

1 Subject: IPP and JMP agreements on job-state and job-state-
 2 reasons
 3 From: Tom Hastings
 4 Date: 65/625/97
 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.

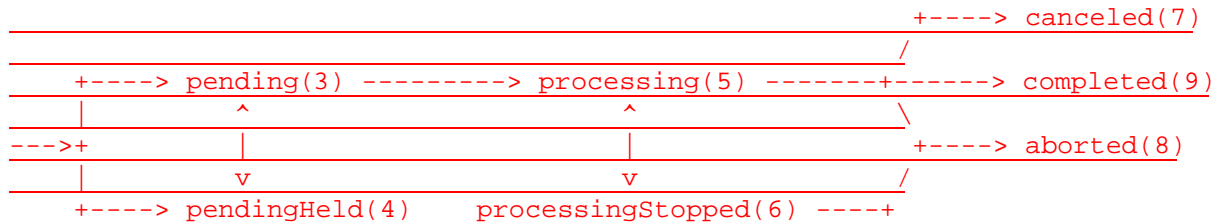
12
 13 1. We agreed to the following states for IPP and JMP:

IPP	JMP
	other(1)
'unknown'	unknown(2)
'pending'	pending(3)
'pending-held'	pendingHeld(4)
'processing'	processing(5)
'processing-stopped'	proessingStopped(6)
'canceled'	canceled(7)
'aborted'	aborted(8)
'completed'	completed(9)

24
 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:



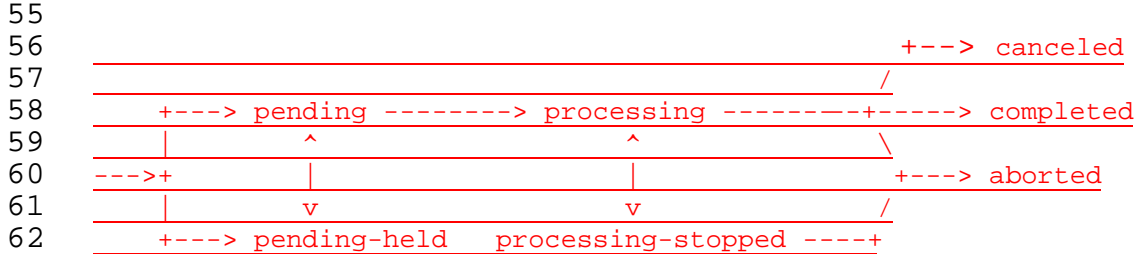
41
 42 Figure 1 - Normal Job State Transitions

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
121 The following are the changes from the previous IPP and JMP
122 published Internet-Draft specifications:

123
124 1. In JMP remove the ~~'held', 'printing', and 'needsAttention'~~
125 ~~states.~~

126
127 2. Add the 'pending-held' and 'processing-stopped' states to IPP.

128
129 3. Rename the JMP 'held' state to 'pendingHeld' and rename the
130 JMP state 'needsAttention' to 'processingStopped'.

131
132 42. In both IPP and JMP add the 'aborted' state and make it a
133 final state.

134
135 53. In both IPP and JMP remove the 'aborted-by-system' job-state-
136 reason,
137 since the new 'aborted' state says it all.

138
139 64. In IPP replace the 'terminating' state with the 'canceled'
140 and 'aborted'
141 states and make 'canceled' and 'aborted' final states, like the
142 JMP 'canceled' and 'completed' states.

143
144 7. Since the **pendingHeld** state has been added, JMP no longer
145 needs a generic jobHeld job state reason.

146

147 8. No job states are MANDATORY in IPP and JMP. However, those
148 that are implemented in the device and are available to the
149 Printer/agent shall be returned.

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

152
153

154
155 Changes to IPP "job-state" attribute
156 -----
157
158 | The following text is copied from the IPP Internet-Draft, 6/3/97.
159 | The revision marks show the changes to reflect the above
160 | agreements to the job-state attribute.
161
162 6.3.2.5 job-state (type1 keyword)

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

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

175 Standard values are:

176 'unknown': The job state is not known, or its state is
177 indeterminate.

178 'pending': The job is a candidate to start processing, but is
179 | not yet processing.

180 | 'pending-~~heldstopped~~': The job is not a candidate for
181 | processing for any number of reasons but will return to the
182 | 'pending' state as soon as the reasons are no longer
183 | present. The job's "job-state-reason" attribute SHALL~~shall~~
184 | indicate why the job is no longer a candidate for
185 | processing.

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

197 ~~ISSUE: The "job-state-reasons" specification is that the job~~
198 ~~is in 'pending' state not 'processing' state.~~

199 When the job is in the 'processing' state, the entire job
200 state includes the detailed status represented in the
201 printer's "printer-state", "printer-state-reasons", and
202 "printer-state-message" attributes.

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

208 'processing-stopped': The job has stopped while processing
209 for any number of reasons and will return to the~~continue~~
210 'processing' state as soon as the reasons are no longer
211 present.

212 The job's "job-state-reason" attribute ~~MAY~~shall indicate why
213 the job has stopped processing. For example, i

214 ~~if~~ the output device is stopped, the 'printer-stopped' value
215 ~~MAY~~shall be included in the job's "job-state-reasons"
216 attribute. For example, if the output device is stopped,
217 the 'printer-stopped' value MAY be included in the job's
218 "job-state-reasons" attribute.

219 NOTE - When ~~an~~the output device is stopped, the device
220 usually indicates its condition in human readable form
221 locally at the device. A client can obtain more complete
222 device status remotely by querying the printer's "printer-
223 state", "printer-state-reasons" and "printer-state-message"
224 attributes.

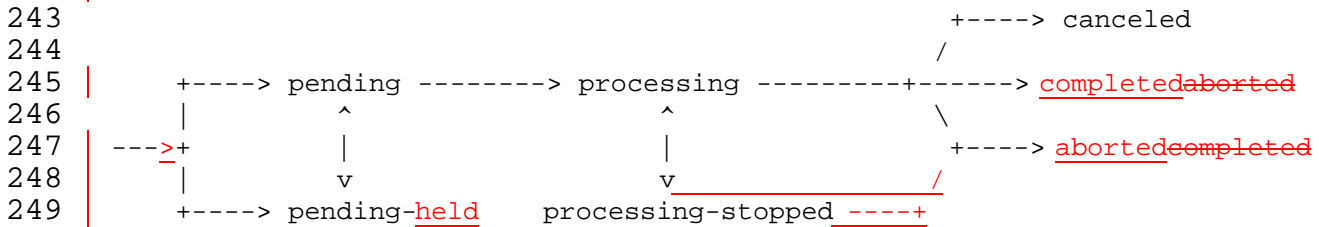
225 'canceled': The job has been canceled by a Cancel-Job
226 operation and is either (1) in the process of terminating or
227 (2) has completed terminating. The job's "job-state-
228 reasons" attribute ~~SHOULD~~shall contain either the 'canceled-
229 by-user' or 'canceled-by-operator' value.

230 'aborted': The job has been aborted by the system, usually
231 while the job was in the 'processing' or 'processing-
232 stopped' state.

233 'completed': The job has completed successfully or with
234 warnings or errors after processing and all of the job media
235 sheets have been ~~successfully~~properly stacked in the
236 appropriate output ~~tray or bin(s)~~. The job's "job-state-
237 reasons" attribute ~~SHOULD~~shall contain one of: 'completed-
238 successfully', 'completed-with-warnings', or 'completed-
239 with-errors' values.

240

241 The following figure shows the normal job state transitions.
 242 ~~Other transitions are unlikely, but are not forbidden.~~



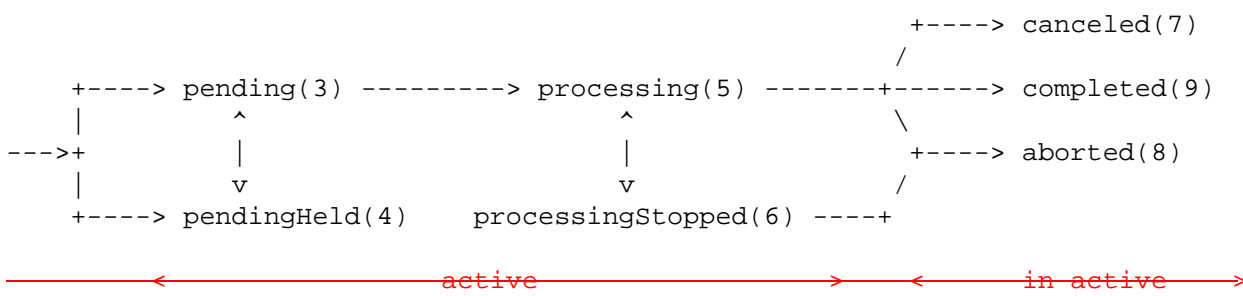
250 **Figure 3 - Normal Jjob State Transitions**

251 Normally a job progresses ~~only~~ from left to right ~~through three~~
 252 states. ~~Other state transitions are unlikely, but are not~~
 253 forbidden. ~~Not shown are the transitions to the 'canceled' state~~
 254 from the 'pending', 'pending-held', 'processing', and
 255 'processing-stopped' states.

256 Changes to JMP **jmJobState** object and **JmJobStateTC** description
257 -----
258 The following shows the changes for the JMP **jmJobState** object and
259 the corresponding **JmJobStateTC** textual-convention for the Job
260 Monitoring MIB (there is no longer a **jobState** attribute):
261
262 **jmJobState** OBJECT-TYPE
263 SYNTAX **JmJobStateTC** -- See page 8
264 MAX-ACCESS read-only
265 STATUS current
266 DESCRIPTION
267 "The current state of the job (**pending**, **processing**,
268 **completed**, etc.). Even though the **JmJobStateTC** textual-
269 convention defines nine values for job states, agents
270 SHALL only implement those states which are appropriate
271 for the particular implementation. In other words, all
272 possible enums for this object SHALL be reported if
273 implemented by the device and available to the agent.
274 However, management applications SHALL be prepared to
275 receive all the standard job states.
276
277 The final value for this object SHALL be one of:
278 **completed**, **canceled**, or **aborted**. The minimum length of
279 time that the agent SHALL keep a job in the **completed**,
280 **canceled**, or **aborted** state before removing the job from
281 the **jmJobIDTable** and **jmJobTable** is specified by the value
282 of the **jmGeneralJobPersistence** object."
283 ::= { jmJobStateEntry 1 }
284
285
286 **JmJobStateTC** ::= TEXTUAL-CONVENTION
287 STATUS current
288 DESCRIPTION
289 "The current state of the job (**pending**, **processing**,
290 **completedheld**, etc.).
291
292 ~~Management applications shall be prepared to receive all~~
293 ~~the standard job states. Agents instrumenting servers~~
294 ~~and devices are not required to generate all job states,~~
295 ~~only those that are indicated as 'mandatory' in the enum~~
296 ~~definitions below. The remaining job states are~~
297 ~~'conditionally mandatory', i.e., an agent for a server or~~
298 ~~device shall implement each of the remaining states if~~
299 ~~server or device jobs have states with the same~~
300 ~~semantics. See Section entitled '' on page for~~
301 ~~additional job state semantics, legal job state~~
302 ~~transitions, and implementation considerations.~~
303
304 ~~Companion textual conventions (**JmJobStateReasonsnTC**,~~
305 ~~**n=1..4**) and corresponding attributes (**jobStateReasonsn**)~~
306 ~~provide additional information about job states. While~~

307 ~~the job states cannot be added to without impacting~~
 308 ~~deployed clients that take actions upon receiving job~~
 309 ~~state values, it is the intent that additional~~
 310 ~~**JmJobStateReasons**TC enums can be defined without~~
 311 ~~impacting deployed clients. In other words, the~~
 312 ~~**JmJobStateReasons**TC TCs are intended to be extensible.~~
 313 ~~See page .~~

315 ~~The following job state standard values are defined. The~~
 316 ~~following figure shows the normal job state transitions:~~



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

329 Normally a job progresses from left to right. Other
 330 state transitions are unlikely, but are not forbidden.
 331 Not shown are the transitions to ~~T~~the **canceled** state ~~can~~
 332 ~~be entered~~ from the **pending**, **pendingHeld**, **processing**,
 333 ~~and~~ **processingStopped** states.

335 Jobs in the **pending**, **processing**, and **processingStopped**
 336 states are called 'active', while jobs in the
 337 **pendingHeld**, **canceled**, **aborted**, and **completed** are called
 338 'in-active'.'

340 -- This is a type 2 enumeration. See Section **Error! Reference source not found.** on page **Error!**
 341 **Bookmark not defined..**

```

342 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(34),
        -- The job is a candidate to start a candidate for
        -- processing, but is not yet processing.

    pendingHeld(43),
        -- The job is not yet 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.—Some
-- reasons are used in other states to give added
-- information about the job state. See the
JmJobStateReasons1TC (n=1..4) textual convention
on page (22) for the specification of each reason
and in which states the reasons are intended to be
used.

processing(55),

MANDATORY

-- Either:

-- 1. The job is using, or is attempting to use, one
-- or more document transforms which include (1)
-- purely software processes that are, ~~such as~~
-- interpreting a PDL, and (2) hardware devices that
-- are interpreting a PDL, ~~but is not yet~~ 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.

-- ~~If an implementation does not distinguish between
-- processing and printing, then the processing state
-- shall be implemented.~~

-- 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.

printing(6),

-- ~~The job is printing, i.e., making marks on a
-- medium.~~

-- ~~If an implementation does not distinguish between~~

~~processing and printing, then the processing state shall be implemented.~~

processingStoppedneedsAttention(67 **MANDATORY**
) ,

-- 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. ~~is~~
-- ~~using one or more devices, but has encountered a~~
-- ~~problem with at least one device that requires~~
-- ~~human intervention before the job can continue~~
-- ~~using that device. Examples include running out~~
-- ~~of paper or a paper jam.~~
--
-- 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
devices usually indicate ~~its~~~~their~~ 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(78), **MANDATORY**

-- 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, either because the client~~
-- ~~canceled the job or because a serious problem was~~
-- ~~encountered by a document transform while~~
-- ~~processing the job.~~ The job's **jobStateReasons1**
-- attribute SHOULD~~shall~~ contain either the
-- **canceledByUser** or **canceledByOperator** value~~the~~
-- ~~reasons that the job was canceled.~~ The job shall
-- remain in the **canceled** state for the same period
-- of time as if the job had completed, before
-- transiting to the **unknown** state. See the
-- **completed** state description.

aborted(8)

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

completed(9) **MANDATORY**

```
-- The job has (1) completed successfully or with
-- warnings or errors after processing/printing and
-- all of the media have been successfully stacked in
-- the appropriate output bin(s).
--
-- The job has completed successfully or with
-- warnings or errors. The job's jobStateReasons1
-- attribute SHOULDshall contain one of:
-- completedSuccessfully, completedWithWarnings, or
-- completedWithErrors valuesthe reasons that the job
-- has entered the completed state.
--
-- The length of time that a job may be in the
-- completed state, before transitioning to unknown,
-- is specified by the value of the
-- jmGeneralJobPersistence object. In addition, the
-- agent shall maintain all of the attributes in the
-- jmAttributeTable for at least the time specified
-- in the jmGeneralAttributePersistence object, so
-- that a management application accounting program
-- can copy all the attributes to an accounting log.
```

```
343     }
344
```

345 Here is the updated Job Monitoring MIB appendix:
346

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

348 **Instrumenting the Job Life Cycle**

349 The job object has well-defined states and client operations that
350 affect the transition between the job states. Internal server
351 and device actions also affect the transitions of the job between
352 the job states. These states and transitions are referred to as
353 the job's *life cycle*.

354 Not all implementations of job submission protocols have all of
355 the states of the job model specified here. The job model
356 specified here is intended to be a superset of most
357 implementations. It is the purpose of the agent to map the
358 particular implementation's job life cycle onto the one specified
359 here. The agent may omit any states not implemented. Only the
360 **processing, ~~needsAttention, canceled,~~ aborted, and **completed****
361 states are required to be implemented by an agent. However, a
362 **conforming** management application shall be prepared to accept any
363 of the states in the job life cycle specified here, so that the
364 management application can interoperate with any conforming
365 agent.

366 The job states are intended to be the user visible. The agent
367 shall make these states visible in the MIB, but only for the
368 subset of job states that the implementation has.
369 Implementations may need to have sub-states of these user-visible
370 states. Such implementation is *not* specified in this model, is
371 not supported by this Job Monitoring MIB, and will vary from
372 implementation to implementation. **In some implementations the**
373 **jmJobStateReasons1 object and the jobStateReasonsn (n=2..4)**
374 **attributes may represent some or all of the sub-states of the**
375 **jobs.**

376 One of the purposes of the job ~~life cycle~~model is to specify what
377 is invariant from implementation to implementation as far as the
378 MIB specification and the **management application user-**
379 **is** concerned. Therefore, job states are all intended to last a
380 user-visible length of time in most implementations. However,
381 some jobs may pass through some states in zero time in some
382 situations and/or in some implementations.

383 The job model does not specify how accounting and auditing is
384 implemented, except to ~~assume~~**require** that accounting and auditing
385 logs are separate from the job life cycle and last longer than
386 job **entries in the MIB**objects. Jobs in the **completed, aborted,**
387 **or canceled** states are not logs, since jobs in these **completed**
388 states are accessible via ~~SNMP job submission and/or job~~
389 **management** protocol operations and ~~shall be~~**removed** from these
390 Job **Monitoring MIB** tables after a site-settable **or**

391 implementation-defined period of time. An accounting application
 392 may copy Aaccounting information ~~may be copied~~ incrementally to
 393 anthe accounting logs as a job processes, or may be copied while
 394 the job is in the canceled, aborted, or completed states,
 395 depending on implementation. The same is true for auditing logs.

396 The jmJobState object ~~and the jobState attribute both specifies~~
 397 the standard job states. The normallegal job state transitions
 398 are shown in the state transition diagram presented in Table 1.
 399 ~~An implementation need not support all legal job state~~
 400 ~~transitions.~~

	New State							
	<u>"active" jobs</u>							
Old state	unkno wn 2	hel d 3	pend ing 4	proce ssing 5	prin ting 6	needsAt tention 7	cance led 8	compl eted 9
unknown(2)		yes	yes	yes	yes			
held(3)			yes	yes	yes		yes	
pending(4)		yes		yes	yes		yes	
processing(5)		yes			yes	yes	yes	yes
printing(6)		yes				yes	yes	yes
needsAttention(7)		yes		yes	yes		yes	
canceled(8)	yes							
completed(9)	yes							

401
 402
 403

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

405 The IPP **job-state-reasons** attribute can contain multiple
 406 keywords, while the JMP **jobStateReasons1** attribute is bit
 407 encoded. The Job Monitoring MIB contains a superset of the IPP
 408 values[8] for the IPP "job-state-reasons" attribute, since the
 409 Job Monitoring MIB is intended to cover other job submission
 410 protocols as well. Also some of the names of the reasons have
 411 been changed from 'printer' to 'device', since the Job Monitoring
 412 MIB is intended to cover additional types of devices, including
 413 input devices, such as scanners.

414 The comparison of the IPP "job-state-reasons" attribute and the
 415 JMP **jmJobStateReasons1** objectattribute after the IPP/JMP telecon,
 416 Wednesday, 5/~~2821~~/97, is as follows:

IPP job-state-reasons values	JMP jobStateReasons1 values	Notes
-	other	
-	unknown	
none	-	<u>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.</u>
<u>job-incoming</u>	<u>jobIncomingdocument</u> <u>sNeeded</u>	<u>Covers both the cases of document transfers in progress and additional SendDocument requests needed since the job isn't closed yet.</u>
<u>job-spooling</u>		
<u>job-queued</u>		
<u>job-sent-to-printer</u>		
<u>job-outgoing</u> <u>job-sending-to-output-</u> <u>device</u>	<u>jobOutgoing</u>	<u>Covers sending to the output device and queued in the output device.</u>

job-printing	<u>jobPrinting</u>	
printer-stopped	<u>deviceStopped</u>	JMP specifies <u>device, not printer, so that it can be used for additional services besides printing.</u>
printer-partly-stopped-partly	<u>deviceStoppedPartly</u>	
job-queued-in-output-device		
	jobHold <u>SpecifiedSet</u>	JMP has to cover <u>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.</u>
job-hold-until-specified	jobHoldUntilSpecified	JMP needs to <u>indicate which reasons prevent a job from being processed.</u>
	jobProcessAfterSpecified	JMP needs to cover <u>(DPA, VMS, Printxchange). Also JMP needs to indicate which reasons prevent a job from being processed.</u>
required-resources-are-not-not-ready	resourcesAreNotReady requiredResourcesNotReady	<u>Renamed so that it can be used in any job state.</u>
job-cancelled-by-user	jobCeanceledByUser	
job-cancelled-by-operator	jobCeanceledByOperator	
aborted-by-system	abortedBySystem	Don't need an <u>'aborted-by-system'</u>

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

417

418 ~~Now both Another difference is that IPP "job-state-reasons"~~
419 ~~attribute and JMP jmJobStateReasons1 object areis Mandatory,~~
420 ~~while the jobStateReasons attribute is conditionally mandatory.~~

421 Another difference is that IPP **job-state-reasons** shall have at
422 least one value, while the JMP **jmJobStateReasons1** object may have
423 no values. This difference is OK, since IPP will have the 'none'
424 value, and the JMP agent will have all job state reasons bits
425 turned off. IPP is striving to eliminate attributes that have no
426 values, in order to avoid ambiguities and inter-working problems.
427 There is no such problem with no bits being on in the
428 jmJobStateReasons1 object in JMP.

429

430 The following is the modified IPP "job-state-reasons" attribute
431 from the Internet-Draft of 6/3/97 following the 6/65/21/97
432 agreements:

433 6.3.2.6 job-state-reasons (lsetOf type2 keyword)

434 This attribute provides additional information about the job's
435 current state, i.e., information that augments the value of the
436 job's "job-state" attribute. ~~identifies the reason or reasons~~
437 ~~that the job is in the state that it is in (e.g., 'pending',~~
438 ~~'processing', 'completed', etc.). The Printer shall indicate the~~
439 ~~particular reason(s) by setting the value of the job's "job-~~
440 ~~state-reasons" attribute.~~

441 Implementation of these values is OPTIONAL, i.e., a Printer NEED
442 NOT implement them, even if (1) the output device supports the
443 functionality represented by the reason and (2) is available to
444 the Printer object implementation. These values MAY be used with
445 any job state or states for which the reason makes sense.
446 Furthermore, when implemented, the Printer SHALL return these
447 values when the reason applies and SHALL NOT return them when the
448 reason no longer applies whether the value of the job's "job-
449 state" attribute changed or not. When the job does not have any
450 reasons for being in its current state, the Printer shall set the
451 value of the job's "job-state-reasons" attribute to 'none'.The
452 Printer MAY provide additional information about each state value
453 by supplying one or more values of the job's companion "job-
454 state-reasons" attribute depending on implementation.

455 NOTE - While values cannot be added to the 'job-state' attribute
456 the job states cannot be added to without impacting deployed
457 clients that take actions upon receiving "job-state" values, it
458 is the intent that additional "job-state-reasons" values can be
459 defined and registered without impacting such deployed clients.
460 In other words, the "job-state-reasons" attribute is intended to
461 be extensible.

462 The following standard values are defined: ~~for use with the job~~
463 ~~states indicated as applicable.~~

464 NOTE - For easy of understanding the order of the reasons is
465 presented in the normal order of job state progression in which
466 the reason is most likely to occur:

467 'none': ~~Mandatory:~~ There are no reasons for the job's
468 current state.

469 ~~Applicable job states:~~ ~~'pending', 'processing', 'aborted'.~~

470 '

471 | ~~job-incoming': Mandatory.—The PrintJob or CreateJob~~
472 | ~~operation has been accepted by the Printer, but the Printer~~
473 | ~~is expectingwaiting for additional SendDocument operations~~
474 | ~~and/or is accessing/accepting the transfer of the remainder~~
475 | ~~of the job or document data.~~

476 | ~~Applicable job states: 'pending-stopped'~~

477 | ~~'~~

478 | ~~job-outgoing': Optional.—The Printer is transmitting the job~~
479 | ~~to the output device.~~

480 | ~~Applicable job states: 'pending'~~

481 | ~~'~~

482 | ~~job-hold-until-specified': Conditionally mandatory.~~
483 | ~~Mandatory if the "job hold until" job attribute is~~
484 | ~~implemented.—The value of the job's "job-hold-until"~~
485 | ~~attribute has specified a time period that is still in the~~
486 | ~~futurehas not yet arrived, so that the Printer shall not~~
487 | ~~consider the job as a candidate for processing. The job~~
488 | ~~SHALL NOT be a candidate for processing until this reason is~~
489 | ~~removed and there are no other reasons to hold the job.~~

490 | ~~Applicable job states: 'pending-stopped'.~~

491 | ~~'~~

492 | ~~job hold until resources-are-not-ready': Optional.—At least~~
493 | ~~one of the resources needed by the job, such as media,~~
494 | ~~fonts, resource objects, etc., is not ready on any of the~~
495 | ~~physical printer's for which the job is a candidate. This~~
496 | ~~condition MAY be detected when the job is accepted, or~~
497 | ~~subsequently while the job is pending or processing,~~
498 | ~~depending on implementation.~~

499 | ~~Usually such a condition is detected while the job is in the~~
500 | ~~'pending' state. If a job starts processing and encounters~~
501 | ~~a resource that is not ready, there are two possible~~
502 | ~~implementations:~~

503 | ~~(1) the device is stopped and no jobs can run until the~~
504 | ~~resource(s) are made ready, in which case the Printer shall~~
505 | ~~keep the job in the 'processing' state and shall add the~~
506 | ~~'printer-stopped' reason to the job's "job-state-reasons"~~
507 | ~~attribute (not the 'job hold until resources are ready'~~
508 | ~~value) OR~~

509 | ~~(2) another job is found to use the device while the current~~
510 | ~~job goes back to waiting for its turn and for the resources~~
511 | ~~to be made ready, in which case the Printer shall change the~~
512 | ~~job's "job-state" attribute to 'pending' and add the 'job-~~
513 | ~~hold-until-resources-are-ready' value to the job's "job-~~
514 | ~~state-reasons" attribute.~~

515 ~~Applicable job states: 'pending stopped'.~~
516 ~~'~~
517 printer-stopped-partly': ~~___~~ ~~Optional.~~ ~~This reason appears in~~
518 ~~all jobs in the 'pending' state when~~ the value of the
519 Printer's "printer-state-reasons" attribute contains the
520 value 'stopped-partly'.
521 ~~Applicable job states: 'pending'~~
522 ~~'~~
523 printer-stopped': ~~___~~ ~~Mandatory.~~ ~~The output device is~~
524 ~~stopped. This reason appears in all jobs in the 'pending'~~
525 ~~and 'processing' states when the~~ value of the Printer's
526 "printer-state" attribute is 'stopped'.
527 ~~Applicable job states: 'pending stopped', 'processing stopped'~~
528 ~~'~~
529 job-printing': ~~___~~ ~~Optional.~~ The output device is marking media.
530 This value is useful for Printers which spend a great deal
531 of time processing when no marking is happening and then
532 want to show that marking is now happening.
533 ~~Applicable job states: 'processing', 'processing stopped'~~
534 ~~'~~
535 job-cancelled-by-user': ~~___~~ ~~Level 2 Mandatory.~~ The job was
536 cancelled by the user using the CancelJob request, i.e., by
537 a user whose name is the same as the value of the job's
538 "job-originating-user" attribute.
539 ~~Applicable job states: 'canceled'.~~
540 ~~'~~
541 job-cancelled-by-operator': ~~___~~ ~~Level 2 Mandatory.~~ The job
542 was cancelled by the operator using the CancelJob request,
543 i.e., by a user whose name is different than the value of
544 the job's "job-originating-user" attribute.
545 ~~Applicable job states: 'canceled'.~~
546 ~~'~~
547 job-completed-successfully': ~~___~~ ~~Mandatory.~~ The job completed
548 successfully.
549 ~~Applicable job states: 'completed'.~~
550 ~~'~~
551 job-completed-with-warnings': ~~___~~ ~~Mandatory.~~ The job completed
552 with warnings.
553 ~~Applicable job states: 'completed'.~~
554 ~~'~~

555 job-completed-with-errors': ~~_____Mandatory.~~ The job completed
556 with errors (and possibly warnings too).

557 ~~Applicable job states: 'completed'.~~

558 ~~'~~

559 logfile-pending '': ~~_____Optional.~~ The job's logfile is pending
560 file transfer.

561 ~~Applicable job states: 'canceled', 'aborted', and 'completed'.~~

562 ~~'~~

563 logfile-transferring': ~~_____Optional.~~ The job's logfile is
564 being transferred.

565 ~~Applicable job states: 'canceled', 'aborted', and 'completed'.~~

566

567

568

569 Here is the Job Monitoring MIB definition of the **jobStateReasons1**
570 **objectattribute** and the defined values updated to reflect the
571 agreements on IPP and with additional reasons that are needed by
572 other job submission protocols, even though they are not needed
573 by IPP:

574 jmJobStateReasons1 OBJECT-TYPE

575 SYNTAX JmJobStateReasons1TC -- See page 22

576 MAX-ACCESS read-only

577 STATUS current

578 DESCRIPTION

579 "Additional information about the job's current state,
580 i.e., information that augments the value of the job's
581 jmJobState/jobState object/attribute. ~~NOTE Companion~~
582 ~~textual conventions, JmJobStateReasons1TC (n=1..4~~ ~~see~~
583 ~~page) and corresponding attributes see page provides~~
584 ~~additional information about job states.~~

585
586 NOTE - The jobStateReasonsn (n=2..4) attributes (see page
587 Error! Bookmark not defined.) provide further additional
588 information about the job's current state.

589
590 Implementation of these values is OPTIONAL, i.e., an
591 agent NEED NOT implement them, even if (1) the device
592 supports the functionality represented by the reason and
593 (2) is available to the agent. These values MAY be used
594 with any job state or states for which the reason makes
595 sense. Furthermore, when implemented, the agent SHALL
596 return these values when the reason applies and SHALL NOT
597 return them when the reason no longer applies whether the
598 value of the job's jmJobState object changed or not.
599 When the job does not have any reasons for being in its
600 current state, the agent SHALL set the value of the
601 jmJobStateReasons1 object and jobStateReasonsn attributes
602 to 0.

603
604 NOTE - While values cannot be added to the jmJobState
605 object the job states cannot be added to without
606 impacting deployed clients that take actions upon
607 receiving jmJobState values, it is the intent that
608 additional JmJobStateReasons1TC enums can be defined and
609 registered without impacting such deployed clients. In
610 other words, the jmJobStateReasons1 object and
611 jobStateReasonsn attributes are intended to be
612 extensible. The jobStateReasons1 attribute identifies the
613 reason or reasons that the job is in the held, pending,
614 processing, needsAttention, canceled, or completed state.
615 The agent shall indicate the particular reason(s) by
616 setting the value of the jobStateReasons1 attribute."

617 ::= { jmJobEntry 2 }

618
619
620 **JmJobStateReasons1TC ::= TEXTUAL-CONVENTION**

621 STATUS current

622 DESCRIPTION

623 "This textual-convention is used with the

624 **jmJobStateReasons1** objectattribute to provides additional

625 information regarding the **jmJobState/jobState** object

626 values/attribute. ~~The **jobStateReasons1** attributes~~

627 ~~identifies the reason or reasons that the job is in the~~

628 ~~**held, pending, processing, printing, needsAttention,**~~

629 ~~**cancelled, or completed** state. The server shall indicate~~

630 ~~the particular reason(s) by setting the value of the~~

631 ~~**jobStateReasons1** attribute. While the job states cannot~~

632 ~~be added to without impacting deployed clients, it is the~~

633 ~~intent that additional **JmJobStateReasons1TC** enums can be~~

634 ~~defined without impacting deployed clients. In other~~

635 ~~words, the **JmJobStateReasons1TC** is intended to be~~

636 ~~extensible.~~

637

638 ~~When the job does not have any reasons for being in its~~

639 ~~current state, the server shall set the value of the~~

640 ~~**jobStateReasons1** attribute to zeros.~~

641

642 ~~Companion job state reasons TCs: **JmJobStateReasons2TC,**~~

643 ~~**JmJobStateReasons3TC, JmJobStateReasons4TC,** are~~

644 ~~defined/reserved for additional 31*3 = 93 job state~~

645 ~~reasons. This is a type 2 bit definition. See section~~

646 ~~on page.~~

647

648 The following standard values are defined (in

649 hexadecimal) as *powers of two*, since multiple values MAY

650 be used at the same time.

651

652 NOTE - The Job Monitoring MIB contains a superset of the

653 IPP values[3] for the IPP 'job-state-reasons' attribute,

654 since the Job Monitoring MIB is intended to cover other

655 job submission protocols as well. Also some of the names

656 of the reasons have been changed from 'printer' to

657 'device', since the Job Monitoring MIB is intended to

658 cover additional types of devices, including input

659 devices, such as scanners.+

660

661 NOTE - For easy of understanding the order of the reasons

662 is presented in the order in which the reason is most

663 likely to occur.

664

665 **other** **0x1**

666 The job state reason is not one of the standardized or

667 registered reasons.

668

669 **unknown** **0x2**

670 The job state reason is not known to the agent or is

671 indeterminent.

672

673 **jobIncomingdocumentsNeeded** **0x4**

674 The job has been accepted by the server or device, but

675 the server or device is in the held state because the

676 ~~server or device is expected waiting for (1) additional~~
677 ~~operations to finish creating the job and/or (2) is~~
678 ~~accessing/accepting document data the job's files to start~~
679 ~~and/or finish being transferred before the job can be~~
680 ~~scheduled to be processed.~~

681
682 **jobOutgoing** **0x8**

683 ~~Configuration 2 only: The server is transmitting the job~~
684 ~~to the device.~~

685
686 **jobHoldSpecifiedSet** **0x108**

687 ~~The job is in the held state because the client specified~~
688 ~~that the job is to be held value of the job's **Error!**~~
689 ~~**Reference source not found.** attribute (see page **Error!**~~
690 ~~**Bookmark not defined.**) is **TRUE**, either set when the job~~
691 ~~was created or subsequently by an explicit modify job~~
692 ~~operation. The job SHALL NOT be a candidate for~~
693 ~~processing until this reason is removed and there are no~~
694 ~~other reasons to hold the job.~~

695
696 **jobHoldUntilSpecified** **0x2040000**

697 ~~The value of the job's **Error! Reference source not found.**~~
698 ~~(see page **Error! Bookmark not defined.**) attribute~~
699 ~~specifies a time period was specified for a named time~~
700 ~~period that is still in the future, either set when the~~
701 ~~job was created or subsequently by an explicit modify job~~
702 ~~operation. The job SHALL NOT be a candidate for~~
703 ~~processing until this reason is removed and there are no~~
704 ~~other reasons to hold the job. The job remains in the~~
705 ~~held state until the time period arrives and there are no~~
706 ~~other reasons to hold the job.~~

707
708 **jobProcessAfterSpecified** **0x4010**

709 ~~The value of the job's **Error! Reference source not found.**~~
710 ~~(see page **Error! Bookmark not defined.**) attribute~~
711 ~~specifies a time job is in the held state because the~~
712 ~~client specified a time specification reflected in the~~
713 ~~value of the job's attribute that is still in the~~
714 ~~future, either set when the job was created or~~
715 ~~subsequently by an explicit modify job operation. The~~
716 ~~job SHALL NOT be a candidate for processing until this~~
717 ~~reason is removed and there are no other reasons to hold~~
718 ~~the job.~~

719
720 **requiredResourcesAreNotNotReady** **0x8020**

721 ~~The job is in the held state because Aat least one of the~~
722 ~~resources needed by the job, such as media, fonts,~~
723 ~~resource objects, etc., is not ready on any of the~~
724 ~~physical devices for which the job is a candidate. This~~
725 ~~condition MAY be detected when the job is accepted, or~~
726 ~~subsequently while the job is pending or processing,~~
727 ~~depending on implementation.~~

728
729

730 **deviceStoppedPartly** **0x100**
731 One or more, but not all, of the devices to which the job
732 is assigned are stopped. If all of the devices are
733 stopped (or the only device is stopped), the
734 **deviceStopped** reason SHALL be used.
735
736 **deviceStopped** **0x200**
737 The device(s) to which the job is assigned is (are all)
738 stopped.
739
740 **jobPrinting** **0x400**
741 The output device is marking media. This attribute is
742 useful for servers and output devices which spend a great
743 deal of time processing when no marking is happening and
744 then want to show that marking is now happening.
745
746 **jobCanceledByUser** **0x800200**
747 The job ~~is in the canceled,~~ state having ~~was~~ been
748 canceled by the user, i.e., by a user whose name is the
749 same as the value of the job's **jobOwner** attribute.
750
751 **jobCanceledByOperator** **0x1000400**
752 The job ~~was~~ in the ~~canceled~~ state having been canceled
753 by the operator, i.e., by a user whose name is different
754 than the value of the job's **jobOwner** attribute.
755
756 **abortedBySystem** **0x2000800**
757 The job ~~was~~ in the ~~canceled,~~ state having been aborted
758 by the system. NOTE - this reason is needed only when
759 the job is not placed in the **aborted** job state.
760
761
762 **jobCompletedSuccessfullyCompletion** **0x400040**
763 The job ~~is in the completed~~ state having completed
764 successfully.
765
766 **jobCompletedWithWarnings** **0x800080**
767 The job ~~is in the canceled or completed~~ states having
768 completed with warnings.
769
770 **jobCompletedWithErrors** **0x10000100**
771 The job ~~is in the canceled or completed~~ states having
772 completed with errors (and possibly warnings too).
773
774 The following additional job state reasons have been added to
775 specify sub-states of the ~~held or completed~~ states that may
776 be used to represent job states that are in ISO DPA[2] and
777 other job submission protocols:
778
779 ~~**jobPreProcessing** **0x4000**~~The job has been created on
780 ~~the server or device but the submitting client is in the~~
781 ~~process of adding additional job components and no~~
782 ~~documents have started processing. The job maybe in the~~
783 ~~process of being checked by the server/device for~~

784 ~~attributes, defaults being applied, a device being~~
785 ~~selected, etc.~~

786
787 **jobPaused** **0x~~200008000~~**

788 The job has been indefinitely suspended by a client
789 issuing an operation to suspend the job so that other
790 jobs may proceed using the same devices. The client MAY
791 issue an operation to resume the paused job at any time,
792 in which case the ~~agent SHALL remove the jobPaused values~~
793 ~~from the job's jmJobStateReasons1 object and the server~~
794 ~~or device places the job in the held or pending states~~
795 ~~and the job is eventually resumed at or near~~ the point
796 where the job was paused.
797

798 **jobInterrupted** **0x~~4000010000~~**

799 The job has been interrupted while processing by a client
800 issuing an operation that specifies another job to be run
801 instead of the current job. The server or device will
802 automatically resume the interrupted job when the
803 interrupting job completes.
804

805 **jobRetained** **0x~~8000020000~~**

806 The job is being retained by the server or device ~~with~~
807 ~~all of the job's document data (and submitted resources,~~
808 ~~such as fonts, logos, and forms, if any). Thus a client~~
809 ~~could issue an operation to resubmit the job (or a copy~~
810 ~~of the job). after processing and all of the media have~~
811 ~~been successfully stacked in the output bin(s).~~

812
813 ~~The job (1) has completed successfully or with warnings~~
814 ~~or errors, (2) has been aborted while printing by the~~
815 ~~server/device, or (3) has been canceled by the submitting~~
816 ~~user or operator before or during processing. The job's~~
817 ~~jobStateReasons1 attribute shall contain the reasons that~~
818 ~~the job has entered the completed state.~~

819
820 ~~While the jobRetained state reason is , all of the job's~~
821 ~~document data (and submitted resources, such as fonts,~~
822 ~~logos, and forms, if any) are retained by the server or~~
823 ~~device; thus a client could issue an operation to~~
824 ~~resubmit the job (or a copy of the job). When a client~~
825 ~~could no longer resubmit the job, such as after the~~
826 ~~document data has been discarded, the agent SHALL remove~~
827 ~~the jobRetained value from the jmJobStateReasons1 object.~~

828
829 These bit definitions are the equivalent of a type 2 enum
830 except that combinations of ~~bitsthem~~ may be used together.
831 See section **Error! Reference source not found.** on page **Error!**
832 **Bookmark not defined..** ~~The remaining bits are reserved for~~
833 ~~future standardization and/or registration."~~

834
835 These bit definitions are the equivalent of a type 2 enum
836 except that combinations of ~~bits them~~ may be used together.

837 See section **Error! Reference source not found.** on page **Error!**
838 **Bookmark not defined.."**
839
840 SYNTAX **INTEGER(0..2147483647)** -- 31 bits, all but sign bit
841
842
843

844 ~~Table 1 JmJobStateReasons1TC: Legal Job States for each Job~~
 845 ~~State~~
 846 ~~Reason~~
 847

—	Descriptive Name	Allowed job states
—	documents-needed(1)	held
—	job-hold-set(2)	held
—	job-process-after-specified(3)	held
—	required-resources-not-ready(4)	held
—	successful-completion(5)	completed
—	completed-with-warnings(6)	completed
—	completed-with-errors(7)	completed
—	canceled-by-user(8)	canceled
—	canceled-by-operator(9)	canceled
—	aborted-by-system(10)	canceled
—	logfile-pending(11)	canceled
—	logfile-transferring(12)	canceled
—	jobPreProcessing(45)	held
—	jobPaused(46)	held
—	jobInterrupted(47)	held
—	jobRetained(48)	canceled, completed
—	jobHoldUntilSpecified(49)	held

848