

1 Subj: Differences between IPP **job-state** and **job-state-reasons** attributes and the Job Monitoring MIB
2 **jmJobState** and **jobStateReason1** attributes.

3 From: Tom Hastings

4 Date: 5/2019/97

5 File: ippstate.doc

6 The Job Monitoring Project (JMP) wanted to have a telecon discussion of the differences between the IPP
7 **job-state** and **job-state-reasons** attributes and the IETF Job Monitoring MIB **jmJobState** object and the
8 **jobStateReasons1** attribute. This document lists the differences and suggests a number of alternatives for
9 resolving the differences. I've also added the IPP **job-state-message** and JMP **processingMessage**
10 attribute (which are about the same) and are also used to describe the state of a job.

11 Both the IPP and JMP texts have changed a little as a result of meetings last week. I've edited in these
12 *unofficial* changes for the IPP telecon using revision marks. This modified text follows the list of
13 differences for each attribute. I've also copied in the Appendix from the Job Monitoring MIB which has
14 more details about the job state and the job life cycle, including a state transition diagram. I've also copied
15 in the table from the MIB showing the job -tate-reasons and which job states each may be used with.

16 The goal of the Job Monitoring MIB is to instrument various printers and servers that accept jobs via a
17 variety of job submission protocols. One of the important job submission protocols will be IPP.
18 Therefore, the Job Monitoring MIB should have a simple mapping from IPP to the Job Monitoring MIB
19 that a JMP agent can implement when instrumenting an IPP server or printer. Since the Job Monitoring
20 MIB is intended to describe other printers and servers, the Job Monitoring MIB will need to be a superset
21 of IPP.

22 The following table shows the differences between the IPP job-state attribute and the Job Monitoring MIB
23 jmJobStateAttribute:

IPP job-state	JMP jmJobState	Discussion
	other(1)	Not a problem that JMP has and IPP doesn't, since JMP needs to map other protocols besides IPP.
unknown	unknown(2)	same
	held(3)	IPP achieves held by using job-state-reasons in the 'pending' state.
pending	pending(4)	same
processing	processing(5)	same
	needsAttention(67)	IPP achieves needsAttention by using the ' printer-stopped ' value of the IPP job-state-reasons attribute
terminating	canceled(78)	same, except IPP 'terminating' goes to retained, while JMP canceled(7) is a final job state.
retained		JMP represents the IPP ' retained ' state using the JMP jobRetained value of the JMP jobStateReasons1 attribute
completed	completed(89)	same, except the IPP ' canceled ' jobs wind up in the ' completed ' state with the IPP ' canceled-by-xxx ' value of the IPP job-state-reasons attribute

24 Another difference is that the IPP **job-state** attribute is a type 1 attribute while JMP **JmJobStateTC** is a
25 type 2 enum. This difference can be justified, in the unlikely event that the JMP may need to add a job

- 26 state for some other protocol, where just adding a value to the **jmJobStateReason1** attribute isn't
 27 appropriate.
- 28 Suggested alternatives for resolving each difference:
- 29 1. IPP - vs. JMP **other**(1) - don't change either standard
- 30 2. IPP - vs. JMP **held**((3) - Problem: The JMP agent can't tell whether any of the IPP **job-state-**
 31 **reasons** value(s) is(are) preventing the job from being a candidate for scheduling or is indicating
 32 some other reason while the job is in the '**pending**' state. Also other job submission systems have
 33 the equivalent of the '**held**' state, where the job is temporarily not a candidate for scheduling.
 34 Possible solutions:
- 35 a) Add the **held** state to IPP
- 36 b) Leave IPP as it is, but add another value to IPP **job-state-reasons** attribute that indicates
 37 that the job is held, i.e., is not a candidate for processing. Then the JMP agent can tell
 38 whether the job should be shown in the **held**(3) or the **pending**(4) state.
- 39 c) Remove the '**held**' state from JMP and add the JMP **jobHeld** value to the JMP
 40 **jobStateReasons1** attribute (and to the IPP **job-state-reasons** attribute).
- 41 3. IPP - vs. JMP **needsAttention**(6) - IPP indicates that the job needs attention by keeping the job in
 42 the '**processing**' state and adding the IPP '**printer-stopped**' value to the IPP **job-state-reasons**
 43 attribute. Suggested alternatives:
- 44 a) No changes to either. A JMP agent can easily map the IPP '**processing**' state when the
 45 IPP '**printer-stopped**' value is present in the IPP **job-state-reasons** attribute.
- 46 b) Add the **needs-attention** state to IPP, instead of using the '**printer-stopped**' value with
 47 the IPP **job-state-reasons** attribute.
- 48 c) Remove the JMP **needsAttention**(6) state and add the '**printer-stopped**' value to the
 49 JMP **jobStateReasons1** attribute.
- 50 4. IPP terminating vs. JMP canceled(7):
- 51 a) Don't change either. The JMP agent can map the IPP '**terminating**' state to JMP
 52 '**canceled**' state and can leave the job in the JMP 'canceled' state, whenever the IPP
 53 **job-state-reasons** attribute contains any of the values: '**canceled-by-user**', '**canceled-**
 54 **by-operator**', or '**aborted-by-system**'.
- 55 b) Change IPP '**terminating**' state to '**canceled**' as a final state.
- 56 c) Change JMP '**canceled**' state to '**terminating**' as an intermediary state to **completed**
 57 (and maybe **retained**, if the **retained** state is kept).
- 58 5. IPP **retained** state vs. JMP -:
- 59 a) Don't change either. The JMP agent can easily map the IPP '**retained**' state to the JMP
 60 '**completed**' state and set the **jobRetained** value in the **jobStateReasons1** attribute.
- 61 b) Drop the IPP '**retained**' state, and add the '**job-retained**' value for use with IPP **job-**
 62 **state-reasons** attribute.
- 63 c) Add the '**retained**' state to JMP.
- 64 6. IPP **job-state** attribute values are type 1 keywords, while JMP **jobState** is a type 2 enum.
- 65 a) Don't change either. IPP can control its own job states, while JMP may need to add a
 66 state in order to map from some job submission protocol.
- 67 b) Change IPP to type 2 keyword

- 68 c) Change JMP jmJobState to a type 1 enum.
- 69 7. JMP **processing**, **needsAttention**, **canceled**, and **completed** states are mandatory, the rest are
- 70 conditionally mandatory. IPP doesn't say.
- 71

72 The IPP descriptions from the 970512 version with revisions from agreements on 5/14:

73 6.3.2.5 job-state (type1 keyword, MANDATORY)

74 This attribute identifies the current state of the job. Standard values are:

- unknown** The job state is not known, or is indeterminate.
- pending** The job is waiting to start processing on a printer. Various job-reasons may keep a job from moving to the ~~processing~~printing state.
- processing** The server is processing the job, or 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.

Or

The server has completed processing the job and the output device is currently printing the job. That is, an output device is either printing pages of the job, or failing in its attempt to print pages of the job because of some wait state, such as, start-wait, end-wait, needs-attention, etc. The complete job state includes the detailed status represented in the printer's printer-state attribute.

As with the printer state, let's let the reason make the distinction as to whether paper is being marked or the printer is just processing. Most printers won't bother with this nuance.

- terminating** The job has been canceled by a Cancel-Job request or aborted by the server and is in the process of terminating. The job's job-state-reasons attribute contains the reasons that the job is being terminated.

- retained** The job is being retained at the server. The job has (1) completed successfully or with warnings or errors, (2) been aborted while ~~processing~~printing by the server, or (3) been canceled by the Cancel-Job request before or during processing. The job's job-state-reasons attribute contains the reasons that the job has been retained.

While in the retained state, all of the job's document data (and resources, if any) shall be retained by the server; thus a job in the retained state could be reprinted, using some means outside the scope of IPP V1.0. If a given implementation does not support this functionality, there is no need to support this value.

completed

The job has:

- (1) completed successfully or with warnings or errors,
- (2) been aborted by the server while processingprinting, or
- (3) been canceled by the Cancel-Job request,

For case 1, "completed " only happens when all job media sheets have been properly stacked in the appropriate output tray or bin.

The job's job-state-reasons attribute contains the reason(s) that the job has been completed.

While in the completed state, a job's document data (and resources if any) need not be retained by the server; thus a job in the completed state could not be reprinted. The length of time that a job may be in this state, before transitioning to unknown, is implementation-dependent. However, servers that implement the completed job-state shall retain, as a minimum, the following attributes for any job in the completed state: job-identifier, job-originator, job-name, current-job-state, output-device-assigned, and job-state-reasons.

75

76

The IPP protocol supports all values for job states, but Printers need only support those states which are appropriate for the particular implementation.

77

78

79 The JMP equivalent specifications to the IPP **job-state** attribute is the **jmJobState** object in the mandatory
80 **jmJobTable**. Here is the version from the Internet-Draft 00 (same as version 0.81) with agreements
81 reached at the 5/16 JMP meeting:

82

83 **jmJobState** OBJECT-TYPE

84 SYNTAX **JmJobStateTC** -- See page 6

85 MAX-ACCESS read-only

86 STATUS current

87 DESCRIPTION

88 "The current state of the job (**pending, processing, held**, etc.). The final value for this attribute
89 shall be either **completed** or **canceled**, before the agent removes the job from the table.

90

91 Management applications shall be prepared to receive all the standard job states. Servers and
92 devices are not required to generate all job states, only those which are appropriate for the
93 particular implementation.

94

95 NOTE - Companion textual conventions, **JmJobStateReasons n TC** ($n=1..4$ - see page 14) and
96 corresponding attributes (**jobStateReasons1(5)** - see page 14) provides additional information
97 about job states. While the job states cannot be added to without impacting deployed clients, it is
98 the intent that additional **JmJobStateReasons n TC** enums can be defined without impacting
99 deployed clients.

100

101 The value of this object shall always be the same as that of the **jobState** attribute, so that this
102 information appears in both the **jmJobStateTable** and the **jmAttributeTable** simultaneously. See
103 the **JmJobStateTC** textual-convention on page **Error! Bookmark not defined.** and the **jobState**
104 attribute on page **Error! Bookmark not defined.** in the **jmAttributeTable** for the full
105 specification of this object/attribute."

106 ::= { jmJobStateEntry 1 }

107

108

109 **JmJobStateTC** ::= TEXTUAL-CONVENTION

110 STATUS current

111 DESCRIPTION

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

113

114 Management applications shall be prepared to receive all the standard job states. Agents
115 instrumenting servers and devices are not required to generate all job states, only those that
116 are indicated as 'mandatory' in the enum definitions below. The remaining job states are
117 'conditionally mandatory', i.e., an agent for a server or device shall implement each of the
118 remaining states if server or device jobs have states with the same semantics. See Section 0
119 entitled '**Job Life Cycle**' on page 9 for additional job state semantics, legal job state
120 transitions, and implementation considerations.

121

122 Companion textual conventions (**JmJobStateReasons n TC**, $n=1..4$) and corresponding
123 attributes (**jobStateReasons n**) provide additional information about job states. While the
124 job states cannot be added to without impacting deployed clients, it is the intent that
125 additional **JmJobStateReasons n TC** enums can be defined without impacting deployed
126 clients that take actions upon receiving job state values. In other words, the
127 **JmJobStateReasons n TC** TCs are intended to be extensible. See page 14.

128

129 The following job state standard values are defined:"

130

131
132
133

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

SYNTAX INTEGER {

other(1),

-- The job state is *not* one of the defined states.

unknown(2),

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

held(3),

-- The job is not yet a candidate for processing for any number of reasons. The reasons are represented as bits in the **jobStateReasons1** attribute. Some reasons are used in other states to give added information about the job state. See the **JmJobStateReasons1TC** textual convention for the specification of each reason and in which states the reasons may be used.

pending(4),

-- The job is a candidate for processing, but is not yet processing.

processing(5),

-- mandatory

-- The job is using one or more document transforms which include purely software processes, such as interpreting a PDL, and hardware devices, but is not yet making marks on a medium.

--

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

~~**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.~~

needsAttention(7),

-- mandatory

-- The job 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.

--

-- Usually devices indicate 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.

canceled(8),

-- mandatory

-- The job is in the process of being terminated by the server or device or 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 shall contain 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.

completed(9)

-- mandatory

-- The job has (1) completed after processing/~~printing~~ and all of the media have been successfully stacked in the output bin(s).

--

- The job has completed successfully or with warnings or errors. The job's
- **jobStateReasons1** attribute shall contain the 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.

134 }

135

Appendix A - Job Life Cycle

136 Job Life Cycle

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

140 Not all implementations of job submission protocols have all of the states of the job model specified here.
 141 The job model specified here is intended to be a superset of most implementations. It is the purpose of the
 142 agent to map the particular implementation's job life cycle onto the one specified here. The agent may
 143 omit any states not implemented. Only the **processing**, **needsAttention**, **canceled**, and **completed** states
 144 are required to be implemented by an agent. However, a management application shall be prepared to
 145 accept any of the states in the job life cycle specified here, so that the management application can
 146 interoperate with any conforming agent.

147 The job states are intended to be the user visible. The agent shall make these states visible in the MIB, but
 148 only for the subset of job states that the implementation has. Implementations may need to have sub-states
 149 of these user-visible states. Such implementation is *not* specified in this model, is not supported by this
 150 Job Monitoring MIB, and will vary from implementation to implementation.

151 One of the purposes of the job model is to specify what is invariant from implementation to
 152 implementation as far as the MIB specification and the user is concerned. Therefore, job states are all
 153 intended to last a user-visible length of time in most implementations. However, some jobs may pass
 154 through some states in zero time in some situations and/or in some implementations.

155 The job model does not specify how accounting and auditing is implemented, except to require that
 156 accounting and auditing logs are separate from the job life cycle and last longer than job objects. Jobs in
 157 the **completed** state are not logs, since jobs in the **completed** state are accessible via job submission
 158 and/or job management protocol operations and are removed from these job tables after a site-settable
 159 period of time. Accounting information may be copied incrementally to the accounting logs as a job
 160 processes, or may be copied while the job is in the **completed** state, depending on implementation. The
 161 same is true for auditing logs.

162 The **jmJobState** object and the **jobState** attribute both specify the standard job states. The legal job state
 163 transitions are shown in the state transition diagram presented in Table 1. An implementation need not
 164 support all legal job state transitions.

165

Table 0-2 - Legal Job State Transition Table

	New State							
		"active" jobs						
Old state	unkn own 2	held 3	pendi ng 4	proce ssing 5	print ing 6	needs Atten tion 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(67)		yes		yes	yes		yes	
canceled(78)	yes							

	New State							
	"active" jobs							
Old state	unkn own 2	held 3	pendi ng 4	proce ssing 5	print ing 6	needs Atten tion 7	cance led 8	compl eted 9
completed(89)	yes							

166 The IPP **job-state-reasons** is multivalued keywords, while the JMP **jobStateReasons1** attribute is bit
 167 encoded. The comparison of the IPP job-state-reasons attribute and the JMP jobStateReason1 attribute is
 168 as follows:

IPP job-state-reasons values	JMP jobStateReasons1 values	Notes
-	other	
-	unknown	
none	-	
	documentsNeeded	
job-spooling		
job-queued		
job-sent-to-printer		
job-held	jobHoldSet	
job-sending-to-output-device		
job-queued-in-output-device		
job-printing		
printer-stopped		
printer-partly-stopped		
job-hold-until-specified	jobHoldUntilSpecified	
	jobProcessAfterSpecified	
required-resources-not-ready	requiredResourcesNotReady	
successful completion	successfulCompletion	
completed-with-warnings	completedWithWarnings	
completed-with-errors	completedWithErrors	
cancelled-by-user	canceledByUser	
cancelled-by-operator	canceledByOperator	
aborted-by-system	abortedBySystem	
logfile-pending	abortedBySystem	
logfile-transferring	logfileTransferring	
	jobPreProcessing	
	jobPaused	
	jobInterrupted	
	jobRetained	

169
 170 Another difference is that IPP **job-state-reasons** is Mandatory, while the **jobStateReasons** attribute is
 171 conditionally mandatory.

172 Another difference is that IPP **job-state-reasons** shall have at least one value, while the JMP
173 **jobStateReason1** attribute may have no values.

174 The IPP **job-state-reasons** attribute is defined as:

175 6.3.2.6 job-state-reasons (1setOf type2 keyword, MANDATORY)

176 This attribute identifies the reason or reasons that the job is in the state that it is in (e.g., held,
177 terminating, retained, completed, etc.). The printer shall indicate the particular reason(s) by setting
178 the value of the job-state-reasons attribute.

179 ISSUE: Should we change job-incomplete to job-spooling and add job-queued, job-sent-to-printer and
180 job-held? Job-queued is really the default. Job-held can only be changed by an operator in an
181 unspecified manner. Should a job move to the printing state when a spooler sends it to a printer or
182 move to the printing state when marking starts. Should it depend on the printer as to whether it
183 announces such nuances. We probably need a state transition diagram.

184 The following standard values are defined:

<u>none</u>	<u>No reasons are currently indicated.</u>
job-incomplete	The job has been accepted by the server, but the Printer is waiting for the transfer of the remainder of the job.
<u>job-spooling</u>	
<u>job-queued</u>	
<u>job-sent-to-printer</u>	
<u>job-held</u>	
job-sending-to-output-device	Optional. The job is in the pending state but is being transmitted to the output device.
job-queued-in-output-device	Optional. The job is in the pending state and is queued on the output device.
job-printing	The job-state is processing, and the printer is marking media. This attribute is optional and useful for printers which spend a great deal of time processing when no marking is happening.
printer-stopped	The printer is stopped. This reason appears in all jobs in the pending state when the value of the Printer's printer-state attribute is stopped. This reason appears in a job in the processing state when the output-device is stopped.
printer-partly-stopped	Optional. This reason appears in all jobs in the pending state when the value of the Printer's printer-state-reasons contains the value partly-stopped.
job-hold-until-specified	The value of the job's job-hold-until attribute has specified a time period that has not yet arrived, so that the Printer shall not consider the job as a candidate for processing.
required-resources-not-ready	At least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.
successful completion	The job completed successfully.
completed-with-warnings	The job completed with warnings.

completed-with-errors	The job completed with errors (and possibly warnings too).
cancelled-by-user	The job was cancelled by the user using the CancelJob request.
cancelled-by-operator	The job was cancelled by the operator using the CancelJob request.
aborted-by-system	The job was aborted by the system.
logfile-pending	The job's logfile is pending file transfer.
logfile-transferring	The job's logfile is being transferred.

185 Here is the Job Monitoring MIB definition of the **jobStateReasons1** attribute and the defined values:
186

jobStateReasons1(5), -- **JmJobStateReasons1TC** (pg 14)
-- OCTETS: Additional information about the job's current state that augments the
-- **jmJobState/jobState** object/attribute. The **jobStateReasons1** attribute identifies the reason
-- or reasons that the job is in the **held, pending, processing, needsAttention, canceled, or**
-- **completed** state. The agent shall indicate the particular reason(s) by setting the value of the
-- **jobStateReasons1** attribute. When the job does not have any reasons for being in its
-- current state, the agent shall set the value of the **jobStateReasons1** attribute to **0**.
--
-- While the job states cannot be added to without impacting deployed clients, it is the intent
-- that additional **JmJobStateReasons1TC** bit values can be defined without impacting
-- deployed clients. In other words, the **JmJobStateReasons1TC** TC is intended to be
-- extensible.
--
-- Companion job state reasons TCs: **JmJobStateReasons2TC, JmJobStateReasons3TC,**
-- **JmJobStateReasons4TC,** are defined/reserved for additional $31*3 = 93$ job state reasons
-- for use with the corresponding attributes: **jobStateReasons2, jobStateReasons3,** and
-- **jobStateReasons4**. This is a type 2 bit definition. See section **Error! Reference source**
-- **not found.** on page **Error! Bookmark not defined..**

187
188 **JmJobStateReasons1TC ::= TEXTUAL-CONVENTION**

189 STATUS current
190 DESCRIPTION

191 "This textual-convention is used with the **jobStateReasons1** attribute to provides additional
192 information regarding the **jmJobState/jobState** object/attribute. The **jobStateReasons1**
193 attributes identifies the reason or reasons that the job is in the **held, pending, processing,**
194 **printing, needsAttention, canceled, or completed** state. The server shall indicate the
195 particular reason(s) by setting the value of the **jobStateReasons1** attribute. While the job
196 states cannot be added to without impacting deployed clients, it is the intent that additional
197 **JmJobStateReasons1TC** enums can be defined without impacting deployed clients. In
198 other words, the **JmJobStateReasons1TC** is intended to be extensible.

199
200 When the job does not have any reasons for being in its current state, the server shall set the
201 value of the **jobStateReasons1** attribute to zeros.

202
203 Companion job state reasons TCs: **JmJobStateReasons2TC, JmJobStateReasons3TC,**
204 **JmJobStateReasons4TC,** are defined/reserved for additional $31*3 = 93$ job state reasons.
205 This is a type 2 bit definition. See section **Error! Reference source not found.** on page
206 **Error! Bookmark not defined..**

207
208 The following standard values are defined (in hexadecimal) as *powers of two*, since multiple
209 values may be used at the same time:

210

other	0x1
The job state reason is not one of the standardized or registered reasons.	
unknown	0x2
The job state reason is not known to the agent.	
documentsNeeded	0x4

The job is in the held state because the server or device is waiting for the job's files to start and/or finish being transferred before the job can be scheduled to be processed.

- jobHoldSet** **0x8**
The job is in the **held** state because the client specified that the job is to be held.
- jobProcessAfterSpecified** **0x10**
The job is in the **held** state because the client specified a time specification reflected in the value of the job's **jobProcessAfterDateAndTime** attribute that has not yet occurred.
- requiredResourcesNotReady** **0x20**
The job is in the **held** state because at least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not ready on any of the physical devices for which the job is a candidate.
- successfulCompletion** **0x40**
The job is in the **completed** state having completed successfully.
- completedWithWarnings** **0x80**
The job is in the **canceled** or **completed** states having completed with warnings.
- completedWithErrors** **0x100**
The job is in the **canceled** or **completed** states having completed with errors (and possibly warnings too).
- canceledByUser** **0x200**
The job is in the **canceled**, state having been canceled by the user.
- canceledByOperator** **0x400**
The job is in the **canceled** state having been canceled by the operator.
- abortedBySystem** **0x800**
The job is in the **canceled**, state having been aborted by the system.
- logfilePending** **0x1000**
The job's logfile is pending file transfer.
- logfileTransferring** **0x2000**
The job is in the **canceled** or **completed** states and the job's logfile is being transferred.

The following additional job state reasons have been added to specify sub-states of the **held** or **completed** states that may be used to represent states that are in ISO DPA[2]:

- jobPreProcessing** **0x4000**
The job has been created on the server or device but the submitting client is in the process of adding additional job components and no documents have started processing. The job maybe in the process of being checked by the server/device for attributes, defaults being applied, a device being selected, etc.
- jobPaused** **0x8000**
The job has been indefinitely suspended by a client issuing an operation to suspend the job so that other jobs may proceed using the same devices. The client may issue an operation to resume the paused job at any time, in which case the server or device places the job in the **held** or **pending**

states and the job is eventually resumed at the point where the job was paused.

jobInterrupted

0x10000

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

jobRetained

0x20000

The job is being retained by the server or device after processing and all of the media have been successfully stacked in the output bin(s).

The job (1) has completed successfully or with warnings or errors, (2) has been aborted while **processingprinting** by the server/device, or (3) has been canceled by the submitting user or operator before or during processing. The job's **jobStateReasons1** attribute shall contain the reasons that the job has entered the **completed** state.

While the jobRetained state reason is , all of the job's document data (and submitted resources, such as fonts, logos, and forms, if any) are retained by the server or device; thus a client could issue an operation to resubmit the job (or a copy of the job).

The following job state reasons have been added to align with the Internet Printing Protocol (IPP):

jobHoldUntil

0x40000

The **Error! Reference source not found.** attribute was specified for a named time period that is still in the future. The job remains in the held state until the time period arrives and there are no other reasons to hold the job.

211
212
213
214

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

215 -- Table 1 - **JmJobStateReasons1TC**: Legal Job States for each Job State
 216 -- Reason
 217

--	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

218 Here are the IPP job-state-message and the JMP processingMessage attributes, which are similar enough.
219 in specification. Both are multi-valued. Should we align the names?

220

221 6.3.2.7 job-state-~~messageas-text~~ (text, MANDATORY)

222 This attributes specifies the job-state and job-state-reasons in human readable text and it shall be set
223 by the Printer.

224

225 Here is the Job Monitoring MIB definition of the processingMessage attribute which corresonds to the
226 IPP **job-state-message** attribute:

```
processingMessage(11), -- OCTET STRING(SIZE(0..63))
-- OCTETS: MULTI-ROW: A coded character set message that is generated during the processing of
-- the job as a simple form of processing log to show progress and any problems.
--
-- There is no restriction on the same message in multiple rows.
--
--
```

227