOfDocuments(33), fileName(34), documentName(35), jobComment(36), documentFormatIndex(37), documentFormatIndex(37), documentFormat(38), Job Parameter attributes: jobPriority(50), jobProcessAfterDateAndTime(51), jobHold(52), jobHoldUntil(53), outputBin(54), sides(55), finishing(56), Image Quality attributes: printQualityRequested(70), printeResolutionRequested(72), printerResolutionUsed(73), tonerEcomonyUsed(75).</pre>
2874	tonerEcomonyUsed(75), tonerDensityRequested(76),
2875 2876	tonerDensityUsed(77),
2877 2878 2879 2880 2881 2882 2883 2883 2884 2885 2886	<pre> Job Progress attributes: jobCopiesRequested(90), jobCopiesCompleted(91), documentCopiesRequested(92), documentCopiesCompleted(93), jobKOctetsTransferred(94), sheetCompletedCopyNumber(95), sheetCompletedDocumentNumber(96), jobCollationType(97),</pre>

2887 2888 2889	Impression attributes: impressionsSpooled(110), impressionsSentToDevice(111),
2890	impressionsInterpreted(112),
2891	impressionsCompletedCurrentCopy(113),
2892	fullColorImpressionsCompleted(114),
2893	highlightColorImpressionsCompleted(115),
2894	
2895	Page attributes:
2896	pagesRequested(130),
2897	pagesCompleted(131),
2898	pagesCompletedCurrentCopy(132),
2899	
2900	Sheet attributes:
2901	<pre>sheetsRequested(150),</pre>
2902	<pre>sheetsCompleted(151),</pre>
2903	<pre>sheetsCompletedCurrentCopy(152),</pre>
2904	
2905	Resource attributes:
2906	mediumRequested(170),
2907	mediumConsumed(171),
2908	colorantRequested(172),
2909	colorantConsumed(173),
2910	mine etterilerter.
2911 2912	Time attributes:
2912	jobSubmissionToServerTime(190), jobSubmissionTime(191),
2914	jobStartedBeingHeldTime(192),
2915	jobStartedProcessingTime(192),
2916	jobCompletionTime(194),
2917	jobProcessingCPUTime(195)
2918	}
2919	J
2920	
2921	

2921 2922

2923 2924	JmJobServiceTypesTC ::= TEXTUAL-CONVENTION STATUS current
2925	DESCRIPTION
2926	"Specifies the type(s) of service to which the job has been
2927	submitted (print, fax, scan, etc.). The service type is
2928	represented as an enum that is bit encoded with each job
2929	service type so that more general and arbitrary services can be
2930 2931	created, such as services with more than one destination type, or ones with only a source or only a destination. For example,
2931	a job service might scan, faxOut, and print a single job. In
2932	this case, three bits would be set in the jobServiceTypes
2934	attribute, corresponding to the hexadecimal values: 0x8 + 0x20
2935	+ 0x4, respectively, yielding: 0x2C.
2936	· oki, respectively, relating onde.
2937	Whether this attribute is set from a job attribute supplied by
2938	the job submission client or is set by the recipient job
2939	submission server or device depends on the job submission
2940	protocol. With either implementation, the agent SHALL return a
2941	non-zero value for this attribute indicating the type of the
2942	job.
2943	
2944	One of the purposes of this attribute is to permit a requester
2945	to filter out jobs that are not of interest. For example, a
2946 2947	printer operator MAY only be interested in jobs that include printing. That is why the attribute is in the job
2947	identification category.
2949	identification category.
2950	The following service component types are defined (in
2951	hexadecimal) and are assigned a separate bit value for use with
2952	the jobServiceTypes attribute:
2953	
2954	other 0x1
2955	The job contains some instructions that are not one of the
2956	identified types.
2957	
2958 2959	unknown 0x2 The job contains some instructions whose type is unknown to
2959	the agent.
2961	che agenic.
2962	print 0x4
2963	The job contains some instructions that specify printing
2964	
2965	scan 0x8
2966	The job contains some instructions that specify scanning
2967	
2968	faxIn 0x10
2969	The job contains some instructions that specify receive fax
2970	
2971	faxOut 0x20
2972	The job contains some instructions that specify sending fax
2973	

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 2974 qetFile 0×40 2975 The job contains some instructions that specify accessing 2976 files or documents 2977 putFile 0×80 2978 2979 The job contains some instructions that specify storing 2980 files or documents 2981 2982 mailList 0x100 2983 The job contains some instructions that specify 2984 distribution of documents using an electronic mail system." 2985 REFERENCE 2986 "These bit definitions are the equivalent of a type 2 enum 2987 except that combinations of them MAY be used together. See section 3.7.1.2." 2988 2989 SYNTAX INTEGER(0..2147483647) -- 31 bits, all but sign bit 2990 2991 2992 2993 JmJobStateReasons1TC ::= TEXTUAL-CONVENTION 2994 STATUS current 2995 DESCRIPTION 2996 "The JmJobStateReasonsNTC (N=1..4) textual-conventions are used 2997 with the jmJobStateReasons1 object and jobStateReasonsN 2998 (N=2..4), respectively, to provide additional information regarding the current jmJobState object value. These values 2999 MAY be used with any job state or states for which the reason 3000 3001 makes sense. 3002 3003 NOTE - While values cannot be added to the jmJobState object 3004 without impacting deployed clients that take actions upon 3005 receiving jmJobState values, it is the intent that additional JmJobStateReasonsNTC enums can be defined and registered 3006 3007 without impacting such deployed clients. In other words, the 3008 jmJobStateReasons1 object and jobStateReasonsN attributes are 3009 intended to be extensible. 3010 NOTE - The Job Monitoring MIB contains a superset of the IPP 3011 values[ipp-model] for the IPP 'job-state-reasons' attribute, 3012 since the Job Monitoring MIB is intended to cover other job 3013 submission protocols as well. Also some of the names of the 3014 3015 reasons have been changed from 'printer' to 'device', since the 3016 Job Monitoring MIB is intended to cover additional types of devices, including input devices, such as scanners. 3017 3018 3019 The following standard values are defined (in hexadecimal) as 3020 powers of two, since multiple values MAY be used at the same 3021 time. For ease of understanding, the JmJobStateReasons1TC reasons are presented in the order in which the reasons are 3022 likely to occur (if implemented), starting with the 3023 3024 'jobIncoming' value and ending with the 3025 'jobCompletedWithErrors' value.

[Page 72]

3026	
3027	other 0x1
3028	The job state reason is not one of the standardized or
3029	registered reasons.
3030	
3031	unknown 0x2
3032	The job state reason is not known to the agent or is
3033	indeterminent.
3034	
3035	jobIncoming 0x4
3036	The job has been accepted by the server or device, but the
3037	server or device is expecting (1) additional operations
3038	from the client to finish creating the job and/or (2) is
3039	accessing/accepting document data.
	accessing/accepting document data.
3040	
3041	submissionInterrupted 0x8
3042	The job was not completely submitted for some unforeseen
3043	reason, such as: (1) the server has crashed before the job
3044	was closed by the client, (2) the server or the document
3045	transfer method has crashed in some non-recoverable way
3046	before the document data was entirely transferred to the
3047	server, (3) the client crashed or failed to close the job
3048	before the time-out period.
3049	
3050	jobOutgoing 0x10
	jobOutgoing 0x10 Configuration 2 only: The server is transmitting the job
3050	
3050 3051 3052	Configuration 2 only: The server is transmitting the job
3050 3051 3052 3053	Configuration 2 only: The server is transmitting the job to the device.
3050 3051 3052 3053 3054	Configuration 2 only: The server is transmitting the job to the device. jobHoldSpecified 0x20
3050 3051 3052 3053 3054 3055	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
3050 3051 3052 3053 3054 3055 3056	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
3050 3051 3052 3053 3054 3055 3056 3057	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
3050 3051 3052 3053 3054 3055 3056 3057 3058	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059	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.
3050 3051 3052 3053 3054 3055 3056 3057 3058	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059	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.
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3061 3062	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065	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.
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066	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
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3065	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)
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068	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.
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3065	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)
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068	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.
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3064 3065 3066 3067 3068 3069 3070	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. The job SHALL NOT be a candidate for processing until this reason is removed and there are no other reasons to hold
3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3062 3063 3064 3065 3066 3067 3068 3069	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. The job SHALL NOT be a candidate for processing until this

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

3073 3074 3075 3076 3077 3078 3079 3080	resourcesAreNotReady 0x100 At least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not ready on any of the physical devices for which the job is a candidate. This condition MAY be detected when the job is accepted, or subsequently while the job is pending or processing, depending on implementation.
3081 3082 3083 3084 3085 3086	deviceStoppedPartly 0x200 One or more, but not all, of the devices to which the job is assigned are stopped. If all of the devices are stopped (or the only device is stopped), the deviceStopped reason SHALL be used.
3087 3088 3089 3090	deviceStopped 0x400 The device(s) to which the job is assigned is (are all) stopped.
3091 3092 3093 3094	jobInterpreting 0x800 The device to which the job is assigned is interpreting the document data.
3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105	jobPrinting 0x1000 The output device to which the job is assigned is marking media. This attribute is useful for servers and output devices which spend a great deal of time processing (1) when no marking is happening and then want to show that marking is now happening or (2) when the job is in the process of being canceled or aborted while the job remains in the processing state, but the marking has not yet stopped so that impression or sheet counts are still increasing for the job.
3106 3107 3108 3109 3110 3111	jobCanceledByUser 0x2000 The job was canceled by the owner of the job, i.e., by a user whose name is the same as the value of the job's jmJobOwner object, or by some other authorized end-user, such as a member of the job owner's security group.
3112 3113 3114 3115	jobCanceledByOperator 0x4000 The job was canceled by the operator, i.e., by a user who has been authenticated as having operator privileges (whether local or remote).
3116 3117 3118 3119 3120	jobCanceledAtDevice 0x8000 The job was canceled by an unidentified local user, i.e., a user at a console at the device.

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

3121 3122 3123 3124 3125 3126 3127	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.
3128 3129 3130 3131 3132 3133 3134	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.
3135 3136 3137 3138 3139 3140 3141	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 resources consumed counters have stopped incrementing, the server/device moves the job from the processing state to the canceled or aborted job states.
3142 3143 3144 3145 3146 3147 3148	serviceOffLine 0x40000 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.
3140 3149 3150 3151	jobCompletedSuccessfully 0x80000 The job completed successfully.
3152 3153 3154	jobCompletedWithWarnings 0x100000 The job completed with warnings.
3155 3156 3157	jobCompletedWithErrors 0x200000 The job completed with errors (and possibly warnings too).
3158 3159 3160 3161 3162	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:
3162 3163 3164 3165 3166 3167 3168 3169 3170 3171	jobPaused 0x400000 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.

	INTERNET-DRAFT	Job Monitoring MIB	, V1.0	January 1998
3172 3173 3174 3175 3176 3177 3178	issuing instead automati	ed has been interrupted w an operation that spec of the current job. T cally resume the inter ting job completes.	cifies another The server or de	job to be run evice will
3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189	of the je as fonts issue an re-do the device of When a c as after SHALL re jmJobSta	is being retained by to bb's document data (ar logos, and forms, if operation to the serve job (or a copy of the r (2) resubmit the job lient could no longer the document data has move the jobRetained w teReasons1 object."	nd submitted rea f any). Thus a ver or device to he job) on the s b to another set re-do/resubmit s been discardeo	sources, such client could o either (1) same server or rver or device. the job, such
3190 3191 3192 3193 3194 3195 3196 3197	except that section 3.7. standardizat	efinitions are the equ combinations of bits r 1.2. The remaining b ion and/or registratio GER(02147483647)	may be used toge its are reserved on."	ether. See d for future
3198 3199 3200 3201 3202 3203 3204 3205 3206 3207	STATUS curr DESCRIPTION "This textua attribute to jmJobState oi	::= TEXTUAL-CONVENTION ent l-convention is used w provides additional is oject. See the descriptions	with the jobStat information rega iption under	arding the
3208 3209 3210 3211 3212 3213 3214 3215 3216	powers of tw time: cascaded An outbo	g standard values are b, since multiple valu und gateway has transm ment attributes and da	ues may be used 0x1 mitted all of th	at the same he job's job
3210 3217 3218 3219 3220 3221 3222 3223	discardTimeA The job	histrator has deleted	0x4 to the fact that	

[Page 76]

3224	
3225	postProcessingFailed 0x8
3226	The post-processing agent failed while trying to log
3227	accounting attributes for the job; therefore the job has
3228	been placed into the completed state with the jobRetained
3229	jmJobStateReasons1 object value for a system-defined period
3230	of time, so the administrator can examine it, resubmit it,
3231	etc.
3232	
3233	jobTransforming 0x10
	J · · · · · · · · · · · · · · ·
3234	The server/device is interpreting document data and
3235	producing another electronic representation.
3236	
3237	maxJobFaultCountExceeded 0x20
3238	The job has faulted several times and has exceeded the
3239	administratively defined fault count limit.
3240	
3241	devicesNeedAttentionTimeOut 0x40
3242	One or more document transforms that the job is using needs
3243	human intervention in order for the job to make progress,
3244	but the human intervention did not occur within the site-
3245	settable time-out value.
3245	Sectable time-out value.
3247	needsKeyOperatorTimeOut 0x80
3248	One or more devices or document transforms that the job is
3249	using need a specially trained operator (who may need a key
3250	to unlock the device and gain access) in order for the job
3251	to make progress, but the key operator intervention did not
3252	occur within the site-settable time-out value.
3253	
3254	jobStartWaitTimeOut 0x100
3255	The server/device has stopped the job at the beginning of
3256	processing to await human action, such as installing a
3257	special cartridge or special non-standard media, but the
3258	job was not resumed within the site-settable time-out value
3259	and the server/device has transitioned the job to the
3260	pendingHeld state.
3261	penamynera scace.
3262	jobEndWaitTimeOut 0x200
3263	The server/device has stopped the job at the end of
3264	processing to await human action, such as removing a
3265	special cartridge or restoring standard media, but the job
3266	was not resumed within the site-settable time-out value and
3267	the server/device has transitioned the job to the completed
3268	state.
3269	
3270	jobPasswordWaitTimeOut 0x400
3271	The server/device has stopped the job at the beginning of
3272	processing to await input of the job's password, but the
3273	password was not received within the site-settable time-out
3274	value.
3275	varac.

	INTERNET-DRAFT	Job Monitoring MIB, V1.0	January 1998
3276 3277 3278 3279		ut 0x800 that the job was using has not pecified by the device's site-se	
3280 3281 3282 3283 3284 3285 3286	The serv which ma with tra	DeviceTimeOut 0x1000 er is attempting to connect to o y be dial-up, polled, or queued, ffic from other systems, but ser to the device within the site-se	and so may be busy ver was unable to
3280 3287 3288 3289 3290		0x2000 is being transferred to a down s am device.	tream server or
3291 3292 3293 3294		ce 0x4000 er/device has queued the job in r downstream device.	a down stream
3295 3296 3297	jobQueued The serv	0x8000 er/device has queued the documen	t data.
3298 3299 3300 3301		0x10000 er/device is performing cleanup ormal processing.	activity as part of
3302 3303 3304 3305 3306 3307 3308	process, the job job to t	ait 0x20000 er/device has selected the job t but instead of assigning resour processing, the server/device ha he pendingHeld state to await en patched another job, if there is	ces and starting s transitioned the try of a password
3309 3310 3311 3312	validating The serv job.	0x40000 er/device is validating the job	after accepting the
3313 3314 3315	queueHeld The oper	0x80000 ator has held the entire job set	or queue.
3316 3317 3318 3319 3320 3321 3322	pendingH operatio	0x100000 has produced a single proof copy eld state waiting for the reques n to release the job to print no document copy attributes that we d.	ter to issue an ormally, obeying any
3323 3324 3325		ostics 0x200000 em is running intrusive diagnost being held.	ics, so that all

[Page 78]

3326	noSpaceOnServer 0x800000
3327	There is no room on the server to store all of the job.
3328	
3329	pinRequired 0x1000000
3330	The System Administrator settable device policy is (1) to
3331	require PINs, and (2) to hold jobs that do not have a pin
3332	supplied as an input parameter when the job was created.
3333	
3334	exceededAccountLimit 0x200000
3335	The account for which this job is drawn has exceeded its
3336	limit. This condition SHOULD be detected before the job is
3337	scheduled so that the user does not wait until his/her job
3338	is scheduled only to find that the account is overdrawn.
3339	This condition MAY also occur while the job is processing
3340	either as processing begins or part way through processing.
3341	eromer as processing segras or pare "a, enroagh processing.
3342	heldForRetry 0x4000000
3343	The job encountered some errors that the server/device
3344	could not recover from with its normal retry procedures,
3345	but the error might not be encountered if the job is
3346	processed again in the future. Example cases are phone
3347	number busy or remote file system in-accessible. For such
3348	a situation, the server/device SHALL transition the job
3349	from the processing to the pendingHeld, rather than to the
3350	aborted state.
3351	
3352	The following values are from the X/Open PSIS draft standard:
3353	
3354	canceledByShutdown 0x8000000
3355	The job was canceled because the server or device was
3356	shutdown before completing the job.
3357	
3358	deviceUnavailable 0x1000000
3359	This job was aborted by the system because the device is
3360	currently unable to accept jobs.
3361	
3362	wrongDevice 0x2000000
3363	This job was aborted by the system because the device is
3364	unable to handle this particular job; the spooler SHOULD
3365	try another device or the user should submit the job to
3366	another device.
3367 3368	badJob 0x4000000
3369	
3370	This job was aborted by the system because this job has a major problem, such as an ill-formed PDL; the spooler
3371	SHOULD not even try another device. "
3372	REFERENCE
3373	"These bit definitions are the equivalent of a type 2 enum
3374	except that combinations of them may be used together. See
3375	section 3.7.1.2. See the description under
3376	JmJobStateReasons1TC and the jobStateReasons2 attribute."
3377	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit

Bergman, Hastings, Isaacson, LewisInformational

[Page 79]

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

3378	
3379	JmJobStateReasons3TC ::= TEXTUAL-CONVENTION
3380	STATUS current
3381	DESCRIPTION
3382	"This textual-convention is used with the jobStateReasons3
3383	attribute to provides additional information regarding the
3384	jmJobState object. See the description under
3385	JmJobStateReasons1TC for additional information that applies to
3386	all reasons.
3387	
3388	The following standard values are defined (in hexadecimal) as
3389	powers of two, since multiple values may be used at the same
3390	time:
3391	
3392	jobInterruptedByDeviceFailure 0x1
3393	A device or the print system software that the job was
3394	using has failed while the job was processing. The server
3395	or device is keeping the job in the pendingHeld state until
3396	an operator can determine what to do with the job."
3397	REFERENCE
3398	"These bit definitions are the equivalent of a type 2 enum
3399	except that combinations of them may be used together. See
3400	section 3.7.1.2. The remaining bits are reserved for future
3401	standardization and/or registration. See the description under
3402	
	JmJobStateReasons1TC and the jobStateReasons3 attribute."
3403	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2424	
3404	
3405	
3405	
3405 3406	
3405 3406 3407 3408	JmJobStateReasons4TC ::= TEXTUAL-CONVENTION
3405 3406 3407 3408 3409	JmJobStateReasons4TC ::= TEXTUAL-CONVENTION
3405 3406 3407 3408 3409 3410	STATUS current
3405 3406 3407 3408 3409 3410 3411	STATUS current DESCRIPTION
3405 3406 3407 3408 3409 3410 3411 3412	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4
3405 3406 3407 3408 3409 3410 3411 3412 3413	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under JmJobStateReasons1TC for additional information that applies to
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under JmJobStateReasons1TC for additional information that applies to all reasons.
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under JmJobStateReasons1TC for additional information that applies to
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under JmJobStateReasons1TC for additional information that applies to all reasons. The following standard values are defined (in hexadecimal) as</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418	STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See the description under JmJobStateReasons1TC for additional information that applies to all reasons.
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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:</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3416 3417 3418 3419 3420 3421 3422 3423	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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."</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3415 3416 3417 3418 3419 3420 3421 3422 3423 3424	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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." REFERENCE</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3412 3413 3414 3415 3416 3417 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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." REFERENCE "These bit definitions are the equivalent of a type 2 enum</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3412 3413 3414 3415 3416 3417 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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." REFERENCE "These bit definitions are the equivalent of a type 2 enum except that combinations of them may be used together. See </pre>
3405 3406 3407 3408 3409 3410 3411 3412 3412 3413 3414 3415 3416 3417 3416 3417 3418 3419 3420 3421 3421 3422 3423 3424 3425 3426 3427	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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." REFERENCE "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. See the description under </pre>
3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3414 3415 3416 3417 3418 3419 3420 3421 3422 3421 3422 3422 3424 3425 3426 3427 3428	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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." REFERENCE "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. See the description under JmJobStateReasons1TC and the jobStateReasons4 attribute."</pre>
3405 3406 3407 3408 3409 3410 3411 3412 3412 3413 3414 3415 3416 3417 3416 3417 3418 3419 3420 3421 3421 3422 3423 3424 3425 3426 3427	<pre>STATUS current DESCRIPTION "This textual-convention is used in the jobStateReasons4 attribute to provides additional information regarding the jmJobState object. See 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." REFERENCE "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. See the description under </pre>

Bergman, Hastings, Isaacson, LewisInformational

```
INTERNET-DRAFT
```

Job Monitoring MIB, V1.0 January 1998

```
3430
      jobmonMIBObjects OBJECT IDENTIFIER ::= { jobmonMIB 1 }
3431
3432
3433
      -- The General Group (MANDATORY)
3434
3435
      -- The jmGeneralGroup consists entirely of the jmGeneralTable.
3436
      jmGeneral OBJECT IDENTIFIER ::= { jobmonMIBObjects 1 }
3437
3438
3439
      jmGeneralTable OBJECT-TYPE
3440
                      SEQUENCE OF JmGeneralEntry
          SYNTAX
3441
          MAX-ACCESS not-accessible
3442
          STATUS
                      current
3443
          DESCRIPTION
3444
              "The jmGeneralTable consists of information of a general nature
3445
              that are per-job-set, but are not per-job. See Section 2
              entitled 'Terminology and Job Model' for the definition of a
3446
3447
              job set."
3448
          REFERENCE
3449
              "The MANDATORY-GROUP macro specifies that this group is
3450
              MANDATORY."
3451
          ::= \{ jmGeneral 1 \}
3452
3453
3454
      jmGeneralEntry OBJECT-TYPE
3455
          SYNTAX
                      JmGeneralEntry
3456
          MAX-ACCESS not-accessible
3457
          STATUS
                      current
3458
          DESCRIPTION
3459
              "Information about a job set (queue).
3460
3461
              An entry SHALL exist in this table for each job set."
3462
          INDEX { jmGeneralJobSetIndex }
          ::= { jmGeneralTable 1 }
3463
3464
3465
      JmGeneralEntry ::= SEQUENCE {
3466
3467
          jmGeneralJobSetIndex
                                                Integer32(1..32767),
3468
          imGeneralNumberOfActiveJobs
                                                Integer32(0..2147483647),
3469
          jmGeneralOldestActiveJobIndex
                                                Integer32(0..2147483647),
          jmGeneralNewestActiveJobIndex
3470
                                                Integer32(0..2147483647),
3471
          jmGeneralJobPersistence
                                                Integer32(15..2147483647),
3472
          jmGeneralAttributePersistence
                                               Integer32(15..2147483647),
3473
          jmGeneralJobSetName
                                                JmUTF8StringTC(SIZE(0..63))
3474
      }
3475
```

	INTERNET-DRAFT	Job Monitoring MIB, V1.0	January 1998
3476 3477 3478 3479 3480 3481	MAX-ACCESS not-ac STATUS curren DESCRIPTION	er32(132767) cessible	. The jmJobTable
3482 3483 3484	and jmAttribut primary index.	eTable tables have this same in	ndex as their
3485 3486 3487 3488 3489	across power c	of the jmGeneralJobSetIndex SHAI cycles, so that clients that hav etIndex values will access the s ver-up.	ve retained
3490 3491 3492	with a single 1."	ion that has only one job set, queue, SHALL hard code this ob	
3493 3494 3495 3496	definition of	entitled 'Terminology and Job a job set. the first index in jmJobTable	
3497 3498 3499 3500	jmAttributeTab ::= { jmGeneralEnt		
3501 3502 3503 3504 3505	jmGeneralNumberOfActiv SYNTAX Intege MAX-ACCESS read-c STATUS curren DESCRIPTION	er32(02147483647) only	
3503 3506 3507 3508 3509 3510 3511 3512 3513	"The current n jmJobTable, an jobs that are states. See t	number of 'active' jobs in the particular in the pending, processing, or the JmJobStateTC textual-convent of the semantics of the job state no jobs	total number of processingStopped tion for the exact

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 3514 jmGeneralOldestActiveJobIndex OBJECT-TYPE 3515 SYNTAX Integer32 (0..2147483647) MAX-ACCESS read-only 3516 3517 STATUS current 3518 DESCRIPTION 3519 "The jmJobIndex of the oldest job that is still in one of the 3520 'active' states (pending, processing, or processingStopped). In other words, the index of the 'active' job that has been in 3521 3522 the job tables the longest. 3523 3524 If there are no active jobs, the agent SHALL set the value of 3525 this object to 0." 3526 REFERENCE "See Section 3.2 entitled 'The Job Tables and the Oldest Active 3527 and Newest Active Indexes' for a description of the usage of 3528 3529 this object." DEFVAL { 0 } -- no active jobs 3530 3531 ::= { jmGeneralEntry 3 } 3532 3533 3534 jmGeneralNewestActiveJobIndex OBJECT-TYPE 3535 SYNTAX Integer32 (0..2147483647) 3536 3537 MAX-ACCESS read-only 3538 STATUS current 3539 DESCRIPTION 3540 "The jmJobIndex of the newest job that is in one of the 3541 'active' states (pending, processing, or processingStopped). In other words, the index of the 'active' job that has been 3542 3543 most recently added to the job tables. 3544 3545 When all jobs become 'inactive', i.e., enter the pendingHeld, completed, canceled, or aborted states, the agent SHALL set the 3546 3547 value of this object to 0." 3548 REFERENCE 3549 "See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes' for a description of the usage of 3550 3551 this object." DEFVAL { 0 } -- no active jobs 3552 3553 ::= { jmGeneralEntry 4 } 3554

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 3555 jmGeneralJobPersistence OBJECT-TYPE 3556 SYNTAX Integer32(15..2147483647) 3557 UNTTS "seconds" 3558 MAX-ACCESS read-only current 3559 STATUS 3560 DESCRIPTION 3561 "The minimum time in seconds for this instance of the Job Set that an entry SHALL remain in the jmJobIDTable and jmJobTable 3562 3563 after processing has *completed*, i.e., the minimum time in 3564 seconds starting when the job enters the completed, canceled, 3565 or aborted state. 3566 3567 Configuring this object is implementation-dependent. 3568 3569 This value SHALL be equal to or greater than the value of 3570 jmGeneralAttributePersistence. This value SHOULD be at least 3571 60 which gives a monitoring application one minute in which to poll for job data." VAL { 60 } 3572 3573 DEFVAL -- one minute 3574 ::= { jmGeneralEntry 5 } 3575 3576 3577 3578 jmGeneralAttributePersistence OBJECT-TYPE 3579 SYNTAX Integer32(15..2147483647) 3580 UNITS "seconds" 3581 MAX-ACCESS read-only 3582 STATUS current 3583 DESCRIPTION 3584 "The minimum time in seconds for this instance of the Job Set 3585 that an entry SHALL remain in the jmAttributeTable after 3586 processing has *completed*, i.e., the time in seconds starting when the job enters the completed, canceled, or aborted state. 3587 3588 3589 Configuring this object is implementation-dependent. 3590 This value SHOULD be at least 60 which gives a monitoring 3591 application one minute in which to poll for job data." 3592 3593 -- one minute $DEFVAL { 60 }$::= { jmGeneralEntry 6 } 3594 3595

Job Monitoring MIB, V1.0

3596 jmGeneralJobSetName OBJECT-TYPE 3597 SYNTAX JmUTF8StringTC(SIZE(0..63)) MAX-ACCESS read-only 3598 3599 STATUS current 3600 DESCRIPTION 3601 "The human readable name of this job set assigned by the system 3602 administrator (by means outside of this MIB). Typically, this name SHOULD be the name of the job queue. If a server or 3603 device has only a single job set, this object can be the 3604 administratively assigned name of the server or device itself. 3605 3606 This name does not need to be unique, though each job set in a 3607 single Job Monitoring MIB SHOULD have distinct names. 3608 NOTE - If the job set corresponds to a single printer and the 3609 Printer MIB is implemented, this value SHOULD be the same as 3610 the prtGeneralPrinterName object in the draft Printer MIB 3611 3612 [print-mib-draft]. If the job set corresponds to an IPP Printer, this value SHOULD be the same as the IPP 'printer-3613 name' Printer attribute. 3614 3615 3616 NOTE - The purpose of this object is to help the user of the job monitoring application distinguish between several job sets 3617 in implementations that support more than one job set." 3618 3619 REFERENCE 3620 "See the OBJECT compliance macro for the minimum maximum length 3621 required for conformance." DEFVAL { ''H } -- empty string 3622 ::= { jmGeneralEntry 7 } 3623 3624 3625 3626 3627 3628

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 3629 -- The Job ID Group (MANDATORY) 3630 3631 -- The jmJobIDGroup consists entirely of the jmJobIDTable. 3632 jmJobID OBJECT IDENTIFIER ::= { jobmonMIBObjects 2 } 3633 3634 3635 jmJobIDTable OBJECT-TYPE 3636 SYNTAX SEQUENCE OF JmJobIDEntry 3637 MAX-ACCESS not-accessible 3638 STATUS current 3639 DESCRIPTION 3640 "The jmJobIDTable provides a correspondence map (1) between the 3641 job submission ID that a client uses to refer to a job and (2) 3642 the jmGeneralJobSetIndex and jmJobIndex that the Job Monitoring MIB agent assigned to the job and that are used to access the 3643 3644 job in all of the other tables in the MIB. If a monitoring 3645 application already knows the jmGeneralJobSetIndex and the 3646 jmJobIndex of the job it is querying, that application NEED NOT use the jmJobIDTable." 3647 REFERENCE 3648 3649 "The MANDATORY-GROUP macro specifies that this group is 3650 MANDATORY." 3651 $::= \{ jmJobID 1 \}$ 3652 3653 3654 3655 jmJobIDEntry OBJECT-TYPE 3656 SYNTAX JmJobIDEntry MAX-ACCESS not-accessible 3657 3658 STATUS current 3659 DESCRIPTION 3660 "The map from (1) the jmJobSubmissionID to (2) the 3661 jmGeneralJobSetIndex and jmJobIndex. 3662 3663 An entry SHALL exist in this table for each job currently known 3664 to the agent for all job sets and job states. There MAY be more than one jmJobIDEntry that maps to a single job. This 3665 many to one mapping can occur when more than one network entity 3666 3667 along the job submission path supplies a job submission ID. 3668 See Section 3.5. However, each job SHALL appear once and in one and only one job set." 3669 INDEX { jmJobSubmissionID } 3670 3671 ::= { jmJobIDTable 1 } 3672 3673 JmJobIDEntry ::= SEQUENCE { 3674 jmJobSubmissionID OCTET STRING(SIZE(48)), 3675 jmJobIDJobSetIndex Integer32(0..32767), 3676 Integer32(0..2147483647) jmJobIDJobIndex 3677 } 3678

3679	jmJobSubmissionID OBJECT-TYPE
3680	SYNTAX OCTET STRING(SIZE(48))
3681	MAX-ACCESS not-accessible
3682	STATUS current
3683	DESCRIPTION
3684	"A quasi-unique 48-octet fixed-length string ID which
3685	identifies the job within a particular client-server
3686	environment. There are multiple formats for the
3687	jmJobSubmissionID. Each format SHALL be uniquely identified.
3688	See the JmJobSubmissionIDTypeTC textual convention. Each
3689	format SHALL be registered using the procedures of a type 2
3690	enum. See section 3.7.3 entitled: 'PWG Registration of Job
3691	Submission Id Formats'.
3692	
3693	If the requester (client or server) does not supply a job
3694	submission ID in the job submission protocol, then the
3695	recipient (server or device) SHALL assign a job submission ID
3696	using any of the standard formats that have been reserved for
3697	agents and adding the final 8 octets to distinguish the ID from
3698	others submitted from the same requester.
3699	
3700	The monitoring application, whether in the client or running
3701	separately, MAY use the job submission ID to help identify
3702	which jmJobIndex was assigned by the agent, i.e., in which row
3703	the job information is in the other tables.
3704	
3705	NOTE - fixed-length is used so that a management application
3706	can use a shortened GetNext varbind (in SNMPv1 and SNMPv2) in
3707	order to get the next submission ID, disregarding the remainder
3708	of the ID in order to access jobs independent of the trailing
3709	identifier part, e.g., to get all jobs submitted by a
3710	particular jmJobOwner or submitted from a particular MAC
3711	address."
3712	REFERENCE
3713	"See the JmJobSubmissionIDTypeTC textual convention.
3714	See APPENDIX B - Support of Job Submission Protocols."
3715	::= { jmJobIDEntry 1 }
3716	

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 3717 jmJobIDJobSetIndex OBJECT-TYPE 3718 SYNTAX Integer32(0..32767) MAX-ACCESS read-only 3719 3720 STATUS current 3721 DESCRIPTION 3722 "This object contains the value of the jmGeneralJobSetIndex for 3723 the job with the jmJobSubmissionID value, i.e., the job set index of the job set in which the job was placed when that 3724 3725 server or device accepted the job. This 16-bit value in 3726 combination with the jmJobIDJobIndex value permits the 3727 management application to access the other tables to obtain the 3728 job-specific objects for this job." 3729 REFERENCE 3730 "See jmGeneralJobSetIndex in the jmGeneralTable." DEFVAL $\{0\}$ -- 0 indicates no job set index 3731 3732 ::= { jmJobIDEntry 2 } 3733 3734 3735 3736 jmJobIDJobIndex OBJECT-TYPE 3737 SYNTAX Integer32(0..2147483647) MAX-ACCESS read-only 3738 3739 STATUS current 3740 DESCRIPTION 3741 "This object contains the value of the jmJobIndex for the job 3742 with the jmJobSubmissionID value, i.e., the job index for the job when the server or device accepted the job. This value, in 3743 combination with the jmJobIDJobSetIndex value, permits the 3744 3745 management application to access the other tables to obtain the 3746 job-specific objects for this job." 3747 REFERENCE 3748 "See jmJobIndex in the jmJobTable." DEFVAL { 0 } -- 0 indicates no jmJobIndex value. 3749 3750 ::= { jmJobIDEntry 3 } 3751 3752 3753 3754

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 3755 -- The Job Group (MANDATORY) 3756 3757 -- The jmJobGroup consists entirely of the jmJobTable. 3758 jmJob OBJECT IDENTIFIER ::= { jobmonMIBObjects 3 } 3759 3760 3761 jmJobTable OBJECT-TYPE 3762 SYNTAX SEQUENCE OF JmJobEntry 3763 MAX-ACCESS not-accessible 3764 STATUS current 3765 DESCRIPTION 3766 "The jmJobTable consists of basic job state and status information for each job in a job set that (1) monitoring 3767 applications need to be able to access in a single SNMP Get 3768 operation, (2) that have a single value per job, and (3) that 3769 3770 SHALL always be implemented." 3771 REFERENCE 3772 "The MANDATORY-GROUP macro specifies that this group is 3773 MANDATORY." ::= { jmJob 1 } 3774 3775 3776 3777 3778 jmJobEntry OBJECT-TYPE SYNTAX 3779 JmJobEntry 3780 MAX-ACCESS not-accessible 3781 STATUS current 3782 DESCRIPTION 3783 "Basic per-job state and status information. 3784 3785 An entry SHALL exist in this table for each job, no matter what 3786 the state of the job is. Each job SHALL appear in one and only 3787 one job set." 3788 REFERENCE 3789 "See Section 3.2 entitled 'The Job Tables'." 3790 INDEX { jmGeneralJobSetIndex, jmJobIndex } ::= { jmJobTable 1 } 3791 3792 3793 JmJobEntry ::= SEQUENCE { 3794 jmJobIndex Integer32(1..2147483647), jmJobState 3795 JmJobStateTC, 3796 jmJobStateReasons1 JmJobStateReasons1TC, 3797 jmNumberOfInterveningJobs Integer32(-2..2147483647), Integer32(-2..2147483647), 3798 jmJobKOctetsPerCopyRequested 3799 jmJobKOctetsProcessed Integer32(-2..2147483647), jmJobImpressionsPerCopyRequested 3800 Integer32(-2..2147483647), 3801 jmJobImpressionsCompleted Integer32(-2..2147483647), JmJobStringTC(SIZE(0..63)) 3802 jmJobOwner 3803 } 3804

3805	jmJobIndex OBJECT-TYPE
3806	SYNTAX Integer32(12147483647)
3807	MAX-ACCESS not-accessible
3808	STATUS current
3809	DESCRIPTION
3810	"The sequential, monatonically increasing identifier index for
3811	the job generated by the server or device when that server or
3812	device accepted the job. This index value permits the
3813	management application to access the other tables to obtain the
3814	job-specific row entries."
3815	REFERENCE
3816	"See Section 3.2 entitled 'The Job Tables and the Oldest Active
3817	and Newest Active Indexes'.
3818	See Section 3.5 entitled 'Job Identification'.
3819	See also
3820	
3821	jmGeneralNewestActiveJobIndex for the largest value of
3822	jmJobIndex.
3823	See JmJobSubmissionIDTypeTC for a limit on the size of this
3824	index if the agent represents it as an 8-digit decimal number."
3825	::= { jmJobEntry 1 }
3826	
3827	
3828	
3829	jmJobState OBJECT-TYPE
3830	SYNTAX JmJobStateTC
3831	MAX-ACCESS read-only
3832	STATUS current
3833	DESCRIPTION
3834	"The current state of the job (pending, processing, completed,
3835	etc.). Agents SHALL implement only those states which are
3836	appropriate for the particular implementation. However,
3837	management applications SHALL be prepared to receive all the
3838	standard job states.
3839	
3840	The final value for this object SHALL be one of: completed,
3841	canceled, or aborted. The minimum length of time that the
3842	agent SHALL maintain MIB data for a job in the completed,
3843	canceled, or aborted state before removing the job data from
3844	the jmJobIDTable and jmJobTable is specified by the value of
3845	the jmGeneralJobPersistence object."
3846	DEFVAL { unknown } default is unknown
3847	::= { jmJobEntry 2 }
3848	

```
3849
      jmJobStateReasons1 OBJECT-TYPE
3850
          SYNTAX JmJobStateReasons1TC
3851
          MAX-ACCESS read-only
3852
          STATUS
                     current
3853
          DESCRIPTION
3854
              "Additional information about the job's current state, i.e.,
3855
              information that augments the value of the job's jmJobState
3856
              object.
3857
3858
              Implementation of any reason values is OPTIONAL, but an agent
3859
              SHOULD return any reason information available. These values
              MAY be used with any job state or states for which the reason
3860
              makes sense. Since the Job State Reasons will be more dynamic
3861
3862
             than the Job State, it is recommended that a job monitoring
              application read this object every time jmJobState is read.
3863
3864
              When the agent cannot provide a reason for the current state of
              the job, the value of the jmJobStateReasons1 object and
3865
3866
              jobStateReasonsN attributes SHALL be 0."
3867
          REFERENCE
3868
              "The jobStateReasonsN (N=2..4) attributes provide further
3869
              additional information about the job's current state."
3870
          DEFVAL { 0 } -- no reasons
3871
          ::= { jmJobEntry 3 }
3872
3873
3874
3875
      jmNumberOfInterveningJobs OBJECT-TYPE
3876
          SYNTAX Integer32(-2..2147483647)
3877
          MAX-ACCESS read-only
3878
          STATUS
                      current
3879
          DESCRIPTION
3880
              "The number of jobs that are expected to complete processing
              before this job has completed processing according to the
3881
              implementation's queuing algorithm, if no other jobs were to be
3882
3883
              submitted. In other words, this value is the job's queue
3884
              position. The agent SHALL return a value of 0 for this
              attribute when the job is the next job to complete processing
3885
              (or has completed processing)."
3886
          DEFVAL { 0 } -- default is no intervening jobs.
3887
          ::= { jmJobEntry 4 }
3888
3889
```

3890 jmJobKOctetsPerCopyRequested OBJECT-TYPE 3891 SYNTAX Integer32(-2..2147483647) 3892 MAX-ACCESS read-only 3893 STATUS current 3894 DESCRIPTION 3895 "The total size in K (1024) octets of the document(s) being 3896 requested to be processed in the job. The agent SHALL round the actual number of octets up to the next highest K. Thus 0 3897 octets SHALL be represented as '0', 1-1024 octets SHALL be 3898 3899 represented as '1', 1025-2048 SHALL be represented as '2', etc. 3900 3901 In computing this value, the server/device SHALL not include 3902 the multiplicative factors contributed by (1) the number of 3903 document copies, and (2) the number of job copies, independent of whether the device can process multiple copies of the job or 3904 3905 document without making multiple passes over the job or document data and independent of whether the output is collated 3906 or not. Thus the server/device computation is independent of 3907 the implementation and indicates the size of the document(s) 3908 3909 measured in K octets independent of the number of copies." 3910 DEFVAL $\{-2\}$ -- the default is unknown(-2) ::= { jmJobEntry 5 } 3911 3912 3913 3914 3915 jmJobKOctetsProcessed OBJECT-TYPE 3916 SYNTAX Integer32(-2..2147483647) 3917 MAX-ACCESS read-only 3918 STATUS current 3919 DESCRIPTION 3920 "The total number of octets processed by the server or device measured in units of K (1024) octets so far. The agent SHALL 3921 round the actual number of octets processed up to the next 3922 3923 higher K. Thus 0 octets SHALL be represented as '0', 1-1024 octets SHALL be represented as '1', 1025-2048 octets SHALL be 3924 3925 '2', etc. For printing devices, this value is the number interpreted by the page description language interpreter rather 3926 than what has been marked on media. 3927 3928 3929 For implementations where multiple copies are produced by the 3930 interpreter with only a single pass over the data, the final 3931 value SHALL be equal to the value of the 3932 jmJobKOctetsPerCopyRequested object. For implementations where multiple copies are produced by the interpreter by processing 3933 the data for each copy, the final value SHALL be a multiple of 3934 3935 the value of the jmJobKOctetsPerCopyRequested object. 3936 3937 NOTE - See the impressionsCompletedCurrentCopy and pagesCompletedCurrentCopy attributes for attributes that are 3938 3939 reset on each document copy. 3940

```
Job Monitoring MIB, V1.0
      INTERNET-DRAFT
                                                                   January 1998
3941
              NOTE - The jmJobKOctetsProcessed object can be used with the
              jmJobKOctetsPerCopyRequested object to provide an indication of
3942
3943
              the relative progress of the job, provided that the
              multiplicative factor is taken into account for some
3944
              implementations of multiple copies."
3945
3946
                                -- default is no octets processed.
          DEFVAL
                      { 0 }
3947
          ::= { jmJobEntry 6 }
3948
3949
3950
      jmJobImpressionsPerCopyRequested OBJECT-TYPE
3951
                       Integer32(-2..2147483647)
          SYNTAX
3952
          MAX-ACCESS read-only
3953
          STATUS
                       current
3954
          DESCRIPTION
               "The total size in number of impressions of the document(s)
3955
3956
              submitted.
3957
3958
              In computing this value, the server/device SHALL not include
              the multiplicative factors contributed by (1) the number of
3959
3960
              document copies, and (2) the number of job copies, independent
3961
              of whether the device can process multiple copies of the job or
              document without making multiple passes over the job or
document data and independent of whether the output is collated
3962
3963
3964
              or not. Thus the server/device computation is independent of
3965
              the implementation and reflects the size of the document(s)
3966
              measured in impressions independent of the number of copies."
3967
          REFERENCE
3968
              "See the definition of the term 'impression' in Section 2."
          DEFVAL
3969
                       \{-2\} -- default is unknown(-2)
          ::= { jmJobEntry 7 }
3970
3971
3972
3973
      jmJobImpressionsCompleted OBJECT-TYPE
3974
                       Integer32(-2..2147483647)
          SYNTAX
3975
          MAX-ACCESS read-only
3976
          STATUS
                      current
3977
          DESCRIPTION
               "The total number of impressions completed for this job so far.
3978
3979
              For printing devices, the impressions completed includes
3980
              interpreting, marking, and stacking the output. For other
              types of job services, the number of impressions completed
3981
3982
              includes the number of impressions processed.
3983
3984
              NOTE - See the impressionsCompletedCurrentCopy and
3985
              pagesCompletedCurrentCopy attributes for attributes that are
3986
              reset on each document copy.
3987
3988
              NOTE - The jmJobImpressionsCompleted object can be used with
              the jmJobImpressionsPerCopyRequested object to provide an
3989
3990
              indication of the relative progress of the job, provided that
3991
              the multiplicative factor is taken into account for some
3992
              implementations of multiple copies."
```

Bergman, Hastings, Isaacson, LewisInformational

[Page 93]

Job Monitoring MIB, V1.0 January 1998

```
3993
         REFERENCE
3994
              "See the definition of the term 'impression' in Section 2 and
3995
              the counting example in Section 3.4 entitled 'Monitoring Job
         Progress'."
DEFVAL { 0 } -- default is no octets
3996
3997
         ::= { jmJobEntry 8 }
3998
3999
4000
4001
4002
      jmJobOwner OBJECT-TYPE
4003
          SYNTAX JmJobStringTC(SIZE(0..63))
4004
          MAX-ACCESS read-only
4005
          STATUS current
4006
          DESCRIPTION
              "The coded character set name of the user that submitted the
4007
4008
              job. The method of assigning this user name will be system
4009
             and/or site specific but the method MUST insure that the name
             is unique to the network that is visible to the client and
4010
4011
             target device.
4012
4013
             This value SHOULD be the most authenticated name of the user
             submitting the job."
4014
        REFERENCE
4015
4016
              "See the OBJECT compliance macro for the minimum maximum length
4017
              required for conformance."
         DEFVAL { ''H } -- empty string
4018
         ::= { jmJobEntry 9 }
4019
4020
4021
4022
4023
```

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 4024 -- The Attribute Group (MANDATORY) 4025 4026 -- The jmAttributeGroup consists entirely of the jmAttributeTable. 4027 _ _ -- Implementation of the two objects in this group is MANDATORY. 4028 4029 -- See Section 3.1 entitled 'Conformance Considerations'. 4030 -- An agent SHALL implement any attribute if (1) the server or device -- supports the functionality represented by the attribute and (2) the 4031 4032 -- information is available to the agent. 4033 4034 jmAttribute OBJECT IDENTIFIER ::= { jobmonMIBObjects 4 } 4035 4036 4037 4038 jmAttributeTable OBJECT-TYPE 4039 SYNTAX SEQUENCE OF JmAttributeEntry 4040 MAX-ACCESS not-accessible 4041 STATUS current 4042 DESCRIPTION 4043 "The jmAttributeTable SHALL contain attributes of the job and 4044 document(s) for each job in a job set. Instead of allocating distinct objects for each attribute, each attribute is 4045 4046 represented as a separate row in the jmAttributeTable." 4047 REFERENCE 4048 "The MANDATORY-GROUP macro specifies that this group is MANDATORY. An agent SHALL implement any attribute if (1) the 4049 server or device supports the functionality represented by the 4050 4051 attribute and (2) the information is available to the agent. " 4052 ::= { jmAttribute 1 } 4053 4054 4055 4056 jmAttributeEntry OBJECT-TYPE 4057 SYNTAX JmAttributeEntry 4058 MAX-ACCESS not-accessible 4059 STATUS current 4060 DESCRIPTION 4061 "Attributes representing information about the job and 4062 document(s) or resources required and/or consumed. 4063 4064 Each entry in the jmAttributeTable is a per-job entry with an 4065 extra index for each type of attribute (jmAttributeTypeIndex) that a job can have and an additional index 4066 (jmAttributeInstanceIndex) for those attributes that can have 4067 4068 multiple instances per job. The jmAttributeTypeIndex object SHALL contain an enum type that indicates the type of attribute 4069 4070 (see the JmAttributeTypeTC textual-convention). The value of the attribute SHALL be represented in either the 4071 jmAttributeValueAsInteger or jmAttributeValueAsOctets objects, 4072 4073 and/or both, as specified in the JmAttributeTypeTC textual-4074 convention. 4075

[Page 95]

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

4076 4077 4078 4079 4080 4081 4082 4083 4083 4084 4085 4085	The agent SHALL create rows in the jmAttributeTable as the server or device is able to discover the attributes either from the job submission protocol itself or from the document PDL. As the documents are interpreted, the interpreter MAY discover additional attributes and so the agent adds additional rows to this table. As the attributes that represent resources are actually consumed, the usage counter contained in the jmAttributeValueAsInteger object is incremented according to the units indicated in the description of the JmAttributeTypeTC enum.
4087	The agent SHALL maintain each row in the jmJobTable for at
4088	least the minimum time after a job completes as specified by
4089	the jmGeneralAttributePersistence object.
4090	
4091	Zero or more entries SHALL exist in this table for each job in
4092	a job set."
4093	REFERENCE
4094	"See Section 3.3 entitled 'The Attribute Mechanism' for a
4095	description of the jmAttributeTable."
4096	INDEX { jmGeneralJobSetIndex, jmJobIndex, jmAttributeTypeIndex,
4097	jmAttributeInstanceIndex }
4098	::= { jmAttributeTable 1 }
4099	
4100	JmAttributeEntry ::= SEQUENCE {
4101	jmAttributeTypeIndex JmAttributeTypeTC,
4102	jmAttributeInstanceIndex Integer32(132767),
4103	jmAttributeValueAsInteger Integer32(-22147483647),
4104	jmAttributeValueAsOctets OCTET STRING(SIZE(063))
4105 4106	}

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 4107 jmAttributeTypeIndex OBJECT-TYPE 4108 SYNTAX JmAttributeTypeTC 4109 MAX-ACCESS not-accessible 4110 STATUS current 4111 DESCRIPTION 4112 "The type of attribute that this row entry represents. 4113 The type MAY identify information about the job or document(s) 4114 4115 or MAY identify a resource required to process the job before 4116 the job start processing and/or consumed by the job as the job 4117 is processed. 4118 Examples of job attributes (i.e., apply to the job as a whole) 4119 4120 that have only one instance per job include: 4121 jobCopiesRequested(90), documentCopiesRequested(92), 4122 jobCopiesCompleted(91), documentCopiesCompleted(93), while 4123 examples of job attributes that may have more than one instance 4124 per job include: documentFormatIndex(37), and 4125 documentFormat(38). 4126 4127 Examples of document attributes (one instance per document) 4128 include: fileName(34), and documentName(35). 4129 4130 Examples of required and consumed resource attributes include: 4131 pagesRequested(130), mediumRequested(170), pagesCompleted(131), 4132 and mediumConsumed(171), respectively." 4133 ::= { jmAttributeEntry 1 } 4134 4135 4136 4137 jmAttributeInstanceIndex OBJECT-TYPE 4138 SYNTAX Integer32(1..32767) 4139 MAX-ACCESS not-accessible 4140 STATUS current 4141 DESCRIPTION 4142 "A running 16-bit index of the attributes of the same type for each job. For those attributes with only a single instance per 4143 job, this index value SHALL be 1. For those attributes that 4144 4145 are a single value per document, the index value SHALL be the document number, starting with 1 for the first document in the 4146 job. Jobs with only a single document SHALL use the index 4147 4148 value of 1. For those attributes that can have multiple values 4149 per job or per document, such as documentFormatIndex(37) or documentFormat(38), the index SHALL be a running index for the 4150 4151 job as a whole, starting at 1." 4152 ::= { jmAttributeEntry 2 }

4153

4154 jmAttributeValueAsInteger OBJECT-TYPE 4155 SYNTAX Integer32(-2..2147483647) 4156 MAX-ACCESS read-only 4157 STATUS current 4158 DESCRIPTION 4159 "The integer value of the attribute. The value of the attribute SHALL be represented as an integer if the enum 4160 description in the JmAttributeTypeTC textual-convention 4161 4162 definition has the tag: 'INTEGER:'. 4163 4164 Depending on the enum definition, this object value MAY be an 4165 integer, a counter, an index, or an enum, depending on the jmAttributeTypeIndex value. The units of this value are 4166 4167 specified in the enum description. 4168 4169 For those attributes that are accumulating job consumption as the job is processed as specified in the JmAttributeTypeTC 4170 4171 textual-convention, SHALL contain the final value after the job 4172 completes processing, i.e., this value SHALL indicate the total 4173 usage of this resource made by the job. 4174 4175 A monitoring application is able to copy this value to a 4176 suitable longer term storage for later processing as part of an 4177 accounting system. 4178 4179 Since the agent MAY add attributes representing resources to 4180 this table while the job is waiting to be processed or being 4181 processed, which can be a long time before any of the resources 4182 are actually used, the agent SHALL set the value of the 4183 jmAttributeValueAsInteger object to 0 for resources that the 4184 job has not yet consumed. 4185 4186 Attributes for which the concept of an integer value is meaningless, such as fileName(34), jobName, and processingMessage, do not have the 'INTEGER:' tag in the JmAttributeTypeTC definition and so an agent SHALL always 4187 4188 4189 return a value of '-1' to indicate 'other' for the value of the 4190 4191 jmAttributeValueAsInteger object for these attributes. 4192 4193 For attributes which do have the 'INTEGER:' tag in the 4194 JmAttributeTypeTC definition, if the integer value is not (yet) 4195 known, the agent either (1) SHALL not materialize the row in 4196 the jmAttributeTable until the value is known or (2) SHALL 4197 return a '-2' to represent an 'unknown' counting integer value, 4198 a '0' to represent an 'unknown' index value, and a '2' to 4199 represent an 'unknown(2)' enum value." -- default value is unknown(-2) 4200 $DEFVAL \left\{ -2 \right\}$::= { jmAttributeEntry 3 } 4201 4202

4203 jmAttributeValueAsOctets OBJECT-TYPE 4204 SYNTAX OCTET STRING(SIZE(0..63)) 4205 MAX-ACCESS read-only 4206 STATUS current 4207 DESCRIPTION "The octet string value of the attribute. The value of the 4208 attribute SHALL be represented as an OCTET STRING if the enum 4209 description in the JmAttributeTypeTC textual-convention 4210 4211 definition has the tag: 'OCTETS:'. 4212 4213 Depending on the enum definition, this object value MAY be a 4214 coded character set string (text), such as 'JmUTF8StringTC', or 4215 a binary octet string, such as 'DateAndTime'. 4216 4217 Attributes for which the concept of an octet string value is 4218 meaningless, such as pagesCompleted, do not have the tag 'OCTETS:' in the JmAttributeTypeTC definition and so the agent 4219 4220 SHALL always return a zero length string for the value of the 4221 jmAttributeValueAsOctets object. 4222 4223 For attributes which do have the 'OCTETS:' tag in the 4224 JmAttributeTypeTC definition, if the OCTET STRING value is not (yet) known, the agent either SHALL not materialize the row in 4225 the jmAttributeTable until the value is known or SHALL return a 4226 4227 zero-length string." DEFVAL $\{ ''H \}$ 4228 -- empty string ::= { jmAttributeEntry 4 } 4229 4230

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998 4231 -- Notifications and Trapping 4232 -- Reserved for the future 4233 jobmonMIBNotifications OBJECT IDENTIFIER ::= { jobmonMIB 2} 4234 4235 4236 4237 -- Conformance Information 4238 4239 jmMIBConformance OBJECT IDENTIFIER ::= { jobmonMIB 3 } 4240 4241 42.42 4243 4244 -- compliance statements 4245 jmMIBCompliance MODULE-COMPLIANCE 4246 STATUS current 4247 DESCRIPTION 4248 "The compliance statement for agents that implement the 4249 job monitoring MIB." 4250 MODULE -- this module 4251 MANDATORY-GROUPS { jmGeneralGroup, jmJobIDGroup, jmJobGroup, jmAttributeGroup } 4252 4253 4254 OBJECT jmGeneralJobSetName 4255 SYNTAX JmUTF8StringTC (SIZE(0..8)) 4256 DESCRIPTION "Only 8 octets maximum string length NEED be supported by the 4257 4258 agent." 4259 4260 OBJECT jmJobOwner 4261 JmJobStringTC (SIZE(0..16)) SYNTAX 4262 DESCRIPTION "Only 16 octets maximum string length NEED be supported by the 4263 agent." 4264 4265 4266 -- There are no CONDITIONALLY MANDATORY or OPTIONAL groups. 4267 ::= { jmMIBConformance 1 } 4268 4269

```
INTERNET-DRAFT
                            Job Monitoring MIB, V1.0 January 1998
4270
      jmMIBGroups OBJECT IDENTIFIER ::= { jmMIBConformance 2 }
4271
4272
      jmGeneralGroup OBJECT-GROUP
          OBJECTS {
4273
4274
              jmGeneralNumberOfActiveJobs, jmGeneralOldestActiveJobIndex,
              jmGeneralNewestActiveJobIndex, jmGeneralJobPersistence,
4275
              jmGeneralAttributePersistence, jmGeneralJobSetName}
4276
4277
          STATUS current
4278
          DESCRIPTION
4279
              "The general group."
4280
          ::= { jmMIBGroups 1 }
4281
4282
4283
      jmJobIDGroup OBJECT-GROUP
4284
4285
          OBJECTS {
4286
              jmJobIDJobSetIndex, jmJobIDJobIndex }
          STATUS current
4287
4288
          DESCRIPTION
4289
            "The job ID group."
4290
          ::= { jmMIBGroups 2 }
4291
4292
4293
4294
      jmJobGroup OBJECT-GROUP
4295
         OBJECTS {
              jmJobState, jmJobStateReasons1, jmNumberOfInterveningJobs,
4296
4297
              jmJobKOctetsPerCopyRequested, jmJobKOctetsProcessed,
4298
              jmJobImpressionsPerCopyRequested, jmJobImpressionsCompleted,
4299
              jmJobOwner }
4300
         STATUS current
4301
        DESCRIPTION
4302
             "The job group."
4303
          ::= { jmMIBGroups 3 }
4304
4305
4306
4307
      jmAttributeGroup OBJECT-GROUP
4308
          OBJECTS {
              jmAttributeValueAsInteger, jmAttributeValueAsOctets }
4309
4310
          STATUS current
4311
         DESCRIPTION
4312
             "The attribute group."
        ::= { jmMIBGroups 4 }
4313
4314
4315
4316
      END
```

4317 5. Appendix A - Implementing the Job Life Cycle

4318 The job object has well-defined states and client operations that 4319 affect the transition between the job states. Internal server and device actions also affect the transitions of the job between the job 4320 states. These states and transitions are referred to as the job's life 4321 4322 cycle.

4323 Not all implementations of job submission protocols have all of the states of the job model specified here. The job model specified here 4324 is intended to be a superset of most implementations. It is the 4325 4326 purpose of the agent to map the particular implementation's job life cycle onto the one specified here. The agent MAY omit any states not 4327 implemented. Only the processing and completed states are required to 4328 4329 be implemented by an agent. However, a conforming management 4330 application SHALL be prepared to accept any of the states in the job 4331 life cycle specified here, so that the management application can 4332 interoperate with any conforming agent.

4333 The job states are intended to be user visible. The agent SHALL make 4334 these states visible in the MIB, but only for the subset of job states 4335 that the implementation has. Some implementations MAY need to have 4336 sub-states of these user-visible states. The jmJobStateReasons1 object and the jobStateReasonsN (N=2..4) attributes can be used to represent 4337 4338 the sub-states of the jobs.

Job states are intended to last a user-visible length of time in most 4339 4340 implementations. However, some jobs may pass through some states in 4341 zero time in some situations and/or in some implementations.

4342 The job model does not specify how accounting and auditing is 4343 implemented, except to assume that accounting and auditing logs are 4344 separate from the job life cycle and last longer than job entries in 4345 the MIB. Jobs in the completed, aborted, or canceled states are not 4346 logs, since jobs in these states are accessible via SNMP protocol 4347 operations and SHALL be removed from the Job Monitoring MIB tables 4348 after a site-settable or implementation-defined period of time. An 4349 accounting application MAY copy accounting information incrementally to 4350 an accounting log as a job processes, or MAY be copied while the job is 4351 in the canceled, aborted, or completed states, depending on implementation. The same is true for auditing logs. 4352

4353 The jmJobState object specifies the standard job states. The normal job state transitions are shown in the state transition diagram 4354 4355 presented in Table 1.

Bergman, Hastings, Isaacson, LewisInformational

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

4356 6. APPENDIX B - Support of Job Submission Protocols

4357 A companion PWG document, entitled "Job Submission Protocol Mapping Recommendations for the Job Monitoring MIB" [protomap] contains the 4358 recommended usage of each of the objects and attributes in this MIB 4359 with a number of job submission protocols. In particular, which job 4360 4361 submission ID format should be used is indicated for each job 4362 submission protocol.

4363 Some job submission protocols have support for the client to specify a job submission ID. A second approach is to enhance the document format 4364 to embed the job submission ID in the document data. This second 4365 approach is independent of the job submission protocol. This appendix 4366 4367 lists some examples of these approaches.

Some PJL implementations wrap a banner page as a PJL job around a job 4368 4369 submitted by a client. If this results in multiple job submission IDs, the agent SHALL create multiple jmJobIDEntry rows in the jmJobIDTable 4370 that each point to the same job entry in the job tables. See the 4371 4372 specification of the jmJobIDEntry.

4373 7. References

4374 [char-set policy] Harald Avelstrand, "IETF Policy on Character Sets and Language", June 1997. Latest draft: <draft-avelstrand-charset-4375 4376 policy-00.txt>

- 4377 [GB2312] GB 2312-1980, "Chinese People's Republic of China (PRC) mixed 4378 one byte and two byte coded character set"
- 4379 [hr-mib] P. Grillo, S. Waldbusser, "Host Resources MIB", RFC 1514, 4380 September 1993
- [iana] J. Reynolds, and J. Postel, "Assigned Numbers", STD 2, RFC 1700, 4381 4382 ISI, October 1994.

[IANA-charsets] Coded Character Sets registered by IANA and assigned an 4383 enum value for use in the CodedCharSet textual convention imported from 4384 4385 the Printer MIB. See ftp://ftp.isi.edu/innotes/iana/assignments/character-sets 4386

4387 [iana-media-types] IANA Registration of MIME media types (MIME content 4388 types/subtypes). See ftp://ftp.isi.edu/in-notes/iana/assignments/

4389 [ISO-639] ISO 639:1988 (E/F) - Code for Representation of names of 4390 languages - The International Organization for Standardization, 1st 4391 edition, 1988.

[ISO 646] ISO/IEC 646:1991, "Information technology -- ISO 7-bit coded 4392 character set for information interchange", JTC1/SC2. 4393

4394 [ISO 8859] ISO/IEC 8859-1:1987, "Information technology -- 8-bit single 4395 byte coded graphic character sets - Part 1: Latin alphabet No. 1, 4396 JTC1/SC2."

[ISO 2022] ISO/IEC 2022:1994 - "Information technology -- Character 4397 4398 code structure and extension techniques", JTC1/SC2.

[ISO-3166] ISO 3166:1988 (E/F) - Codes for representation of names of 4399 countries - The International Organization for Standardization, 3rd 4400 4401 edition, 1988-08-15."

[ISO-10646] ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and 4402 4403 Basic Multilingual Plane, JTC1/SC2. 4404

4405 [iso-dpa] ISO/IEC 10175 Document Printing Application (DPA). See 4406 ftp://ftp.pwq.org/pub/pwq/dpa/

4407 [ipp-model] Internet Printing Protocol/1.0: Model and Semantics, work in progress on the IETF standards track. See draft-ietf-ipp-model-4408 4409 09.txt. See also http://www.pwg.org/ipp/index.html

- 4410 [JIS X0208] JIS X0208-1990, "Japanese two byte coded character set."
- [mib-II] MIB-II, RFC 1213. 4411

[print-mib] Smith, R., Wright, F., Hastings, T., Zilles, S. and 4412 4413 Gyllenskog, J., "Printer MIB", RFC 1759, proposed IETF standard, March 4414 1995. See also [draft-print-mib].

[print-mib-draft] Turner, R., "Printer MIB", work in progress, on the 4415 4416 standards track as a draft standard: <draft-ietf-printmib-mib-info-4417 02.txt>, October 15, 1997.

[protomap] Bergman, R., "Job Submission Protocol Mapping 4418 Recommendations for the Job Monitoring MIB, "work in progress as an 4419 4420 informational RFC. See <draft-bergman-printmib-job-protomap-01.txt>, 4421 January 12, 1998.

[pwq] The Printer Working Group is a printer industry consortium open 4422 to any individuals. For more information, access the PWG web page: 4423 http://www.pwg.org 4424

4425 [req-words] S. Bradner, "Keywords for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997. 4426

[rfc 1738] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform 4427 Resource Locators (URL)", RFC 1738, December 1994. 4428

[RFC-1766] Avelstrand, H., "Tags for the Identification of Languages", 4429 RFC 1766, March 1995. 4430

Bergman, Hastings, Isaacson, LewisInformational

[Page 104]

4431

4432

4433

Atkinson, M. Crispin, and P. Svanberg, "The Report of the IAB Character

[rfc 2130] C. Weider, C. Preston, K. Simonsen, H. Alvestrand, R.

Set Workshop held 29 Feb-1 March, 1997", April 1997, RFC 2130.

4434 [SMIv2-SMI] J. Case, et al. "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 4435 4436 1902, January 1996. 4437 [SMIv2-TC] J. Case, et al. "Textual Conventions for Version 2 of the 4438 Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996. 4439 [tipsi] IEEE 1284.1, Transport-independent Printer System Interface 4440 (TIPSI). 4441 [URI-spec] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform 4442 Resource Locators (URL)", RFC 1738, December, 1994. 4443 [US-ASCII] Coded Character Set - 7-bit American Standard Code for 4444 Information Interchange, ANSI X3.4-1986. 4445 [UTF-8] F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 4446 10646", RFC 2044, October 1996. 8. Author's Addresses 4447 4448 Ron Bergman 4449 Dataproducts Corp. 1757 Tapo Canyon Road 4450 4451 Simi Valley, CA 93063-3394 4452 4453 Phone: 805-578-4421 4454 Fax: 805-578-4001 4455 Email: rbergman@dpc.com 4456 4457 4458 Tom Hastings 4459 Xerox Corporation, ESAE-231 701 S. Aviation Blvd. 4460 4461 El Segundo, CA 90245 4462 4463 Phone: 310-333-6413 Fax: 310-333-5514 4464 4465 EMail: hastings@cp10.es.xerox.com 4466 4467 4468 Scott A. Isaacson 4469 Novell, Inc. 4470 122 E 1700 S Provo, UT 84606 4471 4472 4473 Phone: 801-861-7366 4474 Fax: 801-861-4025 EMail: scott isaacson@novell.com 4475

Bergman, Hastings, Isaacson, LewisInformational

INTERNET-DRAFT Job Monitoring MIB, V1.0 January 1998

4476	
4477	
4478	Harry Lewis
4479	IBM Corporation
4480	6300 Diagonal Hwy
4481	Boulder, CO 80301
4482	
4483	Phone: (303) 924-5337
4484	Fax:
4485	Email: harryl@us.ibm.com
4486	1
4487	
4488	Send questions and comments to the Printer Working Group (PWG)
4489	using the Job Monitoring Project (JMP) Mailing List: jmp@pwg.org
4490	abing the top nonicoring respect (one) narring hists superwy.org
4491	To learn how to subscribe, send email to: jmp-request@pwg.org
4492	TO TEATH HOW CO SUBSCIEDE, SENA EMAIL CO. Jup requestepwg.org
4493	Implementers of this specification are encouraged to join the jmp
4493	
	mailing list in order to participate in discussions on any
4495	clarifications needed and registration proposals for additional
4496	attributes and values being reviewed in order to achieve consensus.
4497	
4498	For further information, access the PWG web page under "JMP":
4499	
4500	http://www.pwg.org/
4501	
4502	Other Participants:
4503	Chuck Adams - Tektronix
4503	Jeff Barnett - IBM
4504	
	Keith Carter, IBM Corporation
4506	Jeff Copeland - QMS
4507	Andy Davidson - Tektronix
4508	Roger deBry - IBM
4509	Mabry Dozier - QMS
4510	Lee Ferrel - Canon
4511	Steve Gebert - IBM
4512	Robert Herriot - Sun Microsystems Inc.
4513	Shige Kanemitsu - Kyocera
4514	David Kellerman – Northlake Software
4515	Rick Landau - Digital
4516	Pete Loya - HP
4517	Ray Lutz - Cognisys
4518	Jay Martin - Underscore
4519	Mike MacKay, Novell, Inc.
4520	Stan McConnell - Xerox
4521	Carl-Uno Manros, Xerox, Corp.
4522	Pat Nogay - IBM
4523	Bob Pentecost - HP
4524	Rob Rhoads - Intel
4525	David Roach - Unisys
4526	Stuart Rowley - Kyocera
-	

Bergman, Hastings, Isaacson, LewisInformational

9. INDEX 4540

This index includes the textual conventions, the objects, and the 4541 4542 attributes. Textual conventions all start with the prefix: "JM" and 4543 end with the suffix: "TC". Objects all starts with the prefix: "jm" 4544 followed by the group name. Attributes are identified with enums, and 4545 so start with any lower case letter and have no special prefix.

4546

4546		
4547	colorantConsumed	67
4548	colorantRequested	66
4549	deviceNameRequested	56
4550	documentCopiesCompleted	61
4551	documentCopiesRequested	61
4552	documentFormat	58
4553	documentFormatIndex	57
4554	documentName	57
4555	fileName	57
4556	finishing	60
4557	fullColorImpressionsCompleted	63
4558	highlightColorImpressionsCompleted	64
4559	impressionsCompletedCurrentCopy	63
4560	impressionsInterpreted	63
4561	impressionsSentToDevice	63
4562	impressionsSpooled	63
4563	jmAttributeInstanceIndex	97
4564	jmAttributeTypeIndex	97
4565	JmAttributeTypeTC	50
4566	jmAttributeValueAsInteger	98
4567	jmAttributeValueAsOctets	99
4568	JmBooleanTC	41
4569	JmFinishingTC	39
4570	jmGeneralAttributePersistence	84
4571	jmGeneralJobPersistence	84
4572	jmGeneralJobSetIndex	82
4573	jmGeneralJobSetName	85
4574	jmGeneralNewestActiveJobIndex	83
4575	jmGeneralNumberOfActiveJobs	82
4576	jmGeneralOldestActiveJobIndex	83
4577	jmJobIDJobIndex	88
4578	jmJobIDJobSetIndex	88
4579	jmJobImpressionsCompleted	93
4580	jmJobImpressionsPerCopyRequested	93
4581	jmJobIndex	90
4582	jmJobKOctetsPerCopyRequested	92
4583	jmJobKOctetsProcessed	92
4584	jmJobOwner	94
4585	JmJobServiceTypesTC	71
4586	JmJobSourcePlatformTypeTC	38
4587	jmJobState	90
4588	jmJobStateReasons1	91
4589	JmJobStateReasons1TC	72
4590	JmJobStateReasons2TC	76
		. •

4591	JmJobStateReasons3TC	80
4592	JmJobStateReasons4TC	80
4593	JmJobStateTC	47
4594	JmJobStringTC	37
4595	jmJobSubmissionID	87
4596	JmJobSubmissionIDTypeTC	43
4597	JmMediumTypeTC	41
4598	JmNaturalLanguageTagTC	37
4599	jmNumberOfInterveningJobs	91
4600	JmPrinterResolutionTC	40
4601	JmPrintQualityTC	40
4602	JmTimeStampTC	38
4603	JmTonerEconomyTC	41
4604	JmUTF8StringTC	37
4605	-	53
	jobAccountName	
4606	jobCodedCharSet	52
4607	jobCollationType	62
4608	jobComment	57
4609	jobCompletionTime	68
4610	jobCopiesCompleted	61
4611	jobCopiesRequested	61
4612	jobHold	59
4613	jobHoldUntil	59
4614	jobKOctetsTransferred	62
4615	jobName	54
4616	jobNaturalLanguageTag	53
4617		56
	jobOriginatingHost	
4618	jobPriority	58
4619	jobProcessAfterDateAndTime	59
4620	jobProcessingCPUTime	68
4621	jobServiceTypes	55
4622	jobSourceChannelIndex	55
4623	jobSourcePlatformType	55
4624	jobStartedBeingHeldTime	68
4625		68
4626	jobStateReasons2	51
4627	jobStateReasons3	51
4628	jobStateReasons4	51
4629	jobSubmissionTime	67
	jobSubmissionToServerTime	67
4630	5	
4631	jobURI	53
4632	mediumConsumed	66
4633	mediumRequested	66
4634	numberOfDocuments	56
4635	other	50
4636	outputBin	59
4637	pagesCompleted	64
4638	pagesCompletedCurrentCopy	65
4639	pagesRequested	64
4640	physicalDevice	56
4641	printerResolutionRequested	60
4642	printerResolutionUsed	60
4643	—	60
CFOF	printQualityRequested	00

Bergman, Hastings, Isaacson, LewisInformational

4647queueNameRequested54648serverAssignedJobName54649sheetCompletedCopyNumber64650sheetCompletedDocumentNumber64651sheetsCompleted64652sheetsCompletedCurrentCopy64653sheetsRequested64654sides64655submittingApplicationName54656submittingServerName54657tonerDensityRequested64658tonerDensityUsed64659tonerEcomonyRequested6	ngTag 52 56 54 62
--	----------------------------

4661