1	Job Monitoring MIB, V0.8 <mark>76</mark>
2	(This cover page is <i>not</i> part of the Internet-Draft)
3	
4 5 6	From: Tom Hastings Date: <u>12/03209/19</u> /97 Version: 0.8 <u>76</u>
7 8 9 10	File: ftp://ftp.pwg.org/pub/jmp/mibs/jmp-mib.doc .pdf jmp-mibr.doc .pdf .pdr Status: TenthNinth draft MIB that incorporates the agreements reached on the DL on issues in V0.865 which was released after the 9/198/8 meeting and the agreements reached at the JMP meeting on 9/19. The changes include:
11	1. <u>use the new PWG OIDs</u>
12 13	2. make the document a PWG draft standard that will be sent as an Internet-Draft that will become an IETF Informational RFC [not done]
14	3. <u>add natural language support like IPP</u>
15	4. add/fix monitoring collated/uncollated implementations [see issues]
16	5. <u>fix impressions completed</u> ,
17 18	6. <u>allows multiple Job Submission Id entries to point to the same jmJobIndex entry</u>
19	7. and add any new Job Submission Ids [not done]
20 21 22 23	In addition to the changes listed in Ron's list, the JMP agreed to remove the finishing enums that IPP removed (because of a lack of a coordinate system specification for stapling), add private enum range for attributes to agree with IPP. See the change history in the separate file: changes.doc .pdf.
24 25	We agreed that the MIB specification is finished except for any editorial comments that people may have. See the separate issues.doc and .pdf file.
26 27 28	I've also produced a variation on this document which has all variable font (mp-mib.doc.pdf) without revision marks. This is the version that the JMP should use to make comments. It has line numbers.
29 30	The MIB has been greatly simplified so that now there are only 18 objects in the MIB. There are 65 attributes.

31	INTERNET-DRAFT Ron Bergman
32	Dataproducts Corp.
33	Tom Hastings
34	Xerox Corporation
35	Scott Isaacson
36	Novell, Inc.
37	Harry Lewis
38	IBM Corp.
39	December 2September 19, 1997
40	
41	Job Monitoring MIB - V0.8 <mark>76</mark>
42	<draft-ietf-printmib-job-monitor-06.txt></draft-ietf-printmib-job-monitor-06.txt>
43	Expires <u>June</u> Mar <u>2</u> 19, 1997
44	
45	Status of this Memo
46	This document is an Internet-Draft. Internet-Drafts are working documents of the
47	Internet Engineering Task Force (IETF), its areas, and its working groups. Note
48	that other groups may also distribute working documents as Internet-Drafts.
49	Internet-Drafts are draft documents valid for a maximum of six months and may
50	be updated, replaced, or obsoleted by other documents at any time. It is
51	inappropriate to use Internet-Drafts as reference material or to cite them other than
52	as "work in progress."
53	To learn the current status of any Internet-Draft, please check the "1id-
54	abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on
55	ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim),
56	ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).
57	Abstract
58	This Internet-Draft specifies a small set of read-only SNMP MIB objects for (1)
59	monitoring the status and progress of print jobs (2) obtaining resource
60	requirements before a job is processed, (3) monitoring resource consumption
61	while a job is being processed and (4) collecting resource accounting data after the
62	completion of a job. This MIB is intended to be implemented (1) in a printer or
63	(2) in a server that supports one or more printers. Use of the object set is not
64	limited to printing. However, support for services other than printing is outside
65	the scope of this Job Monitoring MIB. Future extensions to this MIB may
66	include, but are not limited to, fax machines and scanners.

67 68

TABLE OF CONTENTS

69	1. INTRODUCTION	9
70	1.1 Types of Information in the MIB	9
71	1.2 Types of Job Monitoring Applications	10
72	2. TERMINOLOGY AND JOB MODEL	11
73	2.1 System Configurations for the Job Monitoring MIB	14
74	2.1.1 Configuration 1 - client-printer	
75	2.1.2 Configuration 2 - client-server-printer - agent in the server	
76	2.1.3 Configuration 3 - client-server-printer - client monitors printer agent and server	
77	3. MANAGED OBJECT USAGE	17
78	3.1 Conformance Considerations	17
79	3.1.1 Conformance Terminology	18
80	3.1.2 Agent Conformance Requirements	
81	3.1.2.1 MIB II System Group objects	18
82	3.1.2.2 MIB II Interface Group objects	
83	3.1.2.3 Printer MIB objects	19
84	3.1.3 Job Monitoring Application Conformance Requirements	19
85	3.2 The Job Tables and the Oldest Active and Newest Active Indexes	19
86	3.3 The Attribute Mechanism	21
87	3.3.1 Conformance of Attribute Implementation	
88	3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes	
89	3.3.3 Data Sub-types and Attribute Naming Conventions	23
90	3.3.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes	
91	3.3.5 Requested Attributes	
92	3.3.6 Consumption Attributes	
93	3.3.7 Index Value Attributes	24
94	3.4 Job Identification	25
95	3.5 Internationalization Considerations	28
96	3.5.1 'JmUTF8StringTC' for text generated by the server or device	25
97	3.5.2 'JmJobStringTC' for text generated by the job submitter	
98	3.5.3 'DateAndTime' for representing the date and time	25
99	3.6 IANA Considerations	
100	3.6.1 IANA Registration of enums	
101	3.6.1.1 Type 1 enumerations	30

102	3.6.1.2 Type 2 enumerations	31
103	3.6.1.3 Type 3 enumeration	31
104	3.6.2 IANA Registration of type 2 bit values	31
105	3.6.3 IANA Registration of Job Submission Id Formats	32
106	3.6.4 IANA Registration of MIME types/sub-types for document-formats	32
107	3.7 Security Considerations	32
108	3.7.1 Read-Write objects	
109	3.7.2 Read-Only Objects In Other User's Jobs	
110	3.8 Values for Objects	22
111	3.9 Notifications	32
112	4. MIB SPECIFICATION	33
113	Textual conventions for this MIB module	
114	JmUTF8StringTC	36
115	JmJobStringTC	
116	JmNaturalLanguageTC	
117	JmTimeStampTC	
118	JmJobSourcePlatformTypeTC	
119	JmFinishingTC	
120	JmPrintQualityTC	
121	JmPrinterResolutionTC	
122	JmTonerEconomyTC	
123	JmBooleanTC	
124	JmMediumTypeTC	
125	JmCollationTypeTC	
126	JmJobSubmissionIDTypeTC	
127 128	JmJobStateTC	
128	JmAttributeTypeTC	
130	other (Int32(-2) and/or Octets63)	
131	jobStateReasons2 (JmJobStateReasons2TC)	
132	jobStateReasons3 (JmJobStateReasons3TC)	
133	jobStateReasons4 (JmJobStateReasons4TC)	
134	processingMessage (UTF8String63)	
135	jobCodedCharSet (CodedCharSet)	
136	Job Identification attributes	
137	jobURI (Octets(1255))	
138	jobAccountName (Octets63)	
139	serverAssignedJobName (JobString63)	
140	jobName (JobString63)	
141	jobServiceTypes (JmJobServiceTypesTC)	
142	jobSourceChannelIndex (Int32(0))	
143	jobSourcePlatformType (JmJobSourcePlatformTypeTC)	
144	submittingServerName (JobString63)	
145	submittingApplicationName (JobString63)	

146	jobOriginatingHost (JobString63)	
147	deviceNameRequested (JobString63)	52
148	queueNameRequested (JobString63)	52
149	physicalDevice (hrDeviceIndex and/or UTF8String63)	52
150	numberOfDocuments (Int32(-2))	53
151	fileName (JobString63)	53
152	documentName (JobString63)	53
153	jobComment (JobString63)	
154	documentFormatIndex (Int32(0))	
155	documentFormat (PrtInterpreterLangFamilyTC and/or Octets63)	
156	Job Parameter attributes	
157	jobPriority (Int32(1100))	
158	jobProcessAfterDateAndTime (DateAndTime)	
159	jobHold (JmBooleanTC)	
160	jobHoldUntil (JobString63)	
161	outputBin (Int32(0) and/or JobString63)	
162	sides (Int32(-22))	
163	finishing (JmFinishingTC)	
164	Image Quality attributes (requested and used)	
165	printQualityRequested (JmPrintQualityTC)	
166	printQualityUsed (JmPrintQualityTC)	
167	printerResolutionRequested (JmPrinterResolutionTC)	
168	printerResolutionUsed (JmPrinterResolutionTC)	
169	tonerEcomonyRequested (JmTonerEconomyTC)	
170		
170 171	tonerEcomonyUsed (JmTonerEconomyTC)	
	tonerDensityRequested (Int32(-2100))	
172	tonerDensityUsed (Int32(-2100))	
173	Job Progress attributes (requested and consumed)	
174	jobCopiesRequested (Int32(-2))	
175	jobCopiesCompleted (Int32(-2))	
176	documentCopiesRequested (Int32(-2))	
177	documentCopiesCompleted (Int32(-2))	
178	jobKOctetsTransferred (Int32(-2))	
179	currentCopyNumber (Int32(-2))	
180	currentDocumentNumber (Int32(-2))	
181	collationType (JmCollationTypeTC)	
182	Impression attributes (requested and consumed)	
183	impressionsSpooled (Int32(-2))	
184	impressionsSentToDevice (Int32(-2))	
185	impressionsInterpreted (Int32(-2))	
186	impressionsCompletedCurrentCopy (Int32(-2))	59
187	fullColorImpressionsCompleted (Int32(-2))	59
188	highlightColorImpressionsCompleted (Int32(-2))	59
189	Page attributes (requested and consumed)	
190	pagesRequested (Int32(-2))	
191	pagesCompleted (Int32(-2))	
192	pagesCompletedCurrentCopy (Int32(-2))	
193	Sheet attributes (requested and consumed)	
194	sheetsRequested (Int32(-2))	
195	sheetsCompleted (Int32(-2))	
	1 ',','	

196	sheetsCompletedCurrentCopy (Int32(-2))	61
197	Resource attributes (requested and consumed)	61
198	mediumRequested (JmMediumTypeTC and/or JobString63)	61
199	mediumConsumed (JobString63)	61
200	colorantRequested (Int32(-2) and/or JobString63)	61
201	colorantConsumed (Int32(-2) and/or JobString63)	
202	Time attributes (set by server or device)	
203	jobSubmissionToServerTime (JmTimeStampTC and/or DateAndTime)	
204	jobSubmissionTime (JmTimeStampTC and/or DateAndTime)	
205	jobStartedBeingHeldTime (JmTimeStampTC and/or DateAndTime)	
206	jobStartedProcessingTime (JmTimeStampTC and/or DateAndTime)	
207	jobCompletionTime (JmTimeStampTC and/or DateAndTime)	
208	jobProcessingCPUTime (Int32(-2))	
209	JmJobServiceTypesTC	
210	JmJobStateReasons1TC	
211	JmJobStateReasons2TC	
212	JmJobStateReasons3TC	
213	JmJobStateReasons4TC	
213	JIII OUS LILLOTTO TO THE STATE OF THE STATE	
214	The General Group (MANDATORY)	75
215	jmGeneralJobSetIndex (Int32(132767))	
216	jmGeneralNumberOfActiveJobs (Int32(0))	
217	jmGeneralOldestActiveJobIndex (Int32(0))	
218	jmGeneralNewestActiveJobIndex (Int32(0))	
219	jmGeneralJobPersistence (Int32(15))	
220	jmGeneralAttributePersistence (Int32(15))	
221	jmGeneralJobSetName (UTF8String63)	
221	Jinocherai3005ettvanie (0 11 obtinig03)	/ 0
222	The Job ID Group (MANDATORY)	75
223	jmJobSubmissionID (OCTET STRING(SIZE(48)))	
224	jmJobIDJobSetIndex (Int32(132767))	
225	jmJobIDJobIndex (Int32(1))	
	Jill ooi Dvoomson (Into 2(11.))	
226	The Job Group (MANDATORY)	80
227	jmJobIndex (Int32(1))	
228	jmJobState (JmJobStateTC)	
229	jmJobStateReasons1 (JmJobStateReasons1TC)	
230	jmNumberOfInterveningJobs (Int32(-2))	
231	jmJobKOctetsPerCopyRequested (Int32(-2))	85
232	jmJobKOctetsProcessed (Int32(-2))	
233	jmJobImpressionsPerCopyRequested (Int32(-2))	
234	jmJobImpressionsCompleted (Int32(-2))	
235	jmJobOwner (JobString63)	
236	The Attribute Group (MANDATORY)	87
237	jmAttributeTypeIndex (JmAttributeTypeTC)	
238	jmAttributeInstanceIndex (Int32(132767))	
239	jmAttributeValueAsInteger (Int32(-2))	
240	imAttributeValueAsOctets (Octets63).	

241	5. APPENDIX A - IMPLEMENTING THE JOB LIFE CYCLE	94
242 243	6. APPENDIX B - SUPPORT OF THE JOB SUBMISSION ID IN JOB SUBMISSION PROTOCOLS	94
244	6.1 Hewlett-Packard's Printer Job Language (PJL)	95
245	6.2 ISO DPA	95
246	7. REFERENCES	95
247	8. AUTHOR'S ADDRESSES	97
248	9. INDEX	100

Job Monitoring MIB

251	1. Introduction
252 253 254 255 256 257 258 259 260	The Job Monitoring MIB is intended to be implemented by an agent within a printer or the first server closest to the printer, where the printer is either directly connected to the server only or the printer does not contain the job monitoring MIB agent. It is recommended that implementations place the SNMP agent as close as possible to the processing of the print job. This MIB applies to printers with and without spooling capabilities. This MIB is designed to be compatible with most current commonly-used job submission protocols. In most environments that support high function job submission/job control protocols, like ISO DPA[iso-dpa], those protocols would be used to monitor and manage print jobs rather than using the Job Monitoring MIB.
261 262 263 264 265 266 267 268 269 270 271	The Job Monitoring MIB consists of a General Group, a Job Submission ID Group, a Job Group, and an Attribute Group. Each group is a table. All accessible objects are read-only. The General Group contains general information that applies to all jobs in a job set. The Job Submission ID table maps the job submission ID that the client uses to identify a job to the jmJobIndex that the Job Monitoring Agent uses to identify jobs in the Job and Attribute tables. The Job table contains the MANDATORY integer job state and status objects. The Attribute table consists of multiple entries per job that specify (1) job and document identification and parameters, (2) requested resources, and (3) consumed resources during and after job processing/printing. A larger number of job attributes are defined as textual conventions that an agent SHALL return if the server or device implements the functionality so represented and the agent has access to the information.
272	1.1 Types of Information in the MIB
273 274	The job MIB is intended to provide the following information for the indicated Role Models in the Printer MIB[print-mib] (Appendix D - Roles of Users).
275	User:
276 277 278	Provide the ability to identify the least busy printer. The user will be able to determine the number and size of jobs waiting for each printer. No attempt is made to actually predict the length of time that jobs will take.
279	Provide the ability to identify the current status of the user's job (user queries).
280	Provide a timely indication that the job has completed and where it can be found
281 282	Provide error and diagnostic information for jobs that did not successfully complete.
283	Operator:

284 Provide a presentation of the state of all the jobs in the print system. 285 Provide the ability to identify the user that submitted the print job. 286 Provide the ability to identify the resources required by each job. 287 Provide the ability to define which physical printers are candidates for the print 288 job. 289 Provide some idea of how long each job will take. However, exact estimates of 290 time to process a job is not being attempted. Instead, objects are included that allow the operator to be able to make gross estimates. 291 292 Capacity Planner: 293 Provide the ability to determine printer utilization as a function of time. 294 Provide the ability to determine how long jobs wait before starting to print. 295 Accountant: 296 Provide information to allow the creation of a record of resources consumed and 297 printer usage data for charging users or groups for resources consumed. 298 Provide information to allow the prediction of consumable usage and resource 299 need. 300 The MIB supports printers that can contain more than one job at a time, but still be usable 301 for low end printers that only contain a single job at a time. In particular, the MIB 302 supports the needs of Windows and other PC environments for managing low-end direct-303 connect (serial or parallel) and networked devices without unnecessary overhead or 304 complexity, while also providing for higher end systems and devices. 305 1.2 Types of Job Monitoring Applications 306 The Job Monitoring MIB is designed for the following types of monitoring applications: 307 Monitor a single job starting when the job is submitted and ending a defined 308 period after the job completes. The Job Submission ID table provides the 309 map to find the specific job to be monitored. 310 Monitor all 'active' jobs in a queue, which this specification generalizes to a 311 "job set". End users may use such a program when selecting a least busy 312 printer, so the MIB is designed for such a program to start up quickly and find the information needed quickly without having to read all (completed) jobs in 313 order to find the active jobs. System operators may also use such a program, 314 315 in which case it would be running for a long period of time and may also be 316 interested in the jobs that have completed. Finally such a program may be used to provide an enhanced console and logging capability. 317

- 318 3. Collect resource usage for accounting or system utilization purposes that copy 319 the completed job statistics to an accounting system. It is recognized that 320 depending on accounting programs to copy MIB data during the job-retention period is somewhat unreliable, since the accounting program may not be 321 322 running (or may have crashed). Such a program is also expected to keep a 323 shadow copy of the entire Job Attribute table including completed, 324 canceled, and aborted jobs which the program updates on each polling 325 cycle. Such a program polls at the rate of the persistence of the **Attribute** 326 table. The design is not optimized to help such an application determine 327 which jobs are **completed**, **canceled**, or **aborted**. Instead, the application 328 SHALL query each job that the application's shadow copy shows was not 329 complete, canceled, or aborted at the previous poll cycle to see if it is now 330 **complete** or **canceled**, plus any new jobs that have been submitted.
- 331 The MIB provides a set of objects that represent a compatible subset of job and document 332 attributes of the ISO DPA standard[iso-dpa] and the Internet Printing Protocol (IPP)[ipp-333 model], so that coherence is maintained between these two protocols and the information 334 presented to end users and system operators by monitoring applications. However, the 335 job monitoring MIB is intended to be used with printers that implement other job 336 submitting and management protocols, such as IEEE 1284.1 (TIPSI)[tipsi], as well as 337 with ones that do implement ISO DPA. Thus the job monitoring MIB does not require 338 implementation of either the ISO DPA or IPP protocols.
- The MIB is designed so that an additional MIB(s) can be specified in the future for monitoring multi-function (scan, FAX, copy) jobs as an augmentation to this MIB.

2. Terminology and Job Model

- This section defines the terms that are used in this specification and the general model for jobs.
- NOTE Existing systems use conflicting terms, so these terms are drawn from the ISO
- 345 10175 Document Printing Application (DPA) standard[iso-dpa]. For example,
- PostScript systems use the term session for what is called a job in this specification
- and the term *job* to mean what is called a *document* in this specification.
- 348 Job: A unit of work whose results are expected together without interjection of unrelated
- results. A job contains one or more *documents*.
- Job Set: A group of jobs that are queued and scheduled together according to a specified
- 351 scheduling algorithm for a specified device or set of devices. For implementations that
- embed the SNMP agent in the device, the MIB job set normally represents all the jobs
- known to the device, so that the implementation only implements a single job set. If the
- 354 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 device or (2) set of devices, if the server uses

- a single queue to load balance between several devices. Each job set is disjoint; no job
- 357 SHALL be represented in more than one MIB job set.
- 358 Document: A sub-section within a job that contains print data and *document instructions*
- 359 that apply to just the document.
- 360 Client: The network entity that *end users* use to submit jobs to *spoolers*, *servers*, or
- 361 *printers* and other *devices*, depending on the configuration, using any job submission
- protocol over a serial or parallel port to a directly-connected device or over the network to
- a networked-connected device.
- 364 Server: A network entity that accepts jobs from clients and in turn submits the jobs to
- 365 printers and other devices that may be directly connected to the server via a serial or
- parallel port or may be on the network. A server MAY be a printer *supervisor* control
- program, or a print spooler.
- Device: A hardware entity that (1) interfaces to humans, such as a device that produces
- marks on paper or scans marks on paper to produce an electronic representation, (2)
- accesses digital media, such as CD-ROMs, or (3) interfaces electronically to another
- device, such as sends FAX data to another FAX device.
- 372 Printer: A *device* that puts marks on media.
- 373 Supervisor: A server that contains a control program that controls a printer or other
- device. A supervisor is a client to the printer or other device.
- 375 Spooler: A server that accepts jobs, spools the data, and decides when and on which
- printer to print the job. A spooler is a client to a printer or a printer supervisor, depending
- on implementation.
- 378 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.
- Queuing: The act of a *device* or *server* of ordering (queuing) the jobs for the purposes of
- 381 scheduling the jobs to be processed.
- 382 Monitor or Job Monitoring Application: The SNMP management application that End
- 383 Users, and System Operators use to monitor jobs using SNMP. A monitor MAY be
- either a separate application or MAY be part of the client that also submits jobs.
- 385 Accounting Application: The SNMP management application that copies job
- information to some more permanent medium so that another application can perform
- accounting on the data for Accountants, Asset Managers, and Capacity Planners use.
- 388 Agent: The network entity that accepts SNMP requests from a monitor or accounting
- 389 application and provides access to the instrumentation for managing jobs modeled by the
- 390 management objects defined in the Job Monitoring MIB module for a server or a device.

- 391 Proxy: An agent that acts as a concentrator for one or more other agents by accepting
- 392 SNMP operations on the behalf of one or more other agents, forwarding them on to those
- 393 other agents, gathering responses from those other agents and returning them to the
- 394 original requesting monitor.
- 395 User: A person that uses a client or a monitor.
- 396 End User: A user that uses a client to submit a print job.
- 397 System Operator: A user that uses a monitor to monitor the system and carries out tasks
- 398 to keep the system running.
- 399 System Administrator: A user that specifies policy for the system.
- Job Instruction: An instruction specifying how, when, or where the job is to be
- 401 processed. Job instructions MAY be passed in the job submission protocol or MAY be
- 402 embedded in the document data or a combination depending on the job submission
- 403 protocol and implementation.
- 404 Document Instruction: An instruction specifying how to process the document.
- Document instructions MAY be passed in the job submission protocol separate from the
- actual document data, or MAY be embedded in the document data or a combination,
- 407 depending on the job submission protocol and implementation.
- 408 SNMP Information Object: A name, value-pair that specifies an action, a status, or a
- 409 condition in an SNMP MIB. Objects are identified in SNMP by an OBJECT
- 410 IDENTIFIER.
- 411 Attribute: A name, value-pair that specifies a job or document instruction, a status, or a
- condition of a job or a document that has been submitted to a server or device. A
- 413 particular attribute NEED NOT 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 value, either because
- 415 (1) the client supplied a value in the job submission protocol, (2) the document data
- 416 contained an embedded attribute, or (3) the server or device supplied a default value. An
- agent SHALL represent an attribute as an entry (row) in the Attribute table in this MIB in
- 418 which entries are present only when necessary. Attributes are identified in this MIB by an
- 419 enum.
- 420 Job Monitoring (using SNMP): The activity of a management application of accessing
- 421 the MIB and (1) identifying jobs in the job tables being processed by the server, printer or
- other devices, and (2) displaying information to the user about the processing of the job.
- Job Accounting: The activity of a management application of accessing the MIB and
- recording what happens to the job during and after the processing of the job.

425 **2.1** System Configurations for the Job Monitoring MIB

- This section enumerates the three configurations in which the Job Monitoring MIB is
- intended to be used. To simplify the pictures, the *devices* are shown as *printers*. See
- 428 section 1.1 entitled "Types of Information in the MIB".
- The diagram in the Printer MIB[print-mib] entitled: "One Printer's View of the Network"
- is assumed for this MIB as well. Please refer to that diagram to aid in understanding the
- 431 following system configurations.

432

440

455

458

459

460 461

2.1.1 Configuration 1 - client-printer

- In the **client-printer** configuration 1, the **client**(s) submit jobs directly to the **printer**,
- either by some direct connect, or by network connection.
- The job submitting **client** and/or **monitoring application** monitor jobs by
- communicating directly with an agent that is part of the **printer**. The agent in the **printer**
- SHALL keep the job in the Job Monitoring MIB as long as the job is in the **printer**, plus
- a defined time period after the job enters the **completed** state in which accounting
- programs can copy out the accounting data from the Job Monitoring MIB.

```
441
                                                                                                                                                                                                all
                                                                                                                                                                                                                                                                                                              end-user
                                                                                                                                                                                                                                                                                                                                                                                                                                           ####### SNMP query
                                                                                                                                                                 | total | tota
442
                                                                                                                                                                                                                                                                                                           +----+
                                                                                                                                                                                                                                                                                                                                                                                                                                               --- job submission
443
444
445
446
                                                                                                                                                                                                          # ############
447
                                                                                                                                                                                                          # #
448
                                                                                                                                        +==+===#=#=+==+
449
                                                                                                                                                                     agent
450
                                                                                                                                                                    +----+
451
                                                                                                                                                                            PRINTER
452
                                                                                                                                                                                                                                                                                              Print Job Delivery Channel
453
454
                                                                                                                                        +=======+
```

Figure 2-1 - Configuration 1 - client-printer - agent in the printer

- The Job Monitoring MIB is designed to support the following relationships (not shown in Figure 2-1):
 - 1. Multiple clients MAY submit jobs to a printer.
 - 2. Multiple **clients** MAY monitor a **printer**.
 - 3. Multiple **monitors** MAY monitor a **printer**.
 - 4. A **client** MAY submit jobs to multiple **printers**.
- 5. A **monitor** MAY monitor multiple **printers**.

2.1.2 Configuration 2 - client-server-printer - agent in the server

463464

465

466 467

468

469

470 471

472

473

474

475

476

477 478

502

In the **client-server-printer** configuration 2, the **client**(s) submit jobs to an intermediate **server** by some network connection, *not* directly to the **printer**. While configuration 2 is included, the design center for this MIB is configurations 1 and 3.

The job submitting **client** and/or **monitoring application** monitor jobs by communicating directly with:

A Job Monitoring MIB agent that is part of the **server** (or a front for the server)

There is no SNMP Job Monitoring MIB agent in the **printer** in configuration 2, at least that the client or monitor are aware. In this configuration, the agent SHALL return the current values of the objects in the Job Monitoring MIB both for jobs the server keeps and jobs that the server has submitted to the **printer**. The Job Monitoring MIB agent SHALL obtain the required information from the **printer** by a method that is beyond the scope of this document. The agent in the **server** SHALL keep the job in the Job Monitoring MIB in the server as long as the job is in the **printer**, plus a defined time period after the job enters the **completed** state in which accounting programs can copy out the accounting data from the Job Monitoring MIB.

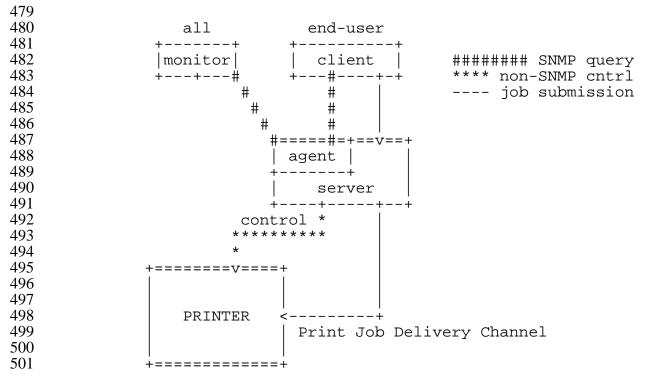


Figure 2-2 - Configuration 2 - client-server-printer - agent in the server

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

505 Multiple clients MAY submit jobs to a server. 506 2. Multiple clients MAY monitor a server. 507 3. Multiple **monitors** MAY monitor a **server**. 508 A **client** MAY submit jobs to multiple **servers**. 509 A monitor MAY monitor multiple servers. Multiple servers MAY submit jobs to a printer. 510 6. 511 7. Multiple servers MAY control a printer. 512 2.1.3 Configuration 3 - client-server-printer - client monitors printer agent and 513 server 514 In the **client-server-printer** configuration 3, the **client**(s) submit jobs to an intermediate 515 **server** by some network connection, *not* directly to the **printer**. That server does *not* 516 contain a Job Monitoring MIB agent. 517 The job submitting **client** and/or **monitoring application** monitor jobs by 518 communicating directly with: 519 The **server** using some undefined protocol to monitor jobs in the server (that 520 does not contain the Job Monitoring MIB) AND 521 A Job Monitoring MIB agent that is part of the **printer** to monitor jobs after 522 the server passes the jobs to the **printer**. In such configurations, the server deletes its copy of the job from the server after submitting the job to the 523 524 printer usually almost immediately (before the job does much processing, if 525 any). 526 In configuration 3, the agent (in the **printer**) SHALL keep the values of the objects in the 527 Job Monitoring MIB that the agent implements updated for a job that the server has 528 submitted to the printer. The agent SHALL obtain information about the jobs submitted 529 to the printer from the server (either in the job submission protocol, in the document data, 530 or by direct query of the server), in order to populate some of the objects the Job 531 Monitoring MIB in the printer. The agent in the printer SHALL keep the job in the Job 532 Monitoring MIB as long as the job is in the Printer, and longer in order to implement the 533 **completed** state in which monitoring programs can copy out the accounting data from the

Job Monitoring MIB.

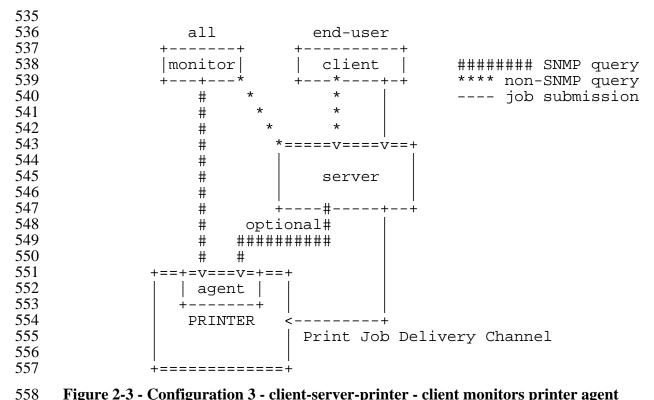


Figure 2-3 - Configuration 3 - client-server-printer - client monitors printer agent and server

- The Job Monitoring MIB is designed to support the following relationships (not shown in Figure 2-3):
 - 1. Multiple **clients** MAY submit jobs to a **server**.
 - 2. Multiple **clients** MAY monitor a **server**.
 - 3. Multiple **monitors** MAY monitor a **server**.
 - 4. A **client** MAY submit jobs to multiple **servers**.
 - 5. A **monitor** MAY monitor multiple **servers**.
 - 6. Multiple **servers** MAY submit jobs to a **printer**.
 - 7. Multiple **servers** MAY control a **printer**.

3. Managed Object Usage

559

562

563

564

565

566

567

568

569

This section describes the usage of the objects in the MIB.

3.1 Conformance Considerations

- In order to achieve interoperability between job monitoring applications and job
- 573 monitoring agents, this specification includes the conformance requirements for both
- monitoring applications and agents.

575 **3.1.1 Conformance Terminology**

- This specification uses the verbs: "SHALL", "SHOULD", "MAY", and "NEED NOT" to specify conformance requirements according to RFC 2119 [req-words] as follows:
- "SHALL": indicates an action that the subject of the sentence must implement in order to claim conformance to this specification
- "MAY": indicates an action that the subject of the sentence does not have to implement in order to claim conformance to this specification, in other words that action is an implementation option
- "NEED NOT": indicates an action that the subject of the sentence does not have to implement in order to claim conformance to this specification. The verb "NEED NOT" is used instead of "may not", since "may not" sounds like a prohibition.
- "SHOULD": indicates an action that is recommended for the subject of the sentence to implement, but is not required, in order to claim conformance to this specification.

3.1.2 Agent Conformance Requirements

590 A conforming agent:

589

591

595

596 597

- 1. SHALL implement *all* MANDATORY groups in this specification.
- 592 2. SHALL implement any attributes if (1) the server or device supports the functionality represented by the attribute and (2) the information is available to the agent.
 - 3. SHOULD implement both forms of an attribute if it implements an attribute that permits a choice of INTEGER and OCTET STRING forms, since implementing both forms may help management applications by giving them a choice of representations, since the representation are equivalent. See the **JmAttributeTypeTC** textual-convention.
- NOTE This MIB, like the Printer MIB, is written following the subset of SMIv2 that can be supported by SMIv1 and SNMPv1 implementations.
- 602 3.1.2.1 MIB II System Group objects
- 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 implemented or not.
- 605 3.1.2.2 MIB II Interface Group objects
- The Job Monitoring MIB agent SHALL implement all objects in the Interfaces Group of
- MIB-II[mib-II], whether the Printer MIB[print-mib] is implemented or not.

608 3.1.2.3 Printer MIB objects

615

617

618

619 620

621

622

623 624

625

626 627

628629

630

631

634

635

638

639

640

641

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

3.1.3 Job Monitoring Application Conformance Requirements

- A conforming job monitoring application:
 - 1. SHALL accept the full syntactic range for all objects in all MANDATORY groups and all MANDATORY attributes that are required to be implemented by an agent according to Section 3.1.2 and SHALL either present them to the user or ignore them.
 - 2. SHALL accept the full syntactic range for *all* attributes, including enum and bit values specified in this specification and additional ones that may be registered with IANA and SHALL either present them to the user or ignore them. In particular, a conforming job monitoring application SHALL not malfunction when receiving any standard or registered enum or bit values. See Section 3.7 entitled "IANA Considerations".
 - 3. SHALL NOT fail when operating with agents that materialize attributes *after* the job has been submitted, as opposed to when the job is submitted.
 - 4. SHALL, if it supports a time attribute, accept either form of the time attribute, since agents are free to implement either time form.

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

- The **jmJobTable** and **jmAttributeTable** contain objects and attributes, respectively, for each job in a job set. These first two indexes are:
 - 1. imGeneralJobSetIndex which job set
 - 2. jmJobIndex which job in the job set
- In order for a monitoring application to quickly find that active jobs (jobs in the **pending**, **processing**, or **processingStopped** states), the MIB contains two indexes:
 - 1. **jmGeneralOldestActiveJobIndex** the index of the active job that has been in the tables the longest.
 - 2. **jmGeneralNewestActiveJobIndex** the index of the active job that 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.
- It is recommended that the largest value for **jmJobIndex** be much larger than the
- maximum number of jobs that the implementation can contain at a single time, so as to
- minimize the premature re-use of a **jmJobIndex** value for a newer job while clients retain
- the same 'stale' value for an older job.
- It is recommended that agents that are providing access to servers/devices that already
- allocate job-identifiers for jobs as integers use the same integer value for the
- **jmJobIndex**. Then management applications using this MIB and applications using
- other protocols will see the same job identifiers for the same jobs. Agents providing
- access to systems that contain jobs with a job identifier of **0** SHALL map the job
- 657 identifier value **0** to a **imJobIndex** value that is one higher than the highest job identifier
- value that any job can have on that system. Then only job 0 will have a different job-
- identifier 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.
- Each 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
- **imGeneralNewestActiveJobIndex** object. If the new job is to be 'inactive'
- 669 (pendingHeld state), the agent SHALL not change the value of
- 670 **imGeneralNewestActiveJobIndex** object (though the agent SHALL assign the next
- incremental **imJobIndex** value to the job).
- 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
- **imGeneralNewestActiveJobIndex** objects, or both, if the job's **imJobIndex** value is

- outside the range between **jmGeneralOldestActiveJobIndex** and
- 683 jmGeneralNewestActiveJobIndex.
- When all jobs become 'inactive', i.e., enter the pending Held, completed, canceled, or
- aborted states, the agent SHALL set the value of both the
- jmGeneralOldestActiveJobIndex and jmGeneralNewestActiveJobIndex objects to 0.
- NOTE Applications that wish to efficiently access all of the active jobs MAY use
- imGeneralOldestActiveJobIndex 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.
- If an application detects that the **jmGeneralNewestActiveJobIndex** is smaller than
- jmGeneralOldestActiveJobIndex, the job index has wrapped. In this case, the
- application SHALL reset the index to 1 when the end of the table is reached and continue
- the GetNext operations to find the rest of the active jobs.
- NOTE Applications detect the end of the **jmAttributeTable** table when the OID
- returned by the GetNext operation is an OID in a different MIB. There is no object in this
- 697 MIB that specifies the maximum value for the **jmJobIndex** supported by the
- 698 implementation.
- When the server or device is power-cycled, the agent SHALL remember the next
- jmJobIndex value to be assigned, so that new jobs are not assigned the same
- 701 **jmJobIndex** as recent jobs before the power cycle.

702 **3.3** The Attribute Mechanism

- Attributes are similar to information objects, except that attributes are identified by an
- enum, instead of an OID, so that attributes may be registered without requiring a new
- MIB. Also an implementation that does not have the functionality represented by the
- attribute can omit the attribute entirely, rather than having to return a distinguished value.
- The agent is free to materialize an attribute in the **jmAttributeTable** as soon as the agent
- is aware of the value of the attribute.
- The agent materializes job attributes in a four-indexed **jmAttributeTable**:
- 710 1. jmGeneralJobSetIndex which job set
- 711 2. jmJobIndex which job in the job set
- 712 3. jmAttributeTypeIndex which attribute
- 713 4. jmAttributeInstanceIndex which attribute instance for those attributes that can have multiple values per job.
- Some attributes represent information about a job, such as a file-name, a document-name,
- a submission-time or a completion time. Other attributes represent resources required,

- e.g., a medium or a colorant, etc. to process the job before the job starts processing OR to
- 718 indicate the amount of the resource consumed during and after processing, e.g., pages
- 719 completed or impressions completed. If both a required and a consumed value of a
- resource is needed, this specification assigns two separate attribute enums in the textual
- 721 convention.
- NOTE The table of contents lists all the attributes in order. This order is the order of
- enum assignments which is the order that the SNMP GetNext operation returns attributes.
- Most attributes apply to all three configurations covered by this MIB specification (see
- section 2.1 entitled "System Configurations for the Job Monitoring MIB"). Those
- attributes that apply to a particular configuration are indicated as Configuration n: and
- 727 SHALL NOT be used with other configurations.

728 **3.3.1 Conformance of Attribute Implementation**

- An agent SHALL implement any attribute if (1) the server or device supports the
- functionality represented by the attribute and (2) the information is available to the agent.
- 731 The agent MAY create the attribute row in the **jmAttributeTable** when the information
- is available or MAY create the row earlier with the designated 'unknown' value
- appropriate for that attribute. See next section.
- 734 If the server or device does not implement or does not provide access to the information
- about an attribute, the agent SHOULD NOT create the corresponding row in the
- 736 **jmAttributeTable**.

737 3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes

- Some attributes have a 'useful' Integer32 value, some have a 'useful' OCTET STRING
- value, some MAY have either or both depending on implementation, and some MUST
- have both. See the **JmAttributeTypeTC** textual convention for the specification of each
- 741 attribute.
- NMP requires that if an object cannot be implemented because its values cannot be
- accessed, then a compliant agent SHALL return an SNMP error in SNMPv1 or an
- exception value in SNMPv2. However, this MIB has been designed so that 'all' objects
- can and SHALL be implemented by an agent, so that neither the SNMPv1 error nor the
- NMPv2 exception value SHALL be generated by the agent. This MIB has also been
- designed so that when an agent materializes an attribute, the agent SHALL materialize a
- 748 row consisting of both the **jmAttributeValueAsInteger** and **jmAttributeValueAsOctets**
- 749 objects.
- 750 In general, values for objects and attributes have been chosen so that a management
- application will be able to determine whether a 'useful', 'unknown', or 'other' value is
- available. When a useful value is not available for an object that agent SHALL return a

- zero-length string for octet strings, the value **unknown(2)**' for enums, a **0**' value for an
- object that represents an index in another table, and a value '-2' for counting integers.
- Since each attribute is represented by a row consisting of both the
- 756 **jmAttributeValueAsInteger** and **jmAttributeValueAsOctets** MANDATORY objects,
- 757 SNMP requires that the agent SHALL always create an attribute row with both objects
- specified. However, for most attributes the agent SHALL return a "useful" value for one
- of the objects and SHALL return the 'other' value for the other object. For integer only
- attributes, the agent SHALL always return a zero-length string value for the
- jmAttributeValueAsOctets object. For octet string only attributes, the agent SHALL
- always return a '-1' value for the jmAttributeValueAsInteger object.

3.3.3 Data Sub-types and Attribute Naming Conventions

- Many attributes are sub-typed to give a more specific data type than **Integer32** or
- 765 **OCTET STRING.** The data sub-type of each attribute is indicated on the first line(s) of
- the description. Some attributes have several different data sub-type representations.
- 767 When an attribute has both an **Integer32** data sub-type and an **OCTET STRING** data
- sub-type, the attribute can be represented in a single row in the **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)** which is both an enum and/or a name. When the
- data sub-types cannot be represented by a single row in the **jmAttributeTable**, each such
- representation is considered a separate attribute and is assigned a separate name and enum
- value. For these attributes, the name of the data sub-type is the last part of the name of
- the attribute: **Name**, **Index**, **DateAndTime**, **TimeStamp**, etc. For example,
- 775 **documentFormatIndex(37)** is an index.

763

- 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 such is defined. The following
- abbreviations are used in the Table of Contents as shown:

'Int32(-2)'	Integer32(-22147483647)
'Int32(0)'	Integer32(02147483647)
'Int32(1)'	Integer32(12147483647)

'Int32(m..n)' For all other Integer ranges, the lower and upper bound of

the range is indicated.

'UTF8String63' JmUTF8StringTC(SIZE(0..63))
'JobString63' JmJobStringTC(SIZE(0..63))
'Octets63' OCTET STRING(SIZE(0..63))

'Octets(m..n)' For all other OCTET STRING ranges, the exact range is

indicated.

779 3.3.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes

- 780 Most attributes SHALL have only one row per job. However, a few attributes can have
- multiple values per job or even per document, where each value is a separate row in the
- 782 **jmAttributeTable**. Unless indicated with 'MULTI-ROW:' in the JmAttributeTypeTC
- description, an agent SHALL ensure that each attribute occurs only once in the
- jmAttributeTable for a job. Most of the 'MULTI-ROW' attributes do not allow
- duplicate values, i.e., the agent SHALL ensure that each value occurs only once for a job.
- Only if the specification of the **MULTI-ROW**' attribute also says "the values NEED
- NOT be unique" can the agent allow duplicate values to occur for the job.
- NOTE Duplicates are allowed for 'extensive' **MULTI-ROW**' attributes, such as
- 789 **fileName(34)** or **documentName(35)** which are specified to be 'per-document' attributes,
- but are *not* allowed for 'intensive' **MULTI-ROW**' attributes, such as
- 791 **mediumConsumed(171)** and **documentFormat(38)** which are specified to be 'per-job'
- 792 attributes.

793 3.3.5 Requested Attributes

- A number of attributes record requirements for the job. Such attribute names end with
- 795 the word '**Requested**'. In the interests of brevity, the phrase 'requested' SHALL mean:
- 796 (1) requested by the client (or intervening server) in the job submission protocol and
- 797 MAY also mean (2) embedded in the submitted document data, and/or (3) defaulted by
- 798 the recipient device or server with the same semantics as if the requester had supplied,
- 799 depending on implementation.

800 **3.3.6 Consumption Attributes**

- 801 A number of attributes record consumption. Such attribute names end with the word
- **'Completed'** or **'Consumed'**. If the job has not yet consumed what that resource is
- 803 metering, the agent either: (1) SHALL return the value **0** or (2) SHALL *not* add this
- attribute to the **imAttributeTable** until the consumption begins. In the interests of
- brevity, the semantics for **0** is specified once here and is *not* repeated for each
- 806 consumptive attribute specification.

807 **3.3.7 Index Value Attributes**

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

814	3.4 Monitoring Job Pr	<u>ogress</u>		
815 816 817 818 819 820	objects and attributes correquested or completed, are objects and attributes currently being processed	ant the number of K octet i.e., processed or stacked, for the overall job and for dor stacked. For the latter	onitoring the progress of a s, impressions, sheets, and depending on implementa or the current copy of the der, the rate at which the varument collation of the job.	pages ation. There ocument
821 822 823	Document collation is de	efined to be collation of de	neets within a document concument concument copies within a masses two types of collation:	ulti-
824	1. External Shee	et Collation		
825	2. <u>Internal Shee</u>	t Collation with Collated	<u>Documents</u>	
826	3. <u>Internal Shee</u>	t Collation with Uncollate	ed Documents	
827 828	Consider the following frameworks:	our variables that are used	d to monitor the progress of	<u>f a job's</u>
829 830	1. jmJobImpre stacked for th		s the total number of impr	<u>essions</u>
831 832	· · · · · · · · · · · · · · · · · · ·	Completed Current Copy e current document copy	- counts the number of im	pressions
833 834	3. currentCopy document bei		number of the copy for the	current
835 836	4. currentDocu that is being s		s the current document wit	hin the job
837 838			two documents (1, 2), who	
839	collation type = External	Sheet Collation		
840				
	jmJobImpressionsCo mpleted	impressionsComplete dCurrentCopy	<u>currentCopyNumber</u>	currentDocumentNu mber
	$\frac{1}{2}$ $\frac{3}{4}$ $\frac{4}{5}$ $\frac{6}{7}$	$ \frac{1}{\frac{1}{2}} $ $ \frac{2}{\frac{2}{3}} $	$\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{2}$ $\frac{3}{1}$	$ \frac{1}{\frac{1}{1}} $ $ \frac{1}{\frac{1}{1}} $ $ \frac{1}{1} $
	<u>/</u>	<u> </u>	<u>1</u>	<u>1</u>

Job Monitoring MIB, V	0.86
-----------------------	------

Sep 19, 1997

89	3 3	$\frac{2}{3}$	<u>1</u>
$\frac{10}{11}$	$\frac{\underline{\underline{\sigma}}}{\underline{1}}$	$\frac{2}{1}$	$\frac{\frac{1}{2}}{2}$
11/12	<u>1</u> 1	$\frac{2}{3}$	$\frac{2}{2}$
<u>13</u>	$\frac{1}{2}$	$\frac{\overline{1}}{2}$	$\frac{\overline{2}}{2}$
$\frac{14}{15}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{2}$
<u>16</u>	$\frac{\overline{3}}{2}$	$\frac{\overline{1}}{2}$	$\frac{\overline{2}}{2}$
$ \frac{8}{9} \\ \underline{10} \\ \underline{11} \\ \underline{12} \\ \underline{13} \\ \underline{14} \\ \underline{15} \\ \underline{16} \\ \underline{17} \\ \underline{18} $	$\frac{\frac{2}{2}}{\frac{2}{3}}$ $\frac{3}{3}$	$\frac{2}{3}$	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

841

842 <u>Collation Type = Internal Collation with document collated within each job copy</u>

843

jmJobImpressionsCo mpleted	impressionsComplete dCurrentCopy	currentCopyNumber	currentDocumentNu mber
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 2 3	1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3	1 1 1 2 2 2 1 1 1 2 2 2 1 1 1 1 2 2 2 2
18	<u>3</u>	<u>3</u>	<u>4</u>

844

Collation Type = Internal Collation with uncollated document copies

846

<u>jmJobImpressionsCo</u> <u>mpleted</u>	impressionsComplete dCurrentCopy	<u>currentCopyNumber</u>	currentDocumentNu mber
$\frac{1}{2}$	$\frac{1}{2}$	<u>1</u>	<u>1</u>
$\frac{\overline{3}}{4}$	$\frac{3}{1}$	$\frac{1}{2}$	$\frac{\overline{1}}{1}$

Job Monitoring	MIB,	V0.86
----------------	------	-------

Sep 19, 1997

$\begin{array}{cccccccccccccccccccccccccccccccccccc$	2 2
1 1 1 1 2 2 2 2 2 2 2 2 2 2 2	1

3.5 Job Identification

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** object that is a text string table index. Being a table index allows a monitoring application to quickly locate and identify a particular job of interest that was submitted from a particular client by the user invoking the monitoring application without having to scan the entire job table. The Job Monitoring MIB needs to provide for identification of the job at both sides of the job submission process. The primary identification point is the client side. The **jmJobSubmissionID** allows the monitoring application to identify the job of interest from all the jobs currently "known" by the server or device. The value of jmJobSubmissionID can be assigned by either the client's local system or a downstream server or device. The point of assignment depends on the job submission protocol in use.

The server/device-side identifier, called the **jmJobIndex** object, SHALL be assigned by the SNMP Job Monitoring MIB agent when the server or device accepts the jobs from submitting clients. The **jmJobIndex** object allows the interested party to obtain all objects desired that relate to a particular job. See Section 3.2, entitled The Job Tables and the Oldest Active and Newest Active Indexes' for the specification of how the agent SHALL assign the **jmJobIndex** values.

The MIB provides a mapping table that maps each **jmJobSubmissionID** value to <u>athe</u> corresponding **jmJobIndex** value generated by the agent, so that an application can determine the correct value for the **jmJobIndex** value for the job of interest in a single Get operation, given the Job Submission ID. See the **jmJobIDGroup**.

In some configurations there may be more than one application program that monitors the same job when the job passes from one network entity to another when it is submitted.

- 873 See configuration 3. In such a case, each application can have its own
- jmJobSubmissionID value. In this case there would be a separate entry in the
- jmJobSubmissionID table, one for each jmJobSubmissionID. Both entries would map
- 876 to the same **jmJobIndex** that contains the job data. When the job is deleted, it is up to
- the agent to remove both entries from the **jmJobSubmissionID** table as well.
- The **jobName** attribute provides a name that the user supplies as a job attribute with the
- job. The **jobName** attribute is not necessarily unique, even for one user, let alone across
- 880 users.

888

889

890

881 **3.6 Internationalization Considerations**

This section describes the internationalization considerations included in this MIB.

883 3.6.1 Text generated by the server or device

- There are a few objects and attributes generated by the server or device that SHALL be
- represented using the Universal Multiple-Octet Coded Character Set (UCS) [ISO-10646].
- These objects and attributes are always supplied (if implemented) by the agent, not by the
- job submitting client:
 - 1. jmGeneralJobSetName object
 - 2. processingMessage(6) attribute
 - 3. physicalDevice(32) (name value) attribute
- The character encoding scheme for representing these objects and attributes SHALL be
- 892 UTF-8 as recommended by RFC 2130 [RFC 2130] and the "IETF Policy on Character
- 893 Sets and Language" [char-set policy]. The 'JmUTF8StringTC' textual convention is used
- to indicate UTF-8 text strings.
- 895 NOTE For strings in 7-bit US-ASCII, there is no impact since the UTF-8 representation
- of 7-bit ASCII is identical to the US-ASCII [US-ASCII] encoding.
- The text contained in the **processingMessage(6)** attribute is generated by the
- 898 server/device. The natural language for the processingMessage(6) attribute is identified
- by the **processingMessageNaturalLanguageTag(7)** attribute. The
- 900 **processingMessageNaturalLanguageTag(7)** attribute uses the
- JmNaturalLanguageTagTC textual convention which SHALL conform to the language
- tag mechanism specified in RFC 1766 [RFC-1766]. The JmNaturalLanguageTagTC
- value is the same as the IPP [IPP-model] 'naturalLanguage' attribute syntax. RFC 1766
- specifies that a US-ASCII string consisting of the natural language followed by an
- optional country field. Both fields use the same two-character codes from ISO 639 [ISO-
- 906 639] and ISO 3166 [ISO-3166], respectively, that are used in the Printer MIB for
- 907 identifying language and country.

908	Examples of the values of the processingMessageNaturalLanguageTag(7) attribute
909	<u>include:</u>
910 911 912 913	 'en' for English 'en-us' for US English 'fr' for French 'de' for German
914	3.6.2 Text suppliedgenerated by the job submitter
915 916 917 918 919 920 921 922 923 924 925	All of the objects and attributes represented by the 'JmJobStringTC' textual-convention are either (1) supplied in the job submission protocol by the client that submits the job to the server or device or (2) are defaulted by the server or device if the job submitting client does not supply values. 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. In any case, the resulting 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, 32 to 127 SHALL be US-ASCII [US-ASCII], 127 SHALL be unused, and the remaining code positions 128 to 255 SHALL represent single-byte or multi-byte graphic characters structured according to ISO 2022 [ISO 2022] or SHALL be unused.
926 927 928 929 930	The coded character set SHALL be one of the ones registered with IANA [IANA] and SHALL be identified by the jobCodedCharSet attribute in the jmJobAttributeTable for the job. If the agent does not know what coded character set was used by the job submitting client, the agent SHALL either (1) return the unknown(2) value for the jobCodedCharSet attribute or (2) not return the jobCodedCharSet attribute for the job.
931 932 933 934 935 936	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, IBM Code Page 850, Windows Default 8-bit set, UTF-8 [UTF-8], US-ASCII plus JIS X0208-1990 Japanese [JIS X0208], US-ASCII plus GB2312-1980 PRC Chinese [GB2312]. See the IANA registry of coded character sets [IANA charsets].
937 938 939 940	Examples of coded character sets which do not meet this criteria are: national 7-bit sets conforming to ISO 646 (except US-ASCII), EBCDIC, and ISO 10646 (Unicode) [ISO-10646]. In order to represent Unicode characters, the UTF-8 [UTF-8] encoding scheme SHALL be used which has been assigned the MIBenum value of '106' by IANA.
941 942	The jobCodedCharSet attribute uses the imported 'CodedCharSet' textual-convention from the Printer MIB [printmib].
943 944 945	The natural language for all attributes represented by the textual-convention JmJobStringTC SHALL be identified by the jobNaturalLanguageTag(8) attribute. The jobNaturalLanguageTag(8) attribute value SHALL have the same syntax and

946	semantics as the processingMessageNaturalLanguageTag (7) attribute, except that the
947	jobNaturalLanguageTag(8) attribute identifies the natural language of attributes
948	supplied by the job submitter instead of the natural language of the
949	processingMessage(6) attribute. See Section 3.5.1.
950	3.6.3 'DateAndTime' for representing the date and time
951	This MIB also contains objects that are represented using the DateAndTime textual
952	convention from SMIv2 [SMIv2-TC]. The job management application SHALL display
953	such objects in the locale of the user running the monitoring application.
954	3.7 IANA Considerations
955	During the development of this standard, the Printer Working Group (PWG) working
956	with IANA [iana] will register additional enums while the standard is in the proposed and
957	draft states according to the procedures described in this section. IANA will handle
958	registration of additional enums after this standard is approved in cooperation with an
959	IANA-appointed registration editor from the PWG according to the procedures described
960	in this section:
961	3.7.1 IANA Registration of enums
962	This specification uses textual conventions to define enumerated values (enums) and bit
963	values. Enumerations (enums) and bit values are sets of symbolic values defined for use
964	with one or more objects or attributes. All enumeration sets and bit value sets are
965	assigned a symbolic data type name (textual convention). As a convention the symbolic
966	name ends in "TC" for textual convention. These enumerations are defined at the
967	beginning of the MIB module specification.
968	This working group has defined several type of enumerations for use in the Job
969	Monitoring MIB and the Printer MIB[print-mib]. These types differ in the method
970	employed to control the addition of new enumerations. Throughout this document,
971	references to "type n enum", where n can be 1, 2 or 3 can be found in the various tables.
972	The definitions of these types of enumerations are:
973	3.7.1.1 Type 1 enumerations
974	Type 1 enumeration: All the values are defined in the Job Monitoring MIB specification
975	(RFC for the Job Monitoring MIB). Additional enumerated values require a new RFC.

There are no type 1 enums in the current draft.

- 977 3.7.1.2 Type 2 enumerations
- 978 Type 2 enumeration: An initial set of values are defined in the Job Monitoring MIB
- 979 specification. Additional enumerated values are registered after review by this working
- group or an editor appointed by IANA after this working group is no longer active.
- The following type 2 enums are contained in the current draft:
- 982 1. JmUTF8StringTC
 - 2. JmJobStringTC
 - 3. <u>JmNaturalLanguageTagTC</u>
- 985 4. JmTimeStampTC

983

1006

- 986 5. JmFinishingTC [same enum values as IPP "finishing" attribute]
- 987 6. JmPrintQualityTC [same enum values as IPP "print-quality" attribute]
- 988 7. JmTonerEconomyTC
- 989 8. JmMediumTypeTC
- 990 9. JmJobSubmission<u>ID</u>TypeTC
- 991 10. <u>JmCollationTypeTC</u>
- 992 11. JmJobStateTC [same enum values as IPP "job-state" attribute]
- 993 12. JmAttributeTypeTC
- 994 For those textual conventions that have the same enum values as the indicated IPP Job
- attribute SHALL be simultaneously registered by IANA for use with IPP [ipp-model] and
- 996 the Job Monitoring MIB.
- 997 3.7.1.3 Type 3 enumeration
- 998 Type 3 enumeration: An initial set of values are defined in the Job Monitoring MIB
- 999 specification. Additional enumerated values are registered through IANA without
- working group review.
- There are no type 3 enums in the current draft.
- 1002 3.7.2 IANA Registration of type 2 bit values
- 1003 This draft contains the following type 2 bit value textual-conventions:
- 1. JmJobServiceTypesTC
- 1005 2. JmJobStateReasons1TC
 - 3. JmJobStateReasons2TC
- 1007 4. JmJobStateReasons3TC
- 5. JmJobStateReasons4TC
- These textual-conventions are defined as bits in an Integer so that they can be used with
- 1010 SNMPv1 SMI. The **jobStateReasons**N (N=1..4) attributes are defined as bit values using
- the corresponding **JmJobStateReasons***N***TC** textual-conventions.
- The registration of **JmJobServiceTypesTC** and **JmJobStateReasonsNTC** bit values
- 1013 SHALL follow the procedures for a type 2 enum as specified in Section 3.7.1.2.

1014	3.7.3 IANA Registration of Job Submission Id Formats
1015 1016 1017	In addition to enums and bit values, this specification assigns a single ASCII digit or letter to various job submission ID formats. See the JmJobSubmissionIDTypeTC textual-convention and the object. The registration of jmJobSubmissionID format
1018	numbers SHALL follow the procedures for a type 2 enum as specified in Section 3.7.1.2.
1019	3.7.4 IANA Registration of MIME types/sub-types for document-formats
1020 1021 1022 1023	The documentFormat (38) attribute has MIME type/sub-type values for indicating document formats which IANA registers as "media type" names. The values of the documentFormat (38) attribute are the same as the corresponding Internet Printing Protocol (IPP) "document-format" Job attribute values [ipp-model].
1024	3.8 Security Considerations
1025	3.8.1 Read-Write objects
1026 1027 1028 1029 1030	All objects are read-only, greatly simplifying the security considerations. If another MIB augments this MIB, that MIB might accept SNMP Write operations to objects in that MIB whose effect is to modify the values of read-only objects in this MIB. However, that MIB SHALL have to support the required access control in order to achieve security, not this MIB.
1031	3.8.2 Read-Only Objects In Other User's Jobs
1032 1033 1034 1035 1036 1037 1038 1039 1040	The security policy of some sites MAY be that unprivileged users can only get the objects from jobs that they submitted, plus a few minimal objects from other jobs, such as the <code>jmJobKOctetsPerCopyRequested</code> and <code>jmJobKOctetsProcessed</code> objects, so that a user can tell how busy a printer is. Other sites MAY allow all unprivileged users to see all objects of all jobs. This MIB does not require, nor does it specify how, such restrictions would be implemented. A monitoring application 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 <code>jmJobOwner</code> object in the <code>jmJobTable</code> does not match the user's user name.
1041 1042	An operator is a privileged user that would be able to see all objects of all jobs, independent of the policy for unprivileged users.
1043	3.9 Notifications
1044 1045	This MIB does not specify any notifications. For simplicity, management applications are expected to poll for status. The jmGeneralJobPersistence and

- jmGeneralAttributePersistence objects assist an application to determine the polling rate. The resulting network traffic is not expected to be significant.
- 1048 **4. MIB specification**
- The following pages constitute the actual Job Monitoring MIB.

```
1050
       Job-Monitoring-MIB DEFINITIONS ::= BEGIN
1051
1052
       IMPORTS
             MODULE-IDENTITY, OBJECT-TYPE, enterprises experimental,
             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
1053
1054
       -- Use the enterprises arc assigned to the PWG (2699).
1055
       -- Assign two arcs under that: standard(1) and experimental(2)
1056
        -- for all PWG usage.
       -- Use the experimental arc until the PWG agrees that the MIB
1057
1058
       -- is approved as a PWG standard.
1059
       experimental (54) OID assigned to the Printer MIB[print-mib]
1060
       -- before it was published as RFC 1759.
1061
       -- Upon publication of the Job Monitoring MIB as a PWG standard
       -- and as an Informational RFC, change the second to last arc
1062
1063
       -- from experimental(2) to standard(1).delete this
        -- comment and the line following this comment and change the
1064
       - reference of { temp 105 } (below) to { mib-2 X }.
1065
1066
       -- This will result in changing:
1067
       -1361354 jobmonMIB(105) to:
1068
       -13612 1 jobmonMIB(X)
       -- This will make it easier to translate prototypes to
1069
1070
       -- the standard namespace because the lengths of the OIDs won't
1071
       -- change.
1072
       temp OBJECT IDENTIFIER ::= { experimental 54 }
1073
1074
       jobmonMIB MODULE-IDENTITY
1075
             LAST-UPDATED "97120209190000Z"
1076
             ORGANIZATION "IETF Printer MIB Working Group"
1077
             CONTACT-INFO
1078
                  "Tom Hastings
                  Postal: Xerox Corp.
1079
                       Mail stop ESAE-231
1080
1081
                       701 S. Aviation Blvd.
1082
                       El Segundo, CA 90245
1083
1084
                  Tel:
                         (301)333-6413
```

(301)333-5514

E-mail: hastings@cp10.es.xerox.com

1085

```
1087
1088
                   Send comments to the printmib WG using the Job Monitoring
1089
                   Project (JMP) Mailing List: jmp@pwg.org
1090
1091
                   To learn how to subscribe to the JMP mailing list,
1092
                   send email to: imp-request@pwg.org
1093
1094
                   For further information, access the PWG web page under 'JMP':
1095
                   http://www.pwg.org/"
1096
             DESCRIPTION
1097
                   "The MIB module for monitoring job in servers, printers, and other devices."
1098
1099
                   File: draft-ietf-printmib-job-monitor-076.txt
                   Version: 0.876"
1100
1101
             ::= { enterprises pwg(2699) experimental(2) jobmon(1)<del>temp 105</del> }
1102
1103
1104
1105
        -- Textual conventions for this MIB module
1106
1107
1108
1109
        JmUTF8StringTC ::= TEXTUAL-CONVENTION
1110
             DISPLAY-HINT "255a"
1111
             STATUS
                         current
             DESCRIPTION
1112
                   "To facilitate internationalization, this TC represents information taken from the ISO/IEC IS
1113
1114
                   10646-1 character set, encoded as an octet string using the UTF-8 character encoding scheme."
1115
             REFERENCE
                   "See section 3.6.1, entitled: Text generated by the server or device'."
1116
                          OCTET STRING (SIZE (0..63))
1117
             SYNTAX
1118
1119
1120
1121
1122
        JmJobStringTC ::= TEXTUAL-CONVENTION
1123
             STATUS
                         current
1124
             DESCRIPTION
                   "To facilitate internationalization, this TC represents information using any coded character set
1125
                   registered by IANA as specified in section 0. While it is recommended that the coded character
1126
1127
                   set be UTF-8 [UTF-8], the actual coded character set SHALL be indicated by the value of the
1128
                   jobCodedCharSet(87) attribute for the job."
             REFERENCE
1129
1130
                   "See section 0, entitled: The text contained in the processing Message (6) attribute is generated
1131
                   by the server/device. The natural language for the processingMessage(6) attribute is identified
1132
                   by the processingMessageNaturalLanguageTag(7) attribute. The
                   processingMessageNaturalLanguageTag(7) attribute uses the JmNaturalLanguageTagTC
1133
```

1134 1135 1136 1137 1138 1139	textual convention which SHALL conform to the language tag mechanism specified in RFC 1766 [RFC-1766]. The JmNaturalLanguageTagTC value is the same as the IPP [IPP-model 'naturalLanguage' attribute syntax. RFC 1766 specifies that a US-ASCII string consisting of the natural language followed by an optional country field. Both fields use the same two-character codes from ISO 639 [ISO-639] and ISO 3166 [ISO-3166], respectively, that are used in the Printer MIB for identifying language and country.
1140	Examples of the values of the processingMessageNaturalLanguageTag(7) attribute include:
1141 1142 1143 1144 1145 1146 1147 1148 1149	1. 'en' for English 2. 'en-us' for US English 3. 'fr' for French 4. 'de' for German Text suppliedgenerated by the job submitter'." SYNTAX OCTET STRING (SIZE (063))
1150	
1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163	STATUS current DESCRIPTION "An IETF RFC 1766-compliant 'language tag', with zero or more sub-tags that identify a natural language. While RFC 1766 specifies that the US-ASCII values are case-insensitive, this MIB specification requires that all characters SHALL be lower case in order to simplify comparing by management applications." REFERENCE "See section 3.6.1, entitled: Text generated by the server or device' and section 3.6.2, entitled: Text supplied generated by the job submitter'." SYNTAX OCTET STRING (SIZE (063))
1164 1165 1166 1167 1168 1169	JmTimeStampTC ::= TEXTUAL-CONVENTION STATUS current DESCRIPTION "The simple time at which an event took place. The units SHALL be in seconds since the system was booted.
1170 1171 1172 1173 1174	NOTE - JmTimeStampTC is defined in units of seconds, rather than 100ths of seconds, so as to be simpler for agents to implement (even if they have to implement the 100ths of a second to comply with implementing sysUpTime in MIB-II[mib-II].)
1175 1176 1177 1178 1179	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 jmAttributeValueAsInteger object. The TimeStamp textual-convention defined in SMNPv2-TC is defined as an APPLICATION 3 IMPLICIT INTEGER tag, not an Integer32 , so cannot be used in this MIB as one of the values of jmAttributeValueAsInteger ."
1180	SYNTAX INTEGER(02147483647)

```
1181
1182
1183
1184
1185
       JmJobSourcePlatformTypeTC ::= TEXTUAL-CONVENTION
1186
            STATUS
                        current
1187
            DESCRIPTION
1188
                  "The source platform type that can submit jobs to servers or devices in any of the 3
1189
                 configurations."
1190
            REFERENCE
1191
                  "This is a type 2 enumeration. See Section 3.7.1.2."
1192
            SYNTAX
                        INTEGER {
                   other(1),
                   unknown(2),
                   sptUNIX(3),
                                                   UNIX
                                                -- OS/2
                   sptOS2(4),
                   sptPCDOS(5),
                                                -- DOS
                                                -- NT
                   sptNT(6),
                   sptMVS(7),
                                                -- MVS
                                               -- VM
                   sptVM(8),
                                               -- OS/400
                   sptOS400(9),
                                                -- VMS
                   sptVMS(10),
                                               -- Windows
                   sptWindows(11),
                   sptNetWare(12)
                                               -- NetWare
1193
            }
1194
1195
1196
1197
1198
1199
       JmFinishingTC ::= TEXTUAL-CONVENTION
1200
            STATUS
                       current
1201
            DESCRIPTION
1202
                  "The type of finishing operation.
1203
                  These values are the same as the enum values of the IPP 'finishings' attribute. See Section
1204
1205
                  3.7.1.2.
1206
1207
                 other(1),
1208
                       Some other finishing operation besides one of the specified or registered values.
1209
1210
                  unknown(2),
1211
                       The finishing is unknown.
1212
1213
                 none(3),
                       Perform no finishing.
1214
1215
```

```
1216
                   staple(4),
1217
                         Bind the document(s) with one or more staples. The exact number and placement of the
1218
                         staples is site-defined.
1219
1220
                   punch(5),
1221
                         This value indicates that holes are required in the finished document. The exact number
                         and placement of the holes is site-defined. The punch specification MAY be satisfied (in
1222
1223
                         a site- and implementation-specific manner) either by drilling/punching, or by
1224
                         substituting pre-drilled media.
1225
1226
                   cover(6).
                         This value is specified when it is desired to select a non-printed (or pre-printed) cover for
1227
1228
                         the document. This does not supplant the specification of a printed cover (on cover stock
1229
                         medium) by the document itself.
1230
1231
                   bind(7)
                         This value indicates that a binding is to be applied to the document; the type and
1232
1233
                         placement of the binding is product-specific."
1234
              REFERENCE
1235
                   "This is a type 2 enumeration. See Section 3.7.1.2."
1236
              SYNTAX
                           INTEGER {
1237
                   other(1).
1238
                   unknown(2),
1239
                   none(3),
1240
                   staple(4),
1241
                   punch(5),
1242
                   cover(6),
1243
                   bind(7)
1244
              }
1245
1246
1247
1248
1249
1250
        JmPrintQualityTC ::= TEXTUAL-CONVENTION
1251
              STATUS
                         current
1252
              DESCRIPTION
1253
                   "Print quality settings.
1254
1255
                   These values are the same as the enum values of the IPP 'print-quality' attribute. See Section
                   3.7.1.2."
1256
1257
              REFERENCE
                   "This is a type 2 enumeration. See Section 3.7.1.2."
1258
1259
              SYNTAX
                           INTEGER {
                                            Not one of the specified or registered values.
                     other(1),
                     unknown(2),
                                            The actual value is unknown.
                     draft(3),
                                            Lowest quality available on the printer.
```

```
normal(4),
                                         Normal or intermediate quality on the printer.
                   high(5)
                                         Highest quality available on the printer.
1260
             }
1261
1262
1263
1264
1265
       JmPrinterResolutionTC ::= TEXTUAL-CONVENTION
1266
             STATUS
                        current
             DESCRIPTION
1267
1268
                  "Printer resolutions.
1269
1270
                  Nine octets consisting of two 4-octet SIGNED-INTEGERs followed by a SIGNED-BYTE. The
                  values are the same as those specified in the Printer MIB [printmib]. The first SIGNED-
1271
1272
                  INTEGER contains the value of prtMarkerAddressabilityXFeedDir. The second SIGNED-
1273
                  INTEGER contains the value of prtMarkerAddressabilityFeedDir. The SIGNED-BYTE
1274
                  contains the value of prtMarkerAddressabilityUnit.
1275
1276
                  Note: the latter value is either 3 (tenThousandsOfInches) or 4 (micrometers) and the
                  addressability is in 10,000 units of measure. Thus the SIGNED-INTEGERs represent integral
1277
1278
                  values in either dots-per-inch or dots-per-centimeter.
1279
                  The syntax is the same as the IPP 'printer-resolution' attribute. See Section3.7.1.2."
1280
1281
                         OCTET STRING (SIZE(9))
             SYNTAX
1282
1283
1284
1285
1286
1287
       JmTonerEconomyTC ::= TEXTUAL-CONVENTION
1288
             STATUS
                        current
1289
             DESCRIPTION
1290
                  "Toner economy settings."
1291
             REFERENCE
1292
                  "This is a type 2 enumeration. See Section 3.7.1.2."
1293
                         INTEGER {
             SYNTAX
                   unknown(2).
                                             unknown.
                                             Off. Normal. Use full toner.
                   off(3),
                                             On. Use less toner than normal.
                   on(4)
1294
             }
1295
1296
1297
1298
1299
       JmBooleanTC ::= TEXTUAL-CONVENTION
1300
```

```
1301
             STATUS
                         current
1302
             DESCRIPTION
                   "Boolean true or false value."
1303
1304
             REFERENCE
1305
                   "This is a type 2 enumeration. See Section 3.7.1.2."
1306
             SYNTAX
                          INTEGER {
                    unknown(2),
                                              unknown.
                    false(3),
                                              FALSE.
                                              TRUE.
                    true(4)
1307
             }
1308
1309
1310
1311
1312
1313
       JmMediumTypeTC ::= TEXTUAL-CONVENTION
1314
             STATUS
                         current
             DESCRIPTION
1315
1316
                   "Identifies the type of medium.
1317
1318
                   other(1),
1319
                        The type is neither one of the values listed in this specification nor a registered value.
1320
1321
                   unknown(2),
1322
                        The type is not known.
1323
1324
                   stationery(3),
1325
                        Separately cut sheets of an opaque material.
1326
1327
                   transparency(4),
1328
                        Separately cut sheets of a transparent material.
1329
1330
1331
                        Envelopes that can be used for conventional mailing purposes.
1332
1333
                   envelopePlain(6),
1334
                        Envelopes that are not preprinted and have no windows.
1335
1336
                   envelopeWindow(7),
1337
                        Envelopes that have windows for addressing purposes.
1338
1339
                   continuousLong(8),
1340
                        Continuously connected sheets of an opaque material connected along the long edge.
1341
1342
                   continuousShort(9).
                        Continuously connected sheets of an opaque material connected along the short edge.
1343
```

```
1345
                   tabStock(10),
                        Media with tabs.
1346
1347
1348
                   multiPartForm(11).
1349
                        Form medium composed of multiple layers not pre-attached to one another; each sheet
                        MAY be drawn separately from an input source.
1350
1351
1352
                   labels(12),
                        Label-stock.
1353
1354
1355
                   multiLaver(13)
1356
                        Form medium composed of multiple layers which are pre-attached to one another, e.g. for
1357
                        use with impact printers."
1358
             REFERENCE
1359
                   "This is a type 2 enumeration. See Section 3.7.1.2."
1360
             SYNTAX
                          INTEGER {
1361
                   other(1),
1362
                   unknown(2),
1363
                   stationery(3),
1364
                   transparency(4),
1365
                   envelope(5),
                   envelopePlain(6),
1366
1367
                   envelopeWindow(7),
1368
                   continuousLong(8),
                   continuousShort(9).
1369
                   tabStock(10).
1370
                   multiPartForm(11),
1371
1372
                   labels(12),
1373
                   multiLayer(13)
1374
             }
1375
1376
1377
1378
1379
1380
        JmCollationTypeTC ::= TECTUAL-CONVENTION
1381
             STATUS current
1382
             DESCRIPTION
1383
                   "This value is the type of sheet and document collation."
1384
1385
                   other(1),
1386
                        Some other collation besides one of the specified or registered values.
1387
1388
                   unknown(2),
1389
                        The collation is unknown.
1390
1391
                   externalSheetCollation(3),
                        Collation of the sheets within a document copy is performed externally to the printing
1392
```

1393	device, either in an attached physical output bin collator or is uncollated (so that the user
1394	does the sheet collation by hand).
1395	
1396	Note that uncollated and collation to a series of output bins are the same in terms of the
1397	behavior of the job MIB Impression and Sheet completed attributes. Therefore, we call
1398	this form External Sheet Collation.
1399	<u> </u>
1400	internalSheetCollationWithCollatedDocs(4),
1401	Collation of the sheets within each document copy is performed within the printing
1402	device by making multiple passes over either the source or an intermediate representation
1403	of the document. In addition, when there are multiple documents per job, the i'th copy of
1404	each document is stacked before the j'th copy of each document, i.e., the documents are
1405	collated within each job copy.
1406	<u>-on-the first for topy:</u>
1407	If jobCopiesRequested or documentCopiesRequested = 1, then collationType is
1408	defined as 4.
1409	
1410	internalSheetCollationWithUnCollatedDocs(5),
1411	Collation of the sheets within each document copy is performed within the printing
1412	device by making multiple passes over either the source or an intermediate representation
1413	of the document. In addition, when there are multiple documents per job, all copies of
1414	the first document in the job are stacked before the any copied of the next document in
1415	the job, i.e., the documents are uncollated within the job.
1416	
1417	REFERENCE
1418	"This is a type 2 enumeration. See Section 3.7.1.2."
1419	SYNTAX INTEGER {
1420	other(1),
1421	$\frac{1}{\text{unknown}(2)}$
1422	externalSheetCollation(3),
1423	internalSheetCollationWithCollatedDocs(4),
1424	internalSheetCollationWithUnCollatedDocs(5),
1425	
1426	
1427	
1428	JmJobSubmissionTypeIDTC ::= TEXTUAL-CONVENTION
1429	STATUS current
1430	DESCRIPTION
1431	"Identifies the format type of a job submission ID.
1432	
1433	Each job submission ID is a fixed-length, 48-octet printable US-ASCII [US-ASCII] coded
1434	character string containing no control characters, consisting of the following fields:
1435	
1436	octet 1 The format letter identifying the format.
1437	The US-ASCII characters '0-9', 'A-Z', and 'a-z'
1438	are assigned in order giving 62 possible
1439	formats.
1440	octets 2-40 A 39-character, US-ASCII trailing SPACE filled
1441	field specified by the format letter, if the

1442 1443 1444 1445	data is less than 39 ASCII characters. octets 41-48 A sequential or random number to make the ID quasi-unique.
1446 1447 1448 1449 1450	If the client does not supply a job submission ID in the job submission protocol, then the agent SHALL assign a job submission ID using any of the standard formats that are reserved for the agent. Clients SHALL not use formats that are reserved for agents and agents SHALL NOT use formats that are reserved for clients, in order to reduce conflicts in ID generation. See the description for which formats are reserved for clients or for agents.
1451 1452 1453 1454	Registration of additional formats may be done following the procedures described in Section 3.7.3.
1455 1456	The format values defined at the time of completion of this specification are:
1457 1458	Format Letter Description
1459 1460 1461 1462 1463	octets 2-40: last 39 bytes of the jmJobOwner object. octets 41-48: 8-decimal-digit sequential number. This format is reserved for agents.
1464 1465 1466 1467 1468	NOTE - Clients wishing to use a job submission ID that incorporates the job owner, SHALL use format '8', not format '0'.
1468 1469 1470 1471 1472	octets 2-40: last 39 bytes of the jobName attribute. octets 41-48: 8-decimal-digit random number. This format is reserved for clients.
1473 1474 1475 1476 1477 1478	 '2' octets 2-40: Client MAC address: in hexadecimal with each nibble of the 6 octet address being '0'-'9' or 'A' - 'F' (uppercase only). Most significant octet first. octets 41-48: 8-decimal-digit sequential number This format is reserved for clients.
1479 1480 1481 1482 1483 1484	octets 2-40: last 39 bytes of the client URL [URI-spec]. octets 41-48: 8-decimal-digit sequential number This format is reserved for clients.
1484 1485 1486 1487 1488 1489 1490	'4' octets 2-40: last 39 bytes of the URI [URI-spec] assigned by the server or device to the job when the job was submitted for processing. octets 41-48: 8-decimal-digit sequential number This format is reserved for agents.

1491	'5' octets 2-40: last 39 bytes of a user number, such
1492	as POSIX user number.
1493	octets 41-48: 8-decimal-digit sequential number
1494	This format is reserved for clients.
1495	1.11.0 101.11.00 10 10 10 10 10 10 10 10 10 10 10 10 1
1496	'6' octets 2-40: last 39 bytes of the user account
1497	number.
1498	octets 41-48: 8-decimal-digit sequential number
1499	This format is reserved for clients.
1500	This format is reserved for elicitis.
1501	7' octets 2-40: last 39 bytes of the DTMF incoming
1502	FAX routing number.
1503	octets 41-48: 8-decimal-digit sequential number
1504	This format is reserved for clients.
1505	This format is reserved for chems.
1505	'8' octets 2-40: last 39 bytes of the job owner name
1507	\mathcal{J}
	(that the agent returns in the jmJobOwner object).
1508	octets 41-48: 8-decimal-digit sequential number
1509	This format is reserved for clients.
1510	'9' octets 2-40: last 39 bytes of the host name with
1511	y sources = . or institute of the most institute with
1512	trailing SPACES that submitted the job to this
1513	server/device using a protocol, such as LPD
1514	[RFC-1179] which includes the host name in the job
1515	submission protocol.
1516	octets 41-48: 8-decimal-digit leading zero
1517	representation of the job id generated by the
1518	by the submitting server (configuration 3)
1519	or the client (configuration 1 and 2), such as in
1520	the LPD protocol.
1521	This format is reserved for clients.
1522	
1523	NOTE - the job submission id is only intended to be unique between a limited set of clients for
1524	a limited duration of time, namely, for the life time of the job in the context of the server or
1525	device that is processing the job. Some of the formats include something that is unique per
1526	client and a random number so that the same job submitted by the same client will have a
1527	different job submission id. For other formats, where part of the id is guaranteed to be unique
1528	for each client, such as the MAC address or URL, a sequential number SHOULD suffice for
1529	each client (and may be easier for each client to manage). Therefore, the length of the job
1530	submission id has been selected to reduce the probability of collision to an extremely low
1531	number, but is not intended to be an absolute guarantee of uniqueness. None-the-less,
1532	collisions are remotely possible, but without bad consequences, since this MIB is intended to be
1533	used only for monitoring jobs, not for controlling and managing them."
1534	REFERENCE
1535	"This is like a type 2 enumeration. See section 3.7.3."
1536	SYNTAX OCTEŤ STRING(SIZE(1)) ASCII '0'-'9', 'A'-'Z', 'a'-'z'
1527	

JmJobStateTC ::= TEXTUAL-CONVENTION STATUS current

DESCRIPTION

"The current state of the job (**pending**, **processing**, **completed**, etc.).

The following figure shows the normal job state transitions:

Figure 4 - Normal Job State Transitions

Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden. Not shown are the transitions to the **canceled** state from the **pending**, pendingHeld, and processingStopped states.

Jobs in the **pending**, **processing**, and **processingStopped** states are called 'active', while jobs in the **pendingHeld**, **canceled**, **aborted**, and **completed** states are called 'inactive'. Jobs reach one of the three terminal states: **completed**, **canceled**, or **aborted**, *after* the jobs have completed all activity, and all MIB objects and attributes have reached their final values for the

These values are the same as the enum values of the IPP 'job-state' job attribute. See Section 3.7.1.2.

unknown(2).

The job state is *not* known, or its state is indeterminate.

pending(3),

The job is a candidate to start processing, but is not yet processing.

pendingHeld(4),

The job is not a candidate for processing for any number of reasons but will return to the **pending** state as soon as the reasons are no longer present. The job's jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4) attributes SHALL indicate why the job is no longer a candidate for processing. The reasons are represented as bits in the jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4) attributes. See the **JmJobStateReasonsNTC** (N=1..4) textual convention for the specification of each reason.

1587	
1587 1588	
1589	
1590	
1591	
1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600	
1594	
1595	
1596	
1597	
1598	
1599	
1601	
1602	
1603	
1604	
1605	
1601 1602 1603 1604 1605 1606 1607	
1608	
1608 1609 1610 1611 1612 1613 1614 1615 1616 1617	
1610	
1611	
1612	
1613	
1615	
1616	
1617	
1618	
1619	
1620 1621	
1621	
1623	
1624	
1625	
1626	
1619 1620 1621 1622 1623 1624 1625 1626 1627	
111/0	

1630

1631

1632 1633

1634

1635

processing(5),

One or more of:

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

OR

3. (configuration 2) the server has made the job ready for printing, but the output device is not yet printing it, either because the job hasn't reached the output device or because the job is queued in the output device or some other spooler, awaiting the output device to print it.

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

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

processingStopped(6),

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

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

NOTE - When an output device is stopped, the device usually indicates its condition in human readable form at the device. The management application can obtain more complete device status remotely by querying the appropriate device MIB using the job's **deviceIndex** attribute(s), if the agent implements such a device MIB

canceled(7).

A client has canceled the job and the server or device has completed canceling the job *AND* all MIB objects and attributes have reached their final values for the job. While the server or device is canceling the job, the job's **jmJobStateReasons1** object SHOULD contain the **processingToStopPoint** value and one of the **canceledByUser**, **canceledByOperator**, or **canceledAtDevice** values. The **canceledByUser**, **canceledByOperator**, or **canceledAtDevice** values remain while the job is in the **canceled** state.

```
1636
1637
                   aborted(8),
1638
                         The job has been aborted by the system, usually while the job was in the processing or
                         processingStopped state and the server or device has completed aborting the job AND all
1639
1640
                         MIB objects and attributes have reached their final values for the job. While the server or
                         device is aborting the job, the job's jmJobStateReasons1 object MAY contain the
1641
1642
                         processing ToStopPoint and abortedBySystem values. If implemented, the
1643
                         abortedBySystem value SHALL remain while the job is in the aborted state.
1644
1645
                   completed(9)
1646
                         The job has completed successfully or with warnings or errors after processing and all of
1647
                         the media have been successfully stacked in the appropriate output bin(s) AND all MIB
                         objects and attributes have reached their final values for the job. The job's
1648
1649
                         imJobStateReasons1 object SHOULD contain one of: completedSuccessfully,
1650
                         completedWithWarnings, or completedWithErrors values."
1651
             REFERENCE
1652
                   "This is a type 2 enumeration. See Section 3.7.1.2."
                          INTEGER {
1653
             SYNTAX
1654
                   unknown(2),
1655
                   pending(3).
                   pendingHeld(4),
1656
1657
                   processing(5),
1658
                   processingStopped(6),
1659
                   canceled(7),
                   aborted(8).
1660
1661
                   completed(9)
1662
              }
1663
1664
1665
        JmAttributeTypeTC ::= TEXTUAL-CONVENTION
1666
              STATUS
                          current
1667
             DESCRIPTION
                   "The type of the attribute which identifies the attribute."
1668
1669
1670
                   In the following definitions of the enums, each description indicates whether the useful value of
                   the attribute SHALL be represented using the jmAttributeValueAsInteger or the
1671
                   jmAttributeValueAsOctets objects by the initial tag: INTEGER: or OCTETS:',
1672
1673
                   respectively.
1674
1675
                   Some attributes allow the agent implementer a choice of useful values of either an integer, an
                   octets representation, or both, depending on implementation. These attributes are indicated
1676
                   with 'INTEGER:' AND/OR OCTETS:' tags.
1677
1678
1679
                   A very few attributes require both objects at the same time to represent a pair of useful values
```

(see mediumConsumed(171)). These attributes are indicated with INTEGER: AND

'OCTETS:' tags. See the **imAttributeGroup** for the descriptions of these two MANDATORY

objects.

1680 1681

1682

1684 NOTE - The enum assignments are grouped logically with values assigned in groups of 20, so that additional values may be registered in the future and assigned a value that is part of their 1685 1686 logical grouping. 1687 Values in the range $2^{**}30$ to $2^{**}31-1$ are reserved for private or experimental usage. This 1688 range corresponds to the same range reserved in IPP. Implementers are warned that use of such 1689 1690 values may conflict with other implementations. Implementers are encouraged to request 1691 registration of enum values following the procedures in Section 3.7.1. 1692 1693 NOTE: No attribute name exceeds 31 characters. 1694 1695 The standard attribute types defined at the time of completion of the specification are: 1696 1697 **jmAttributeTypeIndex Datatype** 1698 1699 1700 other(1), Integer32(-2..2147483647) 1701 AND/OR 1702 OCTET STRING(SIZE(0..63)) 1703 INTEGER: and/or OCTETS: An attribute that is not in the list and/or that has not been 1704 approved and registered with IANA. 1705 1706 1707 1708 + Job State attributes 1709 1710 + The following attributes specify the state of a job. 1711 1712 1713 jobStateReasons2(3), JmJobStateReasons2TC INTEGER: Additional information about the job's current state that augments the 1714 1715 jmJobState object. See the description under the JmJobStateReasons1TC textual-1716 convention. 1717 1718 jobStateReasons3(4), JmJobStateReasons3TC 1719 INTEGER: Additional information about the job's current state that augments the jmJobState object. See the description under JmJobStateReasons1TC textual-1720 1721 convention. 1722 1723 jobStateReasons4(5), JmJobStateReasons4TC 1724 INTEGER: Additional information about the job's current state that augments the 1725 jmJobState object. See the description under JmJobStateReasons1TC textual-1726 convention. 1727 JmUTF8StringTC(SIZE(0..63)) 1728 processingMessage(6), OCTETS: MULTI-ROW: A coded character set message that is generated by the server 1729 1730 or device during the processing of the job as a simple form of processing log to show 1731 progress and any problems. The natural language of each value is specified by the corresponding processing Message Natural Language Tag (7) value. 1732

1733	
1734 1735	
1735 1736	ı
1736	
1737	
1738	
1739	
1740	
1741	
1742	
1743	
1744	
1745	
1746	
1747	
1748	
1749	
1750	
1751	
1752	
1753	
1754	
1755	
1756	
1757	
1758	
1759	
1760	
1761	
1762	
1763	
1764	
1765	
1766	
1767	
1768	
1769	
1770	
1771	
1772	
1773	
1774	
1775	
1776	
1777	
1778	
1779	1
1780	
1781	
1/01	

There is no restriction for the same message occurring in multiple rows.

processingMessageNaturalLanguageTag(7), OCTET STRING(SIZE(2..63))

OCTETS: MULTI-ROW: The natural language of the corresponding processingMessage(6) attribute. See section 3.6.1, entitled Text generated by the server or device.

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 **processingMessageNaturalLanguageTag(7)** attribute or (2) not return the **processingMessageNaturalLanguageTag(7)** attribute for the job.

There is no restriction for the same tag occurring in multiple rows.

jobCodedCharSet(87), 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(87)** attribute. See section 3.6.2, entitled Text suppliedgenerated by the job submitter.

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 used by the job submitting client, the agent SHALL either (1) return the **unknown(2)**' value for the **jobCodedCharSet**_attribute or (2) not return the **jobCodedCharSet**_attribute for the job.

jobNaturalLanguageTag(9), OCTET STRING(SIZE(2..63))

OCTETS: The natural language of the job attributes supplied by the job submitter or defaulted by the server or device for the job, i.e., all objects and attributes represented by the 'JmJobStringTC' textual-convention, such as jobName, mediumRequested, etc. See Section 3.6.2, entitled Text suppliedgenerated by the job submitter.

If the agent does not know what natural language was used by the job submitting client, the agent SHALL either (1) return a zero length string value for the jobNaturalLanguageTag(9) attribute or (2) not return jobNaturalLanguageTag(9) attribute for the job.

1782	+ Job Identification attributes
1783	+
1784	+ The following attributes help an end user, a system
1785	+ operator, or an accounting program identify a job.
1786	+++++++++++++++++++++++++++++++++++++++
1787	
1788	
1789	
1790	jobURI(20), OCTET STRING(SIZE(163255))
1791	OCTETS: MULTI-ROW: The job's Universal Resource Identifier (URI) [RFC-1738].
1792	See IPP [ipp-model] for example usage.
1793	see if I [Ipp moder] for example usuge.
1794	NOTE - The agent may be able to generate this value on each SNMP Get operation from
1795	smaller values, rather than having to store the entire URI.
1796	Smaller values, rather than having to store the entire OKI.
1797	If the LIDI exceeds 62255 potets, the egent SHALL use multiple values, with the part 6t2
	If the URI exceeds 63255 octets, the agent SHALL use multiple values, with the next 6t3
1798	octets coming in the second value, etc. truncate from the beginning (since the end tends to
1799	be more unique than the beginning).
1800	
1801	jobAccountName(21), OCTET STRING(SIZE(063))
1802	OCTETS: Arbitrary binary information which MAY be coded character set data or
1803	encrypted data supplied by the submitting user for use by accounting services to allocate
1804	or categorize charges for services provided, such as a customer account name or number.
1805	
1806	NOTE: This attribute NEED NOT be printable characters.
1807	
1808	serverAssignedJobName(22), JmJobStringTC(SIZE(063))
1809	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the
1809 1810	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is
1809 1810 1811	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the
1809 1810 1811 1812	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB.
1809 1810 1811 1812 1813	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is
1809 1810 1811 1812 1813 1814	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB.
1809 1810 1811 1812 1813	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server
1809 1810 1811 1812 1813 1814	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or
1809 1810 1811 1812 1813 1814 1815	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device.
1809 1810 1811 1812 1813 1814 1815 1816	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device.
1809 1810 1811 1812 1813 1814 1815 1816 1817	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23), JmJobStringTC(SIZE(063)) OCTETS: The human readable string name of the job as assigned by the submitting user
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23), JmJobStringTC(SIZE(063)) OCTETS: The human readable string name of the job as assigned by the submitting user to help the user distinguish between his/her various jobs. This name does not need to be
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23), JmJobStringTC(SIZE(063)) OCTETS: The human readable string name of the job as assigned by the submitting user
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23), JmJobStringTC(SIZE(063)) OCTETS: The human readable string name of the job as assigned by the submitting user to help the user distinguish between his/her various jobs. This name does not need to be unique. This attribute is intended for enabling a user or the user's application to convey a job
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23), JmJobStringTC(SIZE(063)) OCTETS: The human readable string name of the job as assigned by the submitting user to help the user distinguish between his/her various jobs. This name does not need to be unique. This attribute is intended for enabling a user or the user's application to convey a job
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826	OCTĒTS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23), JmJobStringTC(SIZE(063)) OCTETS: The human readable string name of the job as assigned by the submitting user to help the user distinguish between his/her various jobs. This name does not need to be unique. This attribute is intended for enabling a user or the user's application to convey a job name that MAY be printed on a start sheet, returned in a query result, or used in notification or logging messages. In order to assist users to find their jobs for job submission protocols that don't supply a
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),
1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828	OCTETS: Configuration 3 only: The human readable string name, number, or ID of the job as assigned by the server that submitted the job to the device that the agent is providing access to with this MIB. NOTE - This attribute is intended for enabling a user to find his/her job that a server submitted to a device when either the client does not support the jmJobSubmissionID or the server does not pass the jmJobSubmissionID through to the device. jobName(23),

If this attribute is not specified when the job is submitted, no job name is assumed, but implementation specific defaults are allowed, such as the value of the **documentName** attribute of the first document in the job or the **fileName** attribute of the first document in the job.

The **jobName** attribute is distinguished from the **jobComment** attribute, in that the **jobName** attribute is intended to permit the submitting user to distinguish between different jobs that he/she has submitted. The **jobComment** attribute is intended to be free form additional information that a user might wish to use to communicate with himself/herself, such as a reminder of what to do with the results or to indicate a different set of input parameters were tried in several different job submissions.

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.

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.

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.

jobSourceChannelIndex(25),

Integer32(0..2147483647)

INTEGER: The index of the row in the associated Printer MIB[print-mib] of the channel which is the source of the print job.

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.

submittingServerName(27),

JmJobStringTC(SIZE(0..63))

OCTETS: For configuration 3 only: The administrative name of the server that submitted the job to the device.

submittingApplicationName(28),

JmJobStringTC(SIZE(0..63))

OCTETS: The name of the client application (not the server in configuration 3) that submitted the job to the server or device.

1880	
1881	jobOriginatingHost(29), JmJobStringTC(SIZE(063))
1882	OCTETS: The name of the client host (not the server host name in configuration 3) that
1883	submitted the job to the server or device.
1884	swormtow and job to the serior of active.
1885	deviceNameRequested(30), JmJobStringTC(SIZE(063))
1886	OCTETS: The administratively defined coded character set name of the target device
1887	requested by the submitting user. For configuration 1, its value corresponds to the Printer
1888	MIB[print-mib]: prtGeneralPrinterName object. For configuration 2 and 3, its value is
1000	
1889	the name of the logical or physical device that the user supplied to indicate to the server
1890	on which device(s) they wanted the job to be processed.
1891	N D (21)
1892	queueNameRequested(31), JmJobStringTC(SIZE(063))
1893	OCTETS: The administratively defined coded character set name of the target queue
1894	requested by the submitting user. For configuration 1, its value corresponds to the queue
1895	in the device for which the agent is providing access. For configuration 2 and 3, its value
1896	is the name of the queue that the user supplied to indicate to the server on which device(s)
1897	they wanted the job to be processed.
1898	
1899	NOTE - typically an implementation SHOULD support either the deviceNameRequested
1900	or queueNameRequested attribute, but not both.
1901	
1902	physicalDevice(32), hrDeviceIndex
1903	AND/OR
1904	JmUTF8StringTC(SIZE(063))
1905	INTEGER: MULTI-ROW: The index of the physical device MIB instance
1906	requested/used, such as the Printer MIB[print-mib]. This value is an hrDeviceIndex
1907	value. See the Host Resources MIB[hr-mib].
1908	
1909	AND/OR
1910	
1911	OCTETS: MULTI-ROW: The name of the physical device to which the job is assigned.
1912	
1913	numberOfDocuments(33), Integer32(-22147483647)
1914	INTEGER: The number of documents in this job.
1915	J
1916	The agent SHOULD return this attribute if the job has more than one document.
1917	g
1918	fileName(34), JmJobStringTC(SIZE(063))
1919	OCTETS: MULTI-ROW: The coded character set file name or URI[URI-spec] of the
1920	document.
1921	document.
1922	There is no restriction on the same file name occurring in multiple rows.
1923	There is no restriction on the same the name occurring in multiple rows.
1924	documentName(35), JmJobStringTC(SIZE(063))
1924	OCTETS: MULTI-ROW: The coded character set name of the document.
1925	OCTATO. INTOLATINOW. THE COURT CHARACTER SET HATHE OF the document.
1920	There is no restriction on the same document name occurring in multiple rows.
1927	There is no resurction on the same document name occurring in multiple rows.
1/40	

1020	110	T T I C(! T C(CTZT (0 (2))
1929	jobComment(36),	JmJobStringTC(SIZE(063))
1930		-readable coded character text string supplied by the
1931	submitting user or the job subm	nitting application program for any purpose. For example
1932		he is going to do with the printed output or the job
1933	submitting application program	might indicate how the document was produced.
1934		
1935	The iobComment attribute is r	ot intended to be a name; see the jobName attribute.
1936	1110 J 0	37 min 10 37 to 10 10 min 10 m
1937	documentFormatIndex(37),	Integer32(02147483647)
1938		ne index in the prtInterpreterTable in the Printer
1939	MIR[print_mih] of the page des	cription language (PDL) or control language interpreter
1940		ocument or a job MAY use more than one PDL or control
1940	· •	ocument of a job MAT use more than one TDL of control
	language.	
1942	NOTE A!4111 !	(4
1943		ttributes where multiple rows are allowed, there SHALL
1944	be only one distinct row for each	th distinct interpreter; there SHALL be no duplicates.
1945		
1946		ntended to be used with an agent that implements the
1947		e used if the agent does not implement the Printer MIB.
1948	Such an agent SHALL use the	documentFormat attribute instead.
1949		
1950	documentFormat(38),	PrtInterpreterLangFamilyTC
1951	` '/	AND/OR
1952		OCTET STRING(SIZE(063))
1953	INTEGER: MULTI-ROW: TI	ne interpreter language family corresponding to the Printe
1954		LangFamily object, that this job requires/uses. A
1955		ore than one PDL or control language.
1956	document of a job Will a use in	The than one I DL of control language.
1957	AND/OR	
1958	AND/OK	
	OCTETS, MILITIDOW, The	do assessed formered as a sistemad as a media term of ion a media
1959		e document format registered as a media type[iana-media-
1960		ME content-type/subtype. Examples:
1961		ation/vnd.hp-PCL' and 'application/pdf', 'text/plain' (US-
1962	ASCII SHALL be assumed), 't	ext/plain; charset=iso-8859-1, and 'application/octet-
1963	stream'. See the IPP ipp-mode	l] 'mimeMediaType' attribute syntaxand the "document-
1964	format" attribute for further exa	mples and explanation.
1965		
1966		
1967	+++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++
1968	+ Job Parameter attributes	
1969	+	
1970	+ The following attributes represer	t innut parameters
1971	+ supplied by the submitting client	
1972	+ protocol.	in the job submission
1973	-	+++++++++++++++++++++++++++++++++++++++
1973		
1974	iobPriority(50)	Integer 32(1 100)
	jobPriority(50),	Integer 32(1100)
1976		neduling the job. It is used by servers and devices that
1977	employ a priority-based schedu	ing aigorithm.

17/0	
1979	
17/7	
1980	
1000	
1981	
1000	
1982	
1983	
1983	
1004	
1984	
1985	
1903	
1986	
1700	
1987	
1007	
1988	
1000	
1989	
1990	
1990	
1991	
1991	
1992	
1//4	
1993	
1001	
1994	
1005	
1995	
1996	
1990	
1997	
1997	
1998	
1770	
1999	
1///	
2000	
2000	
2001	
2002	
2002	
2002	
2003	
2003	
2003 2004	
2003 2004	
2003 2004 2005	
2003 2004 2005	
2003 2004 2005 2006	
2003 2004 2005 2006	
2003 2004 2005 2006 2007	
2003 2004 2005 2006 2007 2008	
2003 2004 2005 2006 2007 2008	
2003 2004 2005 2006 2007 2008 2009	
2003 2004 2005 2006 2007 2008 2009	
2003 2004 2005 2006 2007 2008 2009 2010	
2003 2004 2005 2006 2007 2008 2009 2010	
2003 2004 2005 2006 2007 2008 2009 2010 2011	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024	
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023	

A higher value specifies a higher priority. The value 1 is defined to indicate the lowest possible priority (a job which a priority-based scheduling algorithm SHALL pass over in favor of higher priority jobs). The value 100 is defined to indicate the highest possible priority. Priority is expected to be evenly or 'normally' distributed across this range. The mapping of vendor-defined priority over this range is implementation-specific.

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.

jobHold(52), JmBooleanTC

INTEGER: If the value is **true(4)**, a client has explicitly specified that the job is to be held until explicitly released. Until the job is explicitly released by a client, the job SHALL be in the **pendingHeld** state with the **jobHoldSpecified** value in the **jmJobStateReasons1** attribute.

jobHoldUntil(53), JmJobStringTC(SIZE(0..63))

OCTETS: The named time period during which the job SHALL become a candidate for processing, such as 'evening', 'night', 'weekend', 'second-shift', 'third-shift', etc., as defined by the system administrator. See IPP [ipp-model] for the standard keyword values. Until that time period arrives, the job SHALL be in the pendingHeld state with the jobHoldUntilSpecified value in the jmJobStateReasons1 object. The value 'no-hold' SHALL indicate explicitly that no time period has been specified; the absence of this attribute SHALL indicate implicitly that no time period has been specified.

outputBin(54), Integer32(0..2147483647)

AND/OR JmJobStringTC(SIZE(0..63))

INTEGER: MULTI-ROW: The output subunit index in the Printer MIB[print-mib]

AND/OR

OCTETS: <u>MULTI-ROW</u>: the name or number (represented as ASCII digits) of the output bin to which all or part of the job is placed in.

sides(55), Integer 32(-2..2)

INTEGER: MULTI-ROW: The number of sides, 1' or 2', that any document in this job requires/used.

finishing(56), JmFinishingTC

INTEGER: MULTI-ROW: Type of finishing that any document in this job requires/used.

2027		
2028	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++
2029	+ Image Quality attributes (requested a	and consumed)
2030	+	,
2031	+ For devices that can vary the image q	uality.
2032		++++++++++++++++++++++++++++++++++++++
2033		
2034	printQualityRequested(70),	JmPrintQualityTC
2035		int quality selection requested for a document in the
2036	job for printers that allow quality di	
2037	job for printers that allow quality di	nerendation.
2038	printQualityUsed(71),	JmPrintQualityTC
2038		int quality selection actually used by a document in
2040	the job for printers that allow qualit	y differentiation.
2041		I. D. C. A. D. a. L. A. a. M.C.
2042	printerResolutionRequested(72),	JmPrinterResolutionTC
2043		nter resolution requested for a document in the job for
2044	printers that support resolution selec	ction.
2045		
2046	printerResolutionUsed(73),	JmPrinterResolutionTC
2047		nter resolution actually used by a document in the job
2048	for printers that support resolution s	election.
2049		
2050	tonerEcomonyRequested(74),	JmTonerEconomyTC
2051	INTEGER: MULTI-ROW: The to	ner economy selection requested for documents in the
2052	job for printers that allow toner eco	
2053	J I	,
2054	tonerEcomonyUsed(75),	JmTonerEconomyTC
2055	INTEGER: MULTI-ROW: The to	ner economy selection actually used by documents in
2056	the job for printers that allow toner	
2057	the job for printers that allow toner	economy differentiation.
2058	tonerDensityRequested(76),	Integer32(-2100)
2059		ner density requested for a document in this job for
2060	davious that can years to nor density	avals. I eval 1 is the legisted density and level 100 is
	the highest density level. Devices y	evels. Level 1 is the lowest density and level 100 is
2061		vith a smaller range, SHALL map the 1-100 range
2062	evenly onto the implemented range.	
2063		7 (70 (70)
2064	tonerDensityUsed(77),	Integer32(-2100)
2065		ner density used by documents in this job for devices
2066		Level 1 is the lowest density and level 100 is the
2067		a smaller range, SHALL map the 1-100 range evenly
2068	onto the implemented range.	
2069		
2070		
2071	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++
2072	+ Job Progress attributes (requested an	nd consumed)
2073	+	,
2074	+ Pairs of these attributes can be used l	ov monitoring
2075	+ applications to show an indication of	

2076	+ to users.
2077	+++++++++++++++++++++++++++++++++++++++
2078	
2079	jobCopiesRequested(90), Integer32(-22147483647)
2080	INTEGER: The number of copies of the entire job that are to be produced.
2081	
2082	jobCopiesCompleted(91), Integer32(-22147483647)
2083	INTEGER: The number of copies of the entire job that have been completed so far.
2084	
2085	documentCopiesRequested(92), Integer32(-22147483647)
2086	INTEGER: The total count of the number of document copies requested for the job as a
2087	whole. If there are documents A, B, and C, and document B is specified to produce 4
2088	copies, the number of document copies requested is 6 for the job.
2089	
2090	This attribute SHALL be used only when a job has multiple documents. The
2091	jobCopiesRequested attribute SHALL be used when the job has only one document.
2092	
2093	ISSUE: Would it be better/simpler to understand for documentCopiesRequested to be
2094	MULTI-VALUED, where each value is for a separate document in the multi-document
2095	<u>job?</u>
2096	
2097	documentCopiesCompleted(93), Integer32(-22147483647)
2098	INTEGER: The total count of the number of document copies completed so far for the
2099	job as a whole. If there are documents A, B, and C, and document B is specified to
2100	produce 4 copies, the number of document copies starts a 0 and runs up to 6 for the job as
2101	the job processes.
2102	
2103	This attribute SHALL be used only when a job has multiple documents. The
2104	jobCopiesCompleted attribute SHALL be used when the job has only one document.
2105	
2106	ISSUE: Would it be better for documentCopiesCompleted to be MULTI-VALUED, where
2107	each value is for a separate document in the multi-document job?
2108	
2109	jobKOctetsTransferred(94), Integer32(-22147483647)
2110	INTEGER: The number of K (1024) octets transferred to the server or device to which
2111	the agent is providing access. This count is independent of the number of copies of the
2112	job or documents that will be produced, but it is only a measure of the number of bytes
2113	transferred to the server or device.
2114	
2115	The agent SHALL round the actual number of octets transferred up to the next higher K.
2116	Thus 0 octets SHALL be represented as 0 ′, 1-1024 octets SHALL BE represented as 1 ′,
2117	1025-2048 SHALL be 2', etc. When the job completes, the values of the
2118	jmJobKOctetsPerCopyRequested object and the jobKOctetsTransferred attribute
2119	SHALL be equal.
2120	NOTE THE LITTO A A TO A STATE OF THE STATE O
2121	NOTE - The jobKOctetsTransferred can be used with the
2122	jmJobKOctetsPerCopyRequested object in order to produce a relative indication of the
2123	progress of the job for agents that do not implement the jmJobKOctetsProcessed object
2124	

2125	currentCopyNumber(95), Integer32(-22147483647)
2126	INTEGER: The number of the copy being stacked for the current document. This
2127	number starts at 0, is set to 1 when the first sheet of the first copy for each document is
2128	being stacked and increases to the value of jobCopiesRequested.
2129	
2130	For External Sheet Collation, this increments as each sheet is stacked, and is reset to 1
2131	when the printer moves to stacking the next sheet number in a document. For internally
2132	collated copies, this number increments when all sheets of the current document have
2133	been stacked. See the collationType (97) attribute.
2134	been stacked. See the conation 1 ypc (77) attribute.
2135	currentDocumentNumber(96), Integer32(-22147483647)
	INTEGER. The ardinal number of the decument in the ich that is approachly being
2136	INTEGER: The ordinal number of the document in the job that is currently being
2137	stacked. This number starts at 0, increments to 1 when the first sheet of the first
2138	document in the job is being stacked, and increases to the value of numberOfDocuments
2139	by the end of the job.
2140	
2141	For uncollated or externally collated copies, this increments as each document is stacked
2142	and wraps back to 1 when the printer moves to printing the next document number in a
2143	copy. For internally collated copies, this number increments when all copies of the
2144	current document have been stacked. See the collationType(97) attribute.
2145	
2146	Implementations that only support one document jobs SHOULD NOT implement this
2147	attribute.
2148	
2149	ISSUE: Instead of having the currentDocumentNumber attribute for the multi-
2150	document job implementation, how about making the jmJobImpressionsCompleted and
2151	the currentCopyNumber attributes multi-valued, one value for each document in the
2152	(multi-document) job? This makes it simpler to understand. The down side is that a
2153	monitoring program would have to get all the values for a multi-document job.
2154	Accounting programs would have to get all the values of the multi-valued attribute and
2155	add them up.
2156	<u>aud them up.</u>
2157	collationType(97), JmCollationTypeTC
2157	INTEGER: The type of sheet and document collation.
2159	INTEGER. The type of sheet and document conation.
2160	
2161	++++++++++++++++++++++++++++++++++++++
2162	+ Impression attributes
2163	+
2164	+ For a print job, an impression is the marking of the
2165	+ entire side of a sheet. Two-sided processing involves two
2166	+ impressions per sheet. Two-up is the placement of two
2167	+ logical pages on one side of a sheet and so is still a
2168	+ single impression.
2169	<u> </u>
2170	+ See also jmJobImpressionsPerCopyRequested and
2171	+ jmJobImpressionsCompleted objects in the jmJobTable.
2172	+++++++++++++++++++++++++++++++++++++++
2173	

2174	impressionsSpooled(110), Integer32(-22147483647)
2175	INTEGER: The number of impressions spooled to the server or device for the job so far.
2176	
2177	impressionsSentToDevice(111), Integer32(-22147483647)
2178	INTEGER: The number of impressions sent to the device for the job so far.
2179	THE number of impressions sent to the device for the job so far.
	:
2180	impressionsInterpreted(112), Integer32(-22147483647)
2181	INTEGER: The number of impressions interpreted for the job so far.
2182	
2183	impressionsCompletedCurrentCopy(113), Integer32(-22147483647)
2184	INTEGER: The number of impressions completed by the device for the current copy of
2185	the current document so far. For printing, the impressions completed includes
2186	interpreting, marking, and stacking the output. For other types of job services, the
2187	number of impressions completed includes the number of impressions processed.
2188	number of impressions completed metades the number of impressions processed.
	This value CIIAII he reset to 0 for each decument in the ich and for each decument
2189	This value SHALL be reset to 0 for each document in the job and for each document
2190	copy.
2191	
2192	fullColorImpressionsCompleted(114), Integer32(-22147483647)
2193	INTEGER: The number of full color impressions completed by the device for this job so
2194	far. For printing, the impressions completed includes interpreting, marking, and stacking
2195	the output. For other types of job services, the number of impressions completed includes
2196	the number of impressions processed. Full color impressions are typically defined as
2197	those requiring 3 or more colorants, but this MAY vary by implementation.
2198	those requiring 5 of more colorants, but this WAT vary by implementation.
	h:-h1:-h4C-1I
2199	highlightColorImpressionsCompleted(115), Integer32(-22147483647)
2200	INTEGER: The number of highlight color impressions completed by the device for this
2201	job so far. For printing, the impressions completed includes interpreting, marking, and
2202	stacking the output. For other types of job services, the number of impressions completed
2203	includes the number of impressions processed. Highlight color impressions are typically
2204	defined as those requiring black plus one other colorant, but this MAY vary by
2205	implementation.
2206	T
2207	
2208	+++++++++++++++++++++++++++++++++++++++
2209	
	+ Page attributes
2210	+
2211	+ A page is a logical page. Number up can impose more than
2212	+ one page on a single side of a sheet. Two-up is the
2213	+ placement of two logical pages on one side of a sheet so
2214	+ that each side counts as two pages.
2215	+++++++++++++++++++++++++++++++++++++++
2216	
2217	pagesRequested(130), Integer32(-22147483647)
2218	INTEGER: The number of logical pages requested by the job to be processed.
2219	11.12.5214. The humber of togical pages requested by the job to be processed.
2220	pagesCompleted(131), Integer32(-22147483647)
2221	INTEGER: The number of logical pages completed for this job so far.
2222	

2223	For implementations where multiple copies are produced by the interpreter with only a
2224	single pass over the data, the final value SHALL be equal to the value of the
2225	pagesRequested object. For implementations where multiple copies are produced by the
2226	interpreter by processing the data for each copy, the final value SHALL be a multiple of
2227	the value of the pagesRequested object.
2228	The state of the s
2229	NOTE - See the impressionsCompletedCurrentCopy and
2230	pagesCompletedCurrentCopy attributes for attributes that are reset on each document
2231	copy.
2232	•• _F ,
2233	NOTE - The pagesCompleted object can be used with the pagesRequested object to
2234	provide an indication of the relative progress of the job, provided that the multiplicative
2235	factor is taken into account for some implementations of multiple copies.
2236	ractor is taken into account for some implementations of material copies.
2237	pagesCompletedCurrentCopy(132), Integer32(-22147483647)
2238	INTEGER: The number of logical pages completed for the current copy of the document
2239	so far. This value SHALL be reset to 0 for each document in the job and for each
2240	document copy.
2241	document copy.
2242	
2243	+++++++++++++++++++++++++++++++++++++++
2244 2244	+ Sheet attributes
22 44 2245	+ Succe attributes +
2245 2246	+ The sheet is a single piece of a medium, whether printing
2240 2247	+ on one or both sides.
2247 2248	
2248 2249	+++++++++++++++++++++++++++++++++++++++
2249 2250	shootsDoguested(150) Integer 22(2.2147492647)
2251	sheetsRequested(150), Integer 32(-22147483647)
2252	INTEGER: The total number of medium sheets requested to be processed for this job.
2252 2253	Unlike the im IchV Octote Dev Conv. Decreated and
2233 2254	Unlike the jmJobKOctetsPerCopyRequested and im JobImpressionsPorCopyPopuested attributes the shootsPopuested (150) attributes
2254	<u>imJobImpressionsPerCopyRequested</u> attributes, the sheetsRequested(150) attribute SHALL include the multiplicative factor contributed by the number of copies.
2255	SHALL include the multiplicative factor contributed by the number of copies.
2256	ghootsCompleted(151) Integer 22(2.2147492447)
2257	sheetsCompleted(151), Integer32(-22147483647) INTEGER: The number of medium sheets that have completed marking and stacking for
2258 2259	
	the entire job so far whether those sheets have been processed on one side or on both.
2260	aboutaCompletedComp(152) Integra 22(2, 2147492447)
2261	sheetsCompletedCurrentCopy(152), Integer32(-22147483647)
2262	INTEGER: The number of medium sheets that have completed marking and stacking for
2263	the current copy of a document in the job so far whether those sheets have been processed
2264	on one side or on both.
2265	TTI 1 CALLACTIANT 1 A O 11 ALLACTIC
2266	The value of this attribute SHALL be reset to 0 as each document in the job starts being
2267	processed and for each document copy as it starts being processed.
2268	
2269	
2270	+++++++++++++++++++++++++++++++++++++++
2271	+ Resources attributes (requested and consumed)

2272	+	
2273	+ Pairs of these attributes can be used by monitoring	
2274	+ applications to show an indication of relative	
2275	+ users.	g-
2276	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++
2277		
2278	mediumRequested(170),	JmMediumTypeTC
2279	medium requested (170),	AND/OR
2280		JmJobStringTC(SIZE(063))
2281	INTECED, MILITIDOW, The type	
2282	INTEGER: MULTI-ROW: The type	
	AND/OR	1' 4 4' ' 11 4 ' 1
2283	OCTETS: MULTI-ROW: the name of the	medium that is required by the job.
2284	W G 1/454)	T
2285	mediumConsumed(171),	Integer32(-22147483647)
2286		AND
2287		JmJobStringTC(SIZE(063))
2288	INTEGER: The number of sheets	
2289	AND	
2290	OCTETS: MULTI-ROW: MULTI-ROW:	the name of the medium that has been
2291	consumed so far whether those sheets have	been processed on one side or on both.
2292		•
2293	This attribute SHALL have both Integer32	and OCTET STRING (represented as
2294	JmJobStringTC) values.	\ 1
2295	55 0% 24 g 1 0) + uo 5	
2296	colorantRequested(172),	Integer32(-22147483647)
2297	colorum tequesteu(1/2),	AND/OR
2298		JmJobStringTC(SIZE(063))
2299	INTEGER: MULTI-ROW: The index (prt	
2300	MIB[print-mib]	warker color antinuex) in the 1 inter
2300	AND/OR	
2302	OCTETS: MULTI-ROW: the name of the	colorant requested
2303	OCTETS. MOETI-ROW. the maine of the	colorant requested.
2304	colorantConsumed(173),	Integer 22 (2. 2147492647)
2305	colorant consumed (173),	Integer32(-22147483647)
		AND/OR
2306	INTEGED. MILITIDOW. The State of the state o	JmJobStringTC(SIZE(063))
2307	INTEGER: MULTI-ROW: The index (prt	Marker Colorant Index) in the Printer
2308	MIB[print-mib]	
2309	AND/OR	
2310	OCTETS: MULTI-ROW: the name of the	colorant consumed.
2311		
2312		
2313	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++
2314	+ Time attributes (set by server or device)	
2315	+	
2316	+ This section of attributes are ones that are se	
2317	+ server or device that accepts jobs. Two forms of time are	
2318	+ provided. Each form is represented in a sepa	arate attribute.
2319	+ See section 3.1.2 and section 3.1.3 for the	
2320	+ conformance requirements for time attribute	e for agents and

Job Monitoring MIB, V0.86

Sep 19, 1997

2321 2322	+ monitoring applications, respectively. The	two forms are:	
	+ 2Data And Time? is an P on 11 actat himoury of		
2323	+ 'DateAndTime' is an 8 or 11 octet binary encoded year,		
2324	+ month, day, hour, minute, second, deci-seco		
2325	+ optional offset from UTC. See SNMPv2-To	C [SMIv2-TC].	
2326	+		
2327	+ NOTE: 'DateAndTime' is not printable cha	aracters; it is	
2328	+ binary.		
2329	+		
2330	+ 'JmTimeStampTC' is the time of day meas	ured in the number of	
2331	+ seconds since the system was booted.		
2332	+++++++++++++++++++++++++++++++++++++++	-++++++++++++++++++++++	
2333			
2334	jobSubmissionToServerTime(190),	JmTimeStampTC	
2335	Jobb domission 1 ober ver 1 mic(170),	AND/OR	
2336		DateAndTime	
	INTECED: Configuration 2 and The ti-		
2337	INTEGER: Configuration 3 only: The time	me	
2338	AND/OR	1 10 10 1	
2339	OCTETS: the date and time that the job v	was submitted to the server (as distinguished	
2340	from the device which uses jobSubmission	nTime).	
2341			
2342	jobSubmissionTime(191),	JmTimeStampTC	
2343		AND/OR	
2344		DateAndTime	
2345	INTEGER: Configurations 1, 2, and 3: T	The time	
2346	AND/OR		
2347		was submitted to the server or device to which	
2348	the agent is providing access.	was satisfactor to the server of device to which	
2349	the agent is providing access.		
2350			
2351			
	inhCtantadDainaHaldTima(102)	ITioCtoTC	
2352	jobStartedBeingHeldTime(192),	JmTimeStampTC	
2353		AND/OR	
2354	73 mm	DateAndTime	
2355	INTEGER: The time		
2356	AND/OR		
2357		ast entered the pendingHeld state. If the job	
2358	has never entered the pendingHeld state,	then the value SHALL be '0' or the attribute	
2359	SHALL not be present in the table.		
2360	•		
2361	jobStartedProcessingTime(193),	JmTimeStampTC	
2362	9 (/)	AND/OR	
2363		DateAndTime	
2364	INTEGER: The time	~ WVVI AIIW A IIIIV	
2365	AND/OR		
2366		started processing	
	OCTETS: the date and time that the job s	marteu processing.	
2367	ich Commission Time (104)	ImTimeStermTC	
2368	jobCompletionTime(194),	JmTimeStampTC	
2369		AND/OR	

```
2370
                                                                 DateAndTime
2371
                        INTEGER: The time
2372
                        AND/OR
2373
                        OCTETS: the date and time that the job entered the completed, canceled, or aborted
2374
                        state.
2375
2376
                  jobProcessingCPUTime(195)
                                                                 Integer32(-2..2147483647)
2377
                        UNITS
                                  'seconds'
2378
                        INTEGER: The amount of CPU time in seconds that the job has been in the processing
2379
                        state. If the job enters the processingStopped state, that elapsed time SHALL not be
2380
                        included. In other words, the jobProcessingCPUTime value SHOULD be relatively
2381
                        repeatable when the same job is processed again on the same device."
2382
2383
             REFERENCE
2384
                   "See Section 3.2 entitled 'The Attribute Mechanism' for a description of this textual-convention
2385
                   and its use in the jmAttributeTable.
2386
2387
                   This is a type 2 enumeration. See Section 3.7.1.2."
2388
             SYNTAX
                          INTEGER {
2389
                  other(1).
2390
                   unknown(2),
2391
                   jobStateReasons2(3),
2392
                  jobStateReasons3(4),
2393
                  jobStateReasons4(5),
2394
                   processingMessage(6),
                  processingMessageNaturalLanguageTag(7),
2395
2396
                  jobCodedCharSet(87),
2397
                  jobNaturalLanguageTag(9),
2398
2399
                  jobURI(20),
2400
                  jobAccountName(21),
2401
                  serverAssignedJobName(22),
2402
                  jobName(23),
2403
                  jobServiceTypes(24),
                  jobSourceChannelIndex(25).
2404
2405
                  jobSourcePlatformType(26),
2406
                   submittingServerName(27),
                   submittingApplicationName(28),
2407
2408
                   jobOriginatingHost(29),
2409
                   deviceNameRequested(30),
2410
                   queueNameRequested(31),
2411
                   physicalDevice(32).
2412
                   numberOfDocuments(33),
2413
                   fileName(34).
2414
                   documentName(35),
2415
                   jobComment(36),
2416
                   documentFormatIndex(37),
2417
                   documentFormat(38),
```

```
2419
                   jobPriority(50),
                   jobProcessAfterDateAndTime(51),
2420
2421
                   jobHold(52),
2422
                   jobHoldUntil(53),
2423
                   outputBin(54),
2424
                   sides(55),
2425
                   finishing(56),
2426
2427
                   printQualityRequested(70),
2428
                   printQualityUsed(71),
2429
                   printerResolutionRequested(72),
2430
                   printerResolutionUsed(73),
2431
                   tonerEcomonyRequested(74),
2432
                   tonerEcomonyUsed(75),
2433
                   tonerDensityRequested(76),
2434
                   tonerDensityUsed(77),
2435
2436
                   jobCopiesRequested(90),
2437
                   jobCopiesCompleted(91),
2438
                   documentCopiesRequested(92),
2439
                   documentCopiesCompleted(93),
2440
                   jobKOctetsTransferred(94),
2441
                   currentCopyNumber(95),
                   currentDocumentNumber(96),
2442
2443
                   collationType(97),
2444
2445
                   impressionsSpooled(110),
2446
                   impressionsSentToDevice(111),
2447
                   impressionsInterpreted(112),
2448
                   impressionsCompletedCurrentCopy(113),
2449
                   fullColorImpressionsCompleted(114),
                   highlightColorImpressionsCompleted(115),
2450
2451
2452
                   pagesRequested(130),
2453
                   pagesCompleted(131),
2454
                   pagesCompletedCurrentCopy(132),
2455
2456
                   sheetsRequested(150),
2457
                   sheetsCompleted(151),
2458
                   sheetsCompletedCurrentCopy(152),
2459
2460
                   mediumRequested(170),
                   mediumConsumed(171),
2461
2462
                   colorantRequested(172),
2463
                   colorantConsumed(173),
2464
2465
                   jobSubmissionToServerTime(190),
2466
                   jobSubmissionTime(191),
2467
                   jobStartedBeingHeldTime(192),
```

```
2468
                    jobStartedProcessingTime(193),
2469
                    jobCompletionTime(194),
2470
                    jobProcessingCPUTime(195)
2471
              }
2472
2473
2474
2475
2476
        JmJobServiceTypesTC ::= TEXTUAL-CONVENTION
2477
              STATUS
                           current
              DESCRIPTION
2478
2479
                    "Specifies the type(s) of service to which the job has been submitted (print, fax, scan, etc.). The
2480
                    service type is represented as an enum that is bit encoded with each job service type so that
2481
                    more general and arbitrary services can be created, such as services with more than one
2482
                    destination type, or ones with only a source or only a destination. For example, a job service
2483
                    might scan, faxOut, and print a single job. In this case, three bits would be set in the
2484
                    jobServiceTypes attribute, corresponding to the hexadecimal values: 0x8 + 0x20 + 0x4,
2485
                    respectively, yielding: 0x2C.
2486
2487
                    Whether this attribute is set from a job attribute supplied by the job submission client or is set
2488
                    by the recipient job submission server or device depends on the job submission protocol. With
2489
                    either implementation, the agent SHALL return a non-zero value for this attribute indicating the
2490
                    type of the job.
2491
2492
                    One of the purposes of this attribute is to permit a requester to filter out jobs that are not of
2493
                    interest. For example, a printer operator MAY only be interested in jobs that include printing.
2494
                    That is why the attribute is in the job identification category.
2495
2496
                    The following service component types are defined (in hexadecimal) and are assigned a
2497
                    separate bit value for use with the jobServiceTypes attribute:
2498
2499
                    other 0x1
2500
                          The job contains some instructions that are not one of the identified types.
2501
2502
2503
                          The job contains some instructions whose type is unknown to the agent.
2504
2505
                    print 0x4
2506
                          The job contains some instructions that specify printing
2507
2508
                    scan 0x8
2509
                          The job contains some instructions that specify scanning
2510
2511
                    faxIn 0x10
2512
                          The job contains some instructions that specify receive fax
2513
2514
                    faxOut
```

The job contains some instructions that specify sending fax

2516	
2517	getFile 0x40
2518	The job contains some instructions that specify accessing files or documents
2519	
2520	putFile 0x80
2521	The job contains some instructions that specify storing files or documents
2522	The job contains some moractions that speerly storing mes of documents
2523	mailList 0x100
2524	The job contains some instructions that specify distribution of documents using an
2525	electronic mail system."
2525 2526	REFERENCE
2527	"These bit definitions are the equivalent of a type 2 enum except that combinations of them
2527 2528	
2528 2529	MAY be used together. See section 3.7.1.2."
2329	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2530	
0501	
2531	
2532	
2332	
2533	
2534	JmJobStateReasons1TC ::= TEXTUAL-CONVENTION
2535	STATUS current
2536	DESCRIPTION
2537	"The JmJobStateReasonsNTC ($N=14$) textual-conventions are used with the
2537 2538	jmJobStateReasons1 object and jobStateReasonsN ($N=24$), respectively, to provide
2538 2539	
2339 2540	additional information regarding the current jmJobState object value. These values MAY be
2540	used with any job state or states for which the reason makes sense.
2541	NOTE. While values council be added to the im Tab State abject without immediate dealessed
2542	NOTE - While values cannot be added to the jmJobState object without impacting deployed
2543	clients that take actions upon receiving jmJobState values, it is the intent that additional
2544	JmJobStateReasonsNTC enums can be defined and registered without impacting such
2545	deployed clients. In other words, the jmJobStateReasons1 object and jobStateReasonsN
2546	attributes are intended to be extensible.
2547	
2548	NOTE - The Job Monitoring MIB contains a superset of the IPP values[ipp-model] for the IPP
2549	'job-state-reasons' attribute, since the Job Monitoring MIB is intended to cover other job
2550	submission protocols as well. Also some of the names of the reasons have been changed from
2551	'printer' to 'device', since the Job Monitoring MIB is intended to cover additional types of
2552	devices, including input devices, such as scanners.
2553	
2554	The following standard values are defined (in hexadecimal) as <i>powers of two</i> , since multiple
2555	values MAY be used at the same time. For ease of understanding, the
2556	JmJobStateReasons1TC reasons are presented in the order in which the reasons are likely to
2557	occur (if implemented), starting with the 'jobIncoming' value and ending with the
2558	'jobCompletedWithErrors' value.
2559	•
2560	other 0x1
2561	The job state reason is not one of the standardized or registered reasons.
	J

2562	
2563	unknown 0x2
2564	The job state reason is not known to the agent or is indeterminent.
2565	
2566	jobIncoming 0x4
2567	The job has been accepted by the server or device, but the server or device is expecting
2568	(1) additional operations from the client to finish creating the job and/or (2) is
2569	accessing/accepting document data.
2570	
2571	submissionInterrupted 0x8
2572	The job was not completely submitted for some unforeseen reason, such as: (1) the server
2573	has crashed before the job was closed by the client, (2) the server or the document
2574	transfer method has crashed in some non-recoverable way before the document data was
2575	entirely transferred to the server, (3) the client crashed or failed to close the job before the
2576	time-out period.
2577	
2578	jobOutgoing 0x10
2579	Configuration 2 only: The server is transmitting the job to the device.
2580	g
2581	jobHoldSpecified 0x20
2582	The value of the job's jobHold(52) attribute is TRUE. The job SHALL NOT be a
2583	candidate for processing until this reason is removed and there are no other reasons to
2584	hold the job.
2585	note the job.
2586	jobHoldUntilSpecified 0x40
2587	The value of the job's jobHoldUntil(53) attribute specifies a time period that is still in the
2588	future. The job SHALL NOT be a candidate for processing until this reason is removed
2589	and there are no other reasons to hold the job.
2590	and there are no other reasons to note the job.
2591	jobProcessAfterSpecified 0x80
2592	The value of the job's jobProcessAfterDateAndTime(51) attribute specifies a time that is
2593	still in the future. The job SHALL NOT be a candidate for processing until this reason is
2594	removed and there are no other reasons to hold the job.
2595	removed and there are no other reasons to note the job.
2596	resourcesAreNotReady 0x100
2597	At least one of the resources needed by the job, such as media, fonts, resource objects,
2598	etc., is not ready on any of the physical devices for which the job is a candidate. This
2599	condition MAY be detected when the job is accepted, or subsequently while the job is
2600	pending or processing, depending on implementation.
2601	pending of processing, depending on implementation.
2602	deviceStoppedPartly 0x200
2603	One or more, but not all, of the devices to which the job is assigned are stopped. If all of
2604	the devices are stopped (or the only device is stopped), the deviceStopped reason
2605	SHALL be used.
2606	SITALL DE USEU.
2607	deviceStopped 0x400
2608	The device(s) to which the job is assigned is (are all) stopped.
2609	The device(s) to which the job is assigned is (are an) stopped.
400 <i>7</i>	

2610	jobInterpreting 0x800
2611	The device to which the job is assigned is interpreting the document data.
2612	
2613	jobPrinting 0x1000
2614	The output device to which the job is assigned is marking media. This attribute is useful
2615	for servers and output devices which spend a great deal of time processing (1) when no
2616	marking is happening and then want to show that marking is now happening or (2) when
2617	the job is in the process of being canceled or aborted while the job remains in the
2618	processing state, but the marking has not yet stopped so that impression or sheet counts
2619	are still increasing for the job.
2620	with some more was job!
2621	jobCanceledByUser 0x2000
2622	The job was canceled by the owner of the job, i.e., by a user whose name is the same as
2623	the value of the job's jmJobOwner object, or by some other authorized end-user, such as
2624	a member of the job owner's security group.
2625	a member of the job owner's security group.
2626	jobCanceledByOperator 0x4000
2627	The job was canceled by the operator, i.e., by a user who has been authenticated as having
2628	
	operator privileges (whether local or remote).
2629	inh Canadad At Daving
2630	jobCanceledAtDevice 0x8000
2631	The job was canceled by an unidentified local user, i.e., a user at a console at the device.
2632	
2633	abortedBySystem 0x10000
2634	The job (1) is in the process of being aborted, (2) has been aborted by the system and
2635	placed in the 'aborted' state, or (3) has been aborted by the system and placed in the
2636	'pendingHeld' state, so that a user or operator can manually try the job again.
2637	penangitera state, so that a user of operator can manually by the job again.
2638	processingToStopPoint 0x20000
2639	The requester has issued an operation to cancel or interrupt the job or the server/device
2640	has aborted the job, but the server/device is still performing some actions on the job until
2641	a specified stop point occurs or job termination/cleanup is completed.
2642	a specified stop point occurs of job termination/eleanup is completed.
2643	This reason is recommended to be used in conjunction with the processing job state to
2644	indicate that the server/device is still performing some actions on the job while the job
2645	remains in the processing state. After all the job's resources consumed counters have
2646	stopped incrementing, the server/device moves the job from the processing state to the
2647	canceled or aborted job states.
2648	Canceled of abouted job states.
2649	serviceOffLine 0x40000
2650	
	The service or document transform is off-line and accepting no jobs. All pending jobs
2651	are put into the pendingHeld state. This situation could be true if the service's or
2652	document transform's input is impaired or broken.
2653	inhCompletedCycoccafully 000000
2654	jobCompletedSuccessfully 0x80000
2655	The job completed successfully.
2656	

2657	jobCo	mpletedWithWarnings	0x100000
2658	,	The job completed with warnings.	
2659			
2660	iobCo	ompletedWithErrors	0x200000
2661		The job completed with errors (and	
2662		The job completes with errors (whe	possion warmings to o).
2663			
2664	The fo	llowing additional job state reason	s have been added to represent job states that are in
2665		PA[iso-dpa] and other job submiss	
	130 D	r A[180-upa] and other job sublinss	ion protocois.
2666	!.LD.	3	0400000
2667	jobPa		0x400000
2668			nded by a client issuing an operation to suspend the
2669			using the same devices. The client MAY issue an
2670			at any time, in which case the agent SHALL remove
2671			s jmJobStateReasons1 object and the job is
2672		eventually resumed at or near the p	oint where the job was paused.
2673		•	•
2674	jobInt	terrupted	0x800000
2675			processing by a client issuing an operation that
2676		specifies another job to be run inste	ad of the current job. The server or device will
2677			d job when the interrupting job completes.
2678	,	automatically regame the interrupte	a job when the interrupting job completes.
2679	iohRe	tained	0x1000000
2680			ver or device with all of the job's document data (and
2681			logos, and forms, if any). Thus a client could issue an
2682			either (1) re-do the job (or a copy of the job) on the
2683			nit the job to another server or device. When a client
2684			job, such as after the document data has been
2685		discarded, the agent SHALL remov	e the jobRetained value from the
2686		jmJobStateReasons1 object."	
2687	REFERENC		
2688			of a type 2 enum except that combinations of bits may
2689	be use	d together. See section 3.7.1.2. The	e remaining bits are reserved for future
2690	standa	rdization and/or registration."	
2691		<u> </u>	
2692	SYNTAX	INTEGER(02147483647) 3	1 bits, all but sign bit
2693		,	,
2694			
2695			
2696			
2697			
2698	Im InhState Deag	ons2TC ::= TEXTUAL-CONVEN	TION .
2699	STATUS	current	HON
2700	DESCRIPT]		ichCtotaDoogons2 atteilante te meet dee eddit!
2701			jobStateReasons2 attribute to provides additional
2702	ıntorn	nation regarding the jmJobState ob	ject. See the description under

JmJobStateReasons1TC for additional information that applies to all reasons.

2703

2705 The following standard values are defined (in hexadecimal) as powers of two, since multiple 2706 values may be used at the same time: 2707 2708 2709 An outbound gateway has transmitted all of the job's job and document attributes and data 2710 to another spooling system. 2711 2712 deletedByAdministrator 0x22713 The administrator has deleted the job. 2714 2715 discardTimeArrived 0x42716 The job has been deleted due to the fact that the time specified by the job's job-discard-2717 time attribute has arrived. 2718 2719 postProcessingFailed 0x82720 The post-processing agent failed while trying to log accounting attributes for the job; therefore the job has been placed into the completed state with the **jobRetained** 2721 2722 jmJobStateReasons1 object value for a system-defined period of time, so the 2723 administrator can examine it, resubmit it, etc. 2724 2725 0x10jobTransforming The server/device is interpreting document data and producing another electronic 2726 2727 representation. 2728 2729 maxJobFaultCountExceeded 2730 The job has faulted several times and has exceeded the administratively defined fault 2731 count limit. 2732 2733 0x40devicesNeedAttentionTimeOut 2734 One or more document transforms that the job is using needs human intervention in order 2735 for the job to make progress, but the human intervention did not occur within the site-2736 settable time-out value. 2737 2738 needsKeyOperatorTimeOut 0x80One or more devices or document transforms that the job is using need a specially trained 2739 2740 operator (who may need a key to unlock the device and gain access) in order for the job to make progress, but the key operator intervention did not occur within the site-settable 2741 2742 time-out value. 2743 2744 **jobStartWaitTimeOut** 0x1002745 The server/device has stopped the job at the beginning of processing to await human action, such as installing a special cartridge or special non-standard media, but the job 2746 2747 was not resumed within the site-settable time-out value and the server/device has transitioned the job to the **pendingHeld** state. 2748 2749 2750 **jobEndWaitTimeOut** The server/device has stopped the job at the end of processing to await human action, 2751

such as removing a special cartridge or restoring standard media, but the job was not

2753	resumed within the site-settable time-out	value and the server/device has transitioned the
2754	job to the completed state.	
2755	J	
2756	jobPasswordWaitTimeOut	0x400
2757		the beginning of processing to await input of the
2758		t received within the site-settable time-out value.
2759	J 1 / 1	
2760	deviceTimedOut	0x800
2761	A device that the job was using has not re	esponded in a period specified by the device's
2762	site-settable attribute.	
2763		
2764	connectingToDeviceTimeOut	0x1000
2765		e or more devices which may be dial-up, polled,
2766		ic from other systems, but server was unable to
2767	connect to the device within the site-setta	
2768		
2769	transferring	0x2000
2770	The job is being transferred to a down str	ream server or downstream device.
2771	3	
2772	queuedInDevice	0x4000
2773	The server/device has queued the job in a	a down stream server or downstream device.
2774		
2775	jobQueued	0x8000
2776	The server/device has queued the document	ent data.
2777	-	
2778	jobCleanup	0x10000
2779	The server/device is performing cleanup	activity as part of ending normal processing.
2780		
2781	jobPasswordWait	0x20000
2782	The server/device has selected the job to	be next to process, but instead of assigning
2783	resources and starting the job processing,	, the server/device has transitioned the job to the
2784	pendingHeld state to await entry of a pa	ssword (and dispatched another job, if there is
2785	one).	
2786		
2787	validating	0x40000
2788	The server/device is validating the job <i>after</i>	ter accepting the job.
2789		
2790	queueHeld	0x80000
2791	The operator has held the entire job set or	r queue.
2792		
2793	jobProofWait	0x100000
2794		and is in the pendingHeld state waiting for the
2795		the job to print normally, obeying any job and
2796	document copy attributes that were origin	nally submitted.
2797		
2798	heldForDiagnostics	0x200000
2799	The system is running intrusive diagnosti	ics, so that all jobs are being held.

2801	noSpaceOnServer 0x800000
2802	There is no room on the server to store all of the job.
2803	J
2804	pinRequired 0x1000000
2805	The System Administrator settable device policy is (1) to require PINs, and (2) to hold
2806	jobs that do not have a pin supplied as an input parameter when the job was created.
2807	jobs that do not have a pin supplied as an input parameter when the job was created.
2807 2808	exceededAccountLimit 0x2000000
2809	The account for which this job is drawn has exceeded its limit. This condition SHOULD
2810	be detected before the job is scheduled so that the user does not wait until his/her job is
2811	scheduled only to find that the account is overdrawn. This condition MAY also occur
2812	while the job is processing either as processing begins or part way through processing.
2813	
2814	heldForRetry 0x4000000
2815	The job encountered some errors that the server/device could not recover from with its
2816	normal retry procedures, but the error might not be encountered if the job is processed
2817	again in the future. Example cases are phone number busy or remote file system in-
2818	accessible. For such a situation, the server/device SHALL transition the job from the
2819	processing to the pendingHeld, rather than to the aborted state.
2820	P-000332 g to the Political graves, runner than to the wool to a state.
2821	The following values are from the X/Open PSIS draft standard:
2822	The following values are from the 14 open 1 sits draft standard.
2823	canceledByShutdown 0x8000000
2823 2824	The job was canceled because the server or device was shutdown before completing the
2825	job.
2826	J::
2827	deviceUnavailable 0x10000000
2828	This job was aborted by the system because the device is currently unable to accept jobs.
2829	
2830	wrongDevice 0x20000000
2831	This job was aborted by the system because the device is unable to handle this particular
2832	job; the spooler SHOULD try another device or the user should submit the job to another
2833	device.
2834	
2835	badJob 0x40000000
2836	This job was aborted by the system because this job has a major problem, such as an ill-
2837	formed PDL; the spooler SHOULD not even try another device. "
2838	REFERENCE
2839	"These bit definitions are the equivalent of a type 2 enum except that combinations of them
2840	may be used together. See section 3.7.1.2. See the description under JmJobStateReasons1TC
2841	and the jobStateReasons2 attribute."
2842	and the jobstate reasons 2 attribute.
2843	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2843 2844	STITIME INTEGER(V ZIT/TOJUT/) STUIIS, all out sign of
2845	
2846	
2847	
2848	
2849	

2850	JmJobStateReasons3TC ::= TEXTUAL-CONVENTION
2851	STATUS current
2852	DESCRIPTION
2853	"This textual-convention is used with the jobStateReasons3 attribute to provides additional
2854	information regarding the jmJobState object. See the description under
2855	JmJobStateReasons1TC for additional information that applies to all reasons.
2856	
2857	The following standard values are defined (in hexadecimal) as powers of two, since multiple
2858	values may be used at the same time:
2859	, and the second se
2860	jobInterruptedByDeviceFailure 0x1
2861	A device or the print system software that the job was using has failed while the job was
2862	processing. The server or device is keeping the job in the pendingHeld state until an
2863	operator can determine what to do with the job."
2864	REFERENCE
2865	"These bit definitions are the equivalent of a type 2 enum except that combinations of them
2866	may be used together. See section 3.7.1.2. The remaining bits are reserved for future
2867	standardization and/or registration. See the description under JmJobStateReasons1TC and
2868	the jobStateReasons3 attribute."
2869	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2870	~
2871	
2872	
2873	
2874	
2875	JmJobStateReasons4TC ::= TEXTUAL-CONVENTION
2876	STATUS current
2877	DESCRIPTION
2878	"This textual-convention is used in the jobStateReasons4 attribute to provides additional
2879	information regarding the jmJobState object. See the description under
2880	JmJobStateReasons1TC for additional information that applies to all reasons.
2881	
2882	The following standard values are defined (in hexadecimal) as powers of two, since multiple
2883	values may be used at the same time:
2884	varies may be used at the same time.
2885	none yet defined. These bits are reserved for future standardization and/or registration."
2886	REFERENCE
2887	"These bit definitions are the equivalent of a type 2 enum except that combinations of them
2888	may be used together. See section 3.7.1.2. See the description under JmJobStateReasons1T 0
2889	and the jobStateReasons4 attribute."
2890	and the John meet componer attribute.
2891	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
<u>_</u> U/1	SIIIIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIII

```
2892
2893
       jobmonMIBObjects OBJECT IDENTIFIER ::= { jobmonMIB 1 }
2894
2895
       -- The General Group (MANDATORY)
2896
2897
       -- The jmGeneralGroup consists entirely of the jmGeneralTable.
2898
2899
       jmGeneral OBJECT IDENTIFIER ::= { jobmonMIBObjects 1 }
2900
2901
       imGeneralTable OBJECT-TYPE
2902
            SYNTAX
                         SEQUENCE OF JmGeneralEntry
2903
            MAX-ACCESS not-accessible
2904
             STATUS
                        current
2905
            DESCRIPTION
2906
                  "The imGeneralTable consists of information of a general nature that are per-job-set, but are
2907
                  not per-job. See Section 2 entitled 'Terminology and Job Model' for the definition of a job set."
2908
            REFERENCE
2909
                  "The MANDATORY-GROUP macro specifies that this group is MANDATORY."
2910
            ::= { jmGeneral 1 }
2911
2912
       imGeneralEntry OBJECT-TYPE
2913
             SYNTAX
                         JmGeneralEntry
2914
            MAX-ACCESS not-accessible
2915
            STATUS
                        current
2916
            DESCRIPTION
2917
                  "Information about a job set (queue).
2918
2919
                  An entry SHALL exist in this table for each job set."
2920
            INDEX { imGeneralJobSetIndex }
2921
            ::= { jmGeneralTable 1 }
2922
2923
       JmGeneralEntry ::= SEQUENCE {
2924
            imGeneralJobSetIndex
                                                              Integer32(1...32767),
2925
            jmGeneralNumberOfActiveJobs
                                                              Integer32(0..2147483647),
2926
            jmGeneralOldestActiveJobIndex
                                                              Integer32(0..2147483647),
2927
            imGeneralNewestActiveJobIndex
                                                              Integer32(0..2147483647),
2928
            imGeneralJobPersistence
                                                              Integer32(15..2147483647),
2929
            imGeneralAttributePersistence
                                                              Integer32(15..2147483647),
2930
            jmGeneralJobSetName
                                                              JmUTF8StringTC(SIZE(0..63))
2931
       }
2932
2933
       imGeneralJobSetIndex OBJECT-TYPE
2934
                        Integer32(1..32767)
            SYNTAX
2935
            MAX-ACCESS not-accessible
2936
            STATUS
                        current
2937
            DESCRIPTION
2938
                  "A unique value for each job set in this MIB. The jmJobTable and jmAttributeTable tables
2939
                  have this same index as their primary index.
2940
```

2941 The value(s) of the **jmGeneralJobSetIndex** SHALL be persistent across power cycles, so that 2942 clients that have retained **imGeneralJobSetIndex** values will access the same job sets upon 2943 subsequent power-up. 2944 2945 An implementation that has only one job set, such as a printer with a single queue, SHALL hard 2946 code this object with the value 1." 2947 **REFERENCE** 2948 "See Section 2 entitled Terminology and Job Model' for the definition of a job set. Corresponds to the first index in **jmJobTable** and **jmAttributeTable**." 2949 2950 ::= { jmGeneralEntry 1 } 2951 2952 jmGeneralNumberOfActiveJobs OBJECT-TYPE 2953 Integer32(0..2147483647) SYNTAX 2954 MAX-ACCESS read-only 2955 STATUS current 2956 DESCRIPTION 2957 "The current number of 'active' jobs in the jmJobIDTable, jmJobTable, and 2958 jmAttributeTable, i.e., the total number of jobs that are in the pending, processing, or 2959 processingStopped states. See the JmJobStateTC textual-convention for the exact 2960 specification of the semantics of the job states." 2961 DEFVAL { 0 } -- no jobs 2962 ::= { jmGeneralEntry 2 } 2963 2964 jmGeneralOldestActiveJobIndex OBJECT-TYPE 2965 SYNTAX Integer32 (0..2147483647) 2966 MAX-ACCESS read-only 2967 STATUS current 2968 DESCRIPTION 2969 "The **imJobIndex** of the oldest job that is still in one of the 'active' states **pending**, 2970 **processing**, or **processingStopped**). In other words, the index of the 'active' job that has been 2971 in the job tables the longest. 2972 2973 If there are no active jobs, the agent SHALL set the value of this object to 0." 2974 REFERENCE 2975 "See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes' for 2976 a description of the usage of this object." 2977 DEFVAL { 0 } -- no active jobs 2978 ::= { jmGeneralEntry 3 } 2979 2980 jmGeneralNewestActiveJobIndex OBJECT-TYPE 2981 SYNTAX Integer32 (0..2147483647) 2982 MAX-ACCESS read-only 2983 STATUS current 2984 DESCRIPTION 2985 "The **jmJobIndex** of the newest job that is in one of the 'active' states **pending**, **processing**, or 2986 **processingStopped**). In other words, the index of the 'active' job that has been most recently 2987 added to the job tables.

```
2989
                   When all jobs become 'inactive', i.e., enter the pending Held, completed, canceled, or aborted
2990
                   states, the agent SHALL set the value of this object to 0."
2991
             REFERENCE
2992
                   "See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes' for
2993
                   a description of the usage of this object."
2994
                        { 0 } -- no active jobs
2995
             ::= { jmGeneralEntry 4 }
2996
2997
        imGeneralJobPersistence OBJECT-TYPE
2998
             SYNTAX
                          Integer32(15..2147483647)
2999
                        "seconds"
             UNITS
3000
             MAX-ACCESS read-only
3001
             STATUS
                         current
3002
             DESCRIPTION
3003
                   "The minimum time in seconds for this instance of the Job Set that an entry SHALL remain in
3004
                   the imJobIDTable and imJobTable after processing has completed, i.e., the minimum time in
                   seconds starting when the job enters the completed, canceled, or aborted state.
3005
3006
3007
                   Configuring this object is implementation-dependent.
3008
3009
                   This value SHALL be equal to or greater than the value of jmGeneralAttributePersistence.
                   This value SHOULD be at least 60 which gives a monitoring application one minute in which
3010
3011
                   to poll for job data."
3012
             DEFVAL
                          { 60 }
                                    -- one minute
3013
             ::= { jmGeneralEntry 5 }
3014
3015
        jmGeneralAttributePersistence OBJECT-TYPE
                          Integer32(15..2147483647)
3016
             SYNTAX
3017
             UNITS
                        "seconds"
3018
             MAX-ACCESS read-only
3019
             STATUS
                         current
3020
             DESCRIPTION
3021
                   "The minimum time in seconds for this instance of the Job Set that an entry SHALL remain in
                   the jmAttributeTable after processing has completed, i.e., the time in seconds starting when
3022
3023
                   the job enters the completed, canceled, or aborted state.
3024
3025
                   Configuring this object is implementation-dependent.
3026
3027
                   This value SHOULD be at least 60 which gives a monitoring application one minute in which
                   to poll for job data."
3028
3029
             DEFVAL
                          { 60 }
                                     -- one minute
3030
             ::= { jmGeneralEntry 6 }
3031
3032
        jmGeneralJobSetName OBJECT-TYPE
3033
             SYNTAX
                          JmUTF8StringTC(SIZE(0..63))
3034
             MAX-ACCESS read-only
3035
             STATUS
                         current
3036
             DESCRIPTION
```

```
3037
                   "The human readable name of this job set assigned by the system administrator (by means
                   outside of this MIB). Typically, this name SHOULD be the name of the job queue. If a server
3038
3039
                   or device has only a single job set, this object can be the administratively assigned name of the
3040
                   server or device itself. This name does not need to be unique, though each job set in a single
3041
                   Job Monitoring MIB SHOULD have distinct names.
3042
3043
                   NOTE - The purpose of this object is to help the user of the job monitoring application
3044
                   distinguish between several job sets in implementations that support more than one job set."
3045
             REFERENCE
3046
                   "See the OBJECT compliance macro for the minimum maximum length required for
                  conformance."
3047
             DEFVAL { "'H }
3048
                                   -- empty string
             ::= { jmGeneralEntry 7 }
3049
3050
3051
3052
3053
3054
3055
        -- The Job ID Group (MANDATORY)
3056
3057
        -- The jmJobIDGroup consists entirely of the jmJobIDTable.
3058
3059
       jmJobID OBJECT IDENTIFIER ::= { jobmonMIBObjects 2 }
3060
3061
       imJobIDTable OBJECT-TYPE
3062
                          SEQUENCE OF JmJobIDEntry
             SYNTAX
3063
             MAX-ACCESS not-accessible
             STATUS
3064
                         current
3065
             DESCRIPTION
3066
                   "The imJobIDTable provides a correspondence map (1) between the job submission ID that a
3067
                   client uses to refer to a job and (2) the jmGeneralJobSetIndex and jmJobIndex that the Job
                   Monitoring MIB agent assigned to the job and that are used to access the job in all of the other
3068
3069
                   tables in the MIB. If a monitoring application already knows the jmGeneralJobSetIndex and
3070
                   the jmJobIndex of the job it is querying, that application NEED NOT use the jmJobIDTable."
3071
             REFERENCE
3072
                   "The MANDATORY-GROUP macro specifies that this group is MANDATORY."
3073
             ::= { jmJobID 1 }
3074
3075
       jmJobIDEntry OBJECT-TYPE
3076
             SYNTAX
                          JmJobIDEntry
3077
             MAX-ACCESS not-accessible
             STATUS
3078
                         current
3079
             DESCRIPTION
3080
                   "The map from (1) the jmJobSubmissionID to (2) the jmGeneralJobSetIndex and
3081
                  jmJobIndex.
3082
3083
                   An entry SHALL exist in this table for each job currently known to the agent for all job sets and
3084
                  job states. There MAY be more than one jmJobIDEntry that maps to a single job. This many
```

to one mapping can occur when more than one application program along the job submission

```
3086
                   path wishes to monitor a job. See Section 3.5. However, Eeach job SHALL appear once and in
3087
                   one and only one job set."
3088
             INDEX { jmJobSubmissionID }
3089
             ::= { jmJobIDTable 1 }
3090
3091
        JmJobIDEntry ::= SEQUENCE {
3092
             jmJobSubmissionID
                                                                  OCTET STRING(SIZE(48)),
3093
             jmJobIDJobSetIndex
                                                                  Integer32(1...32767),
3094
                                                                  Integer32(1..2147483647)
             jmJobIDJobIndex
3095
        }
3096
3097
        jmJobSubmissionID OBJECT-TYPE
3098
                          OCTET STRING(SIZE(48))
             SYNTAX
3099
             MAX-ACCESS not-accessible
3100
             STATUS
                         current
             DESCRIPTION
3101
3102
                   "A quasi-unique 48-octet fixed-length string ID which identifies the job within a particular
3103
                   client-server environment. There are multiple formats for the jmJobSubmissionID. Each
3104
                   format SHALL be uniquely identified. See the JmJobSubmissionIDTypeTC textual
3105
                   convention. Each format SHALL be registered using the procedures of a type 2 enum. See
                   section 3.7.3 entitled: 'IANA Registration of Job Submission Id Formats'.
3106
3107
3108
                   If the requester (client or server) does not supply a job submission ID in the job submission
                   protocol, then the recipient (server or device) SHALL assign a job submission ID using any of
3109
3110
                   the standard formats that have been reserved for agents and adding the final 8 octets to
3111
                   distinguish the ID from others submitted from the same requester.
3112
                   The monitoring application, whether in the client or running separately, MAY use the job
3113
3114
                   submission ID to help identify which jmJobIndex was assigned by the agent, i.e., in which row
3115
                   the job information is in the other tables.
3116
3117
                   NOTE - fixed-length is used so that a management application can use a shortened GetNext
                   varbind (in SNMPv1 and SNMPv2) in order to get the next submission ID, disregarding the
3118
                   remainder of the ID in order to access jobs independent of the trailing identifier part, e.g., to get
3119
3120
                   all jobs submitted by a particular imJobOwner or submitted from a particular MAC address."
3121
             REFERENCE
3122
                   "See the JmJobSubmissionIDTypeTC textual convention.
                   See APPENDIX B - Support of the Job Submission ID in Job Submission Protocols."
3123
3124
             DEFVAL
                        { "H } -- empty string
3125
             ::= { jmJobIDEntry 1 }
3126
3127
        imJobIDJobSetIndex OBJECT-TYPE
3128
             SYNTAX Integer32(1..32767)
3129
             MAX-ACCESS read-only
3130
             STATUS
                          current
3131
             DESCRIPTION
3132
                   "This object contains the value of the jmGeneralJobSetIndex for the job with the
3133
                   jmJobSubmissionID value, i.e., the job set index of the job set in which the job was placed
3134
                   when that server or device accepted the job. This 16-bit value in combination with the
```

```
3135
                  jmJobIDJobIndex value permits the management application to access the other tables to
3136
                  obtain the job-specific objects for this job."
3137
             REFERENCE
3138
                   "See jmGeneralJobSetIndex in the jmGeneralTable."
3139
             DEFVAL { 1 } -- default job set index
3140
             ::= { jmJobIDEntry 2 }
3141
       jmJobIDJobIndex OBJECT-TYPE
3142
3143
             SYNTAX
                         Integer32(1..2147483647)
3144
             MAX-ACCESS read-only
3145
             STATUS
                         current
3146
             DESCRIPTION
                  "This object contains the value of the jmJobIndex for the job with the jmJobSubmissionID
3147
3148
                  value, i.e., the job index for the job when the server or device accepted the job. This value, in
3149
                  combination with the jmJobIDJobSetIndex value, permits the management application to
3150
                  access the other tables to obtain the job-specific objects for this job."
3151
             REFERENCE
3152
                   "See jmJobIndex in the jmJobTable."
3153
                        { 1 } -- default jmJobIndex value.
3154
             ::= { imJobIDEntry 3 }
3155
3156
3157
3158
3159
       -- The Job Group (MANDATORY)
3160
3161
       -- The jmJobGroup consists entirely of the jmJobTable.
3162
3163
       jmJob OBJECT IDENTIFIER ::= { jobmonMIBObjects 3 }
3164
3165
       imJobTable OBJECT-TYPE
3166
                         SEQUENCE OF JmJobEntry
             SYNTAX
             MAX-ACCESS not-accessible
3167
3168
             STATUS
                         current
3169
             DESCRIPTION
3170
                   "The imJobTable consists of basic job state and status information for each job in a job set that
3171
                  (1) monitoring applications need to be able to access in a single SNMP Get operation, (2) that
3172
                  have a single value per job, and (3) that SHALL always be implemented."
3173
             REFERENCE
3174
                   "The MANDATORY-GROUP macro specifies that this group is MANDATORY."
3175
             ::= \{ \text{ imJob } 1 \}
3176
3177
       jmJobEntry OBJECT-TYPE
3178
             SYNTAX
                         JmJobEntry
3179
             MAX-ACCESS not-accessible
3180
             STATUS
                         current
3181
             DESCRIPTION
3182
                   "Basic per-job state and status information.
```

```
3184
                   An entry SHALL exist in this table for each job, no matter what the state of the job is. Each job
                   SHALL appear in one and only one job set."
3185
3186
             REFERENCE
3187
                   "See Section 3.2 entitled 'The Job Tables'."
3188
             INDEX { jmGeneralJobSetIndex, jmJobIndex }
3189
             ::= { jmJobTable 1 }
3190
3191
        JmJobEntry ::= SEQUENCE {
3192
             imJobIndex
                                                                   Integer32(1...2147483647).
3193
             imJobState
                                                                   JmJobStateTC,
3194
             imJobStateReasons1
                                                                   JmJobStateReasons1TC.
3195
             jmNumberOfInterveningJobs
                                                                   Integer32(-2..2147483647),
             jmJobKOctetsPerCopyRequested
                                                                   Integer32(-2..2147483647),
3196
             jmJobKOctetsProcessed
                                                                   Integer32(-2..2147483647),
3197
3198
             jmJobImpressionsPerCopyRequested
                                                                   Integer32(-2..2147483647),
3199
             jmJobImpressionsCompleted
                                                                   Integer32(-2..2147483647),
3200
                                                                   JmJobStringTC(SIZE(0..63))
             jmJobOwner
3201
        }
3202
3203
        imJobIndex OBJECT-TYPE
3204
             SYNTAX
                          Integer32(1..2147483647)
3205
             MAX-ACCESS not-accessible
3206
             STATUS
                          current
             DESCRIPTION
3207
3208
                   "The sequential, monatonically increasing identifier index for the job generated by the server or
                   device when that server or device accepted the job. This index value permits the management
3209
3210
                   application to access the other tables to obtain the job-specific row entries."
             REFERENCE
3211
                   "See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes'.
3212
3213
                   See Section 3.4 entitled 'Monitoring Job Progress
3214
        There are a number of objects and attributes for monitoring the progress of a job. These objects and
3215
        attributes count the number of K octets, impressions, sheets, and pages requested or completed, i.e.,
        processed or stacked, depending on implementation. There are objects and attributes for the overall job and
3216
3217
        for the current copy of the document currently being processed or stacked. For the latter, the rate at which
        the various objects and attributes count depends on the sheet and document collation of the job.
3218
3219
        Sheet collation is defined to be the collations of sheets within a document copy. Document collation is
3220
        defined to be collation of document copies within a multi-document job. There are three combinations of
        these two types of collation:
3221
3222
               1. External Sheet Collation
3223
               2. Internal Sheet Collation with Collated Documents
3224
               3. Internal Sheet Collation with Uncollated Documents
3225
        Consider the following four variables that are used to monitor the progress of a job's impressions:
3226
               1. jmJobImpressionsCompleted - counts the total number of impressions stacked for the job
```

- impressions Completed Current Copy counts the number of impressions stacked for the current document copy
 current Copy Number identifies the number of the copy for the current document being
 - stacked
 - 4. **currentDocumentNumber** identifies the current document within the job that is being stacked.
- For each of the three types of collation, a job with two documents (1, 2), where each document consists of 3 impressions, the four variables would have the following values:
- 3235 <u>collation type = External Sheet Collation</u>

3236

3230

3231

3232

<u>imJobImpressionsCo</u> <u>mpleted</u>	impressionsComplete dCurrentCopy	<u>currentCopyNumber</u>	currentDocumentNu mber
$\frac{1}{2}$	<u>1</u>	$\frac{1}{2}$	<u>1</u>
$\frac{3}{4}$	$\frac{\overline{1}}{2}$	$\frac{3}{1}$	$\frac{\overline{1}}{\underline{1}}$
<u>5</u> <u>6</u>	$\frac{2}{2}$ $\frac{3}{3}$	<u>2</u> <u>3</u> .	$\frac{1}{1}$
$\frac{7}{8}$	<u>3</u> 3	$\frac{1}{2}$	<u>1</u> <u>1</u> 1
10 11	<u>1</u> 1	$\frac{3}{1}$	$\frac{1}{2}$
$\frac{\overline{12}}{\underline{13}}$	$\frac{\overline{1}}{2}$	$\frac{3}{1}$	$\frac{\overline{2}}{2}$
14 15	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{2}$
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	2 2 2 3 3 3 3	$\frac{1}{2}$	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
	<u>-</u>	-	-

3237

Collation Type = Internal Collation with document collated within each job copy

3239

<u>imJobImpressionsCo</u> <u>mpleted</u>	impressionsComplete dCurrentCopy	<u>currentCopyNumber</u>	<u>currentDocumentNu</u> <u>mber</u>
$\frac{1}{2}$ $\frac{3}{2}$	$\frac{1}{2}$ $\frac{3}{2}$	$\frac{1}{1}$	1 1 1
<u>4</u> <u>5</u>	$\frac{1}{2}$	<u>1</u> <u>1</u>	<u>2</u> <u>2</u>

		Job Monitoring M	11D , 10.00	Sep 15, 1557
	$ \begin{array}{c} \frac{6}{7} \\ \frac{8}{8} \\ \underline{9} \\ \underline{10} \\ \underline{11} \\ \underline{12} \\ \underline{13} \\ \underline{14} \\ \underline{15} \\ \underline{16} \\ \underline{17} \\ \underline{18} \end{array} $	3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3	1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3	$ \frac{2}{1} $ $ \frac{1}{1} $ $ \frac{2}{2} $ $ \frac{2}{2} $ $ \frac{1}{1} $ $ \frac{1}{2} $ $ \frac{2}{2} $
3240				
3241	<u>Collation Type = Interna</u>	al Collation with uncollate	d document copies	
3242	jmJobImpressionsCo mpleted	impressionsComplete dCurrentCopy	<u>currentCopyNumber</u>	currentDocumentNu mber
	$ \frac{\frac{1}{2}}{\frac{3}{4}} $ $ \frac{\frac{1}{5}}{\frac{6}{7}} $ $ \frac{\frac{1}{8}}{\frac{9}{2}} $ $ \frac{10}{\frac{11}{12}} $ $ \frac{13}{\frac{14}{15}} $ $ \frac{16}{\frac{17}{18}} $	$\frac{1}{2}$ $\frac{1}$	1 1 1 2 2 2 3 3 1 1 1 1 2 2 2 2 3 3 3 3	$ \frac{\frac{1}{1}}{\frac{1}{1}} $ $ \frac{\frac{1}{1}}{\frac{1}{1}} $ $ \frac{\frac{1}{2}}{\frac{1}{2}} $ $ \frac{\frac{2}{2}}{\frac{2}{2}} $ $ \frac{2}{2} $ $ \frac{2} $ $ \frac{2}{2} $ $ \frac{2}{2$
3243 3244 3245 3246 3247 3248	See also jm (See JmJobS as an 8-digit	Job Identification'. See also jmGeneralNewestActiveJobIndex for the largest value of jmJobIndex . See JmJobSubmission_DTypeTC for a limit on the size of this index if the agent represent as an 8-digit decimal number." ::= { jmJobEntry 1 }		lue of jmJobIndex . is index if the agent represents it
3249 3250 3251	jmJobState OBJECT-T SYNTAX JmJ	YPE IobStateTC		

Job Monitoring MIB, V0.86

Sep 19, 1997

```
3252
             MAX-ACCESS read-only
3253
             STATUS
                         current
3254
             DESCRIPTION
3255
                   "The current state of the job (pending, processing, completed, etc.). Agents SHALL
3256
                   implement only those states which are appropriate for the particular implementation. However,
3257
                   management applications SHALL be prepared to receive all the standard job states.
3258
3259
                   The final value for this object SHALL be one of: completed, canceled, or aborted. The
                   minimum length of time that the agent SHALL maintain MIB data for a job in the completed,
3260
3261
                   canceled, or aborted state before removing the job data from the jmJobIDTable and
                   jmJobTable is specified by the value of the jmGeneralJobPersistence object."
3262
3263
                        { unknown } -- default is unknown
3264
             ::= { jmJobEntry 2 }
3265
3266
       jmJobStateReasons1 OBJECT-TYPE
3267
             SYNTAX
                          JmJobStateReasons1TC
             MAX-ACCESS read-only
3268
3269
             STATUS
                         current
3270
             DESCRIPTION
                   "Additional information about the job's current state, i.e., information that augments the value
3271
3272
                   of the job's jmJobState object.
3273
3274
                   Implementation of any reason values is OPTIONAL, but an agent SHOULD return any reason
                   information available. These values MAY be used with any job state or states for which the
3275
3276
                   reason makes sense. Since the Job State Reasons will be more dynamic than the Job State, it is
3277
                   recommended that a job monitoring application read this object every time jmJobState is read.
3278
                   When the agent cannot provide a reason for the current state of the job, the the value of the
3279
                   jmJobStateReasons1 object and jobStateReasonsN attributes SHALL be 0."
3280
             REFERENCE
3281
                   "The jobStateReasonsN (N=2..4) attributes provide further additional information about the
3282
                   job's current state."
             DEFVAL { 0 } -- no reasons
3283
3284
             ::= { jmJobEntry 3 }
3285
3286
       jmNumberOfInterveningJobs OBJECT-TYPE
3287
                          Integer32(-2..2147483647)
             SYNTAX
3288
             MAX-ACCESS read-only
3289
             STATUS
                         current
3290
             DESCRIPTION
3291
                   "The number of jobs that are expected to complete processing before this job has completed
3292
                   processing according to the implementation's queuing algorithm, if no other jobs were to be
                   submitted. In other words, this value is the job's queue position. The agent SHALL return a
3293
                   value of 0 for this attribute when the job is the next job to complete processing (or has
3294
3295
                   completed processing)."
3296
             DEFVAL { 0 } -- default is no intervening jobs.
3297
             ::= { imJobEntry 4 }
3298
3299
       jmJobKOctetsPerCopyRequested OBJECT-TYPE
3300
             SYNTAX
                          Integer32(-2..2147483647)
```

```
3301
             MAX-ACCESS read-only
3302
              STATUS
                          current
3303
             DESCRIPTION
3304
                   "The total size in K (1024) octets of the document(s) being requested to be processed in the job.
3305
                   The agent SHALL round the actual number of octets up to the next highest K. Thus 0 octets
3306
                   SHALL be represented as 0', 1-1024 octets SHALL be represented as 1', 1025-2048 SHALL
3307
                   be represented as 2', etc.
3308
3309
                   In computing this value, the server/device SHALL not include the multiplicative factors
3310
                   contributed by (1) the number of document copies, and (2) the number of job copies,
3311
                   independent of whether the device can process multiple copies of the job or document without
3312
                   making multiple passes over the job or document data and independent of whether the output is
3313
                   collated or not. Thus the server/device computation is independent of the implementation and
3314
                   reflects the size of the document(s) independent of the number of copies."
3315
             DEFVAL { -2 }
                                   -- the default is unknown(-2)
3316
             ::= { jmJobEntry 5 }
3317
3318
        jmJobKOctetsProcessed OBJECT-TYPE
3319
             SYNTAX
                          Integer32(-2..2147483647)
3320
              MAX-ACCESS read-only
3321
             STATUS
                          current
3322
             DESCRIPTION
3323
                   "The totaleurrent number of octets processed by the server or device measured in units of K
                   (1024) octets so far. The agent SHALL round the actual number of octets processed up to the
3324
3325
                   next higher K. Thus 0 octets SHALL be represented as 0', 1-1024 octets SHALL be
                   represented as 1', 1025-2048 octets SHALL be 2', etc. For printing devices, this value is the
3326
3327
                   number interpreted by the page description language interpreter rather than what has been
3328
                   marked on media.
3329
3330
                   For implementations where multiple copies are produced by the interpreter with only a single
3331
                   pass over the data, the final value SHALL be equal to the value of the
3332
                   jmJobKOctetsPerCopyRequested object. For implementations where multiple copies are
3333
                   produced by the interpreter by processing the data for each copy, the final value SHALL be a
3334
                   multiple of the value of the jmJobKOctetsPerCopyRequested object.
3335
3336
                   NOTE - See the impressionsCompletedCurrentCopy and pagesCompletedCurrentCopy
3337
                   attributes for attributes that are reset on each document copy.
3338
3339
                   NOTE - The jmJobKOctetsProcessed object can be used with the
3340
                   imJobKOctetsPerCopyRequested object to provide an indication of the relative progress of
3341
                   the job, provided that the multiplicative factor is taken into account for some implementations
3342
                   of multiple copies."
3343
             DEFVAL
                           { 0 }
                                  -- default is no octets processed.
3344
             ::= { jmJobEntry 6 }
3345
3346 | jmJobImpressionsPerCopyRequested OBJECT-TYPE
3347
                          Integer32(-2..2147483647)
             SYNTAX
3348
             MAX-ACCESS read-only
3349
             STATUS
                          current
```

```
3350
             DESCRIPTION
3351
                   "The total size in number of impressions of the document(s) being requested by this job to
3352
                   produce.
3353
3354
                   In computing this value, the server/device SHALL not include the multiplicative factors
3355
                   contributed by (1) the number of document copies, and (2) the number of job copies,
3356
                   independent of whether the device can process multiple copies of the job or document without
3357
                   making multiple passes over the job or document data and independent of whether the output is
3358
                   collated or not. Thus the server/device computation is independent of the implementation and
3359
                   reflects the size of the document(s) independent of the number of copies."
3360
                         { -2 } -- default is unknown(-2)
3361
             ::= { jmJobEntry 7 }
3362
3363
        jmJobImpressionsCompleted OBJECT-TYPE
3364
              SYNTAX
                          Integer32(-2..2147483647)
3365
             MAX-ACCESS read-only
3366
             STATUS
                          current
3367
             DESCRIPTION
3368
                   "The totaleurrent number of impressions completed for this job so far. For printing devices, the
3369
                   impressions completed includes interpreting, marking, and stacking the output. For other types
                   of job services, the number of impressions completed includes the number of impressions
3370
3371
                   processed.
3372
3373
                   For implementations where multiple copies are produced by the interpreter with only a single
3374
                   pass over the data, the final value SHALL be equal to the value of the
3375
                   jmJobImpressionsRequested object. For implementations where multiple copies are
3376
                   produced by the interpreter by processing the data for each copy, the final value SHALL be a
3377
                   multiple of the value of the jmJobImpressionsRequested object.
3378
3379
                   NOTE - See the impressionsCompletedCurrentCopy and pagesCompletedCurrentCopy
3380
                   attributes for attributes that are reset on each document copy.
3381
3382
                   NOTE - The imJobImpressionsCompleted object can be used with the
3383
                   jmJobImpressionsPerCopyRequested object to provide an indication of the relative progress
3384
                   of the job, provided that the multiplicative factor is taken into account for some
3385
                   implementations of multiple copies."
3386
              DEFVAL { 0 }
                                  -- default is no octets
3387
             ::= { jmJobEntry 8 }
3388
3389
        jmJobOwner OBJECT-TYPE
3390
                          JmJobStringTC(SIZE(0..63))
             SYNTAX
3391
              MAX-ACCESS read-only
3392
             STATUS
                          current
3393
             DESCRIPTION
3394
                   "The coded character set name of the user that submitted the job. The method of assigning this
3395
                   user name will be system and/or site specific but the method MUST insure that the name is
3396
                   unique to the network that is visible to the client and target device.
3397
```

This value SHOULD be the *authenticated* name of the user submitting the job."

```
3399
             REFERENCE
                   "See the OBJECT compliance macro for the minimum maximum length required for
3400
3401
                   conformance."
3402
             DEFVAL { "H } -- empty string
3403
             ::= { jmJobEntry 9 }
3404
3405
3406
3407
3408
        -- The Attribute Group (MANDATORY)
3409
3410
        -- The jmAttributeGroup consists entirely of the jmAttributeTable.
3411
3412
        -- Implementation of the two objects in this group is MANDATORY.
3413
        -- See Section 3.1 entitled 'Conformance Considerations'.
3414
        -- An agent SHALL implement any attribute if (1) the server or device
3415
        -- supports the functionality represented by the attribute and (2) the
3416
        -- information is available to the agent.
3417
3418
       imAttribute OBJECT IDENTIFIER ::= { jobmonMIBObjects 4 }
3419
3420
       imAttributeTable OBJECT-TYPE
3421
             SYNTAX
                          SEQUENCE OF JmAttributeEntry
3422
             MAX-ACCESS not-accessible
3423
             STATUS
                         current
3424
             DESCRIPTION
3425
                   "The imAttributeTable SHALL contain attributes of the job and document(s) for each job in a
                   job set. Instead of allocating distinct objects for each attribute, each attribute is represented as a
3426
3427
                   separate row in the jmAttributeTable."
3428
             REFERENCE
3429
                   "The MANDATORY-GROUP macro specifies that this group is MANDATORY. An agent
3430
                   SHALL implement any attribute if (1) the server or device supports the functionality
3431
                   represented by the attribute and (2) the information is available to the agent. "
3432
             ::= { jmAttribute 1 }
3433
3434
       jmAttributeEntry OBJECT-TYPE
3435
             SYNTAX
                          JmAttributeEntry
3436
             MAX-ACCESS not-accessible
3437
             STATUS
                         current
3438
             DESCRIPTION
3439
                   "Attributes representing information about the job and document(s) or resources required and/or
3440
                   consumed.
3441
3442
                   Each entry in the jmAttributeTable is a per-job entry with an extra index for each type of
3443
                   attribute (jmAttributeTypeIndex) that a job can have and an additional index
3444
                   (imAttributeInstanceIndex) for those attributes that can have multiple instances per job. The
3445
                   imAttributeTypeIndex object SHALL contain an enum type that indicates the type of attribute
3446
                   (see the JmAttributeTypeTC textual-convention). The value of the attribute SHALL be
```

3447 represented in either the jmAttributeValueAsInteger or jmAttributeValueAsOctets objects, 3448 and/or both, as specified in the **JmAttributeTypeTC** textual-convention. 3449 3450 The agent SHALL create rows in the **jmAttributeTable** as the server or device is able to 3451 discover the attributes either from the job submission protocol itself or from the document 3452 PDL. As the documents are interpreted, the interpreter MAY discover additional attributes and 3453 so the agent adds additional rows to this table. As the attributes that represent resources are 3454 actually consumed, the usage counter contained in the **jmAttributeValueAsInteger** object is 3455 incremented according to the units indicated in the description of the **JmAttributeTypeTC** 3456 enum. 3457 3458 The agent SHALL maintain each row in the **jmJobTable** for at least the minimum time after a 3459 job completes as specified by the **jmGeneralAttributePersistence** object. 3460 3461 Zero or more entries SHALL exist in this table for each job in a job set." 3462 REFERENCE 3463 "See Section 3.3 entitled 'The Attribute Mechanism' for a description of the 3464 jmAttributeTable." 3465 INDEX { jmGeneralJobSetIndex, jmJobIndex, jmAttributeTypeIndex, 3466 imAttributeInstanceIndex } ::= { jmAttributeTable 1 } 3467 3468 3469 JmAttributeEntry ::= SEQUENCE { **jmAttributeTypeIndex** 3470 JmAttributeTypeTC, 3471 **jmAttributeInstanceIndex** Integer32(1..32767), Integer32(-2..2147483647), 3472 **jmAttributeValueAsInteger** 3473 **jmAttributeValueAsOctets** OCTET STRING(SIZE(0..63)) 3474 } 3475 3476 jmAttributeTypeIndex OBJECT-TYPE 3477 **JmAttributeTypeTC** SYNTAX 3478 MAX-ACCESS not-accessible 3479 STATUS current 3480 **DESCRIPTION** 3481 "The type of attribute that this row entry represents. 3482 The type MAY identify information about the job or document(s) or MAY identify a resource 3483 3484 required to process the job before the job start processing and/or consumed by the job as the job 3485 is processed. 3486 3487 Examples of job attributes (i.e., apply to the job as a whole) that have only one instance per job include: jobCopiesRequested(90), documentCopiesRequested(92), 3488 jobCopiesCompleted(91), documentCopiesCompleted(93), while examples of job attributes 3489 3490 that may have more than one instance per job include: **documentFormatIndex(37)**, and 3491 documentFormat(38). 3492 3493 Examples of document attributes (one instance per document) include: **fileName(34)**, and 3494 documentName(35). 3495

3496	Examples of required and consumed resource attributes include: pagesRequested(130),
3497	mediumRequested(170), pagesCompleted(131), and mediumConsumed(171), respectively."
3498	::= { jmAttributeEntry 1 }
3499	
3500	jmAttributeInstanceIndex OBJECT-TYPE
3501	SYNTAX Integer32(132767)
3502	MAX-ACCESS not-accessible
3503	STATUS current
3504	DESCRIPTION
3505	"A running 16-bit index of the attributes of the same type for each job. For those attributes with
3506	only a single instance per job, this index value SHALL be 1. For those attributes that are a
3507	single value per document, the index value SHALL be the document number, starting with 1 for
3508	the first document in the job. Jobs with only a single document SHALL use the index value of
3509	1. For those attributes that can have multiple values per job or per document, such as
3510	documentFormatIndex(37) or documentFormat(38), the index SHALL be a running index
3511	for the job as a whole, starting at 1."
3512	::= { jmAttributeEntry 2 }
3513	" (jim turioucezhu) z j
3514	jmAttributeValueAsInteger OBJECT-TYPE
3515	SYNTAX Integer32(-22147483647)
3516	MAX-ACCESS read-only
3517	STATUS current
3518	DESCRIPTION
3519	"The integer value of the attribute. The value of the attribute SHALL be represented as an
3520	integer if the enum description in the JmAttributeTypeTC textual-convention definition has
3521	the tag: 'INTEGER:'.
3522	the tag. It is a second of the
3523	Depending on the enum definition, this object value MAY be an integer, a counter, an index, or
3524	an enum, depending on the jmAttributeTypeIndex value. The units of this value are specified
3525	in the enum description.
3526	in the chain description.
3527	For those attributes that are accumulating job consumption as the job is processed as specified
3528	in the JmAttributeTypeTC textual-convention, SHALL contain the final value after the job
3529	completes processing, i.e., this value SHALL indicate the total usage of this resource made by
3530	the job.
3531	are job.
3532	A monitoring application is able to copy this value to a suitable longer term storage for later
3533	processing as part of an accounting system.
3534	processing as part of an accounting system.
3535	Since the agent MAY add attributes representing resources to this table while the job is waiting
3536	to be processed or being processed, which can be a long time before any of the resources are
3537	actually used, the agent SHALL set the value of the jmAttributeValueAsInteger object to 0
3538	for resources that the job has not vet consumed.

Attributes for which the concept of an integer value is meaningless, such as **fileName(34)**,

JmAttributeTypeTC definition and so an agent SHALL always return a value of '-1' to

indicate 'other' for the value of the jmAttributeValueAsInteger object for these attributes.

jobName, and processingMessage, do not have the 'INTEGER:' tag in the

3539 3540

3541

3542

```
3545
                   For attributes which do have the 'INTEGER:' tag in the JmAttributeTypeTC definition, if the
                   integer value is not (yet) known, the agent either (1) SHALL not materialize the row in the
3546
                   imAttributeTable until the value is known or (2) SHALL return a '-2' to represent an
3547
                   'unknown' counting integer value, a0' to represent an 'unknown' index value, and 32' to
3548
3549
                   represent an 'unknown(2)' enum value."
3550
             DEFVAL { -2 } -- default value is unknown(-2)
3551
             ::= { jmAttributeEntry 3 }
3552
3553
        jmAttributeValueAsOctets OBJECT-TYPE
3554
             SYNTAX
                          OCTET STRING(SIZE(0..63))
3555
             MAX-ACCESS read-only
3556
             STATUS
                         current
3557
             DESCRIPTION
3558
                   "The octet string value of the attribute. The value of the attribute SHALL be represented as an
3559
                   OCTET STRING if the enum description in the JmAttributeTypeTC textual-convention
3560
                   definition has the tag: 'OCTETS:'.
3561
3562
                   Depending on the enum definition, this object value MAY be a coded character set string (text),
3563
                   such as 'JmUTF8StringTC', or a binary octet string, such as DateAndTime'.
3564
                   Attributes for which the concept of an octet string value is meaningless, such as
3565
3566
                   pagesCompleted, do not have the tag 'OCTETS:' in the JmAttributeTypeTC definition and so
3567
                   the agent SHALL always return a zero length string for the value of the
3568
                   jmAttributeValueAsOctets object.
3569
                   For attributes which do have the 'OCTETS:' tag in the JmAttributeTypeTC definition, if the
3570
3571
                   OCTET STRING value is not (yet) known, the agent either SHALL not materialize the row in
                   the jmAttributeTable until the value is known or SHALL return a zero-length string."
3572
3573
                         { "H }
                                   -- empty string
3574
             ::= { jmAttributeEntry 4 }
3575
```

```
3576
       -- Notifications and Trapping
3577
       -- Reserved for the future
3578
3579
       jobmonMIBNotifications OBJECT IDENTIFIER ::= { jobmonMIB 2}
3580
3581
3582
3583
       -- Conformance Information
3584
3585
       jmMIBConformance OBJECT IDENTIFIER ::= { jobmonMIB 3 }
3586
3587
       -- compliance statements
3588
       imMIBCompliance MODULE-COMPLIANCE
3589
            STATUS current
3590
            DESCRIPTION
3591
                 "The compliance statement for agents that implement the
3592
                 job monitoring MIB."
3593
            MODULE -- this module
3594
            MANDATORY-GROUPS {
3595
                 jmGeneralGroup, jmJobIDGroup, jmJobGroup, jmAttributeGroup }
3596
3597
            OBJECT jmGeneralJobSetName
            SYNTAX JmUTF8StringTC (SIZE(0..8))
3598
3599
            DESCRIPTION
3600
                 "Only 8 octets maximum string length NEED be supported by the agent."
3601
            OBJECT jmJobOwner
SYNTAX JmJobStringTC (SIZE(0..16))
3602
3603
3604
            DESCRIPTION
3605
                 "Only 16 octets maximum string length NEED be supported by the agent."
3606
3607
       -- There are no CONDITIONALLY MANDATORY or OPTIONAL groups.
3608
3609
            ::= { jmMIBConformance 1 }
3610
3611
                       OBJECT IDENTIFIER ::= { jmMIBConformance 2 }
       imMIBGroups
3612
3613
       imGeneralGroup OBJECT-GROUP
3614
            OBJECTS {
                 jmGeneralNumberOfActiveJobs, jmGeneralOldestActiveJobIndex,
3615
3616
                 jmGeneralNewestActiveJobIndex, jmGeneralJobPersistence,
3617
                 jmGeneralAttributePersistence, jmGeneralJobSetName}
3618
            STATUS current
3619
            DESCRIPTION
3620
                 "The general group."
3621
            ::= { jmMIBGroups 1 }
3622
       jmJobIDGroup OBJECT-GROUP
3623
3624
            OBJECTS {
```

```
3625
                 jmJobIDJobSetIndex, jmJobIDJobIndex }
3626
            STATUS current
3627
            DESCRIPTION
3628
                 "The job ID group."
3629
            ::= { jmMIBGroups 2 }
3630
3631
       jmJobGroup OBJECT-GROUP
3632
            OBJECTS {
                 jmJobState, jmJobStateReasons1, jmNumberOfInterveningJobs,
3633
3634
                 jmJobKOctetsPerCopyRequested, jmJobKOctetsProcessed,
                 jmJobImpressionsPerCopyRequested, jmJobImpressionsCompleted, jmJobOwner }
3635
3636
            STATUS current
3637
            DESCRIPTION
                 "The job group."
3638
3639
            ::= { jmMIBGroups 3 }
3640
3641
       jmAttributeGroup OBJECT-GROUP
3642
            OBJECTS {
3643
                 jmAttributeValueAsInteger, jmAttributeValueAsOctets }
3644
            STATUS current
3645
            DESCRIPTION
                 "The attribute group."
3646
3647
            ::= { jmMIBGroups 4 }
3648
3649
3650
       END
```

5. Appendix A - Implementing the Job Life Cycle

- The job object has well-defined states and client operations that affect the transition between the
- job states. Internal server and device actions also affect the transitions of the job between the job
- states. These states and transitions are referred to as the job's *life cycle*.
- Not all implementations of job submission protocols have all of the states of the job model
- specified here. The job model specified here is intended to be a superset of most
- implementations. It is the purpose of the agent to map the particular implementation's job life
- 3658 cycle onto the one specified here. The agent MAY omit any states not implemented. Only the
- processing and completed states are required to be implemented by an agent. However, a
- 3660 conforming management application SHALL be prepared to accept any of the states in the job
- 3661 life cycle specified here, so that the management application can interoperate with any
- 3662 conforming agent.
- The job states are intended to be user visible. The agent SHALL make these states visible in the
- 3664 MIB, but only for the subset of job states that the implementation has. Some implementations
- 3665 MAY need to have sub-states of these user-visible states. The **jmJobStateReasons1** object and
- 3666 the **jobStateReasons**N(N=2..4) attributes can be used to represent the sub-states of the jobs.
- Job states are intended to last a user-visible length of time in most implementations. However,
- some jobs may pass through some states in zero time in some situations and/or in some
- 3669 implementations.
- 3670 The job model does not specify how accounting and auditing is implemented, except to assume
- that accounting and auditing logs are separate from the job life cycle and last longer than job
- entries in the MIB. Jobs in the **completed**, aborted, or canceled states are not logs, since jobs in
- these states are accessible via SNMP protocol operations and SHALL be removed from the Job
- 3674 Monitoring MIB tables after a site-settable or implementation-defined period of time. An
- accounting application MAY copy accounting information incrementally to an accounting log as
- a job processes, or MAY be copied while the job is in the canceled, aborted, or completed
- states, depending on implementation. The same is true for auditing logs.
- 3678 The jmJobState object specifies the standard job states. The normal job state transitions
- are shown in the state transition diagram presented in Table 1.

3680 6. APPENDIX B - Support of the Job Submission ID in Job Submission

- 3681 **Protocols**
- 3682 This appendix lists the job submission protocols that support the concept of a job
- submission ID and indicates the attribute used in that job submission protocol.

3684	6.1 Hewlett-Packard's Printer Job Language (PJL)
3685 3686 3687 3688 3689	Hewlett-Packard's Printer Job Language provides job-level printer control and printer status information to applications. The PJL JOB command is used at the beginning of a print job and can include options applying only to that job. A PJL JOB command option has been defined to facilitate passing the JobSubmissionID with the print job, as required by the Job Monitoring MIB. The option is of the form:
3690 3691 3692	SUBMISSIONID = "id string"
3693 3694	Where the "id string" is a string and SHALL be enclosed in double quotes. The format is as described for the jmJobSubmissionID object.
3695	The entire PJL JOB command with the optional parameter would be of the form:
3696 3697 3698	@PJL JOB SUBMISSIONID = "id string"
3699 3700 3701	See "Printer Job Language Technical Reference Manual", part number 5021-0328, from Hewlett-Packard for complete information on the PJL JOB command and the Printer Job Language.
3702 3703 3704 3705 3706	NOTE - Some PJL implementations wrap a banner page as a PJL job around a job submitted by a client. In this case, there will be two job submission ids. The outer one being the one with the banner page and the inner one being the original user's job. The agent SHALL use the last received job submission ID for the jmJobSubmissionID index, so that the original user's job submission ID will be used, not the banner page job ID.
3707	6.2 ISO DPA
3708 3709	The ISO 10175 Document Printing Application (DPA) protocol specifies the " job-client-id " attribute that allows the client to supply a text string ID for each job.
3710	7. References
3711 3712	[char-set policy] Harald Avelstrand, "IETF Policy on Character Sets and Language", June 1997. Latest draft: <draft-avelstrand-charset-policy-00.txt></draft-avelstrand-charset-policy-00.txt>
3713 3714	[GB2312] GB 2312-1980, "Chinese People's Republic of China (PRC) mixed one byte and two byte coded character set"
3715	[hr-mib] P. Grillo, S. Waldbusser, "Host Resources MIB", RFC 1514, September 1993
3716 3717	[iana] J. Reynolds, and J. Postel, "Assigned Numbers", STD 2, RFC 1700, ISI, October 1994.

- 3718 [IANA-charsets] Coded Character Sets registered by IANA and assigned an enum value
- 3719 for use in the **CodedCharSet** textual convention imported from the Printer MIB. See
- 3720 ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets
- [iana-media-types] IANA Registration of MIME media types (MIME content
- 3722 types/subtypes). See ftp://ftp.isi.edu/in-notes/iana/assignments/
- 3723 [ISO-639] ISO 639:1988 (E/F) Code for Representation of names of languages The
- 3724 <u>International Organization for Standardization, 1st edition, 1988.</u>
- 3725 [ISO 646] ISO/IEC 646:1991, "Information technology -- ISO 7-bit coded character set
- 3726 for information interchange", JTC1/SC2.
- 3727 [ISO 8859] ISO/IEC 8859-1:1987, "Information technology -- 8-bit single byte coded
- 3728 graphic character sets Part 1: Latin alphabet No. 1, JTC1/SC2."
- 3729 [ISO 2022] ISO/IEC 2022:1994 "Information technology -- Character code structure
- and extension techniques", JTC1/SC2.
- 3731 [ISO-3166] ISO 3166:1988 (E/F) Codes for representation of names of countries The
- 3732 <u>International Organization for Standardization, 3rd edition, 1988-08-15."</u>
- 3733 [ISO-10646] ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-
- 3734 Octet Coded Character Set (UCS) Part 1: Architecture and Basic Multilingual Plane,
- 3735 JTC1/SC2.
- 3736 [iso-dpa] ISO/IEC 10175 Document Printing Application (DPA). See
- 3737 ftp://ftp.pwg.org/pub/pwg/dpa/
- 3738 [ipp-model] Internet Printing Protocol (IPP), work in progress on the IETF standards
- track. See draft-ietf-ipp-model-071.txt. See also http://www.pwg.org/ipp/index.html
- 3740 [JIS X0208] JIS X0208-1990, "Japanese two byte coded character set."
- 3741 [mib-II] MIB-II, RFC 1213.
- 3742 [print-mib] The Printer MIB RFC 1759, proposed IETF standard. Also an Internet-
- 3743 Draft on the standards track as a draft standard: **draft-ietf-printmib-mib-info-02.txt**
- [req-words] S. Bradner, "Keywords for use in RFCs to Indicate Requirement Levels",
- 3745 RFC 2119, March 1997.
- 3746 [rfc 1738] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource Locators
- 3747 (URL)", RFC 1738, December 1994.
- 3748 [RFC-1766] Avelstrand H., "Tags for the Identification of Languages", RFC 1766, March
- 3749 1995.

- [rfc 2130] C. Weider, C. Preston, K. Simonsen, H. Alvestrand, R. Atkinson, M. Crispin, and P. Svanberg, "The Report of the IAB Character Set Workshop held 29 Feb-1 March,
- 3752 1997", April 1997, RFC 2130.
- [SMIv2-TC] J. Case, et al. "Textual Conventions for Version 2 of the Simple Network
- 3754 Management Protocol (SNMPv2)", RFC 1903, January 1996.
- 3755 [tipsi] IEEE 1284.1, Transport-independent Printer System Interface (TIPSI).
- 3756 [URI-spec] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource Locators
- 3757 (URL)", RFC 1738, December, 1994.
- 3758 [US-ASCII] Coded Character Set 7-bit American Standard Code for Information
- 3759 Interchange, ANSI X3.4-1986.
- 3760 [UTF-8] F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", RFC
- 3761 2044, October 1996.

3762 **8. Author's Addresses**

- 3763 Ron Bergman
- 3764 Dataproducts Corp.
- 3765 1757 Tapo Canyon Road
- 3766 Simi Valley, CA 93063-3394
- 3767
- 3768 Phone: 805-578-4421
- 3769 Fax: 805-578-4001
- 3770 Email: rbergman@dpc.com
- 3771
- 3772
- 3773 Tom Hastings
- 3774 Xerox Corporation, ESAE-231
- 3775 701 S. Aviation Blvd.
- 3776 El Segundo, CA 90245
- 3777
- 3778 Phone: 310-333-6413
- 3779 Fax: 310-333-5514
- 3780 EMail: hastings@cp10.es.xerox.com
- 3781
- 3782
- 3783 Scott A. Isaacson
- Novell, Inc.
- 3785 122 E 1700 S
- 3786 Provo, UT 84606

```
3787
3788
             Phone: 801-861-7366
3789
             Fax: 801-861-4025
3790
             EMail: scott isaacson@novell.com
3791
3792
3793
             Harry Lewis
3794
             IBM Corporation
             6300 Diagonal Hwy
3795
             Boulder, CO 80301
3796
3797
3798
             Phone: (303) 924-5337
3799
             Fax:
3800
             Email: harryl@us.ibm.com
3801
3802
3803
             Send comments to the printmib WG using the Job Monitoring Project (JMP)
3804
             Mailing List: jmp@pwg.org
3805
3806
             To learn how to subscribe, send email to: jmp-request@pwg.org
3807
             For further information, access the PWG web page under "JMP":
3808
3809
             http://www.pwg.org/
3810
3811
        Other Participants:
3812
             Chuck Adams - Tektronix
             Jeff Barnett - IBM
3813
3814
             Keith Carter, IBM Corporation
             Jeff Copeland - QMS
3815
             Andy Davidson - Tektronix
3816
3817
             Roger deBry - IBM
             Mabry Dozier - OMS
3818
             Lee Ferrel - Canon
3819
3820
             Steve Gebert - IBM
3821
             Robert Herriot - Sun Microsystems Inc.
3822
             Shige Kanemitsu - Kyocera
3823
             David Kellerman - Northlake Software
3824
             Rick Landau - Digital
             Harry Lewis - IBM
3825
3826
             Pete Loya - HP
```

Job Monitoring MIB, V0.86

3827	Ray Lutz - Cognisys
3828	Jay Martin - Underscore
3829	Mike MacKay, Novell, Inc.
3830	Stan McConnell - Xerox
3831	Carl-Uno Manros, Xerox, Corp.
3832	Pat Nogay - IBM
3833	Bob Pentecost - HP
3834	Rob Rhoads - Intel
3835	David Roach - Unisys
3836	Hiroyuki Sato - Canon
3837	Bob Setterbo - Adobe
3838	Gail Songer, EFI
3839	Mike Timperman - Lexmark
3840	Randy Turner - Sharp
3841	William Wagner - Digital Products
3842	Jim Walker - Dazel
3843	Chris Wellens - Interworking Labs
3844	Rob Whittle - Novell
3845	Don Wright - Lexmark
3846	Lloyd Young - Lexmark
3847	Atsushi Yuki - Kyocera
3848	Peter Zehler, Xerox, Corp.

Sep 19, 1997

9. INDEX

3849

3850

3851 3852

3853

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

	· · · · · · · · · · · · · · · · · · ·	2000	1 1	
		3888	jmGeneralNumberOfActiveJobs	
3854	_C _	3889	jmGeneralOldestActiveJobIndex	
		3890	jmJobIDJobIndex	
3855	colorantConsumed	54 3891	jmJobIDJobSetIndex	
3856	colorantRequested	54 3892	jmJobImpressionsCompleted	
		3893	jmJobImpressions PerCopy Requested	
3857	_D_	3894	jmJobIndex	
	_	3895	jmJobKOctetsProcessed	
3858	deviceNameRequested	46 3896	jmJobKOctetsPerCopyRequested	74
3859	documentCopiesCompleted	51 3897	jmJobOwner	
3860	documentCopiesRequested	50 3898	JmJobServiceTypesTC	
3861	documentFormat	48 <i>3</i> 899	JmJobSourcePlatformTypeTC	
3862	documentFormatIndex	47 3900	jmJobState	
3863	documentName	47 3901	jmJobStateReasons1	
		3902	JmJobStateReasons1TC	
3864	F	3903	JmJobStateReasons2TC	
3004	-	3904	JmJobStateReasons3TC	65
3865	fileName	₄₇ 3905	JmJobStateReasons4TC	66
3866	finishing	49 3900	JmJobStateTC	40
3867	fullColorImpressionsCompleted	52 3907	JmJobStringTC	32
	<u>r</u>	3908	jmJobSubmissionID	
2060	—H—	3909	JmJobSubmission <u>ID</u> TypeTC	
3868		3910	JmMediumTypeTC	36
3869	highlight Color Impressions Completed	₅₂ 3911	JmNaturalLanguageTagTC	32
5007	mamancolor impressions completed	3712	jmNumberOfInterveningJobs	74
2070	*	3913	JmPrinterResolutionTC	35
3870	— I —	3914	JmPrintQualityTC	
3871	impressionsCompletedCurrentCopy	₅₂ 3915	JmTimeStampTC	
3872	impressionsCompletedCurrentCopyimpressionsInterpreted	32 3916	JmTonerEconomyTC	35
3873	impressions Sont To Dovice	₅₁ 3917	JmUTF8StringTC	32
3874	impressionsSentToDeviceimpressionsSpooled	₅₁ 3918	jobAccountName	
3014	mipressions/spooled	······· 31 3919	jobCodedCharSet	
		3920	jobComment	47
3875	J	3921	jobCompletionTime	
3876	jmAttributeInstanceIndex	₂₀ 3922	jobCopiesCompleted	
	jmAttributeInstanceIndex	⁷⁸ 3923	jobCopiesRequested	
3877	jmAttributeTypeIndex	⁷⁸ 3923		
3878	JmAttributeTypeTC		jobHoldUntil	
3879	jmAttributeValueAsInteger		jobKOctetsTransferred	
3880	jmAttributeValueAsOctets	······· ⁷⁹ 3927	jobName	
3881	JmBooleanTC	36 3928	jobOriginatingHost	
3882	JmFinishingTC	33 3929	jobPriority	
3883	jmGeneralAttributePersistence	······ ⁶⁹ 3930	jobProcessAfterDateAndTime	
3884	JmAttributeTypeTC jmAttributeValueAsInteger jmAttributeValueAsOctets. JmBooleanTC JmFinishingTC jmGeneralAttributePersistence jmGeneralJobPersistence jmGeneralJobSetIndex.	······· ⁶⁹ 3931	jobProcessingCPUTime	
3885	jmGeneralJobSetIndexjmGeneralJobSetName	67 3932	jobServiceTypes	
3886	jmGeneralJobSetName	69 3933	jobSourceChannelIndex	
3887	jmGeneralNewestActiveJobIndex	68 3934	jobSourcePlatformType	
		3734	jousourcer lauorin rype	40

3935 3936 3937 3938 3939 3940 3941 3942	jobStartedBeingHeldTime 55 3955 jobStartedProcessingTime 55 3956 jobStateReasons2 43 3957 jobStateReasons3 43 3958 jobStateReasons4 44 3959 jobSubmissionTime 55 3960 jobSubmissionToServerTime 55 jobURI 44 3961	physicalDevice 47 printerResolutionRequested 50 printerResolutionUsed 50 printQualityRequested 49 printQualityUsed 49 processingMessage 44
3943	─M ─ 3962	queueNameRequested47
3944 3945	mediumConsumed	_S_
3946	3964 N 3965 3966	serverAssignedJobName
3947	numberOfDocuments	sheetsRequested 53 sides 49
3948	—O — 3969 3970	submittingApplicationName 46 submittingServerName 46
3949 3950	other	—T—
3951	_P 3972 3973	tonerDensityRequested
3952 3953 3954	pagesCompleted	tonerEcomonyRequested
3976		