

1 Subject: IPP and JMP agreements on job-state and job-state-
 2 reasons
 3 From: Tom Hastings
 4 Date: 6/6/97
 5 File: jobstatr.doc (with revision marks) jobstate.doc (without
 6 revisions)
 7

8 This document is the final updated IPP "job-state" and "job-
 9 state-reasons" attributes and the corresponding JMP jmJobState
 10 and jmJobStateReasons1 objects
 11 from the IPP telecon, of Friday, June 6.
 12

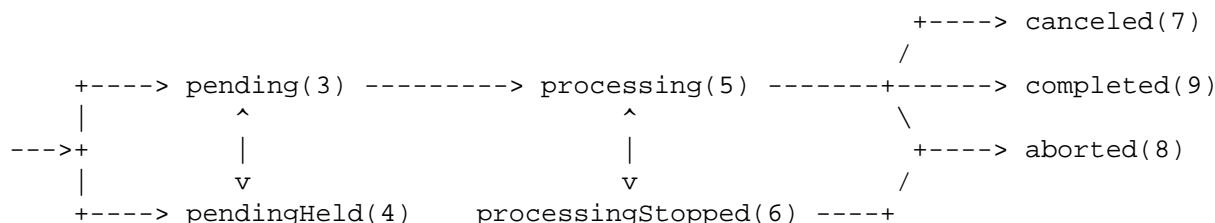
13 1. We agreed to the following states for IPP and JMP:

15 IPP	JMP
16	other(1)
17 'unknown'	unknown(2)
18 'pending'	pending(3)
19 'pending-held'	pendingHeld(4)
20 'processing'	processing(5)
21 'processing-stopped'	proessingStopped(6)
22 'canceled'	canceled(7)
23 'aborted'	aborted(8)
24 'completed'	completed(9)

25
 26
 27 2. We agreed on the simplified job state transition diagram
 28 sentence at the end explaining the transitions into the canceled
 29 state that are not shown:
 30

31 For JMP:

32 The following figure shows the normal job state transitions:



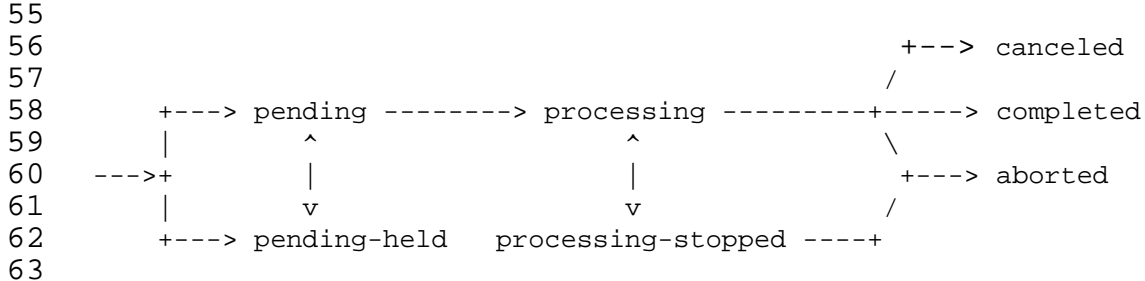
42 **Figure 1 - Normal Job State Transitions**

43
 44 Normally a job progresses only from left to right. Other state
 45 transitions are unlikely, but are not forbidden. Not shown are the
 46 transitions to the **canceled** state from the **pending**, **pendingHeld**,
 47 **processing**, and **processingStopped** states.
 48

49 Jobs in the **pending**, **processing**, and **processingStopped** states are
50 called 'active', while jobs in the **pendingHeld**, **canceled**, **aborted**,
51 and **completed** states are called 'in-active'."
52

53 For IPP:

54 The following figure shows the normal job state transitions:



64 **Figure 2 - Normal Job State Transitions**

65
66 Normally a job progresses only from left to right. Other state
67 transitions are unlikely, but are not forbidden. Not shown are the
68 transitions to the 'canceled' state from the 'pending', 'pending-
69 held', 'processing', and 'processing-stopped' states."
70

71
72 3. Conformance: we agreed that no job states are MANDATORY.
73 For JMP the sentence proposed by Ron will be added:

74
75 All possible enums for this object SHALL be reported if
76 implemented by the device and available to the agent.

77
78 The corresponding sentence for IPP will be:

79
80 All possible job states SHALL be returned by the Printer object
81 if implemented by the output device and available to the Printer
82 object implementation.

83
84
85 4. We agreed to not specify which job state reasons go with which
86 states. An implementation can use the reasons with any state for
87 which the reason makes sense.

88
89 The following JMP sentence will be included in the
90 JmJobStateReasons1TC textual-convention:

91
92 The following standard values are defined (in hexadecimal) as
93 *powers of two*, since multiple values may be used at the same
94 time. These values MAY be used with any job state for which the
95 reason makes sense.

96
97 The corresponding IPP "job-state-reasons" attribute sentence is:
98

99 The following standard values are defined and MAY be used with
100 any job state for which the reason makes sense.
101
102
103 5. We agreed that job-state-reasons are all OPTIONAL.
104
105 The following JMP sentence will be included in the
106 JmJobStateReasons1TC textual-convention:
107
108 Implementation of these values is OPTIONAL, i.e., an agent NEED
109 NOT implement them, even if the device supports the functionality
110 represented by the reason and is available to the agent.
111
112 The corresponding IPP "job-state-reasons" attribute sentence is:
113
114 Implementation of these values is OPTIONAL, i.e., the Printer
115 object NEED NOT return them, even if the output device supports
116 the functionality represented by the reason and is available to
117 the Printer object software.
118
119
120 The following are the changes from the previous IPP and JMP
121 published Internet-Draft specifications:
122
123 1. In JMP remove the 'printing' state.
124
125 2. Add the 'pending-held' and 'processing-stopped' states to IPP.
126
127 3. Rename the JMP 'held' state to 'pendingHeld' and rename the
128 JMP state 'needsAttention' to 'processingStopped'.
129
130 4. In both IPP and JMP add the 'aborted' state and make it a
131 final state.
132
133 5. In both IPP and JMP remove the 'aborted-by-system' job-state-
134 reason,
135 since the new 'aborted' state says it all.
136
137 6. In IPP replace the 'terminating' state with the 'canceled' and
138 'aborted'
139 states and make 'canceled' and 'aborted' final states, like the
140 JMP 'canceled' and 'completed' states.
141
142 7. Since the **pendingHeld** state has been added, JMP no longer
143 needs a generic jobHeld job state reason.
144
145 8. No job states are MANDATORY in IPP and JMP. However, those
146 that are implemented in the device and are available to the
147 Printer/agent shall be returned.
148

149 9. Job state reasons are OPTIONAL in IPP and JMP.
150
151

152
153 Changes to IPP "job-state" attribute
154 -----
155
156 The following text is copied from the IPP Internet-Draft, 6/3/97.
157 The revision marks show the changes to reflect the above
158 agreements to the job-state attribute.
159

160 6.3.2.5 job-state (type1 keyword)

161 This attribute identifies the current state of the job. Even
162 though the IPP protocol defines eight values for job states,
163 Printers SHALL only implement those states which are appropriate
164 for the particular implementation. In other words, all possible
165 job states SHALL be returned by the Printer object if implemented
166 by the output device and available to the Printer object
167 implementation.

168 The final value for this attribute SHALL be one of: 'completed',
169 'canceled', or 'aborted' before the Printer removes the job
170 altogether. The length of time that jobs remain in the
171 'canceled', 'aborted', and 'completed' states depends on
172 implementation.

173 Standard values are:

174 'unknown': The job state is not known, or its state is
175 indeterminate.

176 'pending': The job is a candidate to start processing, but is
177 not yet processing.

178 'pending-held': The job is not a candidate for processing for
179 any number of reasons but will return to the 'pending' state
180 as soon as the reasons are no longer present. The job's
181 "job-state-reason" attribute SHALL indicate why the job is
182 no longer a candidate for processing.

183 'processing ': Either:

- 184 1. the job is using, or is attempting to use, one or more
185 document transforms which include (1) purely software
186 processes that are interpreting a PDL, and (2) hardware
187 devices that are interpreting a PDL, making marks on a
188 medium, and/or performing finishing, such as stapling OR
- 189 2. the server has made the job ready for printing, but the
190 output device is not yet printing it, either because the
191 job hasn't reached the output device or because the job
192 is queued in the output device or some other spooler,
193 awaiting the output device to print it.

194 When the job is in the 'processing' state, the entire job
195 state includes the detailed status represented in the
196 printer's "printer-state", "printer-state-reasons", and
197 "printer-state-message" attributes.

198 Implementations MAY include additional values in the job's
199 "job-state-reasons" attribute to indicate the progress of
200 the job, such as adding the 'job-printing' value to indicate
201 when the output device is actually making marks on paper.
202 Most implementations won't bother with this nuance.

203 'processing-stopped': The job has stopped while processing
204 for any number of reasons and will return to the
205 'processing' state as soon as the reasons are no longer
206 present.

207 The job's "job-state-reason" attribute MAY indicate why the
208 job has stopped processing. For example, if the output
209 device is stopped, the 'printer-stopped' value MAY be
210 included in the job's "job-state-reasons" attribute. For
211 example, if the output device is stopped, the 'printer-
212 stopped' value MAY be included in the job's "job-state-
213 reasons" attribute.

214 NOTE - When an output device is stopped, the device usually
215 indicates its condition in human readable form locally at
216 the device. A client can obtain more complete device status
217 remotely by querying the printer's "printer-state",
218 "printer-state-reasons" and "printer-state-message"
219 attributes.

220 'canceled': The job has been canceled by a Cancel-Job
221 operation and is either (1) in the process of terminating or
222 (2) has completed terminating. The job's "job-state-
223 reasons" attribute SHOULD contain either the 'canceled-by-
224 user' or 'canceled-by-operator' value.

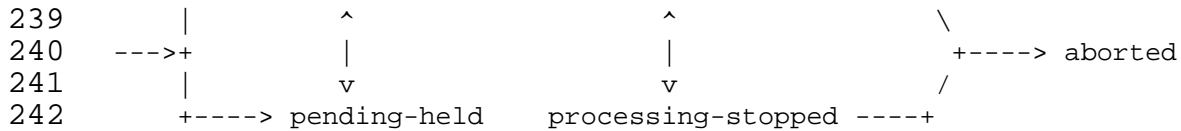
225 'aborted': The job has been aborted by the system, usually
226 while the job was in the 'processing' or 'processing-
227 stopped' state.

228 'completed': The job has completed successfully or with
229 warnings or errors after processing and all of the job media
230 sheets have been successfully stacked in the appropriate
231 output bin(s). The job's "job-state-reasons" attribute
232 SHOULD contain one of: 'completed-successfully', 'completed-
233 with-warnings', or 'completed-with-errors' values.

234

235 The following figure shows the normal job state transitions.

236
237
238 +----> pending -----> processing -----+----> completed
+----> canceled
/



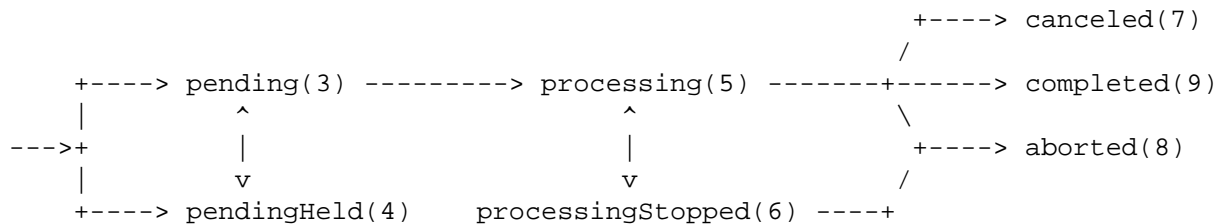
243 **Figure 3 - Normal Job State Transitions**

244 Normally a job progresses from left to right. Other state
245 transitions are unlikely, but are not forbidden. Not shown are
246 the transitions to the 'canceled' state from the 'pending',
247 'pending-held', 'processing', and 'processing-stopped' states.

248 Changes to JMP **jmJobState** object and **JmJobStateTC** description
 249 -----
 250 The following shows the changes for the JMP **jmJobState** object and
 251 the corresponding **JmJobStateTC** textual-convention for the Job
 252 Monitoring MIB (there is no longer a **jobState** attribute):

253
 254 **jmJobState** OBJECT-TYPE
 255 SYNTAX **JmJobStateTC** -- See page 8
 256 MAX-ACCESS read-only
 257 STATUS current
 258 DESCRIPTION
 259 "The current state of the job (**pending**, **processing**,
 260 **completed**, etc.). Even though the **JmJobStateTC** textual-
 261 convention defines nine values for job states, agents
 262 SHALL only implement those states which are appropriate
 263 for the particular implementation. In other words, all
 264 possible enums for this object SHALL be reported if
 265 implemented by the device and available to the agent.
 266 However, management applications SHALL be prepared to
 267 receive all the standard job states.
 268
 269 The final value for this object SHALL be one of:
 270 **completed**, **canceled**, or **aborted**. The minimum length of
 271 time that the agent SHALL keep a job in the **completed**,
 272 **canceled**, or **aborted** state before removing the job from
 273 the **jmJobIDTable** and **jmJobTable** is specified by the value
 274 of the **jmGeneralJobPersistence** object."
 275 ::= { jmJobEntry 1 }

276
 277
 278 **JmJobStateTC** ::= TEXTUAL-CONVENTION
 279 STATUS current
 280 DESCRIPTION
 281 "The current state of the job (**pending**, **processing**,
 282 **completed**, etc.).
 283
 284 The following figure shows the normal job state
 285 transitions:



295 **Figure 4 - Normal Job State Transitions**

296


```

297 Normally a job progresses from left to right. Other
298 state transitions are unlikely, but are not forbidden.
299 Not shown are the transitions to the canceled state from
300 the pending, pendingHeld, processing, and
301 processingStopped states.
302
303 Jobs in the pending, processing, and processingStopped
304 states are called 'active', while jobs in the
305 pendingHeld, canceled, aborted, and completed are called
306 'in-active'."
307
308 -- This is a type 2 enumeration. See Section Error! Reference source not found. on page Error!
309 Bookmark not defined..
310 SYNTAX INTEGER {
    other(1),
        -- The job state is not one of the defined states.

    unknown(2),
        -- The job state is not known, or its state is
        -- indeterminate.

    pending(3),
        -- The job is a candidate to start processing, but is
        -- not yet processing.

    pendingHeld(4),
        -- The job is not a candidate for processing for any
        -- number of reasons but will return to the pending
        -- state as soon as the reasons are no longer
        -- present. The job's jmJobStateReasons1 object
        -- and/or jobStateReasonsn ( $n=2..4$ ) attributes SHALL
        -- indicate why the job is no longer a candidate for
        -- processing. The reasons are represented as bits
        -- in the jobStateReasons1 object and/or
        -- jobStateReasonsn ( $n=2..4$ ) attributes. See the
        -- JmJobStateReasonsnTC ( $n=1..4$ ) textual convention
        -- on page (19) for the specification of each reason.

    processing(5),
        -- Either:
        --
        -- 1. The job is using, or is attempting to use, one
        -- or more document transforms which include (1)
        -- purely software processes that are interpreting a
        -- PDL, and (2) hardware devices that are
        -- interpreting a PDL, making marks on a medium,
        -- and/or performing finishing, such as stapling,
        -- etc.
        --
        -- OR
        --
        -- 2. (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 paper.

processingStopped(6),

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

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

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

canceled(7),

-- A client has canceled the job and the job is
-- either: (1) in the process of being terminated by
-- the server or device or (2) has completed
-- terminating. The job's **jobStateReasons1** attribute
-- SHOULD contain either the **canceledByUser** or
-- **canceledByOperator** value.

aborted(8)

-- The job has been aborted by the system, usually
-- while the job was in the **processing** or
-- **processingStopped** state.

completed(9)

- The job has completed successfully or with
- warnings or errors after processing and all of the
- media have been successfully stacked in the
- appropriate output bin(s). The job's
- **jobStateReasons1** attribute SHOULD contain one of:
- **completedSuccessfully, completedWithWarnings,** or
- **completedWithErrors** values.

311 }
312

313 Here is the updated Job Monitoring MIB appendix:
314

315 **Appendix A - Instrumenting the Job Life Cycle**

316 **Instrumenting the Job Life Cycle**

317 The job object has well-defined states and client operations that
318 affect the transition between the job states. Internal server
319 and device actions also affect the transitions of the job between
320 the job states. These states and transitions are referred to as
321 the job's *life cycle*.

322 Not all implementations of job submission protocols have all of
323 the states of the job model specified here. The job model
324 specified here is intended to be a superset of most
325 implementations. It is the purpose of the agent to map the
326 particular implementation's job life cycle onto the one specified
327 here. The agent may omit any states not implemented. Only the
328 **processing, canceled, aborted, and completed** states are required
329 to be implemented by an agent. However, a conforming management
330 application shall be prepared to accept any of the states in the
331 job life cycle specified here, so that the management application
332 can interoperate with any conforming agent.

333 The job states are intended to be the user visible. The agent
334 shall make these states visible in the MIB, but only for the
335 subset of job states that the implementation has.
336 Implementations may need to have sub-states of these user-visible
337 states. Such implementation is *not* specified in this model, is
338 not supported by this Job Monitoring MIB, and will vary from
339 implementation to implementation. In some implementations the
340 **jmJobStateReasons1** object and the **jobStateReasonsn** ($n=2..4$)
341 attributes may represent some or all of the sub-states of the
342 jobs.

343 One of the purposes of the job life cycle is to specify what is
344 invariant from implementation to implementation as far as the MIB
345 specification and the management application is concerned.
346 Therefore, job states are all intended to last a user-visible
347 length of time in most implementations. However, some jobs may
348 pass through some states in zero time in some situations and/or
349 in some implementations.

350 The job model does not specify how accounting and auditing is
351 implemented, except to assume that accounting and auditing logs
352 are separate from the job life cycle and last longer than job
353 entries in the MIB. Jobs in the **completed, aborted, or canceled**
354 states are not logs, since jobs in these states are accessible
355 via SNMP protocol operations and shall be removed from the Job
356 Monitoring MIB tables after a site-settable or implementation-
357 defined period of time. An accounting application may copy
358 accounting information incrementally to an accounting logs as a

359 job processes, or may be copied while the job is in the **canceled,**
360 **aborted,** or **completed** states, depending on implementation. The
361 same is true for auditing logs.

362 The `jmJobState` object specifies the standard job states. The
363 normal job state transitions are shown in the state transition
364 diagram presented in Table 1.

365

366

367 **Revised IPP job-state-reasons and JMP jobStateReasons1**

368 The IPP **job-state-reasons** attribute can contain multiple
 369 keywords, while the JMP **jobStateReasons1** attribute is bit
 370 encoded. The Job Monitoring MIB contains a superset of the IPP
 371 values[8] for the IPP "job-state-reasons" attribute, since the
 372 Job Monitoring MIB is intended to cover other job submission
 373 protocols as well. Also some of the names of the reasons have
 374 been changed from 'printer' to 'device', since the Job Monitoring
 375 MIB is intended to cover additional types of devices, including
 376 input devices, such as scanners.

377 The comparison of the IPP "job-state-reasons" attribute and the
 378 JMP **jmJobStateReasons1** object after the IPP/JMP telecon,
 379 Wednesday, 5/28/97, is as follows:

IPP job-state-reasons values	JMP jobStateReasons1 values	Notes
-	other	
-	unknown	
none	-	IPP is trying to avoid allowing attributes with no values. No need for JMP to have a none value, since JMP reasons can have no bits on.
job-incoming	jobIncoming	Covers both the cases of document transfers in progress and additional SendDocument requests needed since the job isn't closed yet.
job-outgoing	jobOutgoing	Covers sending to the output device and queued in the output device.

job-printing	jobPrinting	
printer-stopped	deviceStopped	JMP specifies device, not printer, so that it can be used for additional services besides printing.
printer-stopped-partly	deviceStoppedPartly	
	jobHoldSpecified	JMP has to cover job submission protocols (DPA, VMS, Printxchange) where clients can submit a job and explicitly hold it. Such protocols have operations to let the client release the job later.
job-hold-until-specified	jobHoldUntilSpecified	JMP needs to indicate which reasons prevent a job from being processed.
	jobProcessAfterSpecified	JMP needs to cover (DPA, VMS, Printxchange). Also JMP needs to indicate which reasons prevent a job from being processed.
resources-are-not-ready	resourcesAreNotReady	Renamed so that it can be used in any job state.
job-canceled-by-user	jobCanceledByUser	
job-canceled-by-operator	jobCanceledByOperator	
		Don't need an 'aborted-by-system' reason, since there

		is an entire new job state instead: aborted .
job-completed-successfully	jobCompletedSuccessfully	
job-completed-with-warnings	jobCompletedWithWarnings	
job-completed-with-errors	jobCompletedWithErrors	
logfile-pending	logfilePending	
logfile-transferring	logfileTransferring	
		Covered by jobIncoming
	jobPaused	JMP needs to cover other job submission protocols that have these states, such as DPA
	jobInterrupted	ditto
	jobRetained	ditto

380

381 Now both IPP "job-state-reasons" attribute and JMP

382 **jmJobStateReasons1** object are Mandatory.

383 Another difference is that IPP **job-state-reasons** shall have at

384 least one value, while the JMP **jmJobStateReasons1** object may have

385 no values. This difference is OK, since IPP will have the 'none'

386 value, and the JMP agent will have all job state reasons bits

387 turned off. IPP is striving to eliminate attributes that have no

388 values, in order to avoid ambiguities and inter-working problems.

389 There is no such problem with no bits being on in the

390 **jmJobStateReasons1** object in JMP.

391

392 The following is the modified IPP "job-state-reasons" attribute
393 from the Internet-Draft of 6/3/97 following the 6/6/97
394 agreements:

395 6.3.2.6 job-state-reasons (1setOf type2 keyword)

396 This attribute provides additional information about the job's
397 current state, i.e., information that augments the value of the
398 job's "job-state" attribute.

399 Implementation of these values is OPTIONAL, i.e., a Printer NEED
400 NOT implement them, even if (1) the output device supports the
401 functionality represented by the reason and (2) is available to
402 the Printer object implementation. These values MAY be used with
403 any job state or states for which the reason makes sense.
404 Furthermore, when implemented, the Printer SHALL return these
405 values when the reason applies and SHALL NOT return them when the
406 reason no longer applies whether the value of the job's "job-
407 state" attribute changed or not. When the job does not have any
408 reasons for being in its current state, the Printer shall set the
409 value of the job's "job-state-reasons" attribute to 'none'.

410 NOTE - While values cannot be added to the 'job-state' attribute
411 without impacting deployed clients that take actions upon
412 receiving "job-state" values, it is the intent that additional
413 "job-state-reasons" values can be defined and registered without
414 impacting such deployed clients. In other words, the "job-state-
415 reasons" attribute is intended to be extensible.

416 The following standard values are defined:

417 NOTE - For easy of understanding the order of the reasons is
418 presented in the order in which the reason is most likely to
419 occur:

420 'none': There are no reasons for the job's current state.

421 'job-incoming': The CreateJob operation has been accepted by
422 the Printer, but the Printer is expecting additional
423 SendDocument operations and/or is accessing/accepting
424 document data.

425 'job-outgoing': The Printer is transmitting the job to the
426 output device.

427 'job-hold-until-specified': The value of the job's "job-hold-
428 until" attribute specifies a time period that is still in
429 the future. The job SHALL NOT be a candidate for processing
430 until this reason is removed and there are no other reasons
431 to hold the job.

432 'resources-are-not-ready': At least one of the resources
433 needed by the job, such as media, fonts, resource objects,
434 etc., is not ready on any of the physical printer's for
435 which the job is a candidate. This condition MAY be
436 detected when the job is accepted, or subsequently while the
437 job is pending or processing, depending on implementation.

438 'printer-stopped-partly': The value of the Printer's
439 "printer-state-reasons" attribute contains the value
440 'stopped-partly'.

441 'printer-stopped': The value of the Printer's "printer-state"
442 attribute is 'stopped'.

443 'job-printing': The output device is marking media. This
444 value is useful for Printers which spend a great deal of
445 time processing when no marking is happening and then want
446 to show that marking is now happening.

447 'job-cancelled-by-user': The job was cancelled by the user
448 using the CancelJob request, i.e., by a user whose name is
449 the same as the value of the job's "job-originating-user"
450 attribute.

451 'job-cancelled-by-operator': The job was cancelled by the
452 operator using the CancelJob request, i.e., by a user whose
453 name is different than the value of the job's "job-
454 originating-user" attribute.

455 'job-completed-successfully': The job completed successfully.

456 'job-completed-with-warnings': The job completed with
457 warnings.

458 'job-completed-with-errors': The job completed with errors
459 (and possibly warnings too).

460 'logfile-pending ': The job's logfile is pending file
461 transfer.

462 'logfile-transferring': The job's logfile is being
463 transferred.

464 Here is the Job Monitoring MIB definition of the **jobStateReasons1**
465 object and the defined values updated to reflect the agreements
466 on IPP and with additional reasons that are needed by other job
467 submission protocols, even though they are not needed by IPP:

468 **jmJobStateReasons1** OBJECT-TYPE

469 SYNTAX **JmJobStateReasons1TC** -- See page 19

470 MAX-ACCESS read-only

471 STATUS current

472 DESCRIPTION

473 "Additional information about the job's current state,
474 i.e., information that augments the value of the job's
475 **jmJobState** object.

476
477 NOTE - The **jobStateReasonsn** ($n=2..4$) attributes (see page
478 **Error! Bookmark not defined.**) provide further additional
479 information about the job's current state.

480
481 Implementation of these values is OPTIONAL, i.e., an
482 agent NEED NOT implement them, even if (1) the device
483 supports the functionality represented by the reason and
484 (2) is available to the agent. These values MAY be used
485 with any job state or states for which the reason makes
486 sense. Furthermore, when implemented, the agent SHALL
487 return these values when the reason applies and SHALL NOT
488 return them when the reason no longer applies whether the
489 value of the job's **jmJobState** object changed or not.
490 When the job does not have any reasons for being in its
491 current state, the agent SHALL set the value of the
492 **jmJobStateReasons1** object and **jobStateReasonsn** attributes
493 to 0.

494
495 NOTE - While values cannot be added to the **jmJobState**
496 object without impacting deployed clients that take
497 actions upon receiving **jmJobState** values, it is the
498 intent that additional **JmJobStateReasonsnTC** enums can be
499 defined and registered without impacting such deployed
500 clients. In other words, the **jmJobStateReasons1** object
501 and **jobStateReasonsn** attributes are intended to be
502 extensible."

503 ::= { jmJobEntry 2 }

504
505

506 **JmJobStateReasons1TC** ::= TEXTUAL-CONVENTION

507 STATUS current

508 DESCRIPTION

509 "This textual-convention is used with the
510 **jmJobStateReasons1** object to provides additional
511 information regarding the **jmJobState** object values.

512
513 The following standard values are defined (in
514 hexadecimal) as *powers of two*, since multiple values MAY
515 be used at the same time.

516

517 NOTE - The Job Monitoring MIB contains a superset of the
518 IPP values[3] for the IPP 'job-state-reasons' attribute,
519 since the Job Monitoring MIB is intended to cover other
520 job submission protocols as well. Also some of the names
521 of the reasons have been changed from 'printer' to
522 'device', since the Job Monitoring MIB is intended to
523 cover additional types of devices, including input
524 devices, such as scanners.
525

526 NOTE - For easy of understanding the order of the reasons
527 is presented in the order in which the reason is most
528 likely to occur.
529

530 **other** **0x1**
531 The job state reason is not one of the standardized or
532 registered reasons.
533

534 **unknown** **0x2**
535 The job state reason is not known to the agent or is
536 indeterminent.
537

538 **jobIncoming** **0x4**
539 The job has been accepted by the server or device, but
540 the server or device is expected (1) additional
541 operations to finish creating the job and/or (2) is
542 accessing/accepting document data.
543

544 **jobOutgoing** **0x8**
545 Configuration 2 only: The server is transmitting the job
546 to the device.
547

548 **jobHoldSpecified** **0x10**
549 The value of the job's **Error! Reference source not found.**
550 attribute (see page **Error! Bookmark not defined.**) is
551 **TRUE**, either set when the job was created or subsequently
552 by an explicit modify job operation. The job SHALL NOT
553 be a candidate for processing until this reason is
554 removed and there are no other reasons to hold the job.
555

556 **jobHoldUntilSpecified** **0x20**
557 The value of the job's **Error! Reference source not found.**
558 (see page **Error! Bookmark not defined.**) attribute
559 specifies a time period that is still in the future,
560 either set when the job was created or subsequently by an
561 explicit modify job operation. The job SHALL NOT be a
562 candidate for processing until this reason is removed and
563 there are no other reasons to hold the job.
564

565 **jobProcessAfterSpecified** **0x40**
566 The value of the job's **Error! Reference source not found.**
567 (see page **Error! Bookmark not defined.**) attribute
568 specifies a time that is still in the future, either set
569 when the job was created or subsequently by an explicit
570 modify job operation. The job SHALL NOT be a candidate

571 for processing until this reason is removed and there are
572 no other reasons to hold the job.
573

574 **resourcesAreNotReady** **0x80**
575 At least one of the resources needed by the job, such as
576 media, fonts, resource objects, etc., is not ready on any
577 of the physical devices for which the job is a candidate.
578 This condition MAY be detected when the job is accepted,
579 or subsequently while the job is pending or processing,
580 depending on implementation.
581

582 **deviceStoppedPartly** **0x100**
583 One or more, but not all, of the devices to which the job
584 is assigned are stopped. If all of the devices are
585 stopped (or the only device is stopped), the
586 **deviceStopped** reason SHALL be used.
587

588 **deviceStopped** **0x200**
589 The device(s) to which the job is assigned is (are all)
590 stopped.
591

592 **jobPrinting** **0x400**
593 The output device is marking media. This attribute is
594 useful for servers and output devices which spend a great
595 deal of time processing when no marking is happening and
596 then want to show that marking is now happening.
597

598 **jobCanceledByUser** **0x800**
599 The job was canceled by the user, i.e., by a user whose
600 name is the same as the value of the job's **jobOwner**
601 attribute.
602

603 **jobCanceledByOperator** **0x1000**
604 The job was canceled by the operator, i.e., by a user
605 whose name is different than the value of the job's
606 **jobOwner** attribute.
607

608 **abortedBySystem** **0x2000**
609 The job was aborted by the system. NOTE - this reason is
610 needed only when the job is not placed in the **aborted** job
611 state.
612

613 **jobCompletedSuccessfully** **0x4000**
614 The job completed successfully.
615

616 **jobCompletedWithWarnings** **0x8000**
617 The job completed with warnings.
618

619 **jobCompletedWithErrors** **0x10000**
620 The job completed with errors (and possibly warnings
621 too).
622

623 The following additional job state reasons have been added to
624 represent job states that are in ISO DPA[2] and other job
625 submission protocols:
626

627 **jobPaused** **0x20000**

628 The job has been indefinitely suspended by a client
629 issuing an operation to suspend the job so that other
630 jobs may proceed using the same devices. The client MAY
631 issue an operation to resume the paused job at any time,
632 in which case the agent SHALL remove the **jobPaused** values
633 from the job's **jmJobStateReasons1** object and the job is
634 eventually resumed at or near the point where the job was
635 paused.
636

637 **jobInterrupted** **0x40000**

638 The job has been interrupted while processing by a client
639 issuing an operation that specifies another job to be run
640 instead of the current job. The server or device will
641 automatically resume the interrupted job when the
642 interrupting job completes.
643

644 **jobRetained** **0x80000**

645 The job is being retained by the server or device with
646 all of the job's document data (and submitted resources,
647 such as fonts, logos, and forms, if any). Thus a client
648 could issue an operation to resubmit the job (or a copy
649 of the job). When a client could no longer resubmit the
650 job, such as after the document data has been discarded,
651 the agent SHALL remove the **jobRetained** value from the
652 **jmJobStateReasons1** object.
653

654 These bit definitions are the equivalent of a type 2 enum
655 except that combinations of bits may be used together. See
656 section **Error! Reference source not found.** on page **Error!**
657 **Bookmark not defined..** The remaining bits are reserved for
658 future standardization and/or registration."
659

660 These bit definitions are the equivalent of a type 2 enum
661 except that combinations of bits may be used together. See
662 section **Error! Reference source not found.** on page **Error!**
663 **Bookmark not defined..**"
664

665 SYNTAX **INTEGER(0..2147483647)** -- 31 bits, all but sign bit
666
667
668

669