Job Monitoring MIB, V0.86 1 (This cover page is *not* part of the Internet-Draft) 2 3 4 From: Tom Hastings 5 Date: 09/19/97 Version: 0.86 6 7 File: ftp://ftp.pwg.org/pub/jmp/mibs/jmp-mib.doc .pdf imp-mibr.doc .pdf .pdr 8 Status: Ninth draft MIB that incorporates the agreements reached on the DL on issues 9 in V0.85 which was released after the 8/8 meeting and the agreements reached at the JMP 10 meeting on 9/19. In addition to the changes listed in Ron's list, the JMP agreed to remove 11 the finishing enums that IPP removed (because of a lack of a coordinate system 12 specification for stapling), add private enum range for attributes to agree with IPP. See 13 the change history in the separate file: changes.doc .pdf. 14 We agreed that the MIB specification is finished except for any editorial comments that 15 people may have. See the separate issues.doc and .pdf file. 16 I've also produced a variation on this document which has all variable font (jmp-mib.doc .pdf) without revision marks. This is the version that the JMP should use to make 17 18 comments. It has line numbers.

The MIB has been greatly simplified so that now there are only 18 objects in the MIB.

19

20

There are 65 attributes.

21	INTERNET-DRAFT Ron Bergman
22	Dataproducts Corp
23 24	Tom Hastings
24	Xerox Corporation
25	Scott Isaacson
26	Novell, Inc
27	Harry Lewis
28	IBM Corp
29	September 19, 1997
30	
31	Job Monitoring MIB - V0.86
32	<draft-ietf-printmib-job-monitor-06.txt></draft-ietf-printmib-job-monitor-06.txt>
33	Expires Mar 19, 1997
34	•
	Status of this Memo
35	Status of this Memo
36	This document is an Internet-Draft. Internet-Drafts are working documents of the
37	Internet Engineering Task Force (IETF), its areas, and its working groups. Note
38	that other groups may also distribute working documents as Internet-Drafts.
39	Internet-Drafts are draft documents valid for a maximum of six months and may be
40	updated, replaced, or obsoleted by other documents at any time. It is
41	inappropriate to use Internet-Drafts as reference material or to cite them other than
42	as "work in progress."
43	To learn the current status of any Internet-Draft, please check the "1id-
44	abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on
45	ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim),
46	ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).
47	Abstract
48	This Internet-Draft specifies a small set of read-only SNMP MIB objects for (1)
49	monitoring the status and progress of print jobs (2) obtaining resource
50	requirements before a job is processed, (3) monitoring resource consumption while
51	a job is being processed and (4) collecting resource accounting data after the
52	completion of a job. This MIB is intended to be implemented (1) in a printer or
53	(2) in a server that supports one or more printers. Use of the object set is not
54	limited to printing. However, support for services other than printing is outside
55	the scope of this Job Monitoring MIB. Future extensions to this MIB may include
56	but are not limited to, fax machines and scanners.

5	7
5	8

TABLE OF CONTENTS

59	1. INTRODUCTION	9
60	1.1 Types of Information in the MIB	9
61	1.2 Types of Job Monitoring Applications	10
62	2. TERMINOLOGY AND JOB MODEL	11
63	2.1 System Configurations for the Job Monitoring MIB	14
64	2.1.1 Configuration 1 - client-printer	
65	2.1.2 Configuration 2 - client-server-printer - agent in the server	15
66	2.1.3 Configuration 3 - client-server-printer - client monitors printer agent and server	16
67	3. MANAGED OBJECT USAGE	17
68	3.1 Conformance Considerations	17
69	3.1.1 Conformance Terminology	18
70	3.1.2 Agent Conformance Requirements	18
71	3.1.2.1 MIB II System Group objects	18
72	3.1.2.2 MIB II Interface Group objects	18
73	3.1.2.3 Printer MIB objects	19
74	3.1.3 Job Monitoring Application Conformance Requirements	19
75	3.2 The Job Tables and the Oldest Active and Newest Active Indexes	19
76	3.3 The Attribute Mechanism	
77	3.3.1 Conformance of Attribute Implementation	
78	3.3.2 Useful, 'Unknown', and 'Other' Values for Objects and Attributes	
79	3.3.3 Data Sub-types and Attribute Naming Conventions	
80	3.3.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes	
81	3.3.5 Requested Attributes	
82	3.3.6 Consumption Attributes	
83	3.3.7 Index Value Attributes	24
84	3.4 Job Identification	25
85	3.5 Internationalization Considerations	
86	3.5.1 'JmUTF8StringTC' for text generated by the server or device	
87	3.5.2 'JmJobStringTC' for text generated by the job submitter	
88	3.5.3 'DateAndTime' for representing the date and time	25
89	3.6 IANA Considerations	
90	3.6.1 IANA Registration of enums	27

91	3.6.1.1 Type 1 enumerations	27
92	3.6.1.2 Type 2 enumerations	28
93	3.6.1.3 Type 3 enumeration	28
94	3.6.2 IANA Registration of type 2 bit values	28
95	3.6.3 IANA Registration of Job Submission Id Formats	29
96	3.6.4 IANA Registration of MIME types/sub-types for document-formats	29
97	3.7 Security Considerations	29
98	3.7.1 Read-Write objects	
99	3.7.2 Read-Only Objects In Other User's Jobs	29
100	3.8 Values for Objects	22
101	3.9 Notifications	29
102	4. MIB SPECIFICATION	30
103	Textual conventions for this MIB module	32
104	JmUTF8StringTC	
105	JmJobStringTC	32
106	JmTimeStampTC	32
107	JmJobSourcePlatformTypeTC	33
108	JmFinishingTC	33
109	JmPrintQualityTC	
110	JmPrinterResolutionTC	
111	JmTonerEconomyTC	
112	JmBooleanTC	
113	JmMediumTypeTC	
114	JmJobSubmissionIDTypeTC	
115	JmJobStateTC	
116	JmAttributeTypeTC	
117	other (Int32(-2) and/or Octets63)	
118	Job State attributes	
119 120	jobStateReasons2 (JmJobStateReasons2TC)	
120	jobStateReasons3 (JmJobStateReasons3TC)	
121	processingMessage (UTF8String63)	
123	jobCodedCharSet (CodedCharSet)	
123	Job Identification attributes	
125	jobURI (Octets(1255))	
126	jobAccountName (Octets63)	
127	serverAssignedJobName (JobString63)	
128	jobName (JobString63)	
129	jobServiceTypes (JmJobServiceTypesTC)	
130	jobSourceChannelIndex (Int32(0))	
131	jobSourcePlatformType (JmJobSourcePlatformTypeTC)	
132	submittingServerName (JobString63)	
133	submittingApplicationName (JobString63)	

134	jobOriginatingHost (JobString63)	46
135	deviceNameRequested (JobString63)	46
136	queueNameRequested (JobString63)	
137	physicalDevice (hrDeviceIndex and/or UTF8String63)	
138	numberOfDocuments (Int32(-2))	
139	fileName (JobString63)	47
140	documentName (JobString63)	47
141	jobComment (JobString63)	
142	documentFormatIndex (Int32(0))	
143	documentFormat (PrtInterpreterLangFamilyTC and/or Octets63)	48
144	Job Parameter attributes	48
145	jobPriority (Int32(1100))	
146	jobProcessAfterDateAndTime (DateAndTime)	48
147	jobHold (JmBooleanTC)	49
148	jobHoldUntil (JobString63)	49
149	outputBin (Int32(0) and/or JobString63)	49
150	sides (Int32(-22))	49
151	finishing (JmFinishingTC)	49
152	Image Quality attributes (requested and used)	49
153	printQualityRequested (JmPrintQualityTC)	49
154	printQualityUsed (JmPrintQualityTC)	49
155	printerResolutionRequested (JmPrinterResolutionTC)	50
156	printerResolutionUsed (JmPrinterResolutionTC)	50
157	tonerEcomonyRequested (JmTonerEconomyTC)	50
158	tonerEcomonyUsed (JmTonerEconomyTC)	50
159	tonerDensityRequested (Int32(-2100))	50
160	tonerDensityUsed (Int32(-2100))	50
161	Job Progress attributes (requested and consumed)	50
162	jobCopiesRequested (Int32(-2))	50
163	jobCopiesCompleted (Int32(-2))	
164	documentCopiesRequested (Int32(-2))	50
165	documentCopiesCompleted (Int32(-2))	51
166	jobKOctetsTransferred (Int32(-2))	
167	Impression attributes (requested and consumed)	
168	impressionsSpooled (Int32(-2))	51
169	impressionsSentToDevice (Int32(-2))	
170	impressionsInterpreted (Int32(-2))	
171	impressionsCompletedCurrentCopy (Int32(-2))	52
172	fullColorImpressionsCompleted (Int32(-2))	
173	highlightColorImpressionsCompleted (Int32(-2))	52
174	Page attributes (requested and consumed)	52
175	pagesRequested (Int32(-2))	52
176	pagesCompleted (Int32(-2))	52
177	pagesCompletedCurrentCopy (Int32(-2))	53
178	Sheet attributes (requested and consumed)	
179	sheetsRequested (Int32(-2))	
180	sheetsCompleted (Int32(-2))	
181	sheetsCompletedCurrentCopy (Int32(-2))	53
182	Resource attributes (requested and consumed)	53

183	mediumRequested (JmMediumTypeTC and/or JobString63)	53
184	mediumConsumed (JobString63)	54
185	colorantRequested (Int32(-2) and/or JobString63)	54
186	colorantConsumed (Int32(-2) and/or JobString63)	54
187	Time attributes (set by server or device)	54
188	jobSubmissionToServerTime (JmTimeStampTC and/or DateAndTime)	
189	jobSubmissionTime (JmTimeStampTC and/or DateAndTime)	
190	jobStartedBeingHeldTime (JmTimeStampTC and/or DateAndTime)	
191	jobStartedProcessingTime (JmTimeStampTC and/or DateAndTime)	
192	jobCompletionTime (JmTimeStampTC and/or DateAndTime)	
193	jobProcessingCPUTime (Int32(-2))	
194	JmJobServiceTypesTC	
195	JmJobStateReasons1TC	
196	JmJobStateReasons2TC	
197	JmJobStateReasons3TC	
198	JmJobStateReasons4TC	
199	The General Group (MANDATORY)	67
200	jmGeneralJobSetIndex (Int32(132767))	
201	jmGeneralNumberOfActiveJobs (Int32(0))	
202	jmGeneralOldestActiveJobIndex (Int32(0))	
203	jmGeneralNewestActiveJobIndex (Int32(0))	
204	jmGeneralJobPersistence (Int32(15))	
205	jmGeneralAttributePersistence (Int32(15))	
206	jmGeneralJobSetName (UTF8String63)	
207	The Job ID Group (MANDATORY)	70
208	jmJobSubmissionID (OCTET STRING(SIZE(48)))	
209	jmJobIDJobSetIndex (Int32(132767))	
210	jmJobIDJobIndex (Int32(1))	71
211	The Job Group (MANDATORY)	72
212	jmJobIndex (Int32(1))	73
213	jmJobState (JmJobStateTC)	73
214	jmJobStateReasons1 (JmJobStateReasons1TC)	73
215	jmNumberOfInterveningJobs (Int32(-2))	74
216	jmJobKOctetsRequested (Int32(-2))	74
217	jmJobKOctetsProcessed (Int32(-2))	74
218	jmJobImpressionsRequested (Int32(-2))	75
219	jmJobImpressionsCompleted (Int32(-2))	75
220	jmJobOwner (JobString63)	76
221	The Attribute Group (MANDATORY)	76
222	jmAttributeTypeIndex (JmAttributeTypeTC)	77
223	jmAttributeInstanceIndex (Int32(132767))	78
224	jmAttributeValueAsInteger (Int32(-2))	78
225	imAttributeValueAsOctets (Octets63)	70

226	5. APPENDIX A - IMPLEMENTING THE JOB LIFE CYCLE	82
227 228	6. APPENDIX B - SUPPORT OF THE JOB SUBMISSION ID IN JOB SUBMISSION PROTOCOLS	82
229	6.1 Hewlett-Packard's Printer Job Language (PJL)	83
230	6.2 ISO DPA	83
231	7. REFERENCES	83
232	8. AUTHOR'S ADDRESSES	85
233	9. INDEX	88

235 **Job Monitoring MIB** 1. Introduction 236 237 The Job Monitoring MIB is intended to be implemented by an agent within a printer or the 238 first server closest to the printer, where the printer is either directly connected to the 239 server only or the printer does not contain the job monitoring MIB agent. It is 240 recommended that implementations place the SNMP agent as close as possible to the 241 processing of the print job. This MIB applies to printers with and without spooling 242 capabilities. This MIB is designed to be compatible with most current commonly-used job 243 submission protocols. In most environments that support high function job submission/job 244 control protocols, like ISO DPA[iso-dpa], those protocols would be used to monitor and 245 manage print jobs rather than using the Job Monitoring MIB. 246 The Job Monitoring MIB consists of a General Group, a Job Submission ID Group, a Job 247 Group, and an Attribute Group. Each group is a table. All accessible objects are read-248 only. The General Group contains general information that applies to all jobs in a job set. 249 The Job Submission ID table maps the job submission ID that the client uses to identify a 250 job to the **imJobIndex** that the Job Monitoring Agent uses to identify jobs in the Job and 251 Attribute tables. The Job table contains the MANDATORY integer job state and status 252 objects. The Attribute table consists of multiple entries per job that specify (1) job and 253 document identification and parameters, (2) requested resources, and (3) consumed 254 resources during and after job processing/printing. A larger number of job attributes are 255 defined as textual conventions that an agent SHALL return if the server or device 256 implements the functionality so represented and the agent has access to the information. 257 1.1 Types of Information in the MIB 258 The job MIB is intended to provide the following information for the indicated Role 259 Models in the Printer MIB[print-mib] (Appendix D - Roles of Users). 260 User: 261 Provide the ability to identify the least busy printer. The user will be able to 262 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. 263 264 Provide the ability to identify the current status of the user's job (user queries). 265 Provide a timely indication that the job has completed and where it can be found. Provide error and diagnostic information for jobs that did not successfully 266 complete. 267

Operator:

Provide a presentation of the state of all the jobs in the print system.

270 Provide the ability to identify the user that submitted the print job. 271 Provide the ability to identify the resources required by each job. 272 Provide the ability to define which physical printers are candidates for the print 273 job. 274 Provide some idea of how long each job will take. However, exact estimates of 275 time to process a job is not being attempted. Instead, objects are included that 276 allow the operator to be able to make gross estimates. 277 Capacity Planner: 278 Provide the ability to determine printer utilization as a function of time. 279 Provide the ability to determine how long jobs wait before starting to print. 280 Accountant: 281 Provide information to allow the creation of a record of resources consumed and 282 printer usage data for charging users or groups for resources consumed. 283 Provide information to allow the prediction of consumable usage and resource 284 need. 285 The MIB supports printers that can contain more than one job at a time, but still be usable 286 for low end printers that only contain a single job at a time. In particular, the MIB 287 supports the needs of Windows and other PC environments for managing low-end direct-288 connect (serial or parallel) and networked devices without unnecessary overhead or 289 complexity, while also providing for higher end systems and devices. 290 1.2 Types of Job Monitoring Applications 291 The Job Monitoring MIB is designed for the following types of monitoring applications: 292 Monitor a single job starting when the job is submitted and ending a defined 293 period after the job completes. The Job Submission ID table provides the map 294 to find the specific job to be monitored. 295 Monitor all 'active' jobs in a queue, which this specification generalizes to a 296 "job set". End users may use such a program when selecting a least busy printer, so the MIB is designed for such a program to start up quickly and find 297 the information needed quickly without having to read all (completed) jobs in 298 299 order to find the active jobs. System operators may also use such a program,

in which case it would be running for a long period of time and may also be

interested in the jobs that have completed. Finally such a program may be

used to provide an enhanced console and logging capability.

269

300

301

- 303 3. Collect resource usage for accounting or system utilization purposes that copy 304 the completed job statistics to an accounting system. It is recognized that 305 depending on accounting programs to copy MIB data during the job-retention 306 period is somewhat unreliable, since the accounting program may not be 307 running (or may have crashed). Such a program is also expected to keep a 308 shadow copy of the entire Job Attribute table including completed, 309 canceled, and aborted jobs which the program updates on each polling cycle. 310 Such a program polls at the rate of the persistence of the **Attribute** table. 311 The design is not optimized to help such an application determine which jobs 312 are **completed**, **canceled**, or **aborted**. Instead, the application SHALL query 313 each job that the application's shadow copy shows was not complete, 314 canceled, or aborted at the previous poll cycle to see if it is now complete or 315 **canceled**, plus any new jobs that have been submitted.
- 316 The MIB provides a set of objects that represent a compatible subset of job and document 317 attributes of the ISO DPA standard[iso-dpa] and the Internet Printing Protocol (IPP)[ipp-318 model], so that coherence is maintained between these two protocols and the information 319 presented to end users and system operators by monitoring applications. However, the 320 job monitoring MIB is intended to be used with printers that implement other job 321 submitting and management protocols, such as IEEE 1284.1 (TIPSI)[tipsi], as well as 322 with ones that do implement ISO DPA. Thus the job monitoring MIB does not require 323 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
- 330 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.
- 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
- 336 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
- 339 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
- 342 SHALL be represented in more than one MIB job set.
- Document: A sub-section within a job that contains print data and *document instructions*
- that apply to just the document.
- 345 Client: The network entity that end users use to submit jobs to spoolers, servers, or
- 346 *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
- 348 to a networked-connected device.
- 349 Server: A network entity that accepts jobs from clients and in turn submits the jobs to
- 350 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.
- 357 Printer: A *device* that puts marks on media.
- 358 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.
- 360 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.
- 363 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.
- Oueuing: The act of a *device* or *server* of ordering (queuing) the jobs for the purposes of
- scheduling the jobs to be processed.
- 367 Monitor or Job Monitoring Application: The SNMP management application that End
- 368 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.
- 370 Accounting Application: The SNMP management application that copies job information
- 371 to some more permanent medium so that another application can perform accounting on
- the data for Accountants, Asset Managers, and Capacity Planners use.
- 373 Agent: The network entity that accepts SNMP requests from a *monitor* or *accounting*
- 374 application and provides access to the instrumentation for managing jobs modeled by the
- management objects defined in the Job Monitoring MIB module for a *server* or a *device*.

- 376 Proxy: An agent that acts as a concentrator for one or more other agents by accepting
- 377 SNMP operations on the behalf of one or more other agents, forwarding them on to those
- other agents, gathering responses from those other agents and returning them to the
- original requesting monitor.
- 380 User: A person that uses a client or a monitor.
- 381 End User: A user that uses a client to submit a print job.
- 382 System Operator: A user that uses a monitor to monitor the system and carries out tasks
- 383 to keep the system running.
- 384 System Administrator: A user that specifies policy for the system.
- Job Instruction: An instruction specifying how, when, or where the job is to be processed.
- Job instructions MAY be passed in the job submission protocol or MAY be embedded in
- 387 the document data or a combination depending on the job submission protocol and
- 388 implementation.
- 389 Document Instruction: An instruction specifying how to process the document.
- 390 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,
- depending on the job submission protocol and implementation.
- 393 SNMP Information Object: A name, value-pair that specifies an action, a status, or a
- 394 condition in an SNMP MIB. Objects are identified in SNMP by an OBJECT
- 395 IDENTIFIER.
- 396 Attribute: A name, value-pair that specifies a job or document instruction, a status, or a
- 397 condition of a job or a document that has been submitted to a server or device. A
- 398 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
- 400 (1) the client supplied a value in the job submission protocol, (2) the document data
- 401 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
- 403 which entries are present only when necessary. Attributes are identified in this MIB by an
- 404 enum.
- Job Monitoring (using SNMP): The activity of a management application of accessing the
- 406 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.
- 408 Job Accounting: The activity of a management application of accessing the MIB and
- 409 recording what happens to the job during and after the processing of the job.

410 **2.1** System Configurations for the Job Monitoring MIB

- This section enumerates the three configurations in which the Job Monitoring MIB is
- 412 intended to be used. To simplify the pictures, the *devices* are shown as *printers*. See
- 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"
- 415 is assumed for this MIB as well. Please refer to that diagram to aid in understanding the
- 416 following system configurations.

417

425

440

443

444

445

446

447

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.
- 420 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
- 422 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.

```
426
                            end-user ####### SNMP query
                  all
427
                                        ---- job submission
               428
429
430
                   # ###########
431
432
                   # #
433
            +==+===#=#=+==+
434
                agent
435
               +----+
436
                PRINTER
437
                          Print Job Delivery Channel
438
439
            +=======+
```

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

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.

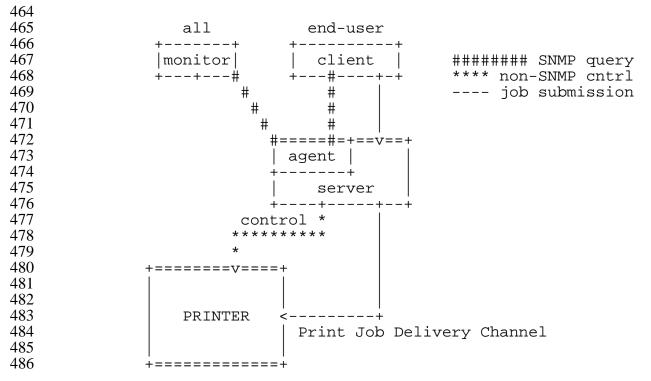


Figure 2-2 - Configuration 2 - client-server-printer - agent in the server

- 488 The Job Monitoring MIB is designed to support the following relationships (not shown in 489 Figure 2-2): 490 1. Multiple **clients** MAY submit jobs to a **server**. 491
 - 2. Multiple clients MAY monitor a server.
 - 3. Multiple monitors MAY monitor a server.
 - A **client** MAY submit jobs to multiple **servers**. 4.
 - 5. A monitor MAY monitor multiple servers.
- 495 Multiple servers MAY submit jobs to a printer. 6.
- 496 7. Multiple servers MAY control a printer.
- 497 2.1.3 Configuration 3 - client-server-printer - client monitors printer agent and 498 server
- 499 In the **client-server-printer** configuration 3, the **client**(s) submit jobs to an intermediate
- 500 **server** by some network connection, *not* directly to the **printer**. That server does *not*
- contain a Job Monitoring MIB agent. 501

492

493

494

504

505

506

507

508

509

- 502 The job submitting **client** and/or **monitoring application** monitor jobs by communicating 503 directly with:
 - The **server** using some undefined protocol to monitor jobs in the server (that does not contain the Job Monitoring MIB) AND
 - A Job Monitoring MIB agent that is part of the **printer** to monitor jobs after 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 printer usually almost immediately (before the job does much processing, if any).
- 511 In configuration 3, the agent (in the **printer**) SHALL keep the values of the objects in the
- 512 Job Monitoring MIB that the agent implements updated for a job that the server has
- 513 submitted to the printer. The agent SHALL obtain information about the jobs submitted
- 514 to the printer from the server (either in the job submission protocol, in the document data,
- 515 or by direct query of the server), in order to populate some of the objects the Job
- 516 Monitoring MIB in the printer. The agent in the printer SHALL keep the job in the Job
- 517 Monitoring MIB as long as the job is in the Printer, and longer in order to implement the
- 518 completed state in which monitoring programs can copy out the accounting data from the
- 519 Job Monitoring MIB.

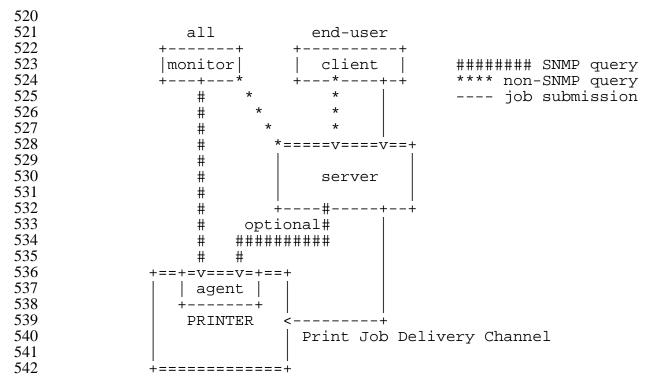


Figure 2-3 - Configuration 3 - client-server-printer - client monitors printer agent and server

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

543

544

547

548

549

550 551

552

553

554

556

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
- 558 monitoring agents, this specification includes the conformance requirements for both
- monitoring applications and agents.

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

575 A conforming agent:

574

576

580

581 582

583

- 1. SHALL implement *all* MANDATORY groups in this specification.
- 577 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.
- 587 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.
- 590 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.

593 3.1.2.3 Printer MIB objects

600

602

603

604

605

606

607

608

609

610

611 612

613

614

615

619

620

623

624

625

626

- 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 MIB[hr-
- 597 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 3.6 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.

616 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. jmGeneralJobSetIndex 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
- 629 incremented value of **imJobIndex** 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
- 636 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 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
- 649 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
- jmGeneralNewestActiveJobIndex object. If the new job is to be 'inactive'
- 654 (pendingHeld state), the agent SHALL not change the value of
- imGeneralNewestActiveJobIndex object (though the agent SHALL assign the next
- incremental **jmJobIndex** 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 **imJobIndex** value that matches the
- imGeneralOldestActiveJobIndex 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 **imGeneralOldestActiveJobIndex** or the
- imGeneralNewestActiveJobIndex objects, or both, if the job's jmJobIndex value is

- outside the range between **jmGeneralOldestActiveJobIndex** and
- 668 jmGeneralNewestActiveJobIndex.
- When all jobs become 'inactive', i.e., enter the **pendingHeld**, **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
- MIB that specifies the maximum value for the **jmJobIndex** supported by the
- 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
- **jmJobIndex** as recent jobs before the power cycle.

687 **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
- 690 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 **imAttributeTable** as soon as the agent
- is aware of the value of the attribute.
- The agent materializes job attributes in a four-indexed **jmAttributeTable**:
- 695 1. jmGeneralJobSetIndex which job set
- 696 2. jmJobIndex which job in the job set
- 3. jmAttributeTypeIndex which attribute
- 698 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
- indicate the amount of the resource consumed during and after processing, e.g., pages
- 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
- 706 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
- 712 SHALL NOT be used with other configurations.

713 **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.
- 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
- 718 for that attribute. See next section.
- 719 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
- 721 **imAttributeTable**.

722 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
- attribute.
- 727 SNMP 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
- 731 SNMPv2 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
- 733 row consisting of both the jmAttributeValueAsInteger and jmAttributeValueAsOctets
- objects.
- 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

- 738 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
- 741 **jmAttributeValueAsInteger** and **jmAttributeValueAsOctets** MANDATORY objects,
- NMP 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
- 746 **jmAttributeValueAsOctets** object. For octet string only attributes, the agent SHALL
- always return a '-1' value for the **jmAttributeValueAsInteger** object.

748 **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
- 750 **OCTET STRING**. The data sub-type of each attribute is indicated on the first line(s) of
- 751 the description. Some attributes have several different data sub-type representations.
- 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
- 759 the attribute: **Name**, **Index**, **DateAndTime**, **TimeStamp**, etc. For example,
- 760 **documentFormatIndex(37)** is an index.
- NOTE: The Table of Contents also lists the data sub-type and/or data sub-types of each
- 762 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.

764 3.3.4 Single-Value (Row) Versus Multi-Value (MULTI-ROW) Attributes

- 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
- jmAttributeTable. Unless indicated with 'MULTI-ROW:' in the JmAttributeTypeTC
- description, an agent SHALL ensure that each attribute occurs only once in the
- 769 **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
- 774 **fileName(34)** or **documentName(35)** which are specified to be 'per-document' attributes,
- but are *not* allowed for 'intensive' 'MULTI-ROW' attributes, such as
- 776 **mediumConsumed(171)** and **documentFormat(38)** which are specified to be 'per-job'
- 777 attributes.

778 3.3.5 Requested Attributes

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

785 **3.3.6 Consumption Attributes**

- 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
- metering, the agent either: (1) SHALL return the value **0** or (2) SHALL *not* add this
- attribute to the **jmAttributeTable** until the consumption begins. In the interests of
- brevity, the semantics for **0** is specified once here and is *not* repeated for each consumptive
- 791 attribute specification.

792 **3.3.7 Index Value Attributes**

- A number of attributes are indexes in other tables. Such attribute names end with the
- word 'Index'. If the agent has not (yet) assigned an index value for a particular index
- attribute for a job, the agent SHALL either: (1) return the value **0** or (2) *not* add this
- 796 attribute to the **imAttributeTable** 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
- 798 attribute specification.

799 **3.4 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. 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
- imJobSubmissionID 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
- 816 SHALL assign the **imJobIndex** values.
- The MIB provides a mapping table that maps each **jmJobSubmissionID** value to the
- 818 corresponding **imJobIndex** value generated by the agent, so that an application can
- 819 determine the correct value for the **imJobIndex** value for the job of interest in a single
- Get operation, given the Job Submission ID. See the **jmJobIDGroup**.
- The **jobName** attribute provides a name that the user supplies as a job attribute with the
- 822 job. The **jobName** attribute is not necessarily unique, even for one user, let alone across
- 823 users.

824 **3.5 Internationalization Considerations**

This section describes the internationalization considerations included in this MIB.

826 3.5.1 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
- 830 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
- UTF-8 as recommended by RFC 2130 [RFC 2130] and the "IETF Policy on Character
- Sets and Language" [char-set policy]. The 'JmUTF8StringTC' textual convention is used
- 837 to indicate UTF-8 text strings.
- 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.

840 3.5.2 Text generated by the job submitter

- 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
- 845 MIB either (1) in the coded character set as they were submitted or (2) MAY convert the
- sign 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
- 850 255 SHALL represent single-byte or multi-byte graphic characters structured according to
- 851 ISO 2022 [ISO 2022] or SHALL be unused.
- The coded character set SHALL be one of the ones registered with IANA [IANA] and
- 853 SHALL be identified by the **iobCodedCharSet** attribute in the **imJobAttributeTable** 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
- 856 **jobCodedCharSet** attribute or (2) not return the **jobCodedCharSet** attribute for the job.
- 857 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
- 859 8859-1], any ISO 8859-n, HP Roman8, IBM Code Page 850, Windows Default 8-bit set,
- 860 UTF-8 [UTF-8], US-ASCII plus JIS X0208-1990 Japanese [JIS X0208], US-ASCII plus
- 861 GB2312-1980 PRC Chinese [GB2312]. See the IANA registry of coded character sets
- [IANA charsets].
- 863 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-
- 865 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.

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

 3.5.3 **'DateAndTime' for representing the date and time**
- This MIB also contains objects that are represented using the **DateAndTime** textual convention from SMIv2 [SMIv2-TC]. The job management application SHALL display
- such objects in the locale of the user running the monitoring application.
- 873 **3.6 IANA Considerations**
- During the development of this standard, the Printer Working Group (PWG) working with
- 875 IANA [iana] will register additional enums while the standard is in the proposed and draft
- states according to the procedures described in this section. IANA will handle registration
- of additional enums after this standard is approved in cooperation with an IANA-
- appointed registration editor from the PWG according to the procedures described in this
- 879 section:
- 880 **3.6.1 IANA Registration of enums**
- This specification uses textual conventions to define enumerated values (enums) and bit
- values. Enumerations (enums) and bit values are sets of symbolic values defined for use
- with one or more objects or attributes. All enumeration sets and bit value sets are
- assigned a symbolic data type name (textual convention). As a convention the symbolic
- name ends in "TC" for textual convention. These enumerations are defined at the
- beginning of the MIB module specification.
- This working group has defined several type of enumerations for use in the Job
- 888 Monitoring MIB and the Printer MIB[print-mib]. These types differ in the method
- employed to control the addition of new enumerations. Throughout this document,
- references to "type n enum", where n can be 1, 2 or 3 can be found in the various tables.
- The definitions of these types of enumerations are:
- 3.6.1.1 Type 1 enumerations
- 893 Type 1 enumeration: All the values are defined in the Job Monitoring MIB specification
- 894 (RFC for the Job Monitoring MIB). Additional enumerated values require a new RFC.
- There are no type 1 enums in the current draft.

- 3.6.1.2 Type 2 enumerations
- 897 Type 2 enumeration: An initial set of values are defined in the Job Monitoring MIB
- 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.
- 900 The following type 2 enums are contained in the current draft:
- 901 1. JmUTF8StringTC

902

903

904

907

908

- 2. JmJobStringTC
- 3. JmTimeStampTC
 - 4. JmFinishingTC [same enum values as IPP "finishing" attribute]
- 5. JmPrintQualityTC [same enum values as IPP "print-quality" attribute]
- 906 6. JmTonerEconomyTC
 - 7. JmMediumTypeTC
 - 8. JmJobSubmissionTypeTC
- 909 9. JmJobStateTC [same enum values as IPP "job-state" attribute]
- 910 10. JmAttributeTypeTC
- 911 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
- 913 the Job Monitoring MIB.
- 914 3.6.1.3 Type 3 enumeration
- Type 3 enumeration: An initial set of values are defined in the Job Monitoring MIB
- 916 specification. Additional enumerated values are registered through IANA without
- 917 working group review.
- There are no type 3 enums in the current draft.
- 919 **3.6.2 IANA Registration of type 2 bit values**
- 920 This draft contains the following type 2 bit value textual-conventions:
- 921 1. JmJobServiceTypesTC
- 922 2. JmJobStateReasons1TC
 - 3. JmJobStateReasons2TC
- 924 4. JmJobStateReasons3TC
- 925 5. JmJobStateReasons4TC
- These textual-conventions are defined as bits in an Integer so that they can be used with
- 927 SNMPv1 SMI. The **jobStateReasons**N (N=1..4) attributes are defined as bit values using
- 928 the corresponding **JmJobStateReasons***N***TC** textual-conventions.
- 929 The registration of **JmJobServiceTypesTC** and **JmJobStateReasonsNTC** bit values
- 930 SHALL follow the procedures for a type 2 enum as specified in Section 3.6.1.2.

931	3.6.3 IANA Registration of Job Submission Id Formats
932	In addition to enums and bit values, this specification assigns a single ASCII digit or letter
933	to various job submission ID formats. See the JmJobSubmissionIDTypeTC textual-
934	convention and the object. The registration of jmJobSubmissionID format numbers
935	SHALL follow the procedures for a type 2 enum as specified in Section 3.6.1.2.
936	3.6.4 IANA Registration of MIME types/sub-types for document-formats
937	The documentFormat(38) attribute has MIME type/sub-type values for indicating
938	document formats which IANA registers as "media type" names. The values of the
939	documentFormat(38) attribute are the same as the corresponding Internet Printing
940	Protocol (IPP) "document-format" Job attribute values [ipp-model].
941	3.7 Security Considerations
942	3.7.1 Read-Write objects
943	All objects are read-only, greatly simplifying the security considerations. If another MIB
944	augments this MIB, that MIB might accept SNMP Write operations to objects in that
945	MIB whose effect is to modify the values of read-only objects in this MIB. However, that
946	MIB SHALL have to support the required access control in order to achieve security, not
947	this MIB.
948	3.7.2 Read-Only Objects In Other User's Jobs
949	The security policy of some sites MAY be that unprivileged users can only get the objects
950	from jobs that they submitted, plus a few minimal objects from other jobs, such as the
951	jmJobKOctetsRequested and jmJobKOctetsProcessed objects, so that a user can tell
952	how busy a printer is. Other sites MAY allow all unprivileged users to see all objects of
953	all jobs. This MIB does not require, nor does it specify how, such restrictions would be
954	implemented. A monitoring application SHOULD enforce the site security policy with
955	respect to returning information to an unprivileged end user that is using the monitoring
956	application to monitor jobs that do not belong to that user, i.e., the jmJobOwner object
957	in the jmJobTable does not match the user's user name.
958	An operator is a privileged user that would be able to see all objects of all jobs,
959	independent of the policy for unprivileged users.
960	3.8 Notifications

This MIB does not specify any notifications. For simplicity, management applications are

expected to poll for status. The jmGeneralJobPersistence and

- 963 **jmGeneralAttributePersistence** objects assist an application to determine the polling
- rate. The resulting network traffic is not expected to be significant.

965 **4. MIB specification**

The following pages constitute the actual Job Monitoring MIB.

```
967
       Job-Monitoring-MIB DEFINITIONS ::= BEGIN
 968
 969
       IMPORTS
             MODULE-IDENTITY, OBJECT-TYPE, 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
 970
 971
       -- Use the experimental (54) OID assigned to the Printer MIB[print-mib]
 972
       -- before it was published as RFC 1759.
 973
       -- Upon publication of the Job Monitoring MIB as an RFC, delete this
 974
       -- comment and the line following this comment and change the
 975
       -- reference of { temp 105 } (below) to { mib-2 X }.
 976
       -- This will result in changing:
 977
       -- 1 3 6 1 3 54 jobmonMIB(105) to:
 978
       -- 1 3 6 1 2 1 jobmonMIB(X)
 979
       -- This will make it easier to translate prototypes to
 980
       -- the standard namespace because the lengths of the OIDs won't
 981
       -- change.
 982
       temp OBJECT IDENTIFIER ::= { experimental 54 }
 983
 984
       jobmonMIB MODULE-IDENTITY
 985
             LAST-UPDATED "9709190000Z"
 986
             ORGANIZATION "IETF Printer MIB Working Group"
 987
             CONTACT-INFO
 988
                  "Tom Hastings
 989
                  Postal: Xerox Corp.
 990
                       Mail stop ESAE-231
                       701 S. Aviation Blvd.
 991
                       El Segundo, CA 90245
 992
 993
 994
                        (301)333-6413
                  Tel:
 995
                         (301)333-5514
                  Fax:
 996
                  E-mail: hastings@cp10.es.xerox.com
 997
 998
                  Send comments to the printmib WG using the Job Monitoring
 999
                  Project (JMP) Mailing List: jmp@pwg.org
1000
1001
                  To learn how to subscribe to the JMP mailing list.
1002
                  send email to: jmp-request@pwg.org
1003
```

```
1004
                   For further information, access the PWG web page under 'JMP':
1005
                   http://www.pwg.org/"
             DESCRÍPTION
1006
1007
                   "The MIB module for monitoring job in servers, printers, and other devices."
1008
1009
                   File: draft-ietf-printmib-job-monitor-06.txt
                   Version: 0.86"
1010
             ::= \{ \text{ temp } 105 \}
1011
1012
1013
1014
1015
        -- Textual conventions for this MIB module
1016
1017
1018
1019
       JmUTF8StringTC ::= TEXTUAL-CONVENTION
1020
             DISPLAY-HINT "255a"
1021
             STATUS
                         current
1022
             DESCRIPTION
1023
                   "To facilitate internationalization, this TC represents information taken from the ISO/IEC IS
1024
                   10646-1 character set, encoded as an octet string using the UTF-8 character encoding scheme."
1025
             REFERENCE
                   "See section 3.5.1, entitled: 'Text generated by the server or device'."
1026
                          OCTET STRING (SIZE (0..63))
1027
             SYNTAX
1028
1029
1030
1031
1032
       JmJobStringTC ::= TEXTUAL-CONVENTION
1033
             STATUS
                         current
1034
             DESCRIPTION
                   "To facilitate internationalization, this TC represents information using any coded character set
1035
                   registered by IANA as specified in section 3.5.2. While it is recommended that the coded
1036
1037
                   character set be UTF-8 [UTF-8], the actual coded character set SHALL be indicated by the
                   value of the jobCodedCharSet(7) attribute for the job."
1038
1039
             REFERENCE
                   "See section 3.5.2, entitled: 'Text generated by the job submitter'."
1040
                          OCTET STRING (SIZE (0..63))
1041
             SYNTAX
1042
1043
1044
1045
        JmTimeStampTC ::= TEXTUAL-CONVENTION
1046
1047
             STATUS
                          current
1048
             DESCRIPTION
```

```
1049
                  "The simple time at which an event took place. The units SHALL be in seconds since the
1050
                  system was booted.
1051
1052
                  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
1053
1054
                  comply with implementing sysUpTime in MIB-II[mib-II].)
1055
1056
                  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-
1057
1058
                  convention defined in SMNPv2-TC is defined as an APPLICATION 3 IMPLICIT INTEGER
1059
                  tag, not an Integer32, so cannot be used in this MIB as one of the values of
                  jmAttributeValueAsInteger."
1060
                         INTEGER(0..2147483647)
1061
             SYNTAX
1062
1063
1064
1065
       JmJobSourcePlatformTypeTC ::= TEXTUAL-CONVENTION
1066
1067
             STATUS
                         current
1068
             DESCRIPTION
                  "The source platform type that can submit jobs to servers or devices in any of the 3
1069
                  configurations."
1070
1071
             REFERENCE
                  "This is a type 2 enumeration. See Section 3.6.1.2."
1072
1073
             SYNTAX
                         INTEGER {
                   other(1),
                   unknown(2),
                   sptUNIX(3),
                                                      UNIX
                   sptOS2(4),
                                                      OS/2
                                                      DOS
                   sptPCDOS(5),
                                                      NT
                   sptNT(6),
                                                      MVS
                   sptMVS(7),
                   sptVM(8),
                                                      VM
                                                      OS/400
                   sptOS400(9),
                                                      VMS
                   sptVMS(10),
                   sptWindows(11),
                                                      Windows
                   sptNetWare(12)
                                                      NetWare
1074
             }
1075
1076
1077
1078
1079
1080
       JmFinishingTC ::= TEXTUAL-CONVENTION
1081
             STATUS
                         current
1082
             DESCRIPTION
1083
                  "The type of finishing operation."
```

```
1084
1085
                    These values are the same as the enum values of the IPP 'finishings' attribute. See Section
1086
                    3.6.1.2.
1087
1088
                    other(1).
1089
                         Some other finishing operation besides one of the specified or registered values.
1090
1091
                    unknown(2).
1092
                          The finishing is unknown.
1093
1094
                    none(3).
1095
                         Perform no finishing.
1096
1097
                    staple(4),
1098
                          Bind the document(s) with one or more staples. The exact number and placement of the
1099
                          staples is site-defined.
1100
1101
                    punch(5),
1102
                          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 a
1103
                          site- and implementation-specific manner) either by drilling/punching, or by substituting
1104
1105
                          pre-drilled media.
1106
1107
                    cover(6).
1108
                          This value is specified when it is desired to select a non-printed (or pre-printed) cover for
1109
                          the document. This does not supplant the specification of a printed cover (on cover stock
1110
                          medium) by the document itself.
1111
1112
                    bind(7)
1113
                          This value indicates that a binding is to be applied to the document; the type and
                         placement of the binding is product-specific."
1114
1115
              REFERENCE
1116
                    "This is a type 2 enumeration. See Section 3.6.1.2."
1117
              SYNTAX
                           INTEGER {
1118
                    other(1).
1119
                    unknown(2).
1120
                    none(3),
1121
                    staple(4),
1122
                    punch(5),
1123
                    cover(6),
1124
                    bind(7)
1125
              }
1126
1127
1128
1129
1130
1131
        JmPrintQualityTC ::= TEXTUAL-CONVENTION
```

```
1132
             STATUS
                          current
1133
             DESCRIPTION
1134
                   "Print quality settings.
1135
1136
                   These values are the same as the enum values of the IPP 'print-quality' attribute. See Section
1137
                   3.6.1.2."
             REFERENCE
1138
1139
                   "This is a type 2 enumeration. See Section 3.6.1.2."
1140
             SYNTAX
                          INTEGER {
                    other(1),
                                           Not one of the specified or registered values.
                    unknown(2),
                                           The actual value is unknown.
                                           Lowest quality available on the printer.
                    draft(3),
                    normal(4),
                                           Normal or intermediate quality on the printer.
                    high(5)
                                           Highest quality available on the printer.
1141
             }
1142
1143
1144
1145
1146
        JmPrinterResolutionTC ::= TEXTUAL-CONVENTION
1147
             STATUS
                         current
             DESCRIPTION
1148
1149
                   "Printer resolutions.
1150
                   Nine octets consisting of two 4-octet SIGNED-INTEGERs followed by a SIGNED-BYTE.
1151
1152
                   The values are the same as those specified in the Printer MIB [printmib]. The first SIGNED-
                   INTEGER contains the value of prtMarkerAddressabilityXFeedDir. The second SIGNED-
1153
1154
                   INTEGER contains the value of prtMarkerAddressabilityFeedDir. The SIGNED-BYTE
1155
                   contains the value of prtMarkerAddressabilityUnit.
1156
1157
                   Note: the latter value is either 3 (tenThousandsOfInches) or 4 (micrometers) and the
1158
                   addressability is in 10,000 units of measure. Thus the SIGNED-INTEGERs represent integral
1159
                   values in either dots-per-inch or dots-per-centimeter.
1160
1161
                   The syntax is the same as the IPP 'printer-resolution' attribute. See Section 3.6.1.2."
1162
             SYNTAX
                          OCTET STRING (SIZE(9))
1163
1164
1165
1166
1167
        JmTonerEconomyTC ::= TEXTUAL-CONVENTION
1168
1169
             STATUS
                         current
1170
             DESCRIPTION
1171
                   "Toner economy settings."
```

```
1172
             REFERENCE
                  "This is a type 2 enumeration. See Section 3.6.1.2."
1173
                         INTEGER {
1174
             SYNTAX
                    unknown(2),
                                             unknown.
                    off(3),
                                             Off. Normal. Use full toner.
                    on(4)
                                             On. Use less toner than normal.
1175
             }
1176
1177
1178
1179
1180
1181
       JmBooleanTC ::= TEXTUAL-CONVENTION
1182
             STATUS
                       current
             DESCRIPTION
1183
1184
                  "Boolean true or false value."
1185
             REFERENCE
1186
                  "This is a type 2 enumeration. See Section 3.6.1.2."
1187
             SYNTAX
                         INTEGER {
                    unknown(2),
                                             unknown.
                                             FALSE.
                    false(3),
                    true(4)
                                             TRUE.
1188
             }
1189
1190
1191
1192
1193
1194
       JmMediumTypeTC ::= TEXTUAL-CONVENTION
1195
             STATUS
                        current
1196
             DESCRIPTION
1197
                  "Identifies the type of medium.
1198
1199
1200
                        The type is neither one of the values listed in this specification nor a registered value.
1201
1202
                  unknown(2),
1203
                        The type is not known.
1204
1205
                  stationery(3),
                        Separately cut sheets of an opaque material.
1206
1207
1208
                  transparency(4),
1209
                        Separately cut sheets of a transparent material.
1210
1211
                  envelope(5),
1212
                        Envelopes that can be used for conventional mailing purposes.
```

```
1213
1214
                   envelopePlain(6),
1215
                         Envelopes that are not preprinted and have no windows.
1216
                   envelopeWindow(7),
1217
1218
                         Envelopes that have windows for addressing purposes.
1219
1220
                   continuousLong(8),
1221
                         Continuously connected sheets of an opaque material connected along the long edge.
1222
1223
                   continuousShort(9),
1224
                         Continuously connected sheets of an opaque material connected along the short edge.
1225
1226
                   tabStock(10).
1227
                         Media with tabs.
1228
1229
                   multiPartForm(11),
1230
                         Form medium composed of multiple layers not pre-attached to one another; each sheet
1231
                         MAY be drawn separately from an input source.
1232
1233
                   labels(12),
1234
                         Label-stock.
1235
1236
                   multiLayer(13)
1237
                         Form medium composed of multiple layers which are pre-attached to one another, e.g. for
1238
                         use with impact printers."
1239
             REFERENCE
1240
                   "This is a type 2 enumeration. See Section 3.6.1.2."
1241
             SYNTAX
                          INTEGER {
1242
                   other(1).
1243
                   unknown(2),
1244
                   stationery(3),
1245
                   transparency(4),
1246
                   envelope(5),
1247
                   envelopePlain(6),
1248
                   envelopeWindow(7),
1249
                   continuousLong(8),
1250
                   continuousShort(9),
1251
                   tabStock(10),
1252
                   multiPartForm(11),
1253
                   labels(12),
1254
                   multiLayer(13)
1255
             }
1256
1257
1258
1259
1260
```

1261 1262	JmJobSubmissionTypeTC ::= TEXTUAL-CONVENTION STATUS current
1263	DESCRIPTION
1264	"Identifies the format type of a job submission ID.
	identifies the format type of a job submission ib.
1265	
1266	Each job submission ID is a fixed-length, 48-octet printable US-ASCII [US-ASCII] coded
1267	character string containing no control characters, consisting of the following fields:
1268	
1269	octet 1 The format letter identifying the format.
1270	The US-ASCII characters '0-9', 'A-Z', and 'a-z'
1271	are assigned in order giving 62 possible
1272	formats.
1273	octets 2-40 A 39-character, US-ASCII trailing SPACE filled
1274	field specified by the format letter, if the
1275	data is less than 39 ASCII characters.
1276	octets 41-48 A sequential or random number to make the ID
1277	quasi-unique.
1278	quali anque.
1279	If the client does not supply a job submission ID in the job submission protocol, then the agent
1280	SHALL assign a job submission ID using any of the standard formats that are reserved for the
1281	agent. Clients SHALL not use formats that are reserved for agents and agents SHALL NOT
1281	
	use formats that are reserved for clients, in order to reduce conflicts in ID generation. See the
1283	description for which formats are reserved for clients or for agents.
1284	Designation of a 1414 and formate many hards of 11 and 41
1285	Registration of additional formats may be done following the procedures described in Section
1286	3.6.3.
1287	
1288	The format values defined at the time of completion of this specification are:
1289	T
1290	Format
1291	Letter Description
1292	
1293	'0' octets 2-40: last 39 bytes of the jmJobOwner
1294	object.
1295	octets 41-48: 8-decimal-digit sequential number.
1296	This format is reserved for agents.
1297	
1298	NOTE - Clients wishing to use a job submission ID
1299	that incorporates the job owner, SHALL use format
1300	'8', not format '0'.
1301	
1302	'1' octets 2-40: last 39 bytes of the jobName attribute.
1303	octets 41-48: 8-decimal-digit random number.
1304	This format is reserved for clients.
1305	
1306	'2' octets 2-40: Client MAC address: in hexadecimal
1307	with each nibble of the 6 octet address being
1308	'0'-'9' or 'A' - 'F' (uppercase only).
1309	Most significant octet first.

1310		octets 41-48: 8-decimal-digit sequential number
1311		This format is reserved for clients.
		This format is reserved for chems.
1312		
1313	'3'	octets 2-40: last 39 bytes of the client URL
1314		[URI-spec].
1315		octets 41-48: 8-decimal-digit sequential number
1316		This format is reserved for clients.
1317		
1318	'4'	octets 2-40: last 39 bytes of the URI [URI-spec]
	4	
1319		assigned by the server or device to the job when
1320		the job was submitted for processing.
1321		octets 41-48: 8-decimal-digit sequential number
1322		This format is reserved for agents.
1323		
1324	'5'	octets 2-40: last 39 bytes of a user number, such
1325		as POSIX user number.
1326		octets 41-48: 8-decimal-digit sequential number
1327		This format is reserved for clients.
1328		
1329	'6'	cotate 2.40 lest 20 bytes of the user eccount
	U	octets 2-40: last 39 bytes of the user account
1330		number.
1331		octets 41-48: 8-decimal-digit sequential number
1332		This format is reserved for clients.
		This format is reserved for chems.
1333		
1334	'7 '	octets 2-40: last 39 bytes of the DTMF incoming
1335		FAX routing number.
1336		octets 41-48: 8-decimal-digit sequential number
1337		This format is reserved for clients.
1338		
1339	'8 '	octets 2-40: last 39 bytes of the job owner name
1340		(that the agent returns in the jmJobOwner object).
1341		octets 41-48: 8-decimal-digit sequential number
1342		This format is reserved for clients.
1343		
1344	'9'	octets 2-40: last 39 bytes of the host name with
1345		trailing SPACES that submitted the job to this
1346		server/device using a protocol, such as LPD
1347		[RFC-1179] which includes the host name in the job
1348		submission protocol.
1349		octets 41-48: 8-decimal-digit leading zero
1350		
		representation of the job id generated by the
1351		by the submitting server (configuration 3)
1352		or the client (configuration 1 and 2), such as in
1353		the LPD protocol.
1354		This format is reserved for clients.
1355		
1356	NOTE	E - the job submission id is only intended to be unique between a limited set of clients for a
1357		d duration of time, namely, for the life time of the job in the context of the server or device
1358		s processing the job. Some of the formats include something that is unique per client and a
1330	11at 18	processing the job. Some of the formats menute something that is unique per chefit and a

random number so that the same job submitted by the same client will have a different job submission id. For other formats, where part of the id is guaranteed to be unique for each client, such as the MAC address or URL, a sequential number SHOULD suffice for each client (and may be easier for each client to manage). Therefore, the length of the job submission id has been selected to reduce the probability of collision to an extremely low number, but is not intended to be an absolute guarantee of uniqueness. None-the-less, collisions are remotely possible, but without bad consequences, since this MIB is intended to be used only for monitoring jobs, not for controlling and managing them."

REFERENCE

"This is like a type 2 enumeration. See section 3.6.3." SYNTAX OCTET STRING(SIZE(1)) -- ASCII '0'-'9', 'A'-'Z', 'a'-'z'

1371 1372 1373

1376

1377

1378

1379 1380

1390

1391 1392

1393

1394

1395 1396

1397 1398

1399

1400 1401

1402

1403 1404

1405

1359

1360

1361 1362

1363 1364

1365 1366

1367

1368 1369

1370

```
1374
1375
      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:

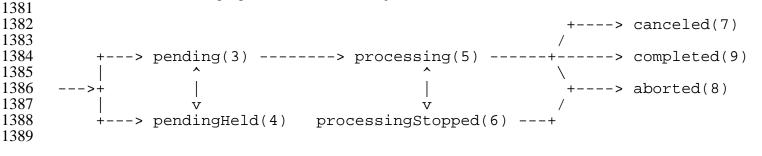


Figure 4 - Normal Job State Transitions

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 job.

These values are the same as the enum values of the IPP 'job-state' job attribute. See Section 3.6.1.2.

unknown(2).

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

1406
1407
1408
1409
1410
1411 1412 1413 1414
1412
1413
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1421 1422 1423 1424
1424
1425
1426
1427
1427
1428
1429
1430 1431 1432
1431
1432
1433
1432 1433 1434 1435 1436
1435
1436
1437
1437 1438
1439
1440
1441
1442
1443
1443
1444
1446
1447
1448
1449
1450
1451
1452

1453

1454

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.

processing(5),

One 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* (*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.

1455 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 1456 1457 complete device status remotely by querying the appropriate device MIB using the job's 1458 **deviceIndex** attribute(s), if the agent implements such a device MIB 1459 1460 canceled(7), A client has canceled the job and the server or device has completed canceling the job and 1461 1462 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 1463 1464 the processing ToStopPoint value and one of the canceled ByUser, 1465 canceledByOperator, or canceledAtDevice values. The canceledByUser, canceledByOperator, or canceledAtDevice values remain while the job is in the 1466 1467 canceled state. 1468 1469 aborted(8), 1470 The job has been aborted by the system, usually while the job was in the **processing** or 1471 **processingStopped** state and the server or device has completed aborting the job and all 1472 MIB objects and attributes have reached their final values for the job. While the server or 1473 device is aborting the job, the job's **jmJobStateReasons1** object MAY contain the processing ToStopPoint and aborted BySystem values. If implemented, the 1474 1475 **abortedBySystem** value SHALL remain while the job is in the **aborted** state. 1476 1477 completed(9) 1478 The job has completed successfully or with warnings or errors after processing and all of 1479 the media have been successfully stacked in the appropriate output bin(s). The job's 1480 jmJobStateReasons1 object SHOULD contain one of: completedSuccessfully, completedWithWarnings, or completedWithErrors values." 1481 1482 REFERENCE 1483 "This is a type 2 enumeration. See Section 3.6.1.2." 1484 SYNTAX INTEGER { 1485 unknown(2). 1486 pending(3), 1487 pendingHeld(4), 1488 processing(5), 1489 processingStopped(6), 1490 canceled(7). 1491 aborted(8), completed(9) 1492 1493 } 1494 1495 1496 **JmAttributeTypeTC** ::= TEXTUAL-CONVENTION 1497 STATUS current 1498 **DESCRIPTION**

"The type of the attribute which identifies the attribute."

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

1499

1500 1501

1502

1503	jmAttributeValueAsOctets objects by the initial	al tag: 'INTEGER:' or 'OCTETS:',
1504	respectively.	
1505 1506	Some attributes allow the agent implementer a al	hoice of useful values of either an integer an
1507	Some attributes allow the agent implementer a cl octets representation, or both, depending on imp	
1507	'INTEGER:' AND/OR 'OCTETS:' tags.	mementation. These attributes are mulcated with
1508	INTEGER. AND/OR OCIETS, tags.	
1510	A very few attributes require both objects at the	same time to represent a pair of useful values
1511	(see mediumConsumed(171)). These attributes	
1512	'OCTETS:' tags. See the jmAttributeGroup for	
1513	objects.	of the descriptions of these two many 2711 of the
1514	oojeets.	
1515	NOTE - The enum assignments are grouped logi	ically with values assigned in groups of 20, so
1516	that additional values may be registered in the fu	
1517	logical grouping.	waze and assigned a value may is part of along
1518	888-	
1519	Values in the range 2**30 to 2**31-1 are reserve	ed for private or experimental usage. This
1520	range corresponds to the same range reserved in	
1521	values may conflict with other implementations.	
1522	registration of enum values following the proced	
1523		
1524	NOTE: No attribute name exceeds 31 characters	8.
1525		
1526	The standard attribute types defined at the time of	of completion of the specification are:
1527		1
1528	jmAttributeTypeIndex	Datatype
1528 1529	jmAttributeTypeIndex 	
1528 1529 1530	<u></u>	Datatype
1528 1529 1530 1531	jmAttributeTypeIndex other(1),	Datatype
1528 1529 1530 1531 1532	<u></u>	Datatype Integer32(-22147483647) AND/OR
1528 1529 1530 1531 1532 1533	other(1),	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063))
1528 1529 1530 1531 1532 1533 1534	other(1), INTEGER: and/or OCTETS: An attribute	Datatype Integer32(-22147483647) AND/OR
1528 1529 1530 1531 1532 1533 1534 1535	other(1),	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063))
1528 1529 1530 1531 1532 1533 1534 1535 1536	other(1), INTEGER: and/or OCTETS: An attribute	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063))
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537	other(1), INTEGER: and/or OCTETS: An attribut approved and registered with IANA.	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538	other(1), INTEGER: and/or OCTETS: An attribut approved and registered with IANA.	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Datatype Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been **** Ta job. **** **JmJobStateReasons2TC **the job's current state that augments the nder the JmJobStateReasons1TC textual-
1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547	other(1), INTEGER: and/or OCTETS: An attribute approved and registered with IANA. ++++++++++++++++++++++++++++++++++	Integer32(-22147483647) AND/OR OCTET STRING(SIZE(063)) te that is not in the list and/or that has not been ++++++++++++++++++++++++++++++++++

1551 1552	jmJobState object. See the description under JmJobStateReasons1TC textual-convention.
1553 1554	inhStateDeagang4(5) Im InhStateDeagang4TC
1555	jobStateReasons4(5), JmJobStateReasons4TC
1556	INTEGER: Additional information about the job's current state that augments the jmJobState object. See the description under JmJobStateReasons1TC textual-
1557	convention.
1558	convention.
1559	processing Massaga (6) Im ITEQC tring TC (CITE (0. 62))
1560	processingMessage(6), JmUTF8StringTC(SIZE(063)) OCTETS: MULTI-ROW: A coded character set message that is generated by the server
1561	or device during the processing of the job as a simple form of processing log to show
1562	progress and any problems.
1563	progress and any problems.
1564	There is no restriction for the same massage occurring in multiple rows
1565	There is no restriction for the same message occurring in multiple rows.
1566	jobCodedCharSet(7), CodedCharSet
1567	jobCodedCharSet(7), CodedCharSet INTEGER: The MIBenum identifier of the coded character set that the agent is using to
1568	represent coded character set objects and attributes of type 'JmJobStringTC'. These
1569	coded character set objects and attributes are either: (1) supplied by the job submitting
1570	client or (2) defaulted by the server or device when omitted by the job submitting client.
1570	The agent SHALL represent these objects and attributes in the MIB either (1) in the coded
1571	character set as they were submitted or (2) MAY convert the coded character set to
1573	another coded character set or encoding scheme as identified by the jobCodedCharSet
1574	attribute.
1574	aunouic.
1576	Those MIP anum values are assigned by IANA [IANA shoreats] when the goded character
1577	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
1578	IANA [IANA] and the enum value uses the CodedCharSet textual-convention from the
1579	Printer MIB. See the JmJobStringTC textual-convention.
1580	Time wid. See the smjobstring i C textual-convention.
1581	If the agent does not know what coded character set was used by the job submitting client,
1582	the agent SHALL either (1) return the 'unknown(2)' value for the jobCodedCharSet
1583	attribute or (2) not return the jobCodedCharSet attribute for the job. See Section 3.5.2,
1584	entitled 'Text generated by the job submitter'.
1585	chitica Text generated by the job submitter.
1586	
1587	
1588	+++++++++++++++++++++++++++++++++++++++
1589	+ Job Identification attributes
1590	+
1591	+ The following attributes help an end user, a system
1592	+ operator, or an accounting program identify a job.
1593	++++++++++++++++++++++++++++++++++++++
1594	
1595	
1596	
1597	jobURI(20), OCTET STRING(SIZE(1255))
1598	OCTETS: The job's Universal Resource Identifier (URI) [RFC-1738]. See IPP for
1599	example usage.

1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1622
1622
1624
1625
1636
1637
1638
1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1630 1631 1632 1633 1634 1635 1638
1640
16/11
1642
1642 1643
1644
1645
1646
-

1647

1648

NOTE - The agent may be able to generate this value on each SNMP Get operation from smaller values, rather than having to store the entire URI.

If the URI exceeds 255 octets, the agent SHALL truncate from the beginning (since the end tends to be more unique than the beginning).

jobAccountName(21),

OCTET STRING(SIZE(0..63))

OCTETS: Arbitrary binary information which MAY be coded character set data or encrypted data supplied by the submitting user for use by accounting services to allocate or categorize charges for services provided, such as a customer account name or number.

NOTE: This attribute NEED NOT be printable characters.

serverAssignedJobName(22),

JmJobStringTC(SIZE(0..63))

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(0..63))

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 **jmJobSubmissionID**, the agent SHOULD maintain the **jobName** attribute for the time specified by the **jmGeneralJobPersistence** object, rather than the (shorter) **jmGeneralAttributePersistence** object.

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.

1649	jobServiceTypes(24), JmJobServiceTypesTC
1650	INTEGER: Specifies the type(s) of service to which the job has been submitted (print,
1651	fax, scan, etc.). The service type is bit encoded with each job service type so that more
1652	general and arbitrary services can be created, such as services with more than one
1653	
	destination type, or ones with only a source or only a destination. For example, a job
1654	service might scan, faxOut, and print a single job. In this case, three bits would be set in
1655	the jobServiceTypes attribute, corresponding to the hexadecimal values: 0x8 + 0x20 +
1656	0x4 , respectively, yielding: 0x2C .
1657	
1658	Whether this attribute is set from a job attribute supplied by the job submission client or is
1659	set by the recipient job submission server or device depends on the job submission
1660	protocol. This attribute SHALL be implemented if the server or device has other types in
1661	addition to or instead of printing.
1662	
1663	One of the purposes of this attribute is to permit a requester to filter out jobs that are not
1664	of interest. For example, a printer operator may only be interested in jobs that include
1665	printing.
1666	Γ -
1667	jobSourceChannelIndex(25), Integer32(02147483647)
1668	INTEGER: The index of the row in the associated Printer MIB[print-mib] of the channel
1669	which is the source of the print job.
1670	which is the source of the print job.
1671	jobSourcePlatformType(26), JmJobSourcePlatformTypeTC
1672	
	INTEGER: The source platform type of the immediate upstream submitter that submitted
1673	the job to the server (configuration 2) or device (configuration 1 and 3) to which the agent
1674	is providing access. For configuration 1, this is the type of the client that submitted the
1675	job to the device; for configuration 2, this is the type of the client that submitted the job
1676	to the server; and for configuration 3, this is the type of the server that submitted the job
1677	to the device.
1678	
1679	submittingServerName(27), JmJobStringTC(SIZE(063))
1680	OCTETS: For configuration 3 only: The administrative name of the server that submitted
1681	the job to the device.
1682	
1683	submittingApplicationName(28), JmJobStringTC(SIZE(063))
1684	OCTETS: The name of the client application (not the server in configuration 3) that
1685	submitted the job to the server or device.
1686	3
1687	jobOriginatingHost(29), JmJobStringTC(SIZE(063))
1688	OCTETS: The name of the client host (not the server host name in configuration 3) that
1689	submitted the job to the server or device.
1690	submitted the job to the server of device.
1691	deviceNameRequested(30), JmJobStringTC(SIZE(063))
1692	OCTETS: The administratively defined coded character set name of the target device
1693	
	requested by the submitting user. For configuration 1, its value corresponds to the Printer
1694 1605	MIB[print-mib]: prtGeneralPrinterName object. For configuration 2 and 3, its value is
1695	the name of the logical or physical device that the user supplied to indicate to the server
1696	on which device(s) they wanted the job to be processed.
1697	

4.400	
1698	queueNameRequested(31), JmJobStringTC(SIZE(063))
1699	OCTETS: The administratively defined coded character set name of the target queue
1700	requested by the submitting user. For configuration 1, its value corresponds to the queue
1701	in the device for which the agent is providing access. For configuration 2 and 3, its value
1702	is the name of the queue that the user supplied to indicate to the server on which device(s)
1703	they wanted the job to be processed.
1704	
1705	NOTE - typically an implementation SHOULD support either the deviceNameRequested
1706	or queueNameRequested attribute, but not both.
1707	
1708	physicalDevice(32), hrDeviceIndex
1709	AND/OR
1710	JmUTF8StringTC(SIZE(063))
1711	INTEGER: MULTI-ROW: The index of the physical device MIB instance
1712	requested/used, such as the Printer MIB[print-mib]. This value is an hrDeviceIndex
1713	value. See the Host Resources MIB[hr-mib].
1714	variation see the frest restources missing in missing.
1715	AND/OR
1716	
1717	OCTETS: MULTI-ROW: The name of the physical device to which the job is assigned.
1718	OCILIS. WOLII-ROW. The hame of the physical device to which the job is assigned.
1719	numberOfDocuments(33), Integer32(-22147483647)
1720	INTEGER: The number of documents in this job.
1721	INTEGER. The number of documents in this job.
1722	fileNeme(24) Im InhStringTC(SI7E(0, 62))
	fileName(34), JmJobStringTC(SIZE(063))
1723	OCTETS: MULTI-ROW: The coded character set file name or URI[URI-spec] of the
1724	document.
1725	
1726	There is no restriction on the same file name occurring in multiple rows.
1727	
1728	documentName(35), JmJobStringTC(SIZE(063))
1729	OCTETS: MULTI-ROW: The coded character set name of the document.
1730	
1731	There is no restriction on the same document name occurring in multiple rows.
1732	
1733	jobComment(36), JmJobStringTC(SIZE(063))
1734	OCTETS: An arbitrary human-readable coded character text string supplied by the
1735	submitting user or the job submitting application program for any purpose. For example,
1736	a user might indicate what he/she is going to do with the printed output or the job
1737	submitting application program might indicate how the document was produced.
1738	
1739	The jobComment attribute is not intended to be a name; see the jobName attribute.
1740	
1741	documentFormatIndex(37), Integer32(02147483647)
1742	INTEGER: MULTI-ROW: The index in the prtInterpreterTable in the Printer
1743	MIB[print-mib] of the page description language (PDL) or control language interpreter
1744	that this job requires/uses. A document or a job MAY use more than one PDL or control
1745	language.
1746	

1747 NOTE - As with all intensive attributes where multiple rows are allowed, there SHALL be 1748 only one distinct row for each distinct interpreter; there SHALL be no duplicates. 1749 1750 NOTE - This attribute type is intended to be used with an agent that implements the Printer MIB and SHALL not be used if the agent does not implement the Printer MIB. 1751 1752 Such an agent SHALL use the **documentFormat** attribute instead. 1753 1754 documentFormat(38), 1755 AND/OR 1756 1757 1758 MIB[print-mib] **prtInterpreterLangFamily** object, that this job requires/uses. A 1759 document or a job MAY use more than one PDL or control language. 1760 AND/OR 1761 1762 1763 1764 types], i.e., the name of the MIME content-type/subtype. Examples: 1765 1766 1767 1768 1769 + Job Parameter attributes 1770 1771 + The following attributes represent input parameters + supplied by the submitting client in the job submission 1772 1773 + protocol. 1774 1775 1776 Integer32(1..100) iobPriority(50). 1777 1778 employ a priority-based scheduling algorithm. 1779 1780 1781 1782 1783 1784 1785 jobProcessAfterDateAndTime(51), 1786 OCTETS: The calendar date and time of day after which the job SHALL become a 1787 1788 1789

PrtInterpreterLangFamilyTC OCTET STRING(SIZE(0..63)) INTEGER: MULTI-ROW: The interpreter language family corresponding to the Printer

OCTETS: MULTI-ROW: The document format registered as a media type[iana-media-'application/postscript', 'application/vnd.hp-PCL', and 'application/pdf'

INTEGER: The priority for scheduling the job. It is used by servers and devices that

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.

DateAndTime (SNMPv2-TC)

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 **imJobState** 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 **imJobStateReasons1** object and, if no other reasons remain, SHALL change the job's **jmJobState** object to **pending**.

1790

1791

1792

1793

1794

1795	jobHold(52), JmBooleanTC
1796	INTEGER: If the value is 'true(4)', a client has explicitly specified that the job is to be
1797	held until explicitly released. Until the job is explicitly released by a client, the job SHALI
1798	be in the pendingHeld state with the jobHoldSpecified value in the
1799	jmJobStateReasons1 attribute.
1800	Jino one tute i comparation i
1801	jobHoldUntil(53), JmJobStringTC(SIZE(063))
1802	OCTETS: The named time period during which the job SHALL become a candidate for
1803	processing, such as 'evening', 'night', 'weekend', 'second-shift', 'third-shift', etc., as
1804	defined by the system administrator. See IPP [ipp-model] for the standard keyword
1805	values. Until that time period arrives, the job SHALL be in the pendingHeld state with
1806	the jobHoldUntilSpecified value in the jmJobStateReasons1 object. The value ' no-
1807	hold' SHALL indicate explicitly that no time period has been specified; the absence of this
1808	attribute SHALL indicate implicitly that no time period has been specified.
1809	autibute STALL indicate implicitly that no time period has been specified.
1810	outputBin(54), Integer32(02147483647)
1811	AND/OR
1812	JmJobStringTC(SIZE(063))
1813	INTEGER: MULTI-ROW: The output subunit index in the Printer MIB[print-mib]
1814	INTEGER. MOETI-ROW. The output subunit index in the Finite Mid[print-init]
1815	AND/OR
1816	AND/OR
1817	OCTETS: the name or number (represented as ASCII digita) of the output him to which
1818	OCTETS: the name or number (represented as ASCII digits) of the output bin to which all or part of the job is placed in.
1819	an or part of the job is praced in.
1820	sides(55) Integer 22(2 2)
1821	sides(55), Integer32(-22) INTEGER: MULTI-ROW: The number of sides, '1' or '2', that any document in this job
1822	
1823	requires/used.
	finishing(56)
1824	finishing(56), JmFinishingTC
1825	INTEGER: MULTI-ROW: Type of finishing that any document in this job requires/used
1826	
1827	
1828	++++++++++++++++++++++++++++++++++++++
1829	+ Image Quality attributes (requested and consumed)
1830	† L'Eau deriese that one recur the image quality
1831	+ For devices that can vary the image quality.
1832	+++++++++++++++++++++++++++++++++++++++
1833	maint Overlitan Decorate of (70) In Decorate of (70)
1834	printQualityRequested(70), JmPrintQualityTC
1835	INTEGER: MULTI-ROW: The print quality selection requested for a document in the
1836	job for printers that allow quality differentiation.
1837	
1838	printQualityUsed(71), JmPrintQualityTC
1839	INTEGER: MULTI-ROW: The print quality selection actually used by a document in the
1840	job for printers that allow quality differentiation.
1841	

10.40	
1842	printerResolutionRequested(72), JmPrinterResolutionTC
1843	OCTETS: MULTI-ROW: The printer resolution requested for a document in the job for
1844	printers that support resolution selection.
1845	
1846	printerResolutionUsed(73), JmPrinterResolutionTC
1847	OCTETS: MULTI-ROW: The printer resolution actually used by a document in the job
1848	for printers that support resolution selection.
1849	r
1850	tonerEcomonyRequested(74), JmTonerEconomyTC
1851	INTEGER: MULTI-ROW: The toner economy selection requested for documents in the
1852	job for printers that allow toner economy differentiation.
	job for printers that anow toner economy unferentiation.
1853	
1854	tonerEcomonyUsed(75), JmTonerEconomyTC
1855	INTEGER: MULTI-ROW: The toner economy selection actually used by documents in
1856	the job for printers that allow toner economy differentiation.
1857	
1858	tonerDensityRequested(76), Integer32(-2100)
1859	INTEGER: MULTI-ROW: The toner density requested for a document in this job for
1860	devices that can vary toner density levels. Level 1 is the lowest density and level 100 is
1861	the highest density level. Devices with a smaller range, SHALL map the 1-100 range
1862	evenly onto the implemented range.
	evenily onto the implemented range.
1863	1.422(2.100)
1864	tonerDensityUsed(77), Integer32(-2100)
1865	INTEGER: MULTI-ROW: The toner density used by documents in this job for devices
1866	that can vary toner density levels. Level 1 is the lowest density and level 100 is the highest
1867	density level. Devices with a smaller range, SHALL map the 1-100 range evenly onto the
1868	implemented range.
1869	
1870	
1871	+++++++++++++++++++++++++++++++++++++++
1872	+ Job Progress attributes (requested and consumed)
1873	+
1874	+ Pairs of these attributes can be used by monitoring
1875	+ applications to show an indication of relative progress
1876	+ to users.
1877	+++++++++++++++++++++++++++++++++++++++
1878	
1879	jobCopiesRequested(90), Integer32(-22147483647)
1880	INTEGER: The number of copies of the entire job that are to be produced.
1881	
1882	jobCopiesCompleted(91), Integer32(-22147483647)
1883	INTEGER: The number of copies of the entire job that have been completed so far.
1884	1.12021. The hamoer of copies of the chine job that have been completed so far.
1885	documentCopiesRequested(92), Integer32(-22147483647)
1886	INTEGER: The total count of the number of document copies requested for the job as a
1887	whole. If there are documents A, B, and C, and document B is specified to produce 4
1888	copies, the number of document copies requested is 6 for the job.
1889	

1890	This attribute SHALL be used only when a job has multiple documents. The
1891	jobCopiesRequested attribute SHALL be used when the job has only one document.
1892	7 (A A A A A A A A A A A A A A A A A A A
1893	documentCopiesCompleted(93), Integer32(-22147483647)
1894	INTEGER: The total count of the number of document copies completed so far for the
1895	job as a whole. If there are documents A, B, and C, and document B is specified to
1896	produce 4 copies, the number of document copies starts a 0 and runs up to 6 for the job as
1897	the job processes.
1898	
1899	This attribute SHALL be used only when a job has multiple documents. The
1900	jobCopiesCompleted attribute SHALL be used when the job has only one document.
1901	
1902	jobKOctetsTransferred(94), Integer32(-22147483647)
1903	INTEGER: The number of K (1024) octets transferred to the server or device to which
1904	the agent is providing access. This count is independent of the number of copies of the
1905	job or documents that will be produced, but it is only a measure of the number of bytes
1906	transferred to the server or device.
1907	
1908	The agent SHALL round the actual number of octets transferred up to the next higher K.
1909	Thus 0 octets SHALL be represented as ' 0 ', 1-1024 octets SHALL BE represented as ' 1 ',
1910	1025-2048 SHALL be '2', etc. When the job completes, the values of the
1911	jmJobKOctetsRequested object and the jobKOctetsTransferred attribute SHALL be
1912	equal.
1913	oqua:
1914	NOTE - The jobKOctetsTransferred can be used with the jmJobKOctetsRequested
1915	object in order to produce a relative indication of the progress of the job for agents that do
1916	not implement the jmJobKOctetsProcessed object.
1917	not implement the jing object.
1918	
1919	+++++++++++++++++++++++++++++++++++++++
1920	+ Impression attributes
1921	1 Impression attributes
1922	+ For a print job, an impression is the marking of the
1923	+ entire side of a sheet. Two-sided processing involves two
1923	+ impressions per sheet. Two-up is the placement of two
1925	+ logical pages on one side of a sheet and so is still a
1925	+ single impression. See also jmJobImpressionsRequested and
1920	+ single impression. See also jinfolding ressions Requested and + jmJobImpressionsCompleted objects in the jmJobTable.
1927	
1928	+++++++++++++++++++++++++++++++++++++++
	impressions Charled (110) Integra 22(2.2147492447)
1930	impressionsSpooled(110), Integer32(-22147483647)
1931	INTEGER: The number of impressions spooled to the server or device for the job so far.
1932	immussionsContToDories(111) Integra 22(2.2147492647)
1933	impressionsSentToDevice(111), Integer32(-22147483647)
1934	INTEGER: The number of impressions sent to the device for the job so far.
1935	
1936	impressionsInterpreted(112), Integer32(-22147483647)
1937	INTEGER: The number of impressions interpreted for the job so far.
1938	

1939	impressionsCompletedCurrentCopy(113), Integer32(-22147483647)
1940	INTEGER: The number of impressions completed by the device for the current copy of
1941	the current document so far. For printing, the impressions completed includes
1942	interpreting, marking, and stacking the output. For other types of job services, the
1943	number of impressions completed includes the number of impressions processed.
1944	number of impressions compresses includes and number of impressions processes.
1945	This value SHALL be reset to 0 for each document in the job and for each document
1946	copy.
1947	copy.
1948	fullColorImpressionsCompleted(114), Integer32(-22147483647)
1949	INTEGER: The number of full color impressions completed by the device for this job so
1950	far. For printing, the impressions completed includes interpreting, marking, and stacking
1951	the output. For other types of job services, the number of impressions completed includes
1952	the number of impressions processed. Full color impressions are typically defined as those
1952	
1953	requiring 3 or more colorants, but this MAY vary by implementation.
	hi-hi-h4C-lI
1955	highlightColorImpressionsCompleted(115), Integer32(-2
1956	2147483647)
1957	INTEGER: The number of highlight color impressions completed by the device for this
1958	job so far. For printing, the impressions completed includes interpreting, marking, and
1959	stacking the output. For other types of job services, the number of impressions completed
1960	includes the number of impressions processed. Highlight color impressions are typically
1961	defined as those requiring black plus one other colorant, but this MAY vary by
1962	implementation.
1963	
1964	
1965	+++++++++++++++++++++++++++++++++++++++
1966	+ Page attributes
1967	+
1968	+ A page is a logical page. Number up can impose more than
1969	+ one page on a single side of a sheet. Two-up is the
1970	+ placement of two logical pages on one side of a sheet so
1971	+ that each side counts as two pages.
1972	++++++++++++++++++++++++++++++++++++++
1973	
1974	pagesRequested(130), Integer32(-22147483647)
1975	INTEGER: The number of logical pages requested by the job to be processed.
1976	S and I also I a
1977	pagesCompleted(131), Integer32(-22147483647)
1978	INTEGER: The number of logical pages completed for this job so far.
1979	11 12 021. The number of logicum puges completed for this job so full
1980	For implementations where multiple copies are produced by the interpreter with only a
1981	single pass over the data, the final value SHALL be equal to the value of the
1982	pagesRequested object. For implementations where multiple copies are produced by the
1983	interpreter by processing the data for each copy, the final value SHALL be a multiple of
1984	the value of the pagesRequested object.
1985	the value of the pagestrequested object.
1703	

1986 1987	NOTE - See the impressionsCon	
1988	pagesCompletedCurrentCopy attributes for attributes that are reset on each document copy.	
1989	сору.	
1990	NOTE - The nages Completed of	ject can be used with the pagesRequested object to
1991		re progress of the job, provided that the multiplicative
1992		ne implementations of multiple copies.
1993	factor is taken into account for son	ne implementations of multiple copies.
1993 1994	pagesCompletedCurrentCopy(132),	Integer32(-22147483647)
1995		l pages completed for the current copy of the document
1996	co far This value SHALL be rese	et to 0 for each document in the job and for each
1997	document copy.	to v for each document in the job and for each
1998	document copy.	
1999		
2000		-++++++++++++++++++++++++++++++++++++++
2000	+ Sheet attributes	
2002	+ Sheet attributes +	
2002	+ The sheet is a single piece of a medi	um whathar printing
2003	+ on one or both sides.	um, whether printing
2005		+++++++++++++++++++++++++++++++++++++++
2005	***************************************	
2007	sheetsRequested(150),	Integer32(-22147483647)
2007		im sheets requested to be processed for this job.
2008 2009	INTEGER. The number of medic	in sheets requested to be processed for this job.
2010	sheetsCompleted(151),	Integer32(-22147483647)
2010		m sheets that have completed marking and stacking for
2012		sheets have been processed on one side or on both.
2012	the entire job so far whether those	sheets have been processed on one side of on both.
2013	sheetsCompletedCurrentCopy(152),	Integer32(-22147483647)
2015		m sheets that have completed marking and stacking for
2016		the job so far whether those sheets have been processed
2017	on one side or on both.	the job so far whether those sheets have been processed
2018	on one side of on both.	
2019	The value of this attribute SHALI	be reset to 0 as each document in the job starts being
2020	processed and for each document	
2021	processed and for each document	copy as it starts being processed.
2022		
2023	+++++++++++++++++++++++++++++++++++++++	-+++++++++++++++++++++++++
2024	+ Resources attributes (requested and	
2025	+	consumeu)
2026	+ Pairs of these attributes can be used	l hy monitoring
2027	+ applications to show an indication of	
2028	+ users.	i relative usage to
2029		-++++++++
2030		
2031	mediumRequested(170),	JmMediumTypeTC
2032		AND/OR
2033		JmJobStringTC(SIZE(063))
2034	INTEGER: MULTI-ROW: The	

2035	AND/OR	
2036	OCTETS: the name of the medium the	nat is required by the job.
2037		
2038	mediumConsumed(171),	Integer32(-22147483647)
2039		AND
2040		JmJobStringTC(SIZE(063))
2041	INTEGER: The number of sheets	
2042	AND	
2043	OCTETS: MULTI-ROW: the name of	of the medium that has been consumed so far
2044	whether those sheets have been proce	
2045	P	
2046	This attribute SHALL have both Inte	ger32 and OCTET STRING (represented as
2047	JmJobStringTC) values.	gorda and o o i b i b i i i to (represented as
2048	onioobstring i e) values.	
2049	colorantRequested(172),	Integer32(-22147483647)
2050	colorantikequesteu(172),	AND/OR
2051	INTEGED MILITIDOM TI ' 1	JmJobStringTC(SIZE(063))
2052		ex (prtMarkerColorantIndex) in the Printer
2053	MIB[print-mib]	
2054	AND/OR	
2055	OCTETS: the name of the colorant re	equested.
2056		
2057	colorantConsumed(173),	Integer32(-22147483647)
2058		AND/OR
2059		JmJobStringTC(SIZE(063))
2060	INTEGER: MULTI-ROW: The inde	ex (prtMarkerColorantIndex) in the Printer
2061	MIB[print-mib]	•
2062	AND/OR	
2063	OCTETS: the name of the colorant co	onsumed.
2064		
2065		
2066	+++++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++
2067	+ Time attributes (set by server or device	
2068	+	
2069	+ This section of attributes are ones that	are set by the
2070		
	+ server or device that accepts jobs. Two	
2071	+ provided. Each form is represented in	
2072	+ See section 3.1.2 and section 3.1.3 for the	
2073	+ conformance requirements for time att	
2074	+ monitoring applications, respectively.	The two forms are:
2075	+	
2076	+ 'DateAndTime' is an 8 or 11 octet bina	
2077	+ month, day, hour, minute, second, deci	
2078	+ optional offset from UTC. See SNMPv	2-TC [SMIv2-TC].
2079	+	
2080	+ NOTE: 'DateAndTime' is not printable	e characters; it is
2081	+ binary.	
2082	+	
2083	+ 'JmTimeStampTC' is the time of day n	neasured in the number of

2084	+ seconds since the system was booted.	
2085	+++++++++++++++++++++++++++++++++++++++	-+++++++++++++++++++++++++
2086		
2087	jobSubmissionToServerTime(190),	JmTimeStampTC
2088		AND/OR
2089		DateAndTime
2090	INTEGER: Configuration 3 only: The	time
2091	AND/OR	
2092		was submitted to the server (as distinguished
2093	from the device which uses jobSubmission	
2094	Troin the device willen dees Jees de lineer	9.1.1 Marie (1)
2095	jobSubmissionTime(191),	JmTimeStampTC
2096	Jobb domission Time (171),	AND/OR
2097		DateAndTime
2098	INTEGER: Configurations 1, 2, and 3:	
2099	AND/OR	The time
2100		was submitted to the server or device to which
		o was submitted to the server or device to which
2101	the agent is providing access.	
2102		
2103		
2104	11G((10A)	T TO GO TOG
2105	jobStartedBeingHeldTime(192),	JmTimeStampTC
2106		AND/OR
2107		DateAndTime
2108	INTEGER: The time	
2109	AND/OR	
2110		o last entered the pendingHeld state. If the job
2111	has never entered the pendingHeld state	e, then the value SHALL be '0' or the attribute
2112	SHALL not be present in the table.	
2113		
2114	jobStartedProcessingTime(193),	JmTimeStampTC
2115		AND/OR
2116		DateAndTime
2117	INTEGER: The time	
2118	AND/OR	
2119	OCTETS: the date and time that the job	o started processing.
2120	3	1 6
2121	jobCompletionTime(194),	JmTimeStampTC
2122	J = = = F = = = = = (=: = =);	AND/OR
2123		DateAndTime
2124	INTEGER: The time	
2125	AND/OR	
2126		entered the completed , canceled , or aborted
2127	state.	o character the completed, canceled, or aborted
2128	state.	
2129	jobProcessingCPUTime(195)	Integer32(-22147483647)
2130	UNITS 'seconds'	III.CgC134(-2417/70307/)
2131		n seconds that the job has been in the processing
2132	ctate. If the job enters the processing of	topped state, that elapsed time SHALL not be
4134	state. If the job enters the processings	opped state, that chapsed time STALL not be

```
2133
                         included. In other words, the jobProcessingCPUTime value SHOULD be relatively
2134
                         repeatable when the same job is processed again on the same device."
2135
2136
             REFERENCE
                   "See Section 3.2 entitled 'The Attribute Mechanism' for a description of this textual-convention
2137
2138
                   and its use in the jmAttributeTable.
2139
2140
                   This is a type 2 enumeration. See Section 3.6.1.2."
2141
             SYNTAX
                          INTEGER {
2142
                   other(1),
2143
                   unknown(2),
2144
                   jobStateReasons2(3),
2145
                   jobStateReasons3(4),
2146
                   jobStateReasons4(5),
2147
                   processingMessage(6),
2148
                   jobCodedCharSet(7),
2149
2150
                   jobURI(20),
2151
                   jobAccountName(21),
2152
                   serverAssignedJobName(22),
2153
                   jobName(23),
2154
                   jobServiceTypes(24),
                   jobSourceChannelIndex(25),
2155
2156
                   jobSourcePlatformType(26),
                   submittingServerName(27),
2157
2158
                   submittingApplicationName(28),
                   jobOriginatingHost(29),
2159
2160
                   deviceNameRequested(30),
2161
                   queueNameRequested(31),
2162
                   physicalDevice(32),
2163
                   numberOfDocuments(33).
2164
                   fileName(34),
2165
                   documentName(35),
2166
                   jobComment(36),
                   documentFormatIndex(37),
2167
2168
                   documentFormat(38),
2169
2170
                   jobPriority(50),
2171
                   jobProcessAfterDateAndTime(51),
2172
                   jobHold(52),
2173
                   jobHoldUntil(53),
2174
                   outputBin(54),
2175
                   sides(55),
2176
                   finishing(56),
2177
                   printQualityRequested(70),
2178
2179
                   printQualityUsed(71),
2180
                   printerResolutionRequested(72),
2181
                   printerResolutionUsed(73),
```

```
2182
                   tonerEcomonyRequested(74),
2183
                   tonerEcomonyUsed(75),
2184
                   tonerDensityRequested(76),
2185
                   tonerDensityUsed(77),
2186
2187
                   jobCopiesRequested(90),
2188
                   jobCopiesCompleted(91),
2189
                   documentCopiesRequested(92),
2190
                   documentCopiesCompleted(93),
2191
                   jobKOctetsTransferred(94),
2192
2193
                   impressionsSpooled(110),
2194
                   impressionsSentToDevice(111),
2195
                   impressionsInterpreted(112),
2196
                   impressionsCompletedCurrentCopy(113),
2197
                   fullColorImpressionsCompleted(114),
2198
                   highlightColorImpressionsCompleted(115),
2199
2200
                   pagesRequested(130),
2201
                   pagesCompleted(131),
2202
                   pagesCompletedCurrentCopy(132),
2203
2204
                   sheetsRequested(150),
2205
                   sheetsCompleted(151),
2206
                   sheetsCompletedCurrentCopy(152),
2207
2208
                   mediumRequested(170),
2209
                   mediumConsumed(171),
2210
                   colorantRequested(172),
2211
                   colorantConsumed(173),
2212
                   jobSubmissionToServerTime(190),
2213
2214
                   jobSubmissionTime(191),
2215
                   jobStartedBeingHeldTime(192),
2216
                   jobStartedProcessingTime(193),
2217
                   jobCompletionTime(194),
2218
                   jobProcessingCPUTime(195)
2219
             }
2220
2221
2222
2223
2224
       JmJobServiceTypesTC ::= TEXTUAL-CONVENTION
2225
             STATUS
                         current
2226
             DESCRIPTION
2227
                   "Specifies the type(s) of service to which the job has been submitted (print, fax, scan, etc.). The
2228
                   service type is represented as an enum that is bit encoded with each job service type so that
2229
                   more general and arbitrary services can be created, such as services with more than one
```

2230 2231	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
2232	jobServiceTypes attribute, corresponding to the hexadecimal values: $0x8 + 0x20 + 0x4$,
2233	respectively, yielding: 0x2C .
2234	
2235	Whether this attribute is set from a job attribute supplied by the job submission client or is set by
2236	the recipient job submission server or device depends on the job submission protocol. With
2237	either implementation, the agent SHALL return a non-zero value for this attribute indicating the
2238	type of the job.
2239	-yp j
2240	One of the purposes of this attribute is to permit a requester to filter out jobs that are not of
2241	interest. For example, a printer operator MAY only be interested in jobs that include printing.
2242	That is why the attribute is in the job identification category.
2243	g, Jou
2244	The following service component types are defined (in hexadecimal) and are assigned a separate
2245	bit value for use with the jobServiceTypes attribute:
2246	on value for use with the gooder vice 1 gpes and is also.
2247	other 0x1
2248	The job contains some instructions that are not one of the identified types.
2249	The for contains some instructions that are not one of the racinities types.
2250	unknown 0x2
2251	The job contains some instructions whose type is unknown to the agent.
2252	The for contains some instructions whose type is untilled in the agent.
2253	print 0x4
2254	The job contains some instructions that specify printing
2255	The job contains some instructions that speetly printing
2256	scan 0x8
2257	The job contains some instructions that specify scanning
2258	The fee commissions management seeming
2259	faxIn 0x10
2260	The job contains some instructions that specify receive fax
2261	j
2262	faxOut 0x20
2263	The job contains some instructions that specify sending fax
2264	j
2265	getFile 0x40
2266	The job contains some instructions that specify accessing files or documents
2267	\mathcal{E}
2268	putFile 0x80
2269	The job contains some instructions that specify storing files or documents
2270	J. J
2271	mailList 0x100
2272	The job contains some instructions that specify distribution of documents using an
2273	electronic mail system."
2274	REFERENCE
2275	"These bit definitions are the equivalent of a type 2 enum except that combinations of them
2276	MAY be used together. See section 3.6.1.2."
2277	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit

2278	
2279	
2280	
2281	
	In Joh Chada Daggarg 1 T.C TEVTI A I. CONVENTION
2282	JmJobStateReasons1TC ::= TEXTUAL-CONVENTION
2283	STATUS current
2284	DESCRIPTION NO. (N. 1. A)
2285	"The JmJobStateReasonsNTC (N=14) textual-conventions are used with the
2286	jmJobStateReasons1 object and jobStateReasonsN (N=24), respectively, to provide
2287	additional information regarding the current jmJobState object value. These values MAY be
2288	used with any job state or states for which the reason makes sense.
2289	NOTE With 1 and 111 and 1 TIO and 12 and 11 and 11 and 12
2290	NOTE - While values cannot be added to the jmJobState object without impacting deployed
2291	clients that take actions upon receiving jmJobState values, it is the intent that additional
2292	JmJobStateReasonsNTC enums can be defined and registered without impacting such
2293	deployed clients. In other words, the jmJobStateReasons1 object and jobStateReasonsN
2294	attributes are intended to be extensible.
2295	
2296	NOTE - The Job Monitoring MIB contains a superset of the IPP values[ipp-model] for the IPP
2297	'job-state-reasons' attribute, since the Job Monitoring MIB is intended to cover other job
2298	submission protocols as well. Also some of the names of the reasons have been changed from
2299	'printer' to 'device', since the Job Monitoring MIB is intended to cover additional types of
2300	devices, including input devices, such as scanners.
2301	
2302	The following standard values are defined (in hexadecimal) as <i>powers of two</i> , since multiple
2303	values MAY be used at the same time. For ease of understanding, the
2304	JmJobStateReasons1TC reasons are presented in the order in which the reasons are likely to
2305	occur (if implemented), starting with the 'jobIncoming' value and ending with the
2306	'jobCompletedWithErrors' value.
2307	
2308	other 0x1
2309	The job state reason is not one of the standardized or registered reasons.
2310	
2311	unknown 0x2
2312	The job state reason is not known to the agent or is indeterminent.
2313	
2314	jobIncoming 0x4
2315	The job has been accepted by the server or device, but the server or device is expecting
2316	(1) additional operations from the client to finish creating the job and/or (2) is
2317	accessing/accepting document data.
2318	
2319	submissionInterrupted 0x8
2320	The job was not completely submitted for some unforeseen reason, such as: (1) the server
2321	has crashed before the job was closed by the client, (2) the server or the document transfer
2322	method has crashed in some non-recoverable way before the document data was entirely

2323	transferred to the server, (3) the client crashed or failed to close the job before the time-
2324	out period.
2325	out period.
2326	jobOutgoing 0x10
2327	Configuration 2 only: The server is transmitting the job to the device.
2328	
2329	jobHoldSpecified 0x20
2330	The value of the job's jobHold(52) attribute is TRUE. The job SHALL NOT be a
2331	candidate for processing until this reason is removed and there are no other reasons to
2332	hold the job.
2333	J
2334	jobHoldUntilSpecified 0x40
2335	The value of the job's jobHoldUntil(53) attribute specifies a time period that is still in the
2336	future. The job SHALL NOT be a candidate for processing until this reason is removed
2337	and there are no other reasons to hold the job.
2338	
2339	jobProcessAfterSpecified 0x80
2340	The value of the job's jobProcessAfterDateAndTime(51) attribute specifies a time that is
2341	still in the future. The job SHALL NOT be a candidate for processing until this reason is
2342	removed and there are no other reasons to hold the job.
2343	J
2344	resourcesAreNotReady 0x100
2345	At least one of the resources needed by the job, such as media, fonts, resource objects,
2346	etc., is not ready on any of the physical devices for which the job is a candidate. This
2347	condition MAY be detected when the job is accepted, or subsequently while the job is
2348	pending or processing , depending on implementation.
2349	
2350	deviceStoppedPartly 0x200
2351	One or more, but not all, of the devices to which the job is assigned are stopped. If all of
2352	the devices are stopped (or the only device is stopped), the deviceStopped reason
2353	SHALL be used.
2354	
2355	deviceStopped 0x400
2356	The device(s) to which the job is assigned is (are all) stopped.
2357	The device(s) to which the job is assigned is (the thi) stopped.
2358	jobInterpreting 0x800
2359	The device to which the job is assigned is interpreting the document data.
2360	1 1 D 1 4
2361	jobPrinting 0x1000
2362	The output device to which the job is assigned is marking media. This attribute is useful
2363	for servers and output devices which spend a great deal of time processing (1) when no
2364	marking is happening and then want to show that marking is now happening or (2) when
2365	the job is in the process of being canceled or aborted while the job remains in the
2366	processing state, but the marking has not yet stopped so that impression or sheet counts
2367	are still increasing for the job.
2368	
2369	jobCanceledByUser 0x2000
2370	The job was canceled by the owner of the job, i.e., by a user whose name is the same as
2310	The job was canceled by the owner of the job, i.e., by a user whose name is the same as

2371	the value of the job's jmJobOwner object, or by some other authorized	end-user, such as
2372	a member of the job owner's security group.	0.000, 50.000 0.5
2373	a memori of the job of more security group.	
2374	jobCanceledByOperator 0x4000	
2375	The job was canceled by the operator, i.e., by a user who has been authorized the control of the	enticated as having
2376	operator privileges (whether local or remote).	mireated as maxing
2377	operator privileges (whether rocar or remote).	
2378	jobCanceledAtDevice 0x8000	
2379	The job was canceled by an unidentified local user, i.e., a user at a conso	ale at the device
	The job was canceled by an undertained total user, i.e., a user at a const	one at the device.
2380		
2381	abortedBySystem 0x10000	
2382	The job (1) is in the process of being aborted, (2) has been aborted by the	e system and
2383	placed in the 'aborted' state, or (3) has been aborted by the system and	
2384	'pendingHeld' state, so that a user or operator can manually try the job again.	1
2385		
2386	processingToStopPoint 0x20000	
2387	The requester has issued an operation to cancel or interrupt the job or the	ne server/device
2388	has aborted the job, but the server/device is still performing some action	s on the job until a
2389	specified stop point occurs or job termination/cleanup is completed.	J
2390		
2391	This reason is recommended to be used in conjunction with the process :	ing job state to
2392	indicate that the server/device is still performing some actions on the job	
2393	remains in the processing state. After all the job's resources consumed	
2394	stopped incrementing, the server/device moves the job from the process	
2395	canceled or aborted job states.	
2396		
2397	serviceOffLine 0x40000	
2398	The service or document transform is off-line and accepting no jobs. Al	l pending jobs are
2399	put into the pendingHeld state. This situation could be true if the servi	
2400	transform's input is impaired or broken.	
2401	www.szs.m.s mpw.s m.pw.ew or cronem	
2402	jobCompletedSuccessfully 0x80000	
2403	The job completed successfully.	
2404	The job completed successian;	
2405	jobCompletedWithWarnings 0x100000	
2406	The job completed with warnings.	
2407	The job completes with warmings.	
2408	jobCompletedWithErrors 0x200000	
2409	The job completed with errors (and possibly warnings too).	
2410	The job completes with errors (and possior) warmings too).	
2411		
2412	The following additional job state reasons have been added to represent job st	ates that are in
2413	ISO DPA[iso-dpa] and other job submission protocols:	ares that are m
2414	100 Dirigino apaj ana omor joe saomission protocols.	
2415	jobPaused 0x400000	
2416	The job has been indefinitely suspended by a client issuing an operation	to suspend the job
2417	so that other jobs may proceed using the same devices. The client MAY	issue an
2418	operation to resume the paused job at any time, in which case the agent	SHALL remove
	-r	

2419	the jobPaused values from the job's jmJobStateReasons1 object and the job is eventually
2420	resumed at or near the point where the job was paused.
2421	
2422	jobInterrupted 0x800000
2423	The job has been interrupted while processing by a client issuing an operation that
2424	specifies another job to be run instead of the current job. The server or device will
2425	automatically resume the interrupted job when the interrupting job completes.
2426	
2427	jobRetained 0x1000000
2428	The job is being retained by the server or device with all of the job's document data (and
2429	submitted resources, such as fonts, logos, and forms, if any). Thus a client could issue an
2430	operation to the server or device to either (1) re-do the job (or a copy of the job) on the
2431	same server or device or (2) resubmit the job to another server or device. When a client
2431	
	could no longer re-do/resubmit the job, such as after the document data has been
2433	discarded, the agent SHALL remove the jobRetained value from the
2434	jmJobStateReasons1 object."
2435	REFERENCE
2436	"These bit definitions are the equivalent of a type 2 enum except that combinations of bits may
2437	be used together. See section 3.6.1.2. The remaining bits are reserved for future
2438	standardization and/or registration."
2439	
2440	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2441	
2442	
2443	
2444	
2445	
2446	JmJobStateReasons2TC ::= TEXTUAL-CONVENTION
2447	STATUS current
2448	DESCRIPTION
2449	"This textual-convention is used with the jobStateReasons2 attribute to provides additional
2450	information regarding the jmJobState object. See the description under
2451	JmJobStateReasons1TC for additional information that applies to all reasons.
2452	singulate reasons in the for additional information that applies to an reasons.
2453	The following standard values are defined (in hexadecimal) as <i>powers of two</i> , since multiple
2454	
	values may be used at the same time:
2455	
2456	cascaded 0x1
2457	An outbound gateway has transmitted all of the job's job and document attributes and data
2458	to another spooling system.
2459	
2460	deletedByAdministrator 0x2
2461	The administrator has deleted the job.
2462	
2463	discardTimeArrived 0x4
2464	The job has been deleted due to the fact that the time specified by the job's job-discard-
2465	time attribute has arrived.
2466	

The post-processing agent failed while trying to log accounting attributes therefore the job has been placed into the completed state with the jobRe jmJobStateReasons1 object value for a system-defined period of time, s administrator can examine it, resubmit it, etc. jobTransforming ox10 The server/device is interpreting document data and producing another el	etained
therefore the job has been placed into the completed state with the jobRe jmJobStateReasons1 object value for a system-defined period of time, s administrator can examine it, resubmit it, etc. jobTransforming ox10 The server/device is interpreting document data and producing another el	etained
jmJobStateReasons1 object value for a system-defined period of time, s administrator can examine it, resubmit it, etc. jobTransforming ox10 The server/device is interpreting document data and producing another el	
2471 administrator can examine it, resubmit it, etc. 2472 2473 jobTransforming 0x10 2474 The server/device is interpreting document data and producing another el	
2472 2473 jobTransforming 2474 The server/device is interpreting document data and producing another el	
jobTransforming 0x10 The server/device is interpreting document data and producing another el	
The server/device is interpreting document data and producing another el	
	ectronic
representation.	
2476	
2477 maxJobFaultCountExceeded 0x20	
The job has faulted several times and has exceeded the administratively de	efined fault count
2479 limit.	crinea raun coam
2480	
2481 devicesNeedAttentionTimeOut 0x40	
One or more document transforms that the job is using needs human inter	vention in order
for the job to make progress, but the human intervention did not occur wi	ithin the cite
settable time-out value.	iumi me site-
2485	
2486 needsKeyOperatorTimeOut 0x80	
One or more devices or document transforms that the job is using need a	
operator (who may need a key to unlock the device and gain access) in or	
make progress, but the key operator intervention did not occur within the	site-settable
2490 time-out value.	
2491	
2491 2492 jobStartWaitTimeOut 0x100	
2491 2492 jobStartWaitTimeOut 0x100 2493 The server/device has stopped the job at the beginning of processing to av	
2491 2492 jobStartWaitTimeOut 0x100 2493 The server/device has stopped the job at the beginning of processing to average action, such as installing a special cartridge or special non-standard median	a, but the job was
jobStartWaitTimeOut 0x100 The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard median not resumed within the site-settable time-out value and the server/device.	a, but the job was
jobStartWaitTimeOut 0x100 The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard median not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state.	a, but the job was
jobStartWaitTimeOut 0x100 The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state.	a, but the job was
jobStartWaitTimeOut 0x100 The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut 0x200	a, but the job was has transitioned
jobStartWaitTimeOut	n, but the job was has transitioned uman action,
jobStartWaitTimeOut	a, but the job was has transitioned uman action, job was not
jobStartWaitTimeOut	a, but the job was has transitioned uman action, job was not
jobStartWaitTimeOut 2493 The server/device has stopped the job at the beginning of processing to average action, such as installing a special cartridge or special non-standard mediate not resumed within the site-settable time-out value and the server/device to the job to the pendingHeld state. jobEndWaitTimeOut yource to action, such as installing a special cartridge or special non-standard mediate not resumed within the site-settable time-out value and the server/device to action action. jobEndWaitTimeOut yource to action to average to action action action action action. jobEndWaitTimeOut action, such as installing a special cartridge or special non-standard media action, such as installing a special cartridge or action action.	a, but the job was has transitioned uman action, job was not
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut ox200 The server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state.	a, but the job was has transitioned uman action, job was not
jobStartWaitTimeOut 2493 The server/device has stopped the job at the beginning of processing to average action, such as installing a special cartridge or special non-standard median not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut performance of processing to await here server/device has stopped the job at the end of processing to await here such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut ox400	uman action, job was not transitioned the
jobStartWaitTimeOut 2493 The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard median not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut for server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut for server/device has stopped the job at the beginning of processing to avait for the server/device has job to the completed state.	uman action, job was not transitioned the
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard median not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut The server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut The server/device has stopped the job at the beginning of processing to avait he server/device has job's password, but the password was not received within the site-settable	uman action, job was not transitioned the
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut the server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut the server/device has stopped the job at the beginning of processing to avait the password was not received within the site-settable to avait the password was not received within the site-settable to avait the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password.	uman action, job was not transitioned the
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut the server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut the server/device has stopped the job at the beginning of processing to avait the password was not received within the site-settable to avait the password was not received within the site-settable to avait the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password, but the password was not received within the site-settable pob's password.	uman action, job was not transitioned the
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut jobPasswordWaitTimeOut for Server/device has stopped the job at the beginning of processing to avait the server/device has job to the completed state. jobPasswordWaitTimeOut for Server/device has stopped the job at the beginning of processing to avait the server/device has job's password, but the password was not received within the site-settable deviceTimedOut deviceTimedOut 0x100 0x200 The server/device has stopped the job at the beginning of processing to avait the password was not received within the site-settable deviceTimedOut deviceTimedOut 0x800	uman action, job was not transitioned the wait input of the etime-out value.
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut The server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut the server/device has stopped the job at the beginning of processing to avait he server/device has job to the completed state. jobPasswordWaitTimeOut the server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has stopped the job at the beginning of processing to avait he server/device has job's password, but the password was not received within the site-settable the password was not	uman action, job was not transitioned the wait input of the etime-out value.
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut jobEndWaitTimeOut the server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut the server/device has stopped the job at the beginning of processing to avait he server/device has job's password, but the password was not received within the site-settable server/device has the password was not received within the site-settable that the job was using has not responded in a period specified by site-settable attribute.	uman action, job was not transitioned the wait input of the etime-out value.
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut The server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut jobPasswordWaitTimeOut jobPasswordWaitTimeOut job's password, but the password was not received within the site-settable attribute. deviceTimedOut A device that the job was using has not responded in a period specified by site-settable attribute.	uman action, job was not transitioned the wait input of the etime-out value.
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut you the server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut you jobPasswordWaitTimeOut jobPasswordWaitTimeOut job's password, but the password was not received within the site-settable deviceTimeOut A deviceTimedOut A device that the job was using has not responded in a period specified by site-settable attribute. connectingToDeviceTimeOut Ox1000	uman action, job was not transitioned the wait input of the etime-out value.
jobStartWaitTimeOut The server/device has stopped the job at the beginning of processing to avaction, such as installing a special cartridge or special non-standard media not resumed within the site-settable time-out value and the server/device the job to the pendingHeld state. jobEndWaitTimeOut you The server/device has stopped the job at the end of processing to await he such as removing a special cartridge or restoring standard media, but the resumed within the site-settable time-out value and the server/device has job to the completed state. jobPasswordWaitTimeOut you jobPasswordWaitTimeOut you jobPasswordWaitTimeOut ox400 The server/device has stopped the job at the beginning of processing to avait he server/device has job's password, but the password was not received within the site-settable story deviceTimedOut A device that the job was using has not responded in a period specified by site-settable attribute. connectingToDeviceTimeOut ox100 ox100 ox100	uman action, job was not transitioned the wait input of the etime-out value. The dial-up, polled,

2516	
2517	transferring 0x2000
2518	The job is being transferred to a down stream server or downstream device.
2519	\mathcal{F}
2520	queuedInDevice 0x4000
2521	The server/device has queued the job in a down stream server or downstream device.
2522	The server device has queded the job in a down stream server or downstream device.
2523	jobQueued 0x8000
2523	
2524	The server/device has queued the document data.
2525	. 1 (1)
2526	jobCleanup 0x10000
2527	The server/device is performing cleanup activity as part of ending normal processing.
2528	
2529	jobPasswordWait 0x20000
2530	The server/device has selected the job to be next to process, but instead of assigning
2531	resources and starting the job processing, the server/device has transitioned the job to the
2532	pendingHeld state to await entry of a password (and dispatched another job, if there is
2533	one).
2534	
2535	validating 0x40000
2536	The server/device is validating the job <i>after</i> accepting the job.
2537	The server/device is vandating the job after accepting the job.
2538	queueHeld 0x80000
2539 2539	4
2540	The operator has held the entire job set or queue.
	:-bD
2541	jobProofWait 0x100000
2542	The job has produced a single proof copy and is in the pendingHeld state waiting for the
2543	requester to issue an operation to release the job to print normally, obeying any job and
2544	document copy attributes that were originally submitted.
2545	
2546	heldForDiagnostics 0x200000
2547	The system is running intrusive diagnostics, so that all jobs are being held.
2548	
2549	noSpaceOnServer 0x800000
2550	There is no room on the server to store all of the job.
2551	
2552	pinRequired 0x1000000
2553	The System Administrator settable device policy is (1) to require PINs, and (2) to hold
2554	jobs that do not have a pin supplied as an input parameter when the job was created.
2555	jobs that do not have a pin supplied as an input parameter when the job was created.
2556	exceededAccountLimit 0x2000000
2557	
	The account for which this job is drawn has exceeded its limit. This condition SHOULD
2558	be detected before the job is scheduled so that the user does not wait until his/her job is
2559	scheduled only to find that the account is overdrawn. This condition MAY also occur
2560	while the job is processing either as processing begins or part way through processing.
2561	1.1IE D.4.
2562	heldForRetry 0x4000000
2563	The job encountered some errors that the server/device could not recover from with its
2564	normal retry procedures, but the error might not be encountered if the job is processed

2565	again in the future. Example cases are phone number busy or remote file system in-
2566	accessible. For such a situation, the server/device SHALL transition the job from the
2567	processing to the pendingHeld, rather than to the aborted state.
2568	
2569	The following values are from the X/Open PSIS draft standard:
2570	The following with from the 12 open 1 818 didniedles
2571	canceledByShutdown 0x8000000
2572	The job was canceled because the server or device was shutdown before completing the
2573	job.
2574	Jou.
2575	deviceUnavailable 0x10000000
2576	This job was aborted by the system because the device is currently unable to accept jobs.
2577	This job was aborted by the system because the device is currently unable to accept jobs.
2578	wrongDevice 0x20000000
2579	
	This job was aborted by the system because the device is unable to handle this particular
2580 2581	job; the spooler SHOULD try another device or the user should submit the job to another
	device.
2582	L - J T - L
2583	badJob 0x40000000
2584	This job was aborted by the system because this job has a major problem, such as an ill-
2585	formed PDL; the spooler SHOULD not even try another device."
2586	REFERENCE
2587	"These bit definitions are the equivalent of a type 2 enum except that combinations of them may
2588	be used together. See section 3.6.1.2. See the description under JmJobStateReasons1TC and
2589	the jobStateReasons2 attribute."
2590	
2591	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2592	
2593	
2594	
2595	
2596	
2597	
2598	JmJobStateReasons3TC ::= TEXTUAL-CONVENTION
2599	STATUS current
2600	DESCRIPTION
2601	"This textual-convention is used with the jobStateReasons3 attribute to provides additional
2602	information regarding the jmJobState object. See the description under
2603	JmJobStateReasons1TC for additional information that applies to all reasons.
2604	
2605	The following standard values are defined (in hexadecimal) as <i>powers of two</i> , since multiple
2606	values may be used at the same time:
2607	
2608	jobInterruptedByDeviceFailure 0x1
2609	A device or the print system software that the job was using has failed while the job was
2610	processing. The server or device is keeping the job in the pendingHeld state until an
2611	operator can determine what to do with the job."
2612	REFERENCE

2613	"These bit definitions are the equivalent of a type 2 enum except that combinations of them ma
2614	be used together. See section 3.6.1.2. The remaining bits are reserved for future
2615	standardization and/or registration. See the description under JmJobStateReasons1TC and the
2616	jobStateReasons3 attribute."
2617	SYNTAX INTEGER(02147483647) 31 bits, all but sign bit
2618	
2619	
2620	
2621	
2622	
2623	JmJobStateReasons4TC ::= TEXTUAL-CONVENTION
2624	STATUS current
2625	DESCRIPTION
2626	"This textual-convention is used in the jobStateReasons4 attribute to provides additional
2627	information regarding the jmJobState object. See the description under
2628	JmJobStateReasons1TC for additional information that applies to all reasons.
2629	
2630	The following standard values are defined (in hexadecimal) as powers of two, since multiple
2631	values may be used at the same time:
2632	
2633	none yet defined. These bits are reserved for future standardization and/or registration."
2634	REFERENCE
2635	"These bit definitions are the equivalent of a type 2 enum except that combinations of them ma
2636	be used together. See section 3.6.1.2. See the description under JmJobStateReasons1TC and
2637	the jobStateReasons4 attribute."
2638	and John Contraction of the Cont
2630	SVNITAV INTECED (0. 2147482647) 21 bits all but sign bit

```
2640
2641
       jobmonMIBObjects OBJECT IDENTIFIER ::= { jobmonMIB 1 }
2642
2643
       -- The General Group (MANDATORY)
2644
2645
       -- The jmGeneralGroup consists entirely of the jmGeneralTable.
2646
2647
       imGeneral OBJECT IDENTIFIER ::= { jobmonMIBObjects 1 }
2648
2649
       imGeneralTable OBJECT-TYPE
2650
             SYNTAX
                         SEQUENCE OF JmGeneralEntry
2651
             MAX-ACCESS not-accessible
2652
             STATUS
                        current
2653
             DESCRIPTION
                  "The jmGeneralTable consists of information of a general nature that are per-job-set, but are
2654
2655
                  not per-job. See Section 2 entitled 'Terminology and Job Model' for the definition of a job set."
2656
             REFERENCE
2657
                  "The MANDATORY-GROUP macro specifies that this group is MANDATORY."
2658
             ::= \{ \text{ imGeneral } 1 \}
2659
2660
       imGeneralEntry OBJECT-TYPE
2661
                         JmGeneralEntry
             SYNTAX
2662
             MAX-ACCESS not-accessible
2663
             STATUS
                        current
             DESCRIPTION
2664
2665
                  "Information about a job set (queue).
2666
2667
                  An entry SHALL exist in this table for each job set."
2668
             INDEX { jmGeneralJobSetIndex }
2669
             ::= { jmGeneralTable 1 }
2670
2671
       JmGeneralEntry ::= SEQUENCE {
2672
             jmGeneralJobSetIndex
                                                               Integer32(1..32767),
             jmGeneralNumberOfActiveJobs
2673
                                                               Integer32(0..2147483647),
2674
             jmGeneralOldestActiveJobIndex
                                                               Integer32(0..2147483647),
2675
             imGeneralNewestActiveJobIndex
                                                               Integer32(0...2147483647),
             jmGeneralJobPersistence
                                                               Integer32(15..2147483647),
2676
                                                               Integer32(15..2147483647),
2677
             imGeneralAttributePersistence
2678
             jmGeneralJobSetName
                                                               JmUTF8StringTC(SIZE(0..63))
2679
       }
2680
2681
       jmGeneralJobSetIndex OBJECT-TYPE
2682
             SYNTAX
                         Integer32(1..32767)
2683
             MAX-ACCESS not-accessible
2684
             STATUS
                        current
             DESCRIPTION
2685
                  "A unique value for each job set in this MIB. The jmJobTable and jmAttributeTable tables
2686
2687
                  have this same index as their primary index.
2688
```

2689	The value(s) of the jmGeneralJobSetIndex SHALL be persistent across power cycles, so that
2690	clients that have retained jmGeneralJobSetIndex values will access the same job sets upon
2691	subsequent power-up.
2692	buobequent power up.
	An implementation that has only one job set such as a mintan with a single group. SHALL hard
2693	An implementation that has only one job set, such as a printer with a single queue, SHALL hard
2694	code this object with the value 1."
2695	REFERENCE
2696	"See Section 2 entitled 'Terminology and Job Model' for the definition of a job set.
2697	Corresponds to the first index in jmJobTable and jmAttributeTable ."
2698	::= { jmGeneralEntry 1 }
	(JindeneralEndy 1)
2699	'. C N L. OM A'. I L. ODJECT TYPE
2700	jmGeneralNumberOfActiveJobs OBJECT-TYPE
2701	SYNTAX Integer32(02147483647)
2702	MAX-ACCESS read-only
2703	STATUS current
2704	DESCRIPTION
2705	"The current number of 'active' jobs in the jmJobIDTable , jmJobTable , and
2706	jmAttributeTable, i.e., the total number of jobs that are in the pending, processing, or
2707	processingStopped states. See the JmJobStateTC textual-convention for the exact
2708	specification of the semantics of the job states."
2709	::= { jmGeneralEntry 2 }
2710	
2711	jmGeneralOldestActiveJobIndex OBJECT-TYPE
2712	SYNTAX Integer32 (02147483647)
2713	MAX-ACCESS read-only
2714	STATUS current
2715	DESCRIPTION
2716	"The jmJobIndex of the oldest job that is still in one of the 'active' states (pending , processing
2717	or processingStopped). In other words, the index of the 'active' job that has been in the job
2718	tables the longest.
2719	
2720	If there are no active jobs, the agent SHALL set the value of this object to 0 ."
2721	REFERENCE
2722	"See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes' for
2723	a description of the usage of this object."
2724	::= { jmGeneralEntry 3 }
2725	
2726	jmGeneralNewestActiveJobIndex OBJECT-TYPE
2727	SYNTAX Integer32 (02147483647)
2728	MAX-ACCESS read-only
2729	
2730	DESCRIPTION
2731	"The jmJobIndex of the newest job that is in one of the 'active' states (pending , processing , or
2732	processingStopped). In other words, the index of the 'active' job that has been most recently
2733	added to the job tables.
2734	
2735	When all jobs become 'inactive', i.e., enter the pendingHeld, completed, canceled, or aborted
2736	states, the agent SHALL set the value of this object to 0 ."
2737	REFERENCE
4131	KLI LIKLINCE

```
2738
                   "See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes' for
2739
                   a description of the usage of this object."
2740
             ::= { jmGeneralEntry 4 }
2741
2742
        jmGeneralJobPersistence OBJECT-TYPE
2743
             SYNTAX
                          Integer32(15..2147483647)
2744
             UNITS
                         "seconds"
2745
             MAX-ACCESS read-only
2746
             STATUS
                          current
2747
             DESCRIPTION
2748
                   "The minimum time in seconds for this instance of the Job Set that an entry SHALL remain in
                   the imJobIDTable and imJobTable after processing has completed, i.e., the minimum time in
2749
2750
                   seconds starting when the job enters the completed, canceled, or aborted state.
2751
2752
                   Configuring this object is implementation-dependent.
2753
2754
                   This value SHALL be equal to or greater than the value of jmGeneralAttributePersistence.
2755
                   This value SHOULD be at least 60 which gives a monitoring application one minute in which to
2756
                   poll for job data."
                          { 60 }
2757
             DEFVAL
                                      -- one minute
             ::= { jmGeneralEntry 5 }
2758
2759
2760
        jmGeneralAttributePersistence OBJECT-TYPE
             SYNTAX
2761
                          Integer32(15..2147483647)
2762
                         "seconds"
             UNITS
2763
             MAX-ACCESS read-only
              STATUS
2764
                          current
2765
             DESCRIPTION
2766
                   "The minimum time in seconds for this instance of the Job Set that an entry SHALL remain in
2767
                   the imAttributeTable after processing has completed, i.e., the time in seconds starting when
2768
                   the job enters the completed, canceled, or aborted state.
2769
2770
                   Configuring this object is implementation-dependent.
2771
2772
                   This value SHOULD be at least 60 which gives a monitoring application one minute in which to
2773
                   poll for job data."
             DEFVAL
2774
                          { 60 }
                                      -- one minute
2775
             ::= { jmGeneralEntry 6 }
2776
2777
        jmGeneralJobSetName OBJECT-TYPE
2778
             SYNTAX
                          JmUTF8StringTC(SIZE(0..63))
2779
             MAX-ACCESS read-only
2780
             STATUS
                          current
2781
             DESCRIPTION
2782
                   "The human readable name of this job set assigned by the system administrator (by means
2783
                   outside of this MIB). Typically, this name SHOULD be the name of the job queue. If a server
2784
                   or device has only a single job set, this object can be the administratively assigned name of the
2785
                   server or device itself. This name does not need to be unique, though each job set in a single
```

2786

Job Monitoring MIB SHOULD have distinct names.

```
2787
2788
                  NOTE - The purpose of this object is to help the user of the job monitoring application
2789
                  distinguish between several job sets in implementations that support more than one job set."
2790
             REFERENCE
                  "See the OBJECT compliance macro for the minimum maximum length required for
2791
2792
                  conformance."
2793
             ::= { jmGeneralEntry 7 }
2794
2795
2796
2797
2798
2799
       -- The Job ID Group (MANDATORY)
2800
2801
       -- The jmJobIDGroup consists entirely of the jmJobIDTable.
2802
2803
       jmJobID OBJECT IDENTIFIER ::= { jobmonMIBObjects 2 }
2804
2805
       imJobIDTable OBJECT-TYPE
2806
                         SEQUENCE OF JmJobIDEntry
             SYNTAX
2807
             MAX-ACCESS not-accessible
2808
             STATUS
                         current
2809
             DESCRIPTION
2810
                  "The imJobIDTable provides a correspondence map (1) between the job submission ID that a
2811
                  client uses to refer to a job and (2) the imGeneralJobSetIndex and imJobIndex that the Job
2812
                  Monitoring MIB agent assigned to the job and that are used to access the job in all of the other
2813
                  tables in the MIB. If a monitoring application already knows the imGeneralJobSetIndex and
2814
                  the jmJobIndex of the job it is querying, that application NEED NOT use the jmJobIDTable."
2815
             REFERENCE
2816
                  "The MANDATORY-GROUP macro specifies that this group is MANDATORY."
2817
             ::= { jmJobID 1 }
2818
2819
       jmJobIDEntry OBJECT-TYPE
2820
             SYNTAX
                         JmJobIDEntry
2821
             MAX-ACCESS not-accessible
2822
             STATUS
                         current
2823
             DESCRIPTION
2824
                  "The map from (1) the jmJobSubmissionID to (2) the jmGeneralJobSetIndex and
2825
                  jmJobIndex.
2826
2827
                  An entry SHALL exist in this table for each job currently known to the agent for all job sets and
2828
                  job states. Each job SHALL appear in one and only one job set."
2829
             INDEX { jmJobSubmissionID }
2830
             ::= { jmJobIDTable 1 }
2831
2832
       JmJobIDEntry ::= SEQUENCE {
                                                                OCTET STRING(SIZE(48)),
             imJobSubmissionID
2833
2834
             jmJobIDJobSetIndex
                                                                Integer32(1..32767),
                                                                Integer32(1..2147483647)
2835
             jmJobIDJobIndex
```

```
2836
        }
2837
        imJobSubmissionID OBJECT-TYPE
2838
2839
             SYNTAX
                          OCTET STRING(SIZE(48))
2840
             MAX-ACCESS not-accessible
2841
             STATUS
                          current
2842
             DESCRIPTION
2843
                   "A quasi-unique 48-octet fixed-length string ID which identifies the job within a particular
2844
                   client-server environment. There are multiple formats for the jmJobSubmissionID. Each
2845
                   format SHALL be uniquely identified. See the JmJobSubmissionIDTypeTC textual convention.
2846
                   Each format SHALL be registered using the procedures of a type 2 enum. See section 3.6.3
2847
                   entitled: 'IANA Registration of Job Submission Id Formats'.
2848
2849
                   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
2850
2851
                   the standard formats that have been reserved for agents and adding the final 8 octets to
2852
                   distinguish the ID from others submitted from the same requester.
2853
2854
                   The monitoring application, whether in the client or running separately, MAY use the job
                   submission ID to help identify which imJobIndex was assigned by the agent, i.e., in which row
2855
2856
                   the job information is in the other tables.
2857
2858
                   NOTE - fixed-length is used so that a management application can use a shortened GetNext
2859
                   varbind (in SNMPv1 and SNMPv2) in order to get the next submission ID, disregarding the
                   remainder of the ID in order to access jobs independent of the trailing identifier part, e.g., to get
2860
                   all jobs submitted by a particular jmJobOwner or submitted from a particular MAC address.
2861
2862
             REFERENCE
2863
                   "See the JmJobSubmissionIDTypeTC textual convention.
2864
                   See APPENDIX B - Support of the Job Submission ID in Job Submission Protocols."
2865
              ::= { jmJobIDEntry 1 }
2866
2867
        jmJobIDJobSetIndex OBJECT-TYPE
2868
             SYNTAX
                           Integer32(1..32767)
             MAX-ACCESS read-only
2869
2870
             STATUS
                          current
2871
             DESCRIPTION
2872
                   "This object contains the value of the jmGeneralJobSetIndex for the job with the
                   jmJobSubmissionID value, i.e., the job set index of the job set in which the job was placed
2873
2874
                   when that server or device accepted the job. This 16-bit value in combination with the
2875
                   jmJobIDJobIndex value permits the management application to access the other tables to
2876
                   obtain the job-specific objects for this job."
2877
             REFERENCE
2878
                   "See jmGeneralJobSetIndex in the jmGeneralTable."
2879
             ::= { jmJobIDEntry 2 }
2880
        jmJobIDJobIndex OBJECT-TYPE
2881
2882
                           Integer32(1..2147483647)
             SYNTAX
2883
             MAX-ACCESS read-only
2884
             STATUS
                          current
```

```
2885
             DESCRIPTION
2886
                   "This object contains the value of the jmJobIndex for the job with the jmJobSubmissionID
2887
                   value, i.e., the job index for the job when the server or device accepted the job. This value, in
2888
                   combination with the jmJobIDJobSetIndex value, permits the management application to
2889
                   access the other tables to obtain the job-specific objects for this job."
2890
             REFERENCE
2891
                   "See jmJobIndex in the jmJobTable."
2892
             ::= { jmJobIDEntry 3 }
2893
2894
2895
2896
2897
        -- The Job Group (MANDATORY)
2898
2899
       -- The jmJobGroup consists entirely of the jmJobTable.
2900
2901
       jmJob OBJECT IDENTIFIER ::= { jobmonMIBObjects 3 }
2902
2903
       imJobTable OBJECT-TYPE
2904
                          SEQUENCE OF JmJobEntry
             SYNTAX
             MAX-ACCESS not-accessible
2905
2906
             STATUS
                         current
2907
             DESCRIPTION
2908
                   "The imJobTable consists of basic job state and status information for each job in a job set that
2909
                   (1) monitoring applications need to be able to access in a single SNMP Get operation, (2) that
2910
                   have a single value per job, and (3) that SHALL always be implemented."
2911
             REFERENCE
2912
                   "The MANDATORY-GROUP macro specifies that this group is MANDATORY."
2913
             ::= \{ \text{ imJob } 1 \}
2914
2915
       jmJobEntry OBJECT-TYPE
2916
             SYNTAX
                          JmJobEntry
2917
             MAX-ACCESS not-accessible
2918
             STATUS
                         current
2919
             DESCRIPTION
2920
                   "Basic per-job state and status information."
2921
2922
                   An entry SHALL exist in this table for each job, no matter what the state of the job is. Each job
2923
                   SHALL appear in one and only one job set."
2924
             REFERENCE
2925
                   "See Section 3.2 entitled 'The Job Tables'."
2926
             INDEX { jmGeneralJobSetIndex, jmJobIndex }
2927
             ::= { jmJobTable 1 }
2928
2929
        JmJobEntry ::= SEQUENCE {
2930
             imJobIndex
                                                                 Integer32(1..2147483647),
2931
             imJobState
                                                                 JmJobStateTC.
2932
             jmJobStateReasons1
                                                                 JmJobStateReasons1TC,
2933
             jmNumberOfInterveningJobs
                                                                 Integer32(-2..2147483647),
```

Integer32(-2..2147483647),

```
2935
             jmJobKOctetsProcessed
                                                                  Integer32(-2..2147483647),
2936
             imJobImpressionsRequested
                                                                  Integer32(-2..2147483647),
2937
             jmJobImpressionsCompleted
                                                                  Integer32(-2..2147483647),
2938
             imJobOwner
                                                                  JmJobStringTC(SIZE(0..63))
2939
        }
2940
2941
        jmJobIndex OBJECT-TYPE
2942
             SYNTAX
                          Integer32(1..2147483647)
2943
             MAX-ACCESS not-accessible
2944
             STATUS
                         current
             DESCRIPTION
2945
2946
                   "The sequential, monatonically increasing identifier index for the job generated by the server or
2947
                   device when that server or device accepted the job. This index value permits the management
                   application to access the other tables to obtain the job-specific row entries."
2948
2949
             REFERENCE
2950
                   "See Section 3.2 entitled 'The Job Tables and the Oldest Active and Newest Active Indexes'.
2951
                   See Section 3.4 entitled 'Job Identification'.
2952
                   See also jmGeneralNewestActiveJobIndex for the largest value of jmJobIndex.
2953
                   See JmJobSubmissionTypeTC for a limit on the size of this index if the agent represents it as
2954
                   an 8-digit decimal number."
2955
             ::= { jmJobEntry 1 }
2956
2957
        jmJobState OBJECT-TYPE
2958
             SYNTAX
                          JmJobStateTC
2959
             MAX-ACCESS read-only
2960
             STATUS
                         current
2961
             DESCRIPTION
2962
                   "The current state of the job (pending, processing, completed, etc.). Agents SHALL
2963
                   implement only those states which are appropriate for the particular implementation. However,
                   management applications SHALL be prepared to receive all the standard job states.
2964
2965
2966
                   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,
2967
2968
                   canceled, or aborted state before removing the job data from the jmJobIDTable and
                   imJobTable is specified by the value of the imGeneralJobPersistence object."
2969
2970
             ::= { jmJobEntry 2 }
2971
2972
        jmJobStateReasons1 OBJECT-TYPE
2973
                          JmJobStateReasons1TC
             SYNTAX
2974
             MAX-ACCESS read-only
2975
             STATUS
                          current
2976
             DESCRIPTION
2977
                   "Additional information about the job's current state, i.e., information that augments the value of
2978
                   the job's jmJobState object.
2979
2980
                   Implementation of any reason values is OPTIONAL, but an agent SHOULD return any reason
2981
                   information available These values MAY be used with any job state or states for which the
```

reason makes sense. Since the Job State Reasons will be more dynamic than the Job State, it is

2934

2982

jmJobKOctetsRequested

2983 recommended that a job monitoring application read this object every time **jmJobState** is read. 2984 When the agent cannot provide a reason for the current state of the job, the the value of the 2985 jmJobStateReasons1 object and jobStateReasonsN attributes SHALL be 0." 2986 REFERENCE "The **jobStateReasons**N (N=2..4) attributes provide further additional information about the 2987 2988 job's current state." 2989 ::= { jmJobEntry 3 } 2990 2991 jmNumberOfInterveningJobs OBJECT-TYPE 2992 Integer32(-2..2147483647) SYNTAX 2993 MAX-ACCESS read-only 2994 **STATUS** current 2995 DESCRIPTION 2996 "The number of jobs that are expected to complete processing before this job has completed 2997 processing according to the implementation's queuing algorithm, if no other jobs were to be 2998 submitted. In other words, this value is the job's queue position. The agent SHALL return a 2999 value of **0** for this attribute when the job is the next job to complete processing (or has 3000 completed processing)." 3001 ::= { jmJobEntry 4 } 3002 3003 jmJobKOctetsRequested OBJECT-TYPE 3004 SYNTAX Integer32(-2..2147483647) MAX-ACCESS read-only 3005 3006 STATUS current 3007 DESCRIPTION "The total size in K (1024) octets of the document(s) being requested to be processed in the job. 3008 The agent SHALL round the actual number of octets up to the next highest K. Thus 0 octets 3009 3010 SHALL be represented as '0', 1-1024 octets SHALL be represented as '1', 1025-2048 SHALL 3011 be represented as '2', etc. 3012 3013 In computing this value, the server/device SHALL *not* include the multiplicative factors contributed by (1) the number of document copies, and (2) the number of job copies, 3014 3015 independent of whether the device can process multiple copies of the job or document without making multiple passes over the job or document data and independent of whether the output is 3016 3017 collated or not. Thus the server/device computation is independent of the implementation." 3018 ::= { jmJobEntry 5 } 3019 3020 jmJobKOctetsProcessed OBJECT-TYPE 3021 SYNTAX Integer32(-2..2147483647) 3022 MAX-ACCESS read-only 3023 STATUS current **DESCRIPTION** 3024 3025 "The current number of octets processed by the server or device measured in units of K (1024) 3026 octets. The agent SHALL round the actual number of octets processed up to the next higher K. 3027 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 number interpreted by the

page description language interpreter rather than what has been marked on media.

3028

3029

3031 For implementations where multiple copies are produced by the interpreter with only a single 3032 pass over the data, the final value SHALL be equal to the value of the 3033 **imJobKOctetsRequested** object. For implementations where multiple copies are produced by 3034 the interpreter by processing the data for each copy, the final value SHALL be a multiple of the value of the **jmJobKOctetsRequested** object. 3035 3036 3037 NOTE - See the impressionsCompletedCurrentCopy and pagesCompletedCurrentCopy 3038 attributes for attributes that are reset on each document copy. 3039 3040 NOTE - The **imJobKOctetsProcessed** object can be used with the **imJobKOctetsRequested** 3041 object to provide an indication of the relative progress of the job, provided that the 3042 multiplicative factor is taken into account for some implementations of multiple copies." 3043 ::= { jmJobEntry 6 } 3044 3045 jmJobImpressionsRequested OBJECT-TYPE 3046 SYNTAX Integer32(-2..2147483647) 3047 MAX-ACCESS read-only 3048 STATUS current 3049 DESCRIPTION "The total size in number of impressions of the document(s) being requested by this job to 3050 3051 produce. 3052 3053 In computing this value, the server/device SHALL *not* include the multiplicative factors 3054 contributed by (1) the number of document copies, and (2) the number of job copies, 3055 independent of whether the device can process multiple copies of the job or document without 3056 making multiple passes over the job or document data and independent of whether the output is 3057 collated or not. Thus the server/device computation is independent of the implementation." 3058 ::= { jmJobEntry 7 } 3059 3060 jmJobImpressionsCompleted OBJECT-TYPE 3061 SYNTAX Integer32(-2..2147483647) MAX-ACCESS read-only 3062 3063 STATUS current **DESCRIPTION** 3064 3065 "The current number of impressions completed for this job so far. For printing devices, the impressions completed includes interpreting, marking, and stacking the output. For other types 3066 of job services, the number of impressions completed includes the number of impressions 3067 3068 processed. 3069 3070 For implementations where multiple copies are produced by the interpreter with only a single 3071 pass over the data, the final value SHALL be equal to the value of the

pass over the data, the final value SHALL be equal to the value of the **jmJobImpressionsRequested** object. For implementations where multiple copies are produced by the interpreter by processing the data for each copy, the final value SHALL be a multiple of the value of the **jmJobImpressionsRequested** object.

NOTE - See the **impressionsCompletedCurrentCopy** and **pagesCompletedCurrentCopy** attributes for attributes that are reset on each document copy.

3072

3073 3074

3075 3076

3077

```
3079
                   NOTE - The jmJobImpressionsCompleted object can be used with the
3080
                   imJobImpressionsRequested object to provide an indication of the relative progress of the job,
3081
                   provided that the multiplicative factor is taken into account for some implementations of
3082
                   multiple copies."
             ::= { jmJobEntry 8 }
3083
3084
3085
        jmJobOwner OBJECT-TYPE
3086
             SYNTAX
                          JmJobStringTC(SIZE(0..63))
             MAX-ACCESS read-only
3087
3088
             STATUS
                          current
3089
             DESCRIPTION
                   "The coded character set name of the user that submitted the job. The method of assigning this
3090
3091
                   user name will be system and/or site specific but the method MUST insure that the name is
3092
                   unique to the network that is visible to the client and target device.
3093
3094
                   This value SHOULD be the authenticated name of the user submitting the job."
3095
             REFERENCE
3096
                   "See the OBJECT compliance macro for the minimum maximum length required for
3097
                   conformance."
             ::= { jmJobEntry 9 }
3098
3099
3100
3101
3102
3103
        -- The Attribute Group (MANDATORY)
3104
3105
        -- The jmAttributeGroup consists entirely of the jmAttributeTable.
3106
3107
        -- Implementation of the two objects in this group is MANDATORY.
3108
        -- See Section 3.1 entitled 'Conformance Considerations'.
3109
        -- An agent SHALL implement any attribute if (1) the server or device
3110
        -- supports the functionality represented by the attribute and (2) the
3111
        -- information is available to the agent.
3112
3113
        jmAttribute OBJECT IDENTIFIER ::= { jobmonMIBObjects 4 }
3114
3115
       imAttributeTable OBJECT-TYPE
3116
                          SEQUENCE OF JmAttributeEntry
             SYNTAX
3117
             MAX-ACCESS not-accessible
3118
             STATUS
                          current
3119
             DESCRIPTION
                   "The imAttributeTable SHALL contain attributes of the job and document(s) for each job in a
3120
3121
                   job set. Instead of allocating distinct objects for each attribute, each attribute is represented as a
3122
                   separate row in the jmAttributeTable."
3123
             REFERÊNCE
3124
                   "The MANDATORY-GROUP macro specifies that this group is MANDATORY. An agent
3125
                   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. "
3126
3127
             := \{ jmAttribute 1 \}
```

```
3128
3129
        jmAttributeEntry OBJECT-TYPE
3130
             SYNTAX
                          JmAttributeEntry
3131
             MAX-ACCESS not-accessible
3132
             STATUS
                          current
3133
             DESCRIPTION
                   "Attributes representing information about the job and document(s) or resources required and/or
3134
3135
                   consumed.
3136
3137
                   Each entry in the imAttributeTable is a per-job entry with an extra index for each type of
3138
                   attribute (jmAttributeTypeIndex) that a job can have and an additional index
                   (jmAttributeInstanceIndex) for those attributes that can have multiple instances per job. The
3139
3140
                   jmAttributeTypeIndex object SHALL contain an enum type that indicates the type of attribute
3141
                   (see the JmAttributeTypeTC textual-convention). The value of the attribute SHALL be
                   represented in either the jmAttributeValueAsInteger or jmAttributeValueAsOctets objects,
3142
3143
                   and/or both, as specified in the JmAttributeTypeTC textual-convention.
3144
                   The agent SHALL create rows in the jmAttributeTable as the server or device is able to
3145
3146
                   discover the attributes either from the job submission protocol itself or from the document PDL.
                   As the documents are interpreted, the interpreter MAY discover additional attributes and so the
3147
                   agent adds additional rows to this table. As the attributes that represent resources are actually
3148
3149
                   consumed, the usage counter contained in the jmAttributeValueAsInteger object is
3150
                   incremented according to the units indicated in the description of the JmAttributeTypeTC
3151
                   enum.
3152
3153
                   The agent SHALL maintain each row in the imJobTable for at least the minimum time after a
3154
                   job completes as specified by the jmGeneralAttributePersistence object.
3155
3156
                   Zero or more entries SHALL exist in this table for each job in a job set."
3157
             REFERENCE
3158
                   "See Section 3.3 entitled 'The Attribute Mechanism' for a description of the imAttributeTable."
3159
             INDEX { jmGeneralJobSetIndex, jmJobIndex, jmAttributeTypeIndex,
3160
             jmAttributeInstanceIndex }
3161
             ::= { jmAttributeTable 1 }
3162
3163
        JmAttributeEntry ::= SEQUENCE {
             jmAttributeTypeIndex
3164
                                                                   JmAttributeTypeTC,
3165
             jmAttributeInstanceIndex
                                                                   Integer32(1..32767),
3166
             jmAttributeValueAsInteger
                                                                   Integer32(-2..2147483647),
3167
             jmAttributeValueAsOctets
                                                                   OCTET STRING(SIZE(0..63))
3168
        }
3169
3170
        jmAttributeTypeIndex OBJECT-TYPE
3171
             SYNTAX
                          JmAttributeTypeTC
3172
             MAX-ACCESS not-accessible
3173
             STATUS
                          current
3174
             DESCRIPTION
3175
                   "The type of attribute that this row entry represents.
3176
```

3177 The type MAY identify information about the job or document(s) or MAY identify a resource 3178 required to process the job before the job start processing and/or consumed by the job as the job 3179 is processed. 3180 Examples of job attributes (i.e., apply to the job as a whole) that have only one instance per job 3181 3182 include: jobCopiesRequested(90), documentCopiesRequested(92), 3183 jobCopiesCompleted(91), documentCopiesCompleted(93), while examples of job attributes 3184 that may have more than one instance per job include: **documentFormatIndex(37)**, and 3185 documentFormat(38). 3186 3187 Examples of document attributes (one instance per document) include: **fileName(34)**, and 3188 documentName(35). 3189 3190 Examples of required and consumed resource attributes include: pagesRequested(130), mediumRequested(170), pagesCompleted(131), and mediumConsumed(171), respectively." 3191 3192 ::= { jmAttributeEntry 1 } 3193 3194 jmAttributeInstanceIndex OBJECT-TYPE 3195 SYNTAX Integer32(1..32767) 3196 MAX-ACCESS not-accessible 3197 STATUS current 3198 DESCRIPTION 3199 "A running 16-bit index of the attributes of the same type for each job. For those attributes with 3200 only a single instance per job, this index value SHALL be 1. For those attributes that are a single value per document, the index value SHALL be the document number, starting with 1 for 3201 3202 the first document in the job. Jobs with only a single document SHALL use the index value of 1. For those attributes that can have multiple values per job or per document, such as 3203 **documentFormatIndex(37)** or **documentFormat(38)**, the index SHALL be a running index 3204 3205 for the job as a whole, starting at 1." 3206 ::= { jmAttributeEntry 2 } 3207 3208 jmAttributeValueAsInteger OBJECT-TYPE SYNTAX 3209 Integer32(-2..2147483647) 3210 MAX-ACCESS read-only 3211 **STATUS** current **DESCRIPTION** 3212 3213 "The integer value of the attribute. The value of the attribute SHALL be represented as an 3214 integer if the enum description in the **JmAttributeTypeTC** textual-convention definition has the 3215 tag: 'INTEGER:'. 3216 3217 Depending on the enum definition, this object value MAY be an integer, a counter, an index, or an enum, depending on the **imAttributeTypeIndex** value. The units of this value are specified 3218 in the enum description. 3219 3220 3221 For those attributes that are accumulating job consumption as the job is processed as specified in 3222 the **JmAttributeTypeTC** textual-convention, SHALL contain the final value after the job

completes processing, i.e., this value SHALL indicate the total usage of this resource made by

the job.

3223

3224

3226 3227	A monitoring application is able to copy this value to a suitable longer term storage for later processing as part of an accounting system.
3228	
3229	Since the agent MAY add attributes representing resources to this table while the job is waiting
3230	to be processed or being processed, which can be a long time before any of the resources are
3231	actually used, the agent SHALL set the value of the jmAttributeValueAsInteger object to 0
3232	for resources that the job has not yet consumed.
3233	j j j j j j j j
3234	Attributes for which the concept of an integer value is meaningless, such as fileName(34) ,
3235	jobName , and processingMessage , do <i>not</i> have the 'INTEGER:' tag in the
3236	JmAttributeTypeTC definition and so an agent SHALL always return a value of '-1' to indicate
3237	'other' for the value of the jmAttributeValueAsInteger object for these attributes.
3238	other for the value of the Jim 2001 battle value 15 mese attributes.
3239	For attributes which do have the 'INTEGER:' tag in the JmAttributeTypeTC definition, if the
3240	integer value is not (yet) known, the agent either (1) SHALL not materialize the row in the
3241	jmAttributeTable until the value is known or (2) SHALL return a '-2' to represent an
3242	'unknown' counting integer value, a '0' to represent an 'unknown' index value, and a '2' to
3243	represent an 'unknown(2)' enum value."
3244	::= { jmAttributeEntry 3 }
3245	·· (J.··· 10010 000 2.1112 / 0)
3246	jmAttributeValueAsOctets OBJECT-TYPE
3247	SYNTAX OCTET STRING(SIZE(063))
3248	MAX-ACCESS read-only
3249	STATUS current
3250	DESCRIPTION
3251	"The octet string value of the attribute. The value of the attribute SHALL be represented as an
3252	OCTET STRING if the enum description in the JmAttributeTypeTC textual-convention
3253	definition has the tag: 'OCTETS:'.
3254	
3255	Depending on the enum definition, this object value MAY be a coded character set string (text),
3256	such as 'JmUTF8StringTC', or a binary octet string, such as 'DateAndTime'.
3257	count us come in opening is a woman's count as in its in i
3258	Attributes for which the concept of an octet string value is meaningless, such as
3259	pagesCompleted, do not have the tag 'OCTETS:' in the JmAttributeTypeTC definition and so
3260	the agent SHALL always return a zero length string for the value of the
3261	jmAttributeValueAsOctets object.
3262	g
3263	For attributes which do have the 'OCTETS:' tag in the JmAttributeTypeTC definition, if the
3264	OCTET STRING value is not (yet) known, the agent either SHALL not materialize the row in
3265	the jmAttributeTable until the value is known or SHALL return a zero-length string."
3266	::= { jmAttributeEntry 4 }
	• •

```
3268
       -- Notifications and Trapping
3269
       -- Reserved for the future
3270
3271
       jobmonMIBNotifications OBJECT IDENTIFIER ::= { jobmonMIB 2}
3272
3273
3274
3275
       -- Conformance Information
3276
3277
       jmMIBConformance OBJECT IDENTIFIER ::= { jobmonMIB 3 }
3278
3279
       -- compliance statements
3280
       imMIBCompliance MODULE-COMPLIANCE
3281
            STATUS current
3282
            DESCRIPTION
3283
                  "The compliance statement for agents that implement the
3284
                 job monitoring MIB."
3285
            MODULE -- this module
3286
            MANDATORY-GROUPS {
                 jmGeneralGroup, jmJobIDGroup, jmJobGroup, jmAttributeGroup }
3287
3288
3289
            OBJECT jmGeneralJobSetName
            SYNTAX JmUTF8StringTC (SIZE(0..8))
3290
3291
            DESCRIPTION
3292
                  "Only 8 octets maximum string length NEED be supported by the agent."
3293
            OBJECT jmJobOwner
SYNTAX JmJobStringTC (SIZE(0..16))
3294
3295
3296
            DESCRIPTION
3297
                  "Only 16 octets maximum string length NEED be supported by the agent."
3298
3299
       -- There are no CONDITIONALLY MANDATORY or OPTIONAL groups.
3300
3301
            ::= { imMIBConformance 1 }
3302
3303
       imMIBGroups
                       OBJECT IDENTIFIER ::= { jmMIBConformance 2 }
3304
       imGeneralGroup OBJECT-GROUP
3305
3306
            OBJECTS {
3307
                 jmGeneralNumberOfActiveJobs, jmGeneralOldestActiveJobIndex,
3308
                  jmGeneralNewestActiveJobIndex, jmGeneralJobPersistence,
3309
                 jmGeneralAttributePersistence, jmGeneralJobSetName}
3310
            STATUS current
3311
            DESCRIPTION
3312
                  "The general group."
3313
            ::= { jmMIBGroups 1 }
3314
       jmJobIDGroup OBJECT-GROUP
3315
3316
            OBJECTS {
```

```
3317
                 jmJobIDJobSetIndex, jmJobIDJobIndex }
            STATUS current
3318
3319
            DESCRIPTION
3320
                 "The job ID group."
3321
            ::= { jmMIBGroups 2 }
3322
3323
       imJobGroup OBJECT-GROUP
3324
            OBJÉCTS {
                 jmJobState, jmJobStateReasons1, jmNumberOfInterveningJobs,
3325
3326
                 jmJobKOctetsRequested, jmJobKOctetsProcessed, jmJobImpressionsRequested,
3327
                 jmJobImpressionsCompleted, jmJobOwner }
3328
            STATUS current
            DESCRIPTION
3329
3330
                 "The job group."
3331
            ::= { jmMIBGroups 3 }
3332
3333
       jmAttributeGroup OBJECT-GROUP
3334
            OBJECTS {
3335
                 jmAttributeValueAsInteger, jmAttributeValueAsOctets }
3336
            STATUS current
3337
            DESCRIPTION
3338
                 "The attribute group."
3339
            ::= { jmMIBGroups 4 }
3340
3341
3342
       END
```

3343 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.
- 3349 It is the purpose of the agent to map the particular implementation's job life cycle onto the one
- specified here. The agent MAY omit any states not implemented. Only the processing and
- completed states are required to be implemented by an agent. However, a conforming
- management application SHALL be prepared to accept any of the states in the job life cycle
- specified here, so that the management application can interoperate with any conforming agent.
- The job states are intended to be user visible. The agent SHALL make these states visible in the
- 3355 MIB, but only for the subset of job states that the implementation has. Some implementations
- MAY need to have sub-states of these user-visible states. The **jmJobStateReasons1** object and
- 3357 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
- implementations.
- 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
- 3365 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.
- 3369 The jmJobState object specifies the standard job states. The normal job state transitions
- are shown in the state transition diagram presented in Table 1.

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

- 3372 **Protocols**
- 3373 This appendix lists the job submission protocols that support the concept of a job
- 3374 submission ID and indicates the attribute used in that job submission protocol.

3375	6.1 Hewlett-Packard's Printer Job Language (PJL)		
3376 3377 3378 3379 3380 3381	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:		
3382 3383	SUBMISSIONID = "id string"		
3384 3385	Where the "id string" is a string and SHALL be enclosed in double quotes. The format is as described for the jmJobSubmissionID object.		
3386 3387	The entire PJL JOB command with the optional parameter would be of the form:		
3388 3389	@PJL JOB SUBMISSIONID = "id string"		
3390 3391 3392	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.		
3393 3394 3395 3396 3397	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.		
3398	6.2 ISO DPA		
3399 3400	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.		
3401	7. References		
3402 3403	[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>		
3404 3405	[GB2312] GB 2312-1980, "Chinese People's Republic of China (PRC) mixed one byte and two byte coded character set"		
3406	[hr-mib] P. Grillo, S. Waldbusser, "Host Resources MIB", RFC 1514, September 1993		
3407 3408	[iana] J. Reynolds, and J. Postel, "Assigned Numbers", STD 2, RFC 1700, ISI, October 1994.		

- 3409 [IANA-charsets] Coded Character Sets registered by IANA and assigned an enum value
- for use in the **CodedCharSet** textual convention imported from the Printer MIB. See
- 3411 ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets
- 3412 [iana-media-types] IANA Registration of MIME media types (MIME content
- 3413 types/subtypes). See ftp://ftp.isi.edu/in-notes/iana/assignments/
- 3414 [ISO 646] ISO/IEC 646:1991, "Information technology -- ISO 7-bit coded character set
- 3415 for information interchange", JTC1/SC2.
- 3416 [ISO 8859] ISO/IEC 8859-1:1987, "Information technology -- 8-bit single byte coded
- 3417 graphic character sets Part 1: Latin alphabet No. 1, JTC1/SC2."
- 3418 [ISO 2022] ISO/IEC 2022:1994 "Information technology -- Character code structure
- and extension techniques", JTC1/SC2.
- 3420 [ISO-10646] ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-
- Octet Coded Character Set (UCS) Part 1: Architecture and Basic Multilingual Plane,
- 3422 JTC1/SC2.
- 3423 [iso-dpa] ISO/IEC 10175 Document Printing Application (DPA). See
- 3424 ftp://ftp.pwg.org/pub/pwg/dpa/
- 3425 [ipp-model] Internet Printing Protocol (IPP), work in progress on the IETF standards
- track. See draft-ietf-ipp-model-01.txt. See also http://www.pwg.org/ipp/index.html
- 3427 [JIS X0208] JIS X0208-1990, "Japanese two byte coded character set."
- 3428 [mib-II] MIB-II, RFC 1213.
- 3429 [print-mib] The Printer MIB RFC 1759, proposed IETF standard. Also an Internet-
- Draft on the standards track as a draft standard: **draft-ietf-printmib-mib-info-02.txt**
- [reg-words] S. Bradner, "Keywords for use in RFCs to Indicate Requirement Levels",
- 3432 RFC 2119, March 1997.
- 3433 [rfc 1738] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource Locators
- 3434 (URL)", RFC 1738, December 1994.
- 3435 [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,
- 3437 1997", April 1997, RFC 2130.
- 3438 [SMIv2-TC] J. Case, et al. "Textual Conventions for Version 2 of the Simple Network
- Management Protocol (SNMPv2)", RFC 1903, January 1996.
- 3440 [tipsi] IEEE 1284.1, Transport-independent Printer System Interface (TIPSI).
- 3441 [URI-spec] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource Locators
- 3442 (URL)", RFC 1738, December, 1994.

3443 3444	[US-ASCII] Coded Character Set - 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.		
3445 3446	[UTF-8] F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.		
3447	8. Author's Addresses		
3448	Ron Bergman		
3449	Dataproducts Corp.		
3450	1757 Tapo Canyon Road		
3451	Simi Valley, CA 93063-3394		
3452			
3453	Phone: 805-578-4421		
3454	Fax: 805-578-4001		
3455	Email: rbergman@dpc.com		
3456			
3457			
3458	Tom Hastings		
3459	Xerox Corporation, ESAE-231		
3460	701 S. Aviation Blvd.		
3461	El Segundo, CA 90245		
3462			
3463	Phone: 310-333-6413		
3464	Fax: 310-333-5514		
3465	EMail: hastings@cp10.es.xerox.com		
3466			
3467			
3468	Scott A. Isaacson		
3469	Novell, Inc.		
3470	122 E 1700 S		
3471	Provo, UT 84606		
3472			
3473	Phone: 801-861-7366		
3474	Fax: 801-861-4025		
3475	EMail: scott_isaacson@novell.com		
3476			
3477			
3478	Harry Lewis		
3479	IBM Corporation		
3480	6300 Diagonal Hwy		
3481	Boulder, CO 80301		

```
3482
3483
             Phone: (303) 924-5337
             Fax:
3484
3485
             Email: harryl@us.ibm.com
3486
3487
3488
             Send comments to the printmib WG using the Job Monitoring Project (JMP)
3489
             Mailing List: jmp@pwg.org
3490
3491
             To learn how to subscribe, send email to: jmp-request@pwg.org
3492
3493
             For further information, access the PWG web page under "JMP":
3494
             http://www.pwg.org/
3495
3496
        Other Participants:
3497
             Chuck Adams - Tektronix
3498
             Jeff Barnett - IBM
             Keith Carter, IBM Corporation
3499
3500
             Jeff Copeland - QMS
             Andy Davidson - Tektronix
3501
3502
             Roger deBry - IBM
             Mabry Dozier - OMS
3503
3504
             Lee Ferrel - Canon
3505
             Steve Gebert - IBM
3506
             Robert Herriot - Sun Microsystems Inc.
             Shige Kanemitsu - Kyocera
3507
             David Kellerman - Northlake Software
3508
3509
             Rick Landau - Digital
3510
             Harry Lewis - IBM
             Pete Lova - HP
3511
             Ray Lutz - Cognisys
3512
3513
             Jay Martin - Underscore
             Mike MacKay, Novell, Inc.
3514
             Stan McConnell - Xerox
3515
             Carl-Uno Manros, Xerox, Corp.
3516
             Pat Nogay - IBM
3517
3518
             Bob Pentecost - HP
3519
             Rob Rhoads - Intel
             David Roach - Unisys
3520
3521
             Hiroyuki Sato - Canon
```

Job Monitoring MIB, V0.86

Sep 19, 1997

3522	Bob Setterbo - Adobe
3523	Gail Songer, EFI
3524	Mike Timperman - Lexmark
3525	Randy Turner - Sharp
3526	William Wagner - Digital Products
3527	Jim Walker - Dazel
3528	Chris Wellens - Interworking Labs
3529	Rob Whittle - Novell
3530	Don Wright - Lexmark
3531	Lloyd Young - Lexmark
3532	Atsushi Yuki - Kyocera
3533	Peter Zehler, Xerox, Corp.

9. INDEX

3534

3535

3536

3537

3538

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.

3330	chamb, and so start with any lower	cuse letter un	ia nave no speciai piena.	
		3572	jmGeneralNewestActiveJobIndex	68
3539	—C—	3573	jmGeneralNumberOfActiveJobs	68
3337	_	3574	jmGeneralOldestActiveJobIndex	68
3540	colorantConsumed		jmJobIDJobIndex	71
3541	colorantRequested	54 3576	jmJobIDJobSetIndex	71
		3577	jmJobImpressionsCompleted	75
3542	D	3578	jmJobImpressionsRequested	75
		3579	jmJobIndex	73
3543	deviceNameRequested		jmJobKOctetsProcessed	
3544	documentCopiesCompleted	51 3581	jmJobKOctetsRequested	74
3545	documentCopiesRequested	50 3582	jmJobOwner	
3546	documentFormat	48 3583	JmJobServiceTypesTC	57
3547	documentFormatIndex	47 3584	JmJobSourcePlatformTypeTC	
3548	documentName	47 3585	jmJobState	
		3586	jmJobStateReasons1	
3549	—F —	3587	JmJobStateReasons1TC	
		3588	JmJobStateReasons2TC	
3550	fileName		JmJobStateReasons3TC	
3551	finishing	49 3590	JmJobStateReasons4TC	
3552	fullColorImpressionsCompleted	52 3591	JmJobStateTC	
		3592	JmJobStringTC	
3553	—H—	3593	jmJobSubmissionID	
2554		3594	JmJobSubmissionTypeTC	
3554	highlight Color Impressions Completed	52 3393	JmMediumTypeTC	
		3596	jmNumberOfInterveningJobs	
3555	— I —	3597 3598	JmPrinterResolutionTC	
2556			JmPrintQualityTC	
3556	impressionsCompletedCurrentCopy	52 3399	JmTimeStampTC JmTonerEconomyTC	
3557 3558	impressionsInterpreted	51 3600	JmUTF8StringTC	
3559	impressionsSentToDevice	31 3601	jobAccountName	
3339	impressionsSpooled	3603	jobCodedCharSet	
	_	3604	jobComment	
3560	_J _	3605	jobCompletionTime	
3561	in A Ataih nto In aton o Ta Jon		jobCompletionTimejobCopiesCompleted	
3562	jmAttributeInstanceIndexjmAttributeTypeIndex	77 3607	jobCopiesRequested	
3563	JmAttributeTypeTC	77 3607	jobHold	
3564	jmAttributeTypeTCjmAttributeValueAsInteger	79 3609	jobHoldUntil	
3565	jmAttributeValueAsOctets	70 3610	jobKOctetsTransferred	
3566	JmBooleanTC	36 3611	jobName	
3567	JmFinishingTC	33 3612	jobOriginatingHost	
3568	jmGeneralAttributePersistence	69 3613	jobPriority	
3569	jmGeneralJobPersistence	69 3614	jobProcessAfterDateAndTime	
3570	jmGeneralJobSetIndex	67 3615	jobProcessingCPUTime	
3571	jmGeneralJobSetName	69 3616	jobServiceTypes	
3311	Jiii Circi ald ODDCH faille		Jane J Pas	

3617	jobSourceChannelIndex	pagesRequested52
3618	jobSourcePlatformType	physicalDevice47
3619	jobStartedBeingHeldTime55 3640	printerResolutionRequested50
3620	jobStartedProcessingTime 55 3641	printerResolutionUsed50
3621	jobStateReasons2	printQualityRequested49
3622	jobStateReasons3	printQualityUsed49
3623	jobStateReasons4	processingMessage44
3624	jobSubmissionTime	
3625	jobSubmissionToServerTime	_0_
3626	jobURI	—v —
	3646	queueNameRequested47
3627	—M —	
2620	3647	<u>—S</u> —
3628	mediumConsumed	
3629	mediumRequested	serverAssignedJobName
	3649	sheetsCompleted53
3630	_N 3650	sheetsCompletedCurrentCopy53
2621	3651	sheetsRequested
3631	numberOfDocuments	sides
	3653	submittingApplicationName46
3632	_O 3654	submittingServerName46
3633	other	T.
3634	outputBin	—T—
3034	3656	tonerDensityRequested50
2625	2757	tonerDensityUsed
3635	—P— 305 / 3658	tonerEcomonyRequested50
3636	pagesCompleted	tonerEcomonyUsed
3637	pagesCompletedCurrentCopy53	50112120111011j Oseanii 11111111111111111111111111111111111
	F8	
3660		