PWG WHITE PAPER

jmp-activity-accounting-01.doc .pdf

T. Hastings
Dan Sorentino
Xerox Corporation

June 30, 1998

Proposed Activity-based Accouting Attributes for the PWG Job Monitoring MIB

Version 0.01

1. Introduction

This white paper introduces accounting attributes that can be used for multiple types of job services, not just printing. Furthermore, these accounting attributes can be used by multi-function services, i.e., by a job service that has more than one type of service. Examples of multi-function services are scan, print, fax-in, and fax-out. Therefore, a single job may have accounting attributes from several types of job services. In addition, the accounting attributes are at a finer grain than the job, so that different charges can be assigned to different usages.

2. Tracking Attributes at the Job Activity Instance Level

There are two levels of job accounting for multi-function jobs, i.e., jobs that (1) scan and print or (2) send fax and print, etc:

(1) At the job level for the combined services

(2) At a more detailed level for each type of service

Rather than change the current single valued attributes in the PWG Job Monitoring MIB from single values to multi-valued, it is useful to add additional attributes to collect this more detailed data. For example, the current Job Monitoring MIB keeps the number of sheets completed at the job level for all sizes and types combined into a single number. The more detailed attributes keep sheets completed separately for each type and size of medium that a job uses. The more detailed attributes permit an accounting application to charge differently for different size and types of media.

Having two levels of accounting attributes also permits two levels of agent implementation. The simpler agent implementation only maintains accounting attributes at the job level, using attributes in the current Job Monitoring MIB. A more advanced agent implementation could maintain both the job level attributes and the detailed attributes.

2.1.1 Separate Activities for each combination of attributes and documents

The additional attributes are defined for what is called an "activity". An "activity instance" is the combination of a document representation instance (hardcopy or electronic) and a set of document service instructions. If a job has n documents, then there are n activity instances for that job. If a job has one document, say a slide presentation, but two sets of document service instructions, such as (1) print one copy of

the document one-sided on transparencies, and (2) print 20 copies of the same document two-sided, two-up, and stapled, there would be two activity instances for the job. Another example of multiple activities, is a job that is sending a document to 20 fax numbers. Each number represents a separate activity instance. Another example, is a single document that uses several different media. Each different use is recorded as a different activity, so that an advanced accounting application can apply different charges for different media.

A number of counts can be associated with a single activity instance for a job using the MULTI-ROW capability of the PWG Job Monitoring MIB attribute mechanism. Each activity instance type can have one attribute for a job, with multiple values, one value for each activity instance. For example the number of sheets completed for a printing activity would be one of the two values for the activitySheetsCompleted attribute and the number of sheets completed for the fax activity would be the second value. Another example of an activity instance would be when a document print one page range on “A” sized sheets another page range on “C” sized sheets—the change of sheet size causes another activity instance to be generated.

As with all integer attributes, if the agent does not know the value of an integer counter until the activity completes, the agent SHALL return the -2 (unknown) value until it knows a correct value.

The types of document activities are (from the PWG Job Monitoring MIB):

Bit Meaning	Bit Value for Activity	Specification
Other	0x1	that are not one of the following identified types:
Unknown	0x2	whose type is unknown to the agent.
Print	0x4	that specify printing
Scan	0x8	that specify scanning
faxIn	0x10	that specify receive fax
faxOut	0x20	that specify sending fax
getFile	0x40	that specify accessing files or documents
putFile	0x80	that specify storing files or documents
mailList	0x100	that specify distribution of documents using an electronic mail

2.2 Attribute Naming Conventions for Activity Attributes

If the attribute name starts with "**activity**" then it is multi-valued and the values are per *activity instance* for the job. So if a job has: 2 scan, 1 faxIn, and 4 print activity instances, then each "activityXxxYxx" attribute has 7 values: the first two for scan, the next one for faxIn, and the last four for print. Example: **activityKOctetsProcessed** would have seven values for such a multi-function job.

If an attribute name contains "**activity**" but does not start with "**activity**", that attribute is a job level attribute that is a summary job attribute for all activities. Example: **numberOfActivitiesCompleted** is a single-valued job attribute that is the total number of activities completed for that job so far.

2.3 Activity Instantiation Rules

Any time that there is a need for a different value for an activity variables (e.g. a different sheet size), the agent instantiates an entire new row for all activity variables for that job. For example, if an activity instance uses one size of paper and then another size, an entire row of activity variables must be instantiated, just because of the different paper sizes. This helps to ensure that there is a consistent characterization of all the activity attributes. With this type of convention the MIB will be able to track the number of sheets completed for size 1, bond, simplex , and the number of sheets completed for size 2, transparency, duplex, for example. The agent shall set up a separate activity for each pair of document instance and set of document service instructions. If there are multiple documents in the job, the processing of each document will be a separate activity, even though the processing instructions are identical.

There is nothing to say that a running activity cannot have more than one change happen in more than one variable *at the same time*. In this case, just one row would be generated, which would accommodate this request.

2.3.1 Linked MULTI-ROW Values for Activity Attributes

All activity attributes are MULTI-ROW attributes and are intended to be used together. The ith value of all activity attributes for a particular job represent one activity instance. Thus the ith value for each activity attribute is related to the ith value of every other activity attribute and the ith values represent one activity instance.

2.4 Comparison of PWG Job and Proposed Activity attributes

Many of the proposed activity attributes have corresponding existing job level attributes. Table 1 shows the relationship between existing PWG Job Monitoring MIB job attributes and the proposed activity attributes. The attributes are grouped as in the current Job Monitoring MIB in order to show the parallelism between the proposed activity attributes and the current job attributes. The enum range is showed in parentheses. All of the MANDATORY job objects are included. Some of the job objects do not have corresponding activity attributes. These are indicated as N/A in the activity column. No job attribute is included, unless there is a corresponding activity attribute proposed.

1

2 **Table 1 - Current and Proposed MIB attributes**

PWG Job Mon Job Object or Attribute	Corresponding Proposed Activity Attribute
Job and state attributes (3-19)	
jobState	activityState
	activityRetriesRequested
	activityRetriesAttemptedg
	numberOfActivitiesRequested
	numberOfActivitiesTransfered
	numberOfActivitiesCompleted
Identification attributes (20-49)	
jmJobIDJobIndex	N/A
jmJobOwner	N/A
jobAccountName	N/A
jmJobName	N/A
jobServiceTypes	activityServiceTypes
deviceNameRequested	activityJobServiceName
	activityRemoteID
	activityDestinationPhoneNumber
	activityURI
filename	activityFileName
Parameter attributes (50-67)	
finishing	
sides	activitySides
	activityInputSourceTypeUsed
	activityDestinationPhoneLineType
	activityLineProtocol
	activityTransmissionSpeed
	activityCompressionType
Image Quality attributes (70-89)	
printerResolutionUsed	activityResolution
	activityMagnificationsUsed
Progress attributes (90-109)	
jobCopiesCompleted	
jmJobKOctetsProcessed	activityKOctetsProcessed
	activityMultipleCopiesOnePass
Impression attributes (110-129)	
jmJobImpressionsCompleted	activityImpressionsCompleted
	activityNonBlankImpressionsCompleted
fullColorImpressionsCompleted	activityFullColorImpressionsCompleted
highlightColorImpressionsCompleted	activityHighlightColorImpressionsComplet

PWG Job Mon Job Object or Attribute	Corresponding Proposed Activity Attribute
	ed
Page attributes (130-149)	
pagesCompleted	activityPagesCompleted
Sheet attributes (150-169)	
sheetsCompleted	activitySheetsCompleted
Resource attributes (170-189)	
mediumTypeConsumed	activityMediumType
mediumSizeConsumed	activityMediumSize
Time attributes (190-209)	
	activityStartedTransferringTime
jobStartedProcessingTime	activityStartedProcessingTime
jobCompletionTime	activityCompletionTime
jobProcessingCPUTime	activityProcessingCPUTime

3. Textual Conventions

The following textual conventions are needed by some of the proposed activity attributes:

The following textual convention is needed:

```
JmInputSourceTypeTC {xe " JmInputSourceTypeTC "} ::=
TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The type of input source used to scan the hardcopy
        document.  Often the type can be sensed by the
        device and/or input scanning software.

        other(1),
            Some other input source type besides one of the
            specified or registered values.

        unknown(2),
            The input source type is unknown.

        singleSheetFeeder(3),
            Single sheet feeder

        automaticDocumentHandler(4),
            An automatic document handler (ADH) capable of
            feeding a stack of sheets and scanning one side
            only.

        duplexAutomaticDocumentHandler(5),
            An automatic document handler (DADH) capable of
            feeding a stack of sheets and scanning both
            sides of each sheet.

        platen(6)
            A single sheet of the input document was placed
            on the platen surface.
    REFERENCE
        "This is a type 2 enumeration.  See Section Error!
        Reference source not found.."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
        singleSheetFeeder(3),
        automaticDocumentHandler(4),
        duplexAutomaticDocumentHandler(5),
        platen(6)
    }


JmDestinationPhoneLineTypeTC {xe "
JmDestinationPhoneLineTypeTC "} ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The physical nature of the phone line, if any.
```

```
1
2           other(1),
3               Some other type of phone line besides one of the
4               specified or registered values.
5
6           unknown(2),
7               The physical phone line type is unknown.
8
9           none(3),
10              There is no physical phone line associated with
11              this activity.
12
13          pstn(4),
14              Public Switched Telephone Network.
15
16          isdn(5),
17              Integrated Services Digital Network.
18
19          internet(6)
20              Internet.
21
22      REFERENCE
23          "This is a type 2 enumeration.  See Section Error!
24          Reference source not found.."
25      SYNTAX      INTEGER {
26          other(1),
27          unknown(2),
28          none(3),
29          pstn(4),
30          isdn(5),
31          internet(6)
32      }
33
34
35
36  JmCompressioTypeTC {xe " JmCompressioTypeTC "} ::= TEXTUAL-
37  CONVENTION
38      STATUS      current
39      DESCRIPTION
40          "The compression type.
41
42          other(1),
43              Some other type of compression besides one of
44              the specified or registered values.
45
46          unknown(2),
47              The compression type is unknown.
48
49          none(3),
50              There is no compression associated with this
51              activity.
52
53          g3mh(4),
54              CCITT Group 3 lossless facsimile standard with
```

```
            Modified Huffman (MH) coding.  The scan runs are
            compressed using a run length coding algorithm
            and a Huffman procedure uses variable length
            codes to represent the run lengths.

        g3mr(5),
            CCITT Group 4 lossless two-dimensional facsimile
            standard called Modified Read (Relative Element
            Address Designate) which incorporates the
            correlation between scan lines for compression
            wherever possible prior to the use of the MH
            algorithm.

        g4(6),
            CCITT Group 4 lossless two-dimensional facsimile
            standard that is often called MMR (Modified
            Modified Read). It is very similar to G3MR, but
            only two-dimensional lines are transmitted
            without any EOL (End-Of-Line).

        jpeg(7),
            An ISO standard lossy compression scheme from
            the Joint Photographic Expert Group (JPEG) that
            is best suited for the photographic images. It
            involves the application of the Discrete Cosine
            transform (DCT) on subblocks of the image and
            quantization and encoding of the transformed
            coefficients.

        jbig(8),
            AN ISO standard from the Joint Bi-Level Image
            Group (JBIG) that is best suited for Binary and
            Halftone images. It is a progressive bi-level
            coding technique that consists of reducing the
            resolution of bi-level images repeatedly so that
            each reduction has half number of pixels. The
            base line is transmitted losslessly and
            consecutive layers are sent by predictive
            coding.

        lzw(9),
            A lossless defacto standard compression
            algorithm (Lempel-Zeeve-Walsh) that looks for
            patterns in the image; builds a codebook of
            patterns, replaces other instances of the
            patterns by codebook entries and modifies
            codebook dynamically.

    REFERENCE
        "This is a type 2 enumeration.  See Section Error!
        Reference source not found.."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
```

```
        none(3),
        g3mh(4),
        g3mr(5),
        g4(6),
        jpeg(7),
        jbig(8),
        lzw(9)
    }


JmLineProtocolTypeTC {xe " JmLineProtocolTypeTC "} ::=
TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The base protocol used by the activity to transmit
        or receive data.

        other(1),
            Some other type of phone line besides one of the
            specified or registered values.

        unknown(2),
            The physical phone line type is unknown.

        none(3),
            There is no physical phone line associated with
            this activity.

        g3(4),
            CCITT Group 3 lossless facsimile standard.

        g4(5),
            CCITT Group 4 lossless two-dimensional facsimile
            standard.

        salutation(6)
            Protocol defined by the Salutation Consortium.

        postScriptFax(7)
            Adobe PostScript FAX.
    REFERENCE
        "This is a type 2 enumeration.  See Section Error!
        Reference source not found.."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
        none(3),
        g3(4),
        g4(5),
        salutation(6),
        postScriptFax(7)
    }
```

4. Attributes Proposed to be registered with the PWG for the Job Monitoring MIB

This sub-section contains the accounting attributes that are to be proposed to the PWG for registration with the PWG Job Monitoring MIB. Attributes that are already in the PWG Job Monitoring MIB are *not* listed here. See the table above.

NOTE: All of the new activity attributes are defined to work with any kind of job service. There are no activity specific attributes (e.g. no attributes *just for* print or fax).

4.1 Job State Attributes (3-19)

4.1.1 activityState

```
activityState(??),{tc "activityState   (JmJobStateTC)"}{xe
"activityState"}                    JmJobStateTC
INTEGER:  MULTI-ROW:  The current state of the activity
(pending, processing, completed, etc.).  Agents SHALL implement
only those states which are appropriate for the particular
implementation.  However, management applications SHALL be
prepared to receive all the standard activity states.

The initial value for this attribute SHALL be one of pending,
pendingHeld, or processing.  The processingStopped MAY be an
intermediate state.

The final value for this attribute SHALL be one of: completed,
canceled, or aborted.  The minimum length of time that the agent
SHALL maintain MIB data for an activity in the completed,
canceled, or aborted state before removing the attribute data
from the jmAttributeTable is specified by the value of the
jmGeneralAttributePersistence object.

This attribute is for activities what jmJobState is for jobs.
```

4.1.2 activityRetriesRequested

```
activityRetriesRequested(??),{tc " activityRetriesRequested
    (Int32(-2..2))"}{xe " activityRetriesRequested "}
                                    Integer32 (-2..2)
    INTEGER:  MULTI-ROW:  The number of times an activity
    instance is requested to be attempted (e.g. phone dialings
    to attempt) at the beginning of processing the activity.
```

4.1.3 activityRetriesAttempted

```
activityRetriesAttempted(??),{tc " activityRetriesAttempted
    (Int32(-2..2))"}{xe " activityRetriesAttempted "}
                                    Integer32 (-2..2)
    INTEGER:  MULTI-ROW:  The number of times an activity
    instance has been attempted so far.
```

Note: Updating this attribute does not create a new activity.

4.1.4 numberOfActivitiesRequested

numberOfActivitiesRequested(??){tc " numberOfActivitiesRequested (Int32(-2..))"}{xe " numberOfActivitiesRequested "} Integer32 (-2..2147483647)

INTEGER: The number of activities that this job will use or is using, i.e., the number of rows representing separate activities for this job.

NOTE: All activities have the same number of rows for a given job, namely, the number specified by this job attribute.

The agent SHALL set up a separate activity for each pair of document instance and set of document service instructions. If there are multiple documents in the job, the processing of each document will be a separate activity, even though the processing instructions are identical.

4.1.5 numberOfActivitiesTransfered

numberOfActivitiesTransfered(??){tc " numberOfActivitiesTransfered (Int32(-2..))"}{xe " numberOfActivitiesTransfered "} Integer32 (-2..2147483647)

INTEGER: The number of activities that have transferred all their data for this job.

Use this attribute only for activity types that are not completed when the data transfer is completed, such as 'print' or 'faxOut'. For document activity types for which transferring and processing are the same, i.e., the activity state is 'processing' during transfer, such as for scan or faxIn, use numberOfActivitiesCompleted, so that all service types will support numberOfActivitiesCompleted, at least.

4.1.6 numberOfActivitiesCompleted

numberOfActivitiesCompleted(??){tc " numberOfActivitiesCompleted (Int32(-2..))"}{xe " numberOfActivitiesCompleted "} Integer32 (-2..2147483647)

INTEGER: The number of activities that have completed for this job, i.e., 'completed', 'canceled', or 'aborted'.

For document activity types for which transferring and processing are the same, i.e., the activity state is 'processing' during transfer, such as for 'scan' or 'faxIn', use this attribute (instead of **numberOfActivitiesTransfered**) so that all service types support this attribute, at least.

4.2 Identification attributes (20-49)

4.2.1 activityServiceTypes

activityServiceTypes(??),{tc "activityServiceTypes (JmJobServiceTypesTC)"}{xe "activityServiceTypes"} JmJobServiceTypesTC

INTEGER: MULTI-ROW: Specifies the type(s) of service of the activity (print, fax, scan, etc.). The service type is bit encoded with each service type so that more general and arbitrary services can be created, such as services with more than one destination type, or ones with only a source or only a destination. For example, an activity might scan, faxOut, and print. In this case, three bits would be set in the activityServiceTypes attribute, corresponding to the hexadecimal values: 0x8 + 0x20 + 0x4, respectively, yielding: 0x2C.

Whether this attribute is set from an attribute supplied by the job submission client or is set by the recipient job submission server or device depends on the job submission protocol. This attribute SHALL be implemented if the server or device has other types in addition to or instead of printing.

One of the purposes of this attribute is to permit a requester to filter out activities that are not of interest. For example, an accounting program may only be interested in activities that include printing.

4.2.2 activityJobServiceName

activityJobServiceName(??),{tc "deviceNameRequested (JobString63)"}{xe "deviceNameRequested"} JmJobStringTC (SIZE(0..63))

OCTETS: MULTI-ROW: The administratively defined coded character set name of the target device or service that the service provider assigned to the activity.

4.2.3 activityRemoteID

activityRemoteID(??),{tc " activityRemoteID (JobString63)"}{xe " activityRemoteID "} JmJobStringTC (SIZE(0..63))

OCTETS: MULTI-ROW: The name or number of a recipient or sender relative to the device from which data is collected:

For outBound Destinations Remote ID = ID of recipient
For inbound Destinations Remote ID = ID of sender

4.2.4 activityDestinationPhoneNumber

activityDestinationPhoneNumber(??),{tc " activityDestinationPhoneNumber (JobString63)"}{xe "

```
            activityDestinationPhoneNumber "}     JmJobStringTC
            (SIZE(0..63))
            OCTETS:  MULTI-ROW:  The phone number as an alphanumeric
            string, including the ITU PSTN characters for delaying the
            dialing of digits.
```

4.2.5 activityURI

```
        activityURI(??),{tc " activityURI   (Octets(0..63))"}{xe "
            activityURI "}                Integer32 (-2..2147483647)
                                          AND
                                          OCTET STRING(SIZE(0..63))
            INTEGER:  MULTI-ROW:  The index of the next 63 octets of
            this URI, for URIs that exceed 63 octets.  A 0 means that
            there is no next part of this URI.  The agent should
            allocate this overflow index with large values that do not
            conflict with index values for other activities.

            AND

            OCTETS:  MULTI-ROW: The Universal Resource Identifier (URI)
            [RFC-1738] of the service that will perform the document
            activity, i.e., the URI (or URL) of the remote FAX service
            that performs Internet Fax or the URI of the remote printer.

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

            If the URI exceeds 63 octets, the agent SHALL create a new
            row and store its index value in the
            jmAttributeValueAsInteger  with the next 63 octets coming in
            the new row, etc.

            NOTE - IPP [ipp-model] has a 1023-octet maximum length for a
            URI, though the URI standard itself and HTTP/1.1 specify no
            maximum length.
```

4.2.6 ActivityFileName

```
        activityFileName(??),{tc "activityFileName   (JobString63)"}{xe
            "activityFileName"}           JmJobStringTC (SIZE(0..63))
            OCTETS:  MULTI-ROW:  The coded character set file name or
            URI[URI-spec] of the document that the activity is producing
            or consuming.

            There is no restriction on the same file name occurring in
            multiple rows.
```

4.3 Parameter attributes (50-67)

4.3.1 activitySides

```
activitySides(??),{tc "activitySides   (Int32(-2..2))"}{xe
    "activitySides"}              Integer32 (-2..2)
    INTEGER:  MULTI-ROW:  The number of sides, '1' or '2', that
    the document that this activity is producing or consuming
    requires/used.
```

4.3.2 activityInputSourceTypeUsed

```
activityInputSourceTypeUsed(??),{tc "
    activityInputSourceTypeUsed   (JmJobStateReasons2TC)"}{xe "
    activityInputSourceTypeUsed "}   JmInputSourceTypeTC
    INTEGER:  MULTI-ROW:  The type of input source used to scan
    the hardcopy document.
```

4.3.3 activityDestinationPhoneLineType

```
activityDestinationPhoneLineType(??),{tc "
    activityDestinationPhoneLineType
    (JmDestinationPhoneLineTypeTC)"}{xe "
    activityDestinationPhoneLineType "}
                                  JmDestinationPhoneLineTypeTC
    INTEGER:  MULTI-ROW:  The physical nature of the phone line.
```

4.3.4 activityLineProtocol

```
activityLineProtocol(??),{tc " activityLineProtocol
    (JmLineProtocolTypeTC)"}{xe " activityLineProtocol "}
                                  JmLineProtocolTypeTC
    INTEGER:  MULTI-ROW:  The base protocol used by the
    activity.
```

4.3.5 activityTransmissionSpeed

```
activityTransmissionSpeed(??),{tc " activityTransmissionSpeed
    (Int32(-2..))"}{xe " activityTransmissionSpeed "}
                                  Integer32 (-2..2147483647)
    INTEGER:  MULTI-ROW:  The average or nominal speed in bits
    per second that the activity uses to transmit or receive
    data.
```

4.3.6 activityCompressionType

```
activityCompressionType(??),{tc " activityCompressionType
    (JmCompressioTypeTC)"}{xe " activityCompressionType "}
                                  JmCompressioTypeTC
    INTEGER:  MULTI-ROW:  The compression scheme produced by the
```

document activity as an output image or consumed by the document activity from an input image.

4.4 Image Quality attributes (70-89)

4.4.1 activityResolution

activityResolution(??),{tc "activityResolution (JmPrinterResolutionTC)"}{xe "activityResolution"} JmPrinterResolutionTC
OCTETS: MULTI-ROW: The cross-feed and along-feed resolution requested/used for an activity.

4.4.2 activityMagnificationsUsed

activityMagnificationsUsed(??),{tc "activityMagnificationsUsed (Int32(-2..))"}{xe "activityMagnificationsUsed"} Integer32 (-2..2147483647)
INTEGER: MULTI-ROW: The percentage of magnification used by the activity. If < 100, then reduction, if = 100 then no magnification, if > 100 then enlargement.

4.5 Progress attributes (90-109)

4.5.1 activityKOctetsProcessed

activityKOctetsProcessed(??),{tc "activityKOctetsProcessed (Int32(-2..))"}{xe "activityKOctetsProcessed"} Integer32 (-2..2147483647)
INTEGER: MULTI-ROW: The total number of octets processed by the activity instance in units of K (1024) octets so far.

The agent SHALL round the actual number of octets processed up to the next higher K. Thus 0 octets SHALL be represented as '0', 1-1024 octets SHALL be represented as '1', 1025-2048 octets SHALL be '2', etc. For printing devices, this value is the number interpreted by the page description language interpreter rather than what has been marked on media.

4.5.2 activityMultipleCopiesOnePass

activityMultipleCopiesOnePass(??),{tc "activityMultipleCopiesOnePass (JmBooleanTC)"}{xe "activityMultipleCopiesOnePass "} JmBooleanTC
INTEGER: MULTI-VALUE: Indicates when the activity is making multiple copies of a document whether the activity is collating sheets in each document copy by making multiple passes (false(3)) or by using multiple output bins (true(4)).

4.6 Impression attributes (110-129)

4.6.1 activityImpressionsCompleted

activityImpressionsCompleted(??),{tc "
activityImpressionsCompleted (Int32(-2..))"}{xe "
activityImpressionsCompleted "} Integer32 (-2..2147483647)
INTEGER: MULTI-ROW: The number of impressions of media that were impressed and/or processed, in each activity instance. See Section 2 for a definition of the term "impression" and how it applies to two-sided documents. Also see the counting example in Section 3.4 entitled 'Monitoring Job Progress'

For printing devices, the impressions completed includes interpreting, marking, and stacking the output. For other types of job services, the number of impressions completed includes the number of impressions processed.

4.6.2 activityNonBlankImpressionsCompleted

activityNonBlankImpressionsCompleted(??),{tc "
activityNonBlankImpressionsCompleted (Int32(-2..))"}{xe "
activityNonBlankImpressionsCompleted "} Integer32 (-2..2147483647)
INTEGER: MULTI-ROW: The number of non-blank impressions of media that were impressed and/or processed, in each activity instance.

This attribute is defined to count the same as 'impressions', except that the value does NOT include any blank impressions.

For activities which mark the medium (print, faxOut), the value is not incremented when an impressions places no marks on the medium. For activities which do not mark the medium (scan, faxIn,, file), the value is the number of impressions for which data was transformed. For example, if a blank impression is scanned, it is not included in the number; if a blank impression is stored to a file, it is not included in the number.

4.6.3 activityFullColorImpressionsCompleted

activityFullColorImpressionsCompleted(??),{tc
"activityFullColorImpressionsCompleted (Int32(-2..))"}{xe
"activityFullColorImpressionsCompleted"} Integer32 (-2..2147483647)
INTEGER: MULTI-ROW: The number of full color impressions completed by the activity instance so far. For printing, the impressions completed includes interpreting, marking, and stacking the output. For other types of job services,

the number of impressions completed includes the number of impressions processed.

Full color impressions are typically defined as those requiring 3 or more colorants, but this MAY vary by implementation. In any case, the value of this attribute counts by 1 for each side that has full color, not by the number of colors per side (and the other activity impression counters are incremented, except activityHighlightColorImpressionsCompleted(xxx)).

4.6.4 activityHighlightColorImpressionsCompleted

activityHighlightColorImpressionsCompleted(??),{tc "activityHighlightColorImpressionsCompleted(Int32(-2..))"}{xe "activityHighlightColorImpressionsCompleted"} Integer32 (-2..2147483647)

INTEGER: MULTI-ROW: The number of highlight color impressions completed by the activity instance so far. For printing, the impressions completed includes interpreting, marking, and stacking the output. For other types of job services, the number of impressions completed includes the number of impressions processed.

Highlight color impressions are typically defined as those requiring black plus one other colorant, but this MAY vary by implementation. In any case, the value of this attribute counts by 1 for each side that has highlight color (and the other activity impression counters are incremented, except for activityFullColorImpressionsCompleted(xxx))

4.7 Page attributes (130-149)

4.7.1 activityPagesCompleted

activityPagesCompleted(??),{tc "activityPagesCompleted (Int32(-2..))"}{xe "activityPagesCompleted"} Integer32 (-2..2147483647)

INTEGER: The number of logical pages that the activity has processed (consumed or produced) so far, whether one-sided or two-sided.

4.8 Sheet attributes (150-169)

4.8.1 activitySheetsCompleted

activitySheetsCompleted(??),{tc "activitySheetsCompleted (Int32(-2..))"}{xe "activitySheetsCompleted"} Integer32 (-2..2147483647)

INTEGER: The number of medium sheets that the activity has processed (consumed or produced) so far, whether one-sided or two-sided.

4.9 Resource attributes (170-189)

4.9.1 activityMediumType

activityMediumType(??),{tc "activityMediumType (JobString63)"}{xe "activityMediumType"} JmJobStringTC (SIZE(0..63))

OCTETS: MULTI-ROW: the name of the medium type that has been consumed or produced so far whether those sheets have been processed on one side or on both.

NOTE - The type name (JmJobStringTC) values correspond to the type name values of the prtInputMediaType object in the Printer MIB [print-mib]. Values are: 'stationery', 'transparency', 'envelope', etc. These medium type names correspond to the enum values of JmMediumTypeTC used in the mediumRequested attribute.

4.9.2 activityMediumSize

activityMediumSize(??),{tc "activityMediumSize (JobString63)"}{xe "activityMediumSize"} JmJobStringTC (SIZE(0..63))
OCTETS: MULTI-ROW: the name of the medium size that the activity is consuming or producing whether those sheets have been processed on one side or on both.

NOTE - The size name (JmJobStringTC) values correspond to the size name values in the Printer MIB [print-mib] Appendix B. Values are: 'letter', 'a', 'iso-a4', 'jis-b4', etc., which are drawn from ISO DPA.

4.10 Time attributes (190-209)

4.10.1 activityStartedTransferringTime

activityStartedTransferringTime(??),{tc "activityStartedTransferringTime (JmTimeStampTC and/or DateAndTime)"}{xe "activityStartedTransferringTime"}
JmTimeStampTC
AND/OR
DateAndTime
INTEGER: MULTI-ROW: The time
AND/OR
OCTETS: MULTI-ROW: the date and time that the activity started transferring data. Use this attribute only for activity types that are not completed when the data transfer is completed, such as 'print' or 'faxOut'. For document activity types for which transferring and processing are the same, i.e., the activity state is 'processing' during

transfer, such as for scan or faxIn, use **activityStartedProcessingTime**, so that all service types support **activityStartedProcessingTime**, at least.

4.10.2 activityStartedProcessingTime

activityStartedProcessingTime(??),**{**tc "activityStartedProcessingTime (JmTimeStampTC and/or DateAndTime)"**}{**xe "activityStartedProcessingTime"**}**

JmTimeStampTC
AND/OR
DateAndTime

INTEGER: MULTI-ROW: The time
AND/OR
OCTETS: MULTI-ROW: the date and time that the activity started processing data. For document activity types for which transferring and processing are the same, such as 'faxIn' or 'scan', use this attribute (instead of the **activityStartedTransferringTime** attribute) so that that all service types support the **activityStartedProcessingTime** attribute, at least.

4.10.3 activityCompletionTime

activityCompletionTime(??),**{**tc "activityCompletionTime (JmTimeStampTC and/or DateAndTime)"**}{**xe "activityCompletionTime"**}**

JmTimeStampTC
AND/OR
DateAndTime

INTEGER: MULTI-ROW: The time
AND/OR
OCTETS: MULTI-ROW: the date and time that the activity was over, i.e., the activity entered the completed, canceled, or aborted activity state. See the **activityState** attribute.

4.10.4 activityProcessingCPUTime

activityProcessingCPUTime(??)**{**tc "activityProcessingCPUTime (Int32(-2..))"**}{**xe "activityProcessingCPUTime"**}** Integer32 (-2..2147483647)
UNITS 'seconds'
INTEGER: MULTI-ROW: The amount of CPU time in seconds that the activity has been in the processing state. If the activity enters the processingStopped state, that elapsed time SHALL not be included. In other words, the **activityProcessingCPUTime** value SHOULD be relatively repeatable when the same activity is performed again on the same device.