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

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

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)**

2. We agreed on the simplified job state transition diagram sentence at the end explaining the transitions into the canceled state that are not shown:

For JMP:

The following figure shows the normal job state transitions:

```
                                                        +----> canceled(7)
                                                       /
    +----> pending(3) ---------> processing(5) -------+------> completed(9)
    |         ^                       ^                \
--->+         |                       |                 +----> aborted(8)
    |         v                       v                /
    +----> pendingHeld(4)    processingStopped(6) ----+
```

Figure 1 - Normal Job State Transitions

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

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

For IPP:

The following figure shows the normal job state transitions:

```
                                                  +--> canceled
                                                 /
    +---> pending --------> processing ---------+-----> completed
    |        ^                   ^               \
--->+        |                   |                +---> aborted
    |        v                   v               /
    +---> pending-held   processing-stopped ----+
```

Figure 2 - Normal Job State Transitions

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

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

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

The corresponding sentence for IPP will be:

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

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

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

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

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

The following standard values are defined and MAY be used with any job state for which the reason makes sense.

5. We agreed that job-state-reasons are all OPTIONAL.

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

Implementation of these values is OPTIONAL, i.e., an agent NEED NOT implement them, even if the device supports the functionality represented by the reason and is available to the agent.

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

Implementation of these values is OPTIONAL, i.e., the Printer object NEED NOT return them, even if the output device supports the functionality represented by the reason and is available to the Printer object software.

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

1. In JMP remove the 'printing' state.

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

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

4. In both IPP and JMP add the 'aborted' state and make it a final state.

5. In both IPP and JMP remove the 'aborted-by-system' job-state-reason, since the new 'aborted' state says it all.

6. In IPP replace the 'terminating' state with the 'canceled' and 'aborted' states and make 'canceled' and 'aborted' final states, like the JMP 'canceled' and 'completed' states.

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

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

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

Changes to IPP "job-state" attribute

The following text is copied from the IPP Internet-Draft, 6/3/97. The revision marks show the changes to reflect the above agreements to the job-state attribute.

6.3.2.5 job-state (type1 keyword)

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

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

Standard values are:

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

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

'pending-held': 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 "job-state-reason" attribute SHALL indicate why the job is no longer a candidate for processing.

'processing ': 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 OR

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 printer's "printer-state", "printer-state-reasons", and "printer-state-message" attributes.

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

'processing-stopped': 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 "job-state-reason" attribute MAY indicate why the job has stopped processing. For example, if the output device is stopped, the 'printer-stopped' value MAY be included in the job's "job-state-reasons" attribute. For example, if the output device is stopped, the 'printer-stopped' value MAY be included in the job's "job-state-reasons" attribute.

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

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

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

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

The following figure shows the normal job state transitions.

```
                                                +----> canceled
                                               /
+----> pending --------> processing ---------+------> completed
```

```
    |         ^                   ^                \
--->+         |                   |                 +----> aborted
    |         v                   v                /
    +----> pending-held    processing-stopped ----+
```

Figure 3 - Normal Job State Transitions

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

Changes to JMP **jmJobState** object and **JmJobStateTC** description

The following shows the changes for the JMP **jmJobState** object and the corresponding **JmJobStateTC** textual-convention for the Job Monitoring MIB (there is no longer a **jobState** attribute):

```
jmJobState OBJECT-TYPE
    SYNTAX     JmJobStateTC          -- See page 8
    MAX-ACCESS  read-only
    STATUS     current
    DESCRIPTION
        "The current state of the job (pending, processing,
        completed, etc.).  Even though the JmJobStateTC textual-
        convention defines nine values for job states, agents
        SHALL only implement those states which are appropriate
        for the particular implementation.  In other words, all
        possible enums for this object SHALL be reported if
        implemented by the device and available to the agent.
        However, management applications SHALL be prepared to
        receive all the standard job states.

        The final value for this object SHALL be one of:
        completed, canceled, or aborted.  The minimum length of
        time that the agent SHALL keep a job in the completed,
        canceled, or aborted state before removing the job from
        the jmJobIDTable and jmJobTable is specified by the value
        of the jmGeneralJobPersistence object."
    ::= { jmJobEntry 1 }


JmJobStateTC ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "The current state of the job (pending, processing,
        completed, etc.).

        The following figure shows the normal job state
        transitions:

                                                           +----> canceled(7)
                                                          /
    +----> pending(3) ---------> processing(5) -------+------> completed(9)
    |        ^                       ^                \
--->+        |                       |                 +----> aborted(8)
    |        v                       v                /
    +----> pendingHeld(4)    processingStopped(6) ----+
```

Figure 4 - Normal Job State Transitions

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

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

-- 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 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.

         }
```

Here is the updated Job Monitoring MIB appendix:

Appendix A - Instrumenting the Job Life Cycle

Instrumenting the Job Life Cycle

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

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

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

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

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

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

Revised IPP job-state-reasons and JMP jobStateReasons1

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

The comparison of the IPP "job-state-reasons" attribute and the JMP **jmJobStateReasons1** object after the IPP/JMP telecon, 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

Now both IPP "job-state-reasons" attribute and JMP **jmJobStateReasons1** object are Mandatory.

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

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

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

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

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

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

The following standard values are defined:

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

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

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

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

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

'resources-are-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. This condition MAY be detected when the job is accepted, or subsequently while the job is pending or processing, depending on implementation.

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

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

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

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

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

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

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

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

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

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

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

```
jmJobStateReasons1 OBJECT-TYPE
    SYNTAX     JmJobStateReasons1TC          -- See page 19
    MAX-ACCESS  read-only
    STATUS     current
    DESCRIPTION
        "Additional information about the job's current state,
        i.e., information that augments the value of the job's
        jmJobState object.

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

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

        NOTE - While values cannot be added to the jmJobState
        object without impacting deployed clients that take
        actions upon receiving jmJobState values, it is the
        intent that additional JmJobStateReasonsnTC enums can be
        defined and registered without impacting such deployed
        clients.  In other words, the jmJobStateReasons1 object
        and jobStateReasonsn attributes are intended to be
        extensible."
    ::= { jmJobEntry 2 }


JmJobStateReasons1TC ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "This textual-convention is used with the
        jmJobStateReasons1 object to provides additional
        information regarding the jmJobState object values.

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

```

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

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

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 or is indeterminent.

jobIncoming **0x4**

The job has been accepted by the server or device, but the server or device is expected (1) additional operations to finish creating the job and/or (2) is accessing/accepting document data.

jobOutgoing **0x8**

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

jobHoldSpecified **0x10**

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

jobHoldUntilSpecified **0x20**

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

jobProcessAfterSpecified **0x40**

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

resourcesAreNotReady **0x80**

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. This condition MAY be detected when the job is accepted, or subsequently while the job is pending or processing, depending on implementation.

deviceStoppedPartly **0x100**

One or more, but not all, of the devices to which the job is assigned are stopped. If all of the devices are stopped (or the only device is stopped), the **deviceStopped** reason SHALL be used.

deviceStopped **0x200**

The device(s) to which the job is assigned is (are all) stopped.

jobPrinting **0x400**

The output device is marking media. This attribute is useful for servers and output devices which spend a great deal of time processing when no marking is happening and then want to show that marking is now happening.

jobCanceledByUser **0x800**

The job was canceled by the user, i.e., by a user whose name is the same as the value of the job's **jobOwner** attribute.

jobCanceledByOperator **0x1000**

The job was canceled by the operator, i.e., by a user whose name is different than the value of the job's **jobOwner** attribute.

abortedBySystem **0x2000**

The job was aborted by the system. NOTE - this reason is needed only when the job is not placed in the **aborted** job state.

jobCompletedSuccessfully **0x4000**

The job completed successfully.

jobCompletedWithWarnings **0x8000**

The job completed with warnings.

jobCompletedWithErrors **0x10000**

The job completed with errors (and possibly warnings too).

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

jobPaused **0x20000**

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 agent SHALL remove the **jobPaused** values from the job's **jmJobStateReasons1** object and the job is eventually resumed at or near the point where the job was paused.

jobInterrupted **0x40000**

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 0x80000

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

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

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

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