

```

1 Job-Monitoring-MIB DEFINITIONS ::= BEGIN
2 -- Version 1.1 - adds Service and Job traps for IPP Notifications
3 -- (see revision history in MODULE-IDENTITY macro below)
4
5 IMPORTS
6     MODULE-IDENTITY, OBJECT-TYPE,
7     OBJECT-IDENTITY, NOTIFICATION-TYPE,
8     enterprises, Integer32, TimeTicks      FROM SNMPv2-SMI
9     TEXTUAL-CONVENTION                     FROM SNMPv2-TC
10    MODULE-COMPLIANCE, OBJECT-GROUP,
11    NOTIFICATION-GROUP                     FROM SNMPv2-CONF
12    SnmpAdminString                        FROM SNMP-FRAMEWORK-MIB;
13
14    -- The following textual-conventions are needed to implement
15    -- certain attributes, but are not needed to compile this MIB.
16    -- They are provided here for convenience:
17    -- hrDeviceIndex                        FROM HOST-RESOURCES-MIB
18    -- DateAndTime                         FROM SNMPv2-TC
19    -- PrtInterpreterLangFamilyTC,
20    -- CodedCharSet                        FROM Printer-MIB
21
22    -- Use the enterprises arc assigned to the PWG which is pwg(2699).
23    -- Group all PWG mibs under mibs(1).
24
25    jobmonMIB MODULE-IDENTITY
26        LAST-UPDATED "0007060000Z" -- 6 July 2000
27        ORGANIZATION "Printer Working Group (PWG)"
28        CONTACT-INFO
29            "Tom Hastings
30             Postal: Xerox Corp.
31                   Mail stop ESAE-231
32                   701 S. Aviation Blvd.
33                   El Segundo, CA 90245
34
35             Tel:   (301)333-6413
36             Fax:   (301)333-5514
37             E-mail: hastings@cpl10.es.xerox.com
38
39             Ira McDonald
40             Postal: High North Inc.
41                   PO Box 221
42                   Grand Marais, MI 49839
43
44             Tel:   (906)494-2434
45             Email: imcdonal@sdsp.mc.xerox.com
46
47             Send questions and comments to the Printer Working Group (PWG)
48             using the Job Monitoring Project (JMP) Mailing List:
49             jmp@pwg.org
50
51             For further information, including how to subscribe to the
52             jmp mailing list, access the PWG web page under 'JMP':
53
54             http://www.pwg.org/
55
56             Implementers of this specification are encouraged to join the
57             jmp mailing list in order to participate in discussions on any

```

```

58         clarifications needed and registration proposals being reviewed
59         in order to achieve consensus."
60     DESCRIPTION
61         "The MIB module for monitoring job in servers, printers, and
62         other devices.
63
64         Version: 1.1"
65
66     -- revision history
67     REVISION      "0007060000Z" -- 6 July 2000
68     DESCRIPTION
69         "Version 1.1 - adds support for Service and Job events:
70
71         1) Added 'SnmpAdminString' to IMPORTS clause for new objects.
72         2) Corrected OID in MODULE-IDENTITY to use forward reference to
73         definition of 'pwg' from 'enterprises' and 'mibs' from 'pwg'
74         3) Added 'JmServiceStateTC' textual convention
75         4) Added 'jmMirrorAttr' and 'jmSystem' object identifiers
76         reserved for future extensions
77         5) Added Service, Job, Job Completed, and Job Progress traps
78         for job notifications (aligned with IPP Notifications)
79         6) Added Service, Service Event, Job Event, and Job Progress
80         object groups to support above traps"
81     REVISION      "9902190000Z" -- 19 February 1999
82     DESCRIPTION
83         "Version 1.0 - published as RFC 2707"
84
85     ::= { mibs 1 }                -- forward reference
86
87     pwg          OBJECT IDENTIFIER ::= { enterprises 2699 }
88     mibs         OBJECT IDENTIFIER ::= { pwg 1 }
89
90     -- Textual conventions for this MIB module
91
92     JmUTF8StringTC ::= TEXTUAL-CONVENTION
93         DISPLAY-HINT "255a"
94         STATUS      current
95         DESCRIPTION
96             "To facilitate internationalization, this TC represents
97             information taken from the ISO/IEC IS 10646-1 character set,
98             encoded as an octet string using the UTF-8 character encoding
99             scheme.
100
101             See section 3.6.1, entitled: 'Text generated by the server or
102             device'."
103     SYNTAX      OCTET STRING (SIZE (0..63))
104
105     JmJobStringTC ::= TEXTUAL-CONVENTION
106         STATUS      current
107         DESCRIPTION
108             "To facilitate internationalization, this TC represents
109             information using any coded character set registered by IANA as
110             specified in section 3.7. While it is recommended that the
111             coded character set be UTF-8 [UTF-8], the actual coded
112             character set SHALL be indicated by the value of the
113             jobCodedCharSet(8) attribute for the job.
114

```

```

115         See section 3.6.2, entitled: 'Text supplied by the job
116         submitter'."
117     SYNTAX         OCTET STRING (SIZE (0..63))
118
119
120 JmNaturalLanguageTagTC ::= TEXTUAL-CONVENTION
121     STATUS         current
122     DESCRIPTION
123         "An IETF RFC 1766-compliant 'language tag', with zero or more
124         sub-tags that identify a natural language. While RFC 1766
125         specifies that the US-ASCII values are case-insensitive, this
126         MIB specification requires that all characters SHALL be lower
127         case in order to simplify comparing by management applications.
128
129         See section 3.6.1, entitled: 'Text generated by the server or
130         device' and section 3.6.2, entitled: 'Text supplied by the job
131         submitter'."
132     SYNTAX         OCTET STRING (SIZE (0..63))
133
134 JmTimeStampTC ::= TEXTUAL-CONVENTION
135     STATUS         current
136     DESCRIPTION
137         "The simple time at which an event took place. The units are
138         in seconds since the system was booted.
139
140         NOTE - JmTimeStampTC is defined in units of seconds, rather
141         than 100ths of seconds, so as to be simpler for agents to
142         implement (even if they have to implement the 100ths of a
143         second to comply with implementing sysUpTime in MIB-II[mib-
144         II].)
145
146         NOTE - JmTimeStampTC is defined as an Integer32 so that it can
147         be used as a value of an attribute, i.e., as a value of the
148         jmAttributeValueAsInteger object. The TimeStamp textual-
149         convention defined in SNMPv2-TC [SMIV2-TC] is defined as an
150         APPLICATION 3 IMPLICIT INTEGER tag, not an Integer32 which is
151         defined in SNMPv2-SMI [SMIV2-TC] as UNIVERSAL 2 IMPLICIT
152         INTEGER, so cannot be used in this MIB as one of the values of
153         jmAttributeValueAsInteger."
154     SYNTAX         INTEGER (0..2147483647)
155
156 JmJobSourcePlatformTypeTC ::= TEXTUAL-CONVENTION
157     STATUS         current
158     DESCRIPTION
159         "The source platform type that can submit jobs to servers or
160         devices in any of the 3 configurations.
161
162         This is a type 2 enumeration. See Section 3.7.1.2. See also
163         IANA operating-system-names registry."
164     SYNTAX         INTEGER {
165         other(1),
166         unknown(2),
167         sptUNIX(3),           -- UNIX
168         sptOS2(4),           -- OS/2
169         sptPCDOS(5),         -- DOS
170         sptNT(6),           -- NT
171         sptMVS(7),          -- MVS

```

```

172         sptVM(8),           -- VM
173         sptOS400(9),        -- OS/400
174         sptVMS(10),         -- VMS
175         sptWindows(11),     -- Windows
176         sptNetWare(12)     -- NetWare
177     }
178
179 JmFinishingTC ::= TEXTUAL-CONVENTION
180     STATUS         current
181     DESCRIPTION
182         "The type of finishing operation.
183
184         These values are the same as the enum values of the IPP
185         'finishings' attribute.  See Section 3.7.1.2.
186
187         other(1),
188             Some other finishing operation besides one of the specified
189             or registered values.
190
191         unknown(2),
192             The finishing is unknown.
193
194         none(3),
195             Perform no finishing.
196
197         staple(4),
198             Bind the document(s) with one or more staples. The exact
199             number and placement of the staples is site-defined.
200
201         punch(5),
202             Holes are required in the finished document. The exact
203             number and placement of the holes is site-defined. The
204             punch specification MAY be satisfied (in a site- and
205             implementation-specific manner) either by
206             drilling/punching, or by substituting pre-drilled media.
207
208         cover(6),
209             Select a non-printed (or pre-printed) cover for the
210             document. This does not supplant the specification of a
211             printed cover (on cover stock medium) by the document
212             itself.
213
214         bind(7)
215             Binding is to be applied to the document; the type and
216             placement of the binding is product-specific.
217
218         This is a type 2 enumeration.  See Section 3.7.1.2."
219     SYNTAX         INTEGER {
220         other(1),
221         unknown(2),
222         none(3),
223         staple(4),
224         punch(5),
225         cover(6),
226         bind(7)
227     }
228

```

```

229 JmPrintQualityTC ::= TEXTUAL-CONVENTION
230     STATUS      current
231     DESCRIPTION
232         "Print quality settings.
233
234         These values are the same as the enum values of the IPP 'print-
235         quality' attribute.  See Section 3.7.1.2.
236
237         This is a type 2 enumeration.  See Section 3.7.1.2."
238     SYNTAX      INTEGER {
239         other(1),    -- Not one of the specified or registered
240                    -- values.
241         unknown(2), -- The actual value is unknown.
242         draft(3),   -- Lowest quality available on the printer.
243         normal(4),  -- Normal or intermediate quality on the
244                    -- printer.
245         high(5)     -- Highest quality available on the printer.
246     }
247
248 JmPrinterResolutionTC ::= TEXTUAL-CONVENTION
249     STATUS      current
250     DESCRIPTION
251         "Printer resolutions.
252
253         Nine octets consisting of two 4-octet SIGNED-INTEGERS followed
254         by a SIGNED-BYTE.  The values are the same as those specified
255         in the Printer MIB [printmib].  The first SIGNED-INTEGER
256         contains the value of prtMarkerAddressabilityXFeedDir.  The
257         second SIGNED-INTEGER contains the value of
258         prtMarkerAddressabilityFeedDir.  The SIGNED-BYTE contains the
259         value of prtMarkerAddressabilityUnit.
260
261         Note: the latter value is either 3 (tenThousandsOfInches) or 4
262         (micrometers) and the addressability is in 10,000 units of
263         measure.  Thus the SIGNED-INTEGERS represent integral values in
264         either dots-per-inch or dots-per-centimeter.
265
266         The syntax is the same as the IPP 'printer-resolution'
267         attribute.  See Section 3.7.1.2."
268     SYNTAX      OCTET STRING (SIZE(9))
269
270 JmTonerEconomyTC ::= TEXTUAL-CONVENTION
271     STATUS      current
272     DESCRIPTION
273         "Toner economy settings.
274
275         This is a type 2 enumeration.  See Section 3.7.1.2."
276     SYNTAX      INTEGER {
277         unknown(2), -- unknown.
278         off(3),     -- Off.  Normal.  Use full toner.
279         on(4)       -- On.  Use less toner than normal.
280     }
281
282 JmBooleanTC ::= TEXTUAL-CONVENTION
283     STATUS      current
284     DESCRIPTION
285         "Boolean true or false value.

```

```

286
287     This is a type 2 enumeration.  See Section 3.7.1.2."
288 SYNTAX     INTEGER {
289     unknown(2),    -- unknown.
290     false(3),     -- FALSE.
291     true(4)       -- TRUE.
292 }
293
294 JmMediumTypeTC ::= TEXTUAL-CONVENTION
295     STATUS     current
296     DESCRIPTION
297         "Identifies the type of medium.
298
299     other(1),
300         The type is neither one of the values listed in this
301         specification nor a registered value.
302
303     unknown(2),
304         The type is not known.
305
306     stationery(3),
307         Separately cut sheets of an opaque material.
308
309     transparency(4),
310         Separately cut sheets of a transparent material.
311
312     envelope(5),
313         Envelopes that can be used for conventional mailing
314         purposes.
315
316     envelopePlain(6),
317         Envelopes that are not preprinted and have no windows.
318
319     envelopeWindow(7),
320         Envelopes that have windows for addressing purposes.
321
322     continuousLong(8),
323         Continuously connected sheets of an opaque material
324         connected along the long edge.
325
326     continuousShort(9),
327         Continuously connected sheets of an opaque material
328         connected along the short edge.
329
330     tabStock(10),
331         Media with tabs.
332
333     multiPartForm(11),
334         Form medium composed of multiple layers not pre-attached to
335         one another; each sheet MAY be drawn separately from an
336         input source.
337
338     labels(12),
339         Label-stock.
340
341     multiLayer(13)
342         Form medium composed of multiple layers which are pre-

```

```

343         attached to one another, e.g. for use with impact printers.
344
345
346         This is a type 2 enumeration.  See Section 3.7.1.2.  These enum
347         values correspond to the keyword name strings of the
348         prtInputMediaType object in the Printer MIB [print-mib].  There
349         is no printer description attribute in IPP/1.0 that represents
350         these values."
351     SYNTAX      INTEGER {
352         other(1),
353         unknown(2),
354         stationery(3),
355         transparency(4),
356         envelope(5),
357         envelopePlain(6),
358         envelopeWindow(7),
359         continuousLong(8),
360         continuousShort(9),
361         tabStock(10),
362         multiPartForm(11),
363         labels(12),
364         multiLayer(13)
365     }
366
367     JmJobCollationTypeTC ::= TEXTUAL-CONVENTION
368         STATUS      current
369         DESCRIPTION
370             "This value is the type of job collation.  Implementations that
371             don't support multiple documents or don't support multiple
372             copies SHALL NOT support the uncollatedDocuments(5) value.
373
374             This is a type 2 enumeration.  See Section 3.7.1.2.  See also
375             Section 3.4, entitled 'Monitoring Job Progress'."
376     SYNTAX      INTEGER {
377         other(1),
378         unknown(2),
379         uncollatedSheets(3),      -- sheets within each document copy
380                                 -- are not collated: 1 1 ..., 2 2 ...,
381                                 -- No corresponding value of IPP
382                                 -- "multiple-document-handling"
383         collatedDocuments(4),    -- internal collated sheets,
384                                 -- documents: A, B, A, B, ...
385                                 -- Corresponds to IPP "multiple-
386                                 -- document-handling"='separate-
387                                 -- documents-collated-copies'
388         uncollatedDocuments(5)  -- internal collated sheets,
389                                 -- documents: A, A, ..., B, B, ...
390                                 -- Corresponds to IPP "multiple-
391                                 -- document-handling"='separate-
392                                 -- documents-uncollated-copies'
393
394     }
395
396
397     JmJobSubmissionIDTypeTC ::= TEXTUAL-CONVENTION
398         STATUS      current
399         DESCRIPTION

```

400 "Identifies the format type of a job submission ID.  
 401  
 402 Each job submission ID is a fixed-length, 48-octet printable  
 403 US-ASCII [US-ASCII] coded character string containing no  
 404 control characters, consisting of the fields defined in section  
 405 3.5.1.

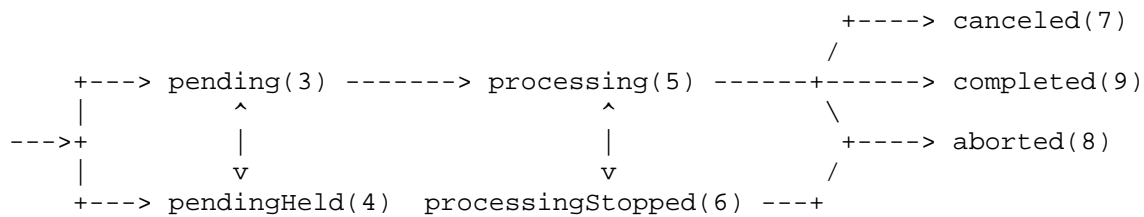
406 This is like a type 2 enumeration. See section 3.7.3."  
 407  
 408 SYNTAX OCTET STRING(SIZE(1)) -- ASCII '0'-'9', 'A'-'Z', 'a'-'z'  
 409

410 JmJobStateTC ::= TEXTUAL-CONVENTION

411 STATUS current

412 DESCRIPTION

413 "The current state of the job (pending, processing, completed,  
 414 etc.). The following figure shows the normal job state  
 415 transitions:  
 416



425 Figure 4 - Normal Job State Transitions

426  
 427 Normally a job progresses from left to right. Other state  
 428 transitions are unlikely, but are not forbidden. Not shown are  
 429 the transitions to the canceled state from the pending,  
 430 pendingHeld, and processingStopped states.

431  
 432 Jobs in the pending, processing, and processingStopped states  
 433 are called 'active', while jobs in the pendingHeld, canceled,  
 434 aborted, and completed states are called 'inactive'. Jobs  
 435 reach one of the three terminal states: completed, canceled, or  
 436 aborted, after the jobs have completed all activity, and all  
 437 MIB objects and attributes have reached their final values for  
 438 the job.  
 439

440  
 441 These values are the same as the enum values of the IPP 'job-  
 442 state' job attribute. See Section 3.7.1.2.  
 443

444 unknown(2),  
 445 The job state is not known, or its state is indeterminate.  
 446

447 pending(3),  
 448 The job is a candidate to start processing, but is not yet  
 449 processing.  
 450

451 pendingHeld(4),  
 452 The job is not a candidate for processing for any number of  
 453 reasons but will return to the pending state as soon as the  
 454 reasons are no longer present. The job's  
 455 jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4)  
 456 attributes SHALL indicate why the job is no longer a



457 candidate for processing. The reasons are represented as  
458 bits in the jmJobStateReasons1 object and/or  
459 jobStateReasonsN (N=2..4) attributes. See the  
460 JmJobStateReasonsNTC (N=1..4) textual convention for the  
461 specification of each reason.

462 processing(5),

463 One or more of:

464  
465 1. the job is using, or is attempting to use, one or  
466 more purely software processes that are analyzing,  
467 creating, or interpreting a PDL, etc.,

468  
469 2. the job is using, or is attempting to use, one or  
470 more hardware devices that are interpreting a PDL,  
471 making mark on a medium, and/or performing finishing,  
472 such as stapling, etc., OR

473  
474 3. (configuration 2) the server has made the job ready  
475 for printing, but the output device is not yet printing  
476 it, either because the job hasn't reached the output  
477 device or because the job is queued in the output  
478 device or some other spooler, awaiting the output  
479 device to print it.

480  
481 When the job is in the processing state, the entire job  
482 state includes the detailed status represented in the  
483 IETF Host MIB indicated by the hrDeviceIndex value of the  
484 job's physicalDevice attribute, if the agent implements  
485 such a device MIB.

486  
487 Implementations MAY, though they NEED NOT, include  
488 additional values in the job's jmJobStateReasons1 object  
489 to indicate the progress of the job, such as adding the  
490 jobPrinting value to indicate when the device is actually  
491 making marks on a medium and/or the processingToStopPoint  
492 value to indicate that the server or device is in the  
493 process of canceling or aborting the job.

494  
495  
496 processingStopped(6),

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

500  
501 The job's jmJobStateReasons1 object and/or the job's  
502 jobStateReasonsN (N=2..4) attributes MAY indicate why the  
503 job has stopped processing. For example, if the output  
504 device is stopped, the deviceStopped value MAY be  
505 included in the job's jmJobStateReasons1 object.

506  
507 NOTE - When an output device is stopped, the device  
508 usually indicates its condition in human readable form  
509 at the device. The management application can obtain  
510 more complete device status remotely by querying the  
511 appropriate device MIB using the job's deviceIndex  
512 attribute(s), if the agent implements such a device MIB

513

514 canceled(7),  
515 A client has canceled the job and the server or device  
516 has completed canceling the job AND all MIB objects and  
517 attributes have reached their final values for the job.  
518 While the server or device is canceling the job, the  
519 job's jmJobStateReasons1 object SHOULD contain the  
520 processingToStopPoint value and one of the  
521 canceledByUser, canceledByOperator, or canceledAtDevice  
522 values. The canceledByUser, canceledByOperator, or  
523 canceledAtDevice values remain while the job is in the  
524 canceled state.  
525

526 aborted(8),  
527 The job has been aborted by the system, usually while the  
528 job was in the processing or processingStopped state and  
529 the server or device has completed aborting the job AND  
530 all MIB objects and attributes have reached their final  
531 values for the job. While the server or device is  
532 aborting the job, the job's jmJobStateReasons1 object MAY  
533 contain the processingToStopPoint and abortedBySystem  
534 values. If implemented, the abortedBySystem value SHALL  
535 remain while the job is in the aborted state.  
536

537

538 completed(9)  
539 The job has completed successfully or with warnings or  
540 errors after processing and all of the media have been  
541 successfully stacked in the appropriate output bin(s) AND  
542 all MIB objects and attributes have reached their final  
543 values for the job. The job's jmJobStateReasons1 object  
544 SHOULD contain one of: completedSuccessfully,  
545 completedWithWarnings, or completedWithErrors values.  
546

547 This is a type 2 enumeration. See Section 3.7.1.2."  
548 SYNTAX INTEGER {  
549 unknown(2),  
550 pending(3),  
551 pendingHeld(4),  
552 processing(5),  
553 processingStopped(6),  
554 canceled(7),  
555 aborted(8),  
556 completed(9)  
557 }  
558

559 JmAttributeTypeTC ::= TEXTUAL-CONVENTION  
560 STATUS current  
561 DESCRIPTION  
562 "The type of the attribute which identifies the attribute.  
563

564 NOTE - The enum assignments are grouped logically with values  
565 assigned in groups of 20, so that additional values may be  
566 registered in the future and assigned a value that is part of  
567 their logical grouping.  
568

569 Values in the range 2\*\*30 to 2\*\*31-1 are reserved for private  
570 or experimental usage. This range corresponds to the same

```

571 range reserved in IPP. Implementers are warned that use of
572 such values may conflict with other implementations.
573 Implementers are encouraged to request registration of enum
574 values following the procedures in Section 3.7.1.
575
576 See Section 3.2 entitled 'The Attribute Mechanism' for a
577 description of this textual-convention and its use in the
578 jmAttributeTable. See Section 3.3.8 for the specification of
579 each attribute. The comment(s) after each enum assignment
580 specifies the data type(s) of the attribute.
581
582 This is a type 2 enumeration. See Section 3.7.1.2."
583
584
585 SYNTAX      INTEGER {
586     other(1),                -- Integer32 (-2..2147483647)
587                               -- AND/OR
588                               -- OCTET STRING(SIZE(0..63))
589
590     -- Job State attributes:
591     jobStateReasons2(3),     -- JmJobStateReasons2TC
592     jobStateReasons3(4),     -- JmJobStateReasons3TC
593     jobStateReasons4(5),     -- JmJobStateReasons4TC
594     processingMessage(6),    -- JmUTF8StringTC (SIZE(0..63))
595     processingMessageNaturalLangTag(7),
596                               -- OCTET STRING(SIZE(0..63))
597     jobCodedCharSet(8),     -- CodedCharSet
598     jobNaturalLanguageTag(9), -- OCTET STRING(SIZE(0..63))
599
600     -- Job Identification attributes:
601     jobURI(20),              -- OCTET STRING(SIZE(0..63))
602     jobAccountName(21),     -- OCTET STRING(SIZE(0..63))
603     serverAssignedJobName(22), -- JmJobStringTC (SIZE(0..63))
604     jobName(23),            -- JmJobStringTC (SIZE(0..63))
605     jobServiceTypes(24),    -- JmJobServiceTypesTC
606     jobSourceChannelIndex(25), -- Integer32 (0..2147483647)
607     jobSourcePlatformType(26), -- JmJobSourcePlatformTypeTC
608     submittingServerName(27), -- JmJobStringTC (SIZE(0..63))
609     submittingApplicationName(28), -- JmJobStringTC (SIZE(0..63))
610     jobOriginatingHost(29),  -- JmJobStringTC (SIZE(0..63))
611     deviceNameRequested(30), -- JmJobStringTC (SIZE(0..63))
612     queueNameRequested(31),  -- JmJobStringTC (SIZE(0..63))
613     physicalDevice(32),     -- hrDeviceIndex
614                               -- AND/OR
615                               -- JmUTF8StringTC (SIZE(0..63))
616     numberOfDocuments(33),   -- Integer32 (-2..2147483647)
617     fileName(34),           -- JmJobStringTC (SIZE(0..63))
618     documentName(35),       -- JmJobStringTC (SIZE(0..63))
619     jobComment(36),         -- JmJobStringTC (SIZE(0..63))
620     documentFormatIndex(37), -- Integer32 (0..2147483647)
621     documentFormat(38),     -- PrtInterpreterLangFamilyTC
622                               -- AND/OR
623                               -- OCTET STRING(SIZE(0..63))
624
625     -- Job Parameter attributes:
626     jobPriority(50),         -- Integer32 (-2..100)
627     jobProcessAfterDateAndTime(51), -- DateAndTime (SNMPv2-TC)

```

```

628     jobHold(52),                -- JmBooleanTC
629     jobHoldUntil(53),          -- JmJobStringTC (SIZE(0..63))
630     outputBin(54),             -- Integer32 (0..2147483647)
631                                 -- AND/OR
632                                 -- JmJobStringTC (SIZE(0..63))
633     sides(55),                 -- Integer32 (-2..2)
634     finishing(56),             -- JmFinishingTC
635
636     -- Image Quality attributes:
637     printQualityRequested(70),  -- JmPrintQualityTC
638     printQualityUsed(71),       -- JmPrintQualityTC
639     printerResolutionRequested(72), -- JmPrinterResolutionTC
640     printerResolutionUsed(73),  -- JmPrinterResolutionTC
641     tonerEcomonyRequested(74),  -- JmTonerEconomyTC
642     tonerEcomonyUsed(75),       -- JmTonerEconomyTC
643     tonerDensityRequested(76),  -- Integer32 (-2..100)
644     tonerDensityUsed(77),       -- Integer32 (-2..100)
645
646     -- Job Progress attributes:
647     jobCopiesRequested(90),      -- Integer32 (-2..2147483647)
648     jobCopiesCompleted(91),     -- Integer32 (-2..2147483647)
649     documentCopiesRequested(92), -- Integer32 (-2..2147483647)
650     documentCopiesCompleted(93), -- Integer32 (-2..2147483647)
651     jobKOctetsTransferred(94),  -- Integer32 (-2..2147483647)
652     sheetCompletedCopyNumber(95), -- Integer32 (-2..2147483647)
653     sheetCompletedDocumentNumber(96), -- Integer32 (-2..2147483647)
654
655     jobCollationType(97),       -- JmJobCollationTypeTC
656
657     -- Impression attributes:
658     impressionsSpooled(110),     -- Integer32 (-2..2147483647)
659     impressionsSentToDevice(111), -- Integer32 (-2..2147483647)
660     impressionsInterpreted(112), -- Integer32 (-2..2147483647)
661     impressionsCompletedCurrentCopy(113), -- Integer32 (-2..2147483647)
662
663     fullColorImpressionsCompleted(114), -- Integer32 (-2..2147483647)
664
665     highlightColorImpressionsCompleted(115), -- Integer32 (-2..2147483647)
666
667
668     -- Page attributes:
669     pagesRequested(130),         -- Integer32 (-2..2147483647)
670     pagesCompleted(131),        -- Integer32 (-2..2147483647)
671     pagesCompletedCurrentCopy(132), -- Integer32 (-2..2147483647)
672
673     -- Sheet attributes:
674     sheetsRequested(150),        -- Integer32 (-2..2147483647)
675     sheetsCompleted(151),        -- Integer32 (-2..2147483647)
676     sheetsCompletedCurrentCopy(152), -- Integer32 (-2..2147483647)
677
678     -- Resource attributes:
679     mediumRequested(170),        -- JmMediumTypeTC
680                                 -- AND/OR
681                                 -- JmJobStringTC (SIZE(0..63))
682     mediumConsumed(171),        -- Integer32 (-2..2147483647)
683                                 -- AND
684                                 -- JmJobStringTC (SIZE(0..63))

```

```

685         colorantRequested(172),           -- Integer32 (-2..2147483647)
686                                         -- AND/OR
687                                         -- JmJobStringTC (SIZE(0..63))
688         colorantConsumed(173),           -- Integer32 (-2..2147483647)
689                                         -- AND/OR
690                                         -- JmJobStringTC (SIZE(0..63))
691         mediumTypeConsumed(174),         -- Integer32 (-2..2147483647)
692                                         -- AND
693                                         -- JmJobStringTC (SIZE(0..63))
694         mediumSizeConsumed(175),         -- Integer32 (-2..2147483647)
695                                         -- AND
696                                         -- JmJobStringTC (SIZE(0..63))
697
698         -- Time attributes:
699         jobSubmissionToServerTime(190),   -- JmTimeStampTC
700                                         -- AND/OR
701                                         -- DateAndTime
702         jobSubmissionTime(191),          -- JmTimeStampTC
703                                         -- AND/OR
704                                         -- DateAndTime
705         jobStartedBeingHeldTime(192),    -- JmTimeStampTC
706                                         -- AND/OR
707                                         -- DateAndTime
708         jobStartedProcessingTime(193),   -- JmTimeStampTC
709                                         -- AND/OR
710                                         -- DateAndTime
711         jobCompletionTime(194),          -- JmTimeStampTC
712                                         -- AND/OR
713                                         -- DateAndTime
714         jobProcessingCPUTime(195)        -- Integer32 (-2..2147483647)
715     }
716
717 JmJobServiceTypesTC ::= TEXTUAL-CONVENTION
718     STATUS         current
719     DESCRIPTION
720         "Specifies the type(s) of service to which the job has been
721         submitted (print, fax, scan, etc.). The service type is
722         represented as an enum that is bit encoded with each job
723         service type so that more general and arbitrary services can be
724         created, such as services with more than one destination type,
725         or ones with only a source or only a destination. For example,
726         a job service might scan, faxOut, and print a single job. In
727         this case, three bits would be set in the jobServiceTypes
728         attribute, corresponding to the hexadecimal values: 0x8 + 0x20
729         + 0x4, respectively, yielding: 0x2C.
730
731         Whether this attribute is set from a job attribute supplied by
732         the job submission client or is set by the recipient job
733         submission server or device depends on the job submission
734         protocol. With either implementation, the agent SHALL return a
735         non-zero value for this attribute indicating the type of the
736         job.
737
738         One of the purposes of this attribute is to permit a requester
739         to filter out jobs that are not of interest. For example, a
740         printer operator MAY only be interested in jobs that include
741         printing. That is why the attribute is in the job

```

742 identification category.

743

744 The following service component types are defined (in  
745 hexadecimal) and are assigned a separate bit value for use with  
746 the jobServiceTypes attribute:

747

748	other	0x1	
749			The job contains some instructions that are not one of the
750			identified types.
751			
752	unknown	0x2	
753			The job contains some instructions whose type is unknown to
754			the agent.
755			
756	print	0x4	
757			The job contains some instructions that specify printing
758			
759	scan	0x8	
760			The job contains some instructions that specify scanning
761			
762	faxIn	0x10	
763			The job contains some instructions that specify receive fax
764			
765	faxOut	0x20	
766			The job contains some instructions that specify sending fax
767			
768	getFile	0x40	
769			The job contains some instructions that specify accessing
770			files or documents
771			
772	putFile	0x80	
773			The job contains some instructions that specify storing
774			files or documents
775			
776	mailList	0x100	
777			The job contains some instructions that specify
778			distribution of documents using an electronic mail system.
779			

780 These bit definitions are the equivalent of a type 2 enum  
781 except that combinations of them MAY be used together. See  
782 section 3.7.1.2."

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

784

785 JmJobStateReasons1TC ::= TEXTUAL-CONVENTION

786	STATUS	current
787	DESCRIPTION	
788		"The JmJobStateReasonsNTC (N=1..4) textual-conventions are used
789		with the jmJobStateReasons1 object and jobStateReasonsN
790		(N=2..4), respectively, to provide additional information
791		regarding the current jmJobState object value. These values
792		MAY be used with any job state or states for which the reason
793		makes sense. See section 3.3.9.1 for the specification of each
794		bit value defined for use with the JmJobStateReasons1TC.
795		
796		These bit definitions are the equivalent of a type 2 enum
797		except that combinations of bits may be used together. See
798		section 3.7.1.2."

```

799     SYNTAX      INTEGER (0..2147483647)  -- 31 bits, all but sign bit
800
801 JmJobStateReasons2TC ::= TEXTUAL-CONVENTION
802     STATUS      current
803     DESCRIPTION
804         "This textual-convention is used with the jobStateReasons2
805         attribute to provides additional information regarding the
806         jmJobState object. See section 3.3.9.2 for the specification
807         of JmJobStateReasons2TC. See section 3.3.9.1 for the
808         description under JmJobStateReasons1TC for additional
809         information that applies to all reasons.
810
811         These bit definitions are the equivalent of a type 2 enum
812         except that combinations of them may be used together. See
813         section 3.7.1.2."
814     SYNTAX      INTEGER (0..2147483647)  -- 31 bits, all but sign bit
815
816 JmJobStateReasons3TC ::= TEXTUAL-CONVENTION
817     STATUS      current
818     DESCRIPTION
819         "This textual-convention is used with the jobStateReasons3
820         attribute to provides additional information regarding the
821         jmJobState object. See section 3.3.9.3 for the specification
822         of JmJobStateReasons3TC. See section 3.3.9.1 for the
823         description under JmJobStateReasons1TC for additional
824         information that applies to all reasons.
825
826         These bit definitions are the equivalent of a type 2 enum
827         except that combinations of them may be used together. See
828         section 3.7.1.2. "
829     SYNTAX      INTEGER (0..2147483647)  -- 31 bits, all but sign bit
830
831 JmJobStateReasons4TC ::= TEXTUAL-CONVENTION
832     STATUS      current
833     DESCRIPTION
834         "This textual-convention is used in the jobStateReasons4
835         attribute to provides additional information regarding the
836         jmJobState object. See section 3.3.9.4 for the specification
837         of JmJobStateReasons4TC. See section 3.3.9.1 for the
838         description under JmJobStateReasons1TC for additional
839         information that applies to all reasons.
840
841         These bit definitions are the equivalent of a type 2 enum
842         except that combinations of them may be used together. See
843         section 3.7.1.2."
844     SYNTAX      INTEGER (0..2147483647)  -- 31 bits, all but sign bit
845
846 JmServiceStateTC ::= TEXTUAL-CONVENTION
847     STATUS      current
848     DESCRIPTION
849         "The current state of the service.
850
851         See:   Section 4.4.11 'printer-state' in [IPP-MOD].
852
853         This is a type 2 enumeration. See Section 3.7.1.2."
854     SYNTAX      INTEGER {
855         unknown(2),

```

```

856         idle(3),           -- new jobs start immediately
857         processing(4),     -- jobs are processing
858         stopped(5)        -- intervention required
859     }
860
861 jobmonMIBObjects OBJECT IDENTIFIER ::= { jobmonMIB 1 }
862
863 -- The General Group (MANDATORY)
864
865 -- The jmGeneralGroup consists entirely of the jmGeneralTable.
866
867 jmGeneral OBJECT IDENTIFIER ::= { jobmonMIBObjects 1 }
868
869 jmGeneralTable OBJECT-TYPE
870     SYNTAX      SEQUENCE OF JmGeneralEntry
871     MAX-ACCESS  not-accessible
872     STATUS      current
873     DESCRIPTION
874         "The jmGeneralTable consists of information of a general nature
875         that are per-job-set, but are not per-job. See Section 2
876         entitled 'Terminology and Job Model' for the definition of a
877         job set.
878
879         The MANDATORY-GROUP macro specifies that this group is
880         MANDATORY."
881     ::= { jmGeneral 1 }
882
883 jmGeneralEntry OBJECT-TYPE
884     SYNTAX      JmGeneralEntry
885     MAX-ACCESS  not-accessible
886     STATUS      current
887     DESCRIPTION
888         "Information about a job set (queue).
889
890         An entry SHALL exist in this table for each job set."
891     INDEX { jmGeneralJobSetIndex }
892     ::= { jmGeneralTable 1 }
893
894 JmGeneralEntry ::= SEQUENCE {
895     jmGeneralJobSetIndex      Integer32 (1..32767),
896     jmGeneralNumberOfActiveJobs Integer32 (0..2147483647),
897     jmGeneralOldestActiveJobIndex Integer32 (0..2147483647),
898     jmGeneralNewestActiveJobIndex Integer32 (0..2147483647),
899     jmGeneralJobPersistence   Integer32 (15..2147483647),
900     jmGeneralAttributePersistence Integer32 (15..2147483647),
901     jmGeneralJobSetName      JmUTF8StringTC (SIZE(0..63))
902 }
903
904 jmGeneralJobSetIndex OBJECT-TYPE
905     SYNTAX      Integer32 (1..32767)
906     MAX-ACCESS  not-accessible
907     STATUS      current
908     DESCRIPTION
909         "A unique value for each job set in this MIB. The jmJobTable
910         and jmAttributeTable tables have this same index as their
911         primary index.
912

```



913 The value(s) of the jmGeneralJobSetIndex SHALL be persistent  
914 across power cycles, so that clients that have retained  
915 jmGeneralJobSetIndex values will access the same job sets upon  
916 subsequent power-up.

917  
918 An implementation that has only one job set, such as a printer  
919 with a single queue, SHALL hard code this object with the value  
920 1.

921  
922 See Section 2 entitled 'Terminology and Job Model' for the  
923 definition of a job set.  
924 Corresponds to the first index in jmJobTable and  
925 jmAttributeTable."  
926 ::= { jmGeneralEntry 1 }

927  
928 jmGeneralNumberOfActiveJobs OBJECT-TYPE  
929 SYNTAX Integer32 (0..2147483647)  
930 MAX-ACCESS read-only  
931 STATUS current  
932 DESCRIPTION  
933 "The current number of 'active' jobs in the jmJobIDTable,  
934 jmJobTable, and jmAttributeTable, i.e., the total number of  
935 jobs that are in the pending, processing, or processingStopped  
936 states. See the JmJobStateTC textual-convention for the exact  
937 specification of the semantics of the job states."  
938 DEFVAL { 0 } -- no jobs  
939 ::= { jmGeneralEntry 2 }

940  
941 jmGeneralOldestActiveJobIndex OBJECT-TYPE  
942 SYNTAX Integer32 (0..2147483647)  
943 MAX-ACCESS read-only  
944 STATUS current  
945 DESCRIPTION  
946 "The jmJobIndex of the oldest job that is still in one of the  
947 'active' states (pending, processing, or processingStopped).  
948 In other words, the index of the 'active' job that has been in  
949 the job tables the longest.

950  
951 If there are no active jobs, the agent SHALL set the value of  
952 this object to 0.

953  
954 See Section 3.2 entitled 'The Job Tables and the Oldest Active  
955 and Newest Active Indexes' for a description of the usage of  
956 this object."  
957 DEFVAL { 0 } -- no active jobs  
958 ::= { jmGeneralEntry 3 }

959  
960 jmGeneralNewestActiveJobIndex OBJECT-TYPE  
961 SYNTAX Integer32 (0..2147483647)  
962 MAX-ACCESS read-only  
963  
964 STATUS current  
965 DESCRIPTION  
966 "The jmJobIndex of the newest job that is in one of the  
967 'active' states (pending, processing, or processingStopped).  
968 In other words, the index of the 'active' job that has been  
969

```

970         most recently added to the job tables.
971
972         When all jobs become 'inactive', i.e., enter the pendingHeld,
973         completed, canceled, or aborted states, the agent SHALL set the
974         value of this object to 0.
975
976         See Section 3.2 entitled 'The Job Tables and the Oldest Active
977         and Newest Active Indexes' for a description of the usage of
978         this object."
979     DEFVAL      { 0 }          -- no active jobs
980     ::= { jmGeneralEntry 4 }
981
982 jmGeneralJobPersistence OBJECT-TYPE
983     SYNTAX      Integer32 (15..2147483647)
984     UNITS       "seconds"
985     MAX-ACCESS  read-only
986     STATUS      current
987     DESCRIPTION
988         "The minimum time in seconds for this instance of the Job Set
989         that an entry SHALL remain in the jmJobIDTable and jmJobTable
990         after processing has completed, i.e., the minimum time in
991         seconds starting when the job enters the completed, canceled,
992         or aborted state.
993
994         Configuring this object is implementation-dependent.
995
996         This value SHALL be equal to or greater than the value of
997         jmGeneralAttributePersistence. This value SHOULD be at least
998         60 which gives a monitoring or accounting application one
999         minute in which to poll for job data."
1000     DEFVAL      { 60 }          -- one minute
1001     ::= { jmGeneralEntry 5 }
1002
1003 jmGeneralAttributePersistence OBJECT-TYPE
1004     SYNTAX      Integer32 (15..2147483647)
1005     UNITS       "seconds"
1006     MAX-ACCESS  read-only
1007     STATUS      current
1008     DESCRIPTION
1009         "The minimum time in seconds for this instance of the Job Set
1010         that an entry SHALL remain in the jmAttributeTable after
1011         processing has completed , i.e., the time in seconds starting
1012         when the job enters the completed, canceled, or aborted state.
1013
1014         Configuring this object is implementation-dependent.
1015
1016         This value SHOULD be at least 60 which gives a monitoring or
1017         accounting application one minute in which to poll for job
1018         data."
1019     DEFVAL      { 60 }          -- one minute
1020     ::= { jmGeneralEntry 6 }
1021
1022 jmGeneralJobSetName OBJECT-TYPE
1023     SYNTAX      JmUTF8StringTC (SIZE(0..63))
1024     MAX-ACCESS  read-only
1025     STATUS      current
1026     DESCRIPTION

```

```

1027     "The human readable name of this job set assigned by the system
1028 administrator (by means outside of this MIB). Typically, this
1029 name SHOULD be the name of the job queue. If a server or
1030 device has only a single job set, this object can be the
1031 administratively assigned name of the server or device itself.
1032 This name does not need to be unique, though each job set in a
1033 single Job Monitoring MIB SHOULD have distinct names.
1034
1035     NOTE - If the job set corresponds to a single printer and the
1036 Printer MIB is implemented, this value SHOULD be the same as
1037 the prtGeneralPrinterName object in the draft Printer MIB
1038 [print-mib-draft]. If the job set corresponds to an IPP
1039 Printer, this value SHOULD be the same as the IPP 'printer-
1040 name' Printer attribute.
1041
1042     NOTE - The purpose of this object is to help the user of the
1043 job monitoring application distinguish between several job sets
1044 in implementations that support more than one job set.
1045
1046     See the OBJECT compliance macro for the minimum maximum length
1047 required for conformance."
1048     DEFVAL      { 'H }      -- empty string
1049     ::= { jmGeneralEntry 7 }
1050
1051 -- The Job ID Group (MANDATORY)
1052
1053 -- The jmJobIDGroup consists entirely of the jmJobIDTable.
1054
1055
1056 jmJobID OBJECT IDENTIFIER ::= { jobmonMIBObjects 2 }
1057
1058 jmJobIDTable OBJECT-TYPE
1059     SYNTAX      SEQUENCE OF JmJobIDEntry
1060     MAX-ACCESS  not-accessible
1061     STATUS      current
1062     DESCRIPTION
1063         "The jmJobIDTable provides a correspondence map (1) between the
1064         job submission ID that a client uses to refer to a job and (2)
1065         the jmGeneralJobSetIndex and jmJobIndex that the Job Monitoring
1066         MIB agent assigned to the job and that are used to access the
1067         job in all of the other tables in the MIB. If a monitoring
1068         application already knows the jmGeneralJobSetIndex and the
1069         jmJobIndex of the job it is querying, that application NEED NOT
1070         use the jmJobIDTable.
1071
1072         The MANDATORY-GROUP macro specifies that this group is
1073         MANDATORY."
1074     ::= { jmJobID 1 }
1075
1076 jmJobIDEntry OBJECT-TYPE
1077     SYNTAX      JmJobIDEntry
1078     MAX-ACCESS  not-accessible
1079     STATUS      current
1080     DESCRIPTION
1081         "The map from (1) the jmJobSubmissionID to (2) the
1082         jmGeneralJobSetIndex and jmJobIndex.
1083

```

1084 An entry SHALL exist in this table for each job currently known  
1085 to the agent for all job sets and job states. There MAY be  
1086 more than one jmJobIDEntry that maps to a single job. This  
1087 many to one mapping can occur when more than one network entity  
1088 along the job submission path supplies a job submission ID.  
1089 See Section 3.5. However, each job SHALL appear once and in  
1090 one and only one job set."

```

1091 INDEX { jmJobSubmissionID }
1092 ::= { jmJobIDTable 1 }
1093
1094 JmJobIDEntry ::= SEQUENCE {
1095     jmJobSubmissionID          OCTET STRING(SIZE(48)),
1096     jmJobIDJobSetIndex        Integer32 (0..32767),
1097     jmJobIDJobIndex           Integer32 (0..2147483647)
1098 }
1099
1100 jmJobSubmissionID OBJECT-TYPE
1101     SYNTAX          OCTET STRING(SIZE(48))
1102     MAX-ACCESS      not-accessible
1103     STATUS          current
1104     DESCRIPTION
1105         "A quasi-unique 48-octet fixed-length string ID which
1106         identifies the job within a particular client-server
1107         environment. There are multiple formats for the
1108         jmJobSubmissionID. Each format SHALL be uniquely identified.
1109         See the JmJobSubmissionIDTypeTC textual convention. Each
1110         format SHALL be registered using the procedures of a type 2
1111         enum. See section 3.7.3 entitled: 'PWG Registration of Job
1112         Submission Id Formats'.


1113



1114         If the requester (client or server) does not supply a job
1115         submission ID in the job submission protocol, then the
1116         recipient (server or device) SHALL assign a job submission ID
1117         using any of the standard formats that have been reserved for
1118         agents and adding the final 8 octets to distinguish the ID from
1119         others submitted from the same requester.



1120



1121         The monitoring application, whether in the client or running
1122         separately, MAY use the job submission ID to help identify
1123         which jmJobIndex was assigned by the agent, i.e., in which row
1124         the job information is in the other tables.



1125



1126         NOTE - fixed-length is used so that a management application
1127         can use a shortened GetNext varbind (in SNMPv1 and SNMPv2) in
1128         order to get the next submission ID, disregarding the remainder
1129         of the ID in order to access jobs independent of the trailing
1130         identifier part, e.g., to get all jobs submitted by a
1131         particular jmJobOwner or submitted from a particular MAC
1132         address.



1133



1134         See the JmJobSubmissionIDTypeTC textual convention.



1135         See APPENDIX B - Support of Job Submission Protocols."



```

1136     ::= { jmJobIDEntry 1 }
1137
1138 jmJobIDJobSetIndex OBJECT-TYPE
1139     SYNTAX          Integer32 (0..32767)
1140     MAX-ACCESS      read-only

```


```

```

1141     STATUS      current
1142     DESCRIPTION
1143         "This object contains the value of the jmGeneralJobSetIndex for
1144         the job with the jmJobSubmissionID value, i.e., the job set
1145         index of the job set in which the job was placed when that
1146         server or device accepted the job. This 16-bit value in
1147         combination with the jmJobIDJobIndex value permits the
1148         management application to access the other tables to obtain the
1149         job-specific objects for this job.
1150
1151         See jmGeneralJobSetIndex in the jmGeneralTable."
1152     DEFVAL      { 0 }      -- 0 indicates no job set index
1153     ::= { jmJobIDEntry 2 }
1154
1155 jmJobIDJobIndex OBJECT-TYPE
1156     SYNTAX      Integer32 (0..2147483647)
1157     MAX-ACCESS  read-only
1158     STATUS      current
1159     DESCRIPTION
1160         "This object contains the value of the jmJobIndex for the job
1161         with the jmJobSubmissionID value, i.e., the job index for the
1162         job when the server or device accepted the job. This value, in
1163         combination with the jmJobIDJobSetIndex value, permits the
1164         management application to access the other tables to obtain the
1165         job-specific objects for this job.
1166
1167         See jmJobIndex in the jmJobTable."
1168     DEFVAL      { 0 }      -- 0 indicates no jmJobIndex value.
1169     ::= { jmJobIDEntry 3 }
1170
1171 -- The Job Group (MANDATORY)
1172
1173 -- The jmJobGroup consists entirely of the jmJobTable.
1174
1175 jmJob OBJECT IDENTIFIER ::= { jobmonMIBObjects 3 }
1176
1177 jmJobTable OBJECT-TYPE
1178     SYNTAX      SEQUENCE OF JmJobEntry
1179     MAX-ACCESS  not-accessible
1180     STATUS      current
1181     DESCRIPTION
1182         "The jmJobTable consists of basic job state and status
1183         information for each job in a job set that (1) monitoring
1184         applications need to be able to access in a single SNMP Get
1185         operation, (2) that have a single value per job, and (3) that
1186         SHALL always be implemented.
1187
1188         The MANDATORY-GROUP macro specifies that this group is
1189         MANDATORY."
1190     ::= { jmJob 1 }
1191
1192
1193 jmJobEntry OBJECT-TYPE
1194     SYNTAX      JmJobEntry
1195     MAX-ACCESS  not-accessible
1196     STATUS      current
1197     DESCRIPTION

```

```

1198         "Basic per-job state and status information.
1199
1200         An entry SHALL exist in this table for each job, no matter what
1201         the state of the job is. Each job SHALL appear in one and only
1202         one job set.
1203
1204         See Section 3.2 entitled 'The Job Tables'."
1205     INDEX { jmGeneralJobSetIndex, jmJobIndex }
1206     ::= { jmJobTable 1 }
1207
1208     JmJobEntry ::= SEQUENCE {
1209         jmJobIndex                Integer32 (1..2147483647),
1210         jmJobState                JmJobStateTC,
1211         jmJobStateReasons1       JmJobStateReasons1TC,
1212         jmNumberOfInterveningJobs Integer32 (-2..2147483647),
1213         jmJobKOctetsPerCopyRequested Integer32 (-2..2147483647),
1214         jmJobKOctetsProcessed    Integer32 (-2..2147483647),
1215         jmJobImpressionsPerCopyRequested Integer32 (-2..2147483647),
1216         jmJobImpressionsCompleted Integer32 (-2..2147483647),
1217         jmJobOwner                JmJobStringTC (SIZE(0..63))
1218     }
1219
1220     jmJobIndex OBJECT-TYPE
1221         SYNTAX      Integer32 (1..2147483647)
1222         MAX-ACCESS  not-accessible
1223         STATUS      current
1224         DESCRIPTION
1225             "The sequential, monotonically increasing identifier index for
1226             the job generated by the server or device when that server or
1227             device accepted the job. This index value permits the
1228             management application to access the other tables to obtain the
1229             job-specific row entries.
1230
1231             See Section 3.2 entitled 'The Job Tables and the Oldest Active
1232             and Newest Active Indexes'.
1233             See Section 3.5 entitled 'Job Identification'.
1234             See also jmGeneralNewestActiveJobIndex for the largest value of
1235             jmJobIndex.
1236             See JmJobSubmissionIDTypeTC for a limit on the size of this
1237             index if the agent represents it as an 8-digit decimal number."
1238         ::= { jmJobEntry 1 }
1239
1240
1241     jmJobState OBJECT-TYPE
1242         SYNTAX      JmJobStateTC
1243         MAX-ACCESS  read-only
1244         STATUS      current
1245         DESCRIPTION
1246             "The current state of the job (pending, processing, completed,
1247             etc.). Agents SHALL implement only those states which are
1248             appropriate for the particular implementation. However,
1249             management applications SHALL be prepared to receive all the
1250             standard job states.
1251
1252             The final value for this object SHALL be one of: completed,
1253             canceled, or aborted. The minimum length of time that the
1254             agent SHALL maintain MIB data for a job in the completed,

```

```

1255         canceled, or aborted state before removing the job data from
1256         the jmJobIDTable and jmJobTable is specified by the value of
1257         the jmGeneralJobPersistence object."
1258     DEFVAL      { unknown }      -- default is unknown
1259     ::= { jmJobEntry 2 }
1260
1261 jmJobStateReasons1 OBJECT-TYPE
1262     SYNTAX      JmJobStateReasons1TC
1263     MAX-ACCESS  read-only
1264     STATUS      current
1265     DESCRIPTION
1266         "Additional information about the job's current state, i.e.,
1267         information that augments the value of the job's jmJobState
1268         object.
1269
1270         Implementation of any reason values is OPTIONAL, but an agent
1271         SHOULD return any reason information available. These values
1272         MAY be used with any job state or states for which the reason
1273         makes sense. Since the Job State Reasons will be more dynamic
1274         than the Job State, it is recommended that a job monitoring
1275         application read this object every time jmJobState is read.
1276         When the agent cannot provide a reason for the current state of
1277         the job, the value of the jmJobStateReasons1 object and
1278         jobStateReasonsN attributes SHALL be 0.
1279
1280         The jobStateReasonsN (N=2..4) attributes provide further
1281         additional information about the job's current state."
1282     DEFVAL      { 0 }      -- no reasons
1283     ::= { jmJobEntry 3 }
1284
1285
1286 jmNumberOfInterveningJobs OBJECT-TYPE
1287     SYNTAX      Integer32 (-2..2147483647)
1288     MAX-ACCESS  read-only
1289     STATUS      current
1290     DESCRIPTION
1291         "The number of jobs that are expected to complete processing
1292         before this job has completed processing according to the
1293         implementation's queuing algorithm, if no other jobs were to be
1294         submitted. In other words, this value is the job's queue
1295         position. The agent SHALL return a value of 0 for this
1296         attribute when the job is the next job to complete processing
1297         (or has completed processing)."
1298     DEFVAL      { 0 }      -- default is no intervening jobs.
1299     ::= { jmJobEntry 4 }
1300
1301 jmJobKOctetsPerCopyRequested OBJECT-TYPE
1302     SYNTAX      Integer32 (-2..2147483647)
1303     MAX-ACCESS  read-only
1304     STATUS      current
1305     DESCRIPTION
1306         "The total size in K (1024) octets of the document(s) being
1307         requested to be processed in the job. The agent SHALL round
1308         the actual number of octets up to the next highest K. Thus 0
1309         octets is represented as '0', 1-1024 octets is represented as
1310         '1', 1025-2048 is represented as '2', etc.
1311

```

1312 In computing this value, the server/device SHALL NOT include  
1313 the multiplicative factors contributed by (1) the number of  
1314 document copies, and (2) the number of job copies, independent  
1315 of whether the device can process multiple copies of the job or  
1316 document without making multiple passes over the job or  
1317 document data and independent of whether the output is collated  
1318 or not. Thus the server/device computation is independent of  
1319 the implementation and indicates the size of the document(s)  
1320 measured in K octets independent of the number of copies."  
1321 DEFVAL { -2 } -- the default is unknown(-2)  
1322 ::= { jmJobEntry 5 }  
1323  
1324 jmJobKOctetsProcessed OBJECT-TYPE  
1325 SYNTAX Integer32 (-2..2147483647)  
1326 MAX-ACCESS read-only  
1327 STATUS current  
1328 DESCRIPTION  
1329 "The total number of octets processed by the server or device  
1330 measured in units of K (1024) octets so far. The agent SHALL  
1331 round the actual number of octets processed up to the next  
1332 higher K. Thus 0 octets is represented as '0', 1-1024 octets  
1333 is represented as '1', 1025-2048 octets is '2', etc. For  
1334 printing devices, this value is the number interpreted by the  
1335 page description language interpreter rather than what has been  
1336 marked on media.  
1337  
1338 For implementations where multiple copies are produced by the  
1339 interpreter with only a single pass over the data, the final  
1340 value SHALL be equal to the value of the  
1341 jmJobKOctetsPerCopyRequested object. For implementations where  
1342 multiple copies are produced by the interpreter by processing  
1343 the data for each copy, the final value SHALL be a multiple of  
1344 the value of the jmJobKOctetsPerCopyRequested object.  
1345  
1346 NOTE - See the impressionsCompletedCurrentCopy and  
1347 pagesCompletedCurrentCopy attributes for attributes that are  
1348 reset on each document copy.  
1349  
1350 NOTE - The jmJobKOctetsProcessed object can be used with the  
1351 jmJobKOctetsPerCopyRequested object to provide an indication of  
1352 the relative progress of the job, provided that the  
1353 multiplicative factor is taken into account for some  
1354 implementations of multiple copies."  
1355 DEFVAL { 0 } -- default is no octets processed.  
1356 ::= { jmJobEntry 6 }  
1357  
1358 jmJobImpressionsPerCopyRequested OBJECT-TYPE  
1359 SYNTAX Integer32 (-2..2147483647)  
1360 MAX-ACCESS read-only  
1361 STATUS current  
1362 DESCRIPTION  
1363 "The total size in number of impressions of the document(s)  
1364 submitted.  
1365  
1366 In computing this value, the server/device SHALL NOT include  
1367 the multiplicative factors contributed by (1) the number of  
1368 document copies, and (2) the number of job copies, independent



1369 of whether the device can process multiple copies of the job or  
1370 document without making multiple passes over the job or  
1371 document data and independent of whether the output is collated  
1372 or not. Thus the server/device computation is independent of  
1373 the implementation and reflects the size of the document(s)  
1374 measured in impressions independent of the number of copies.  
1375

1376 See the definition of the term 'impression' in Section 2."  
1377 DEFVAL { -2 } -- default is unknown(-2)  
1378 ::= { jmJobEntry 7 }  
1379

1380 jmJobImpressionsCompleted OBJECT-TYPE  
1381 SYNTAX Integer32 (-2..2147483647)  
1382 MAX-ACCESS read-only  
1383 STATUS current  
1384 DESCRIPTION  
1385 "The total number of impressions completed for this job so far.  
1386 For printing devices, the impressions completed includes  
1387 interpreting, marking, and stacking the output. For other  
1388 types of job services, the number of impressions completed  
1389 includes the number of impressions processed.  
1390  
1391 NOTE - See the impressionsCompletedCurrentCopy and  
1392 pagesCompletedCurrentCopy attributes for attributes that are  
1393 reset on each document copy.  
1394  
1395 NOTE - The jmJobImpressionsCompleted object can be used with  
1396 the jmJobImpressionsPerCopyRequested object to provide an  
1397 indication of the relative progress of the job, provided that  
1398 the multiplicative factor is taken into account for some  
1399 implementations of multiple copies.  
1400  
1401 See the definition of the term 'impression' in Section 2 and  
1402 the counting example in Section 3.4 entitled 'Monitoring Job  
1403 Progress'."

1404 DEFVAL { 0 } -- default is no octets  
1405 ::= { jmJobEntry 8 }  
1406

1407 jmJobOwner OBJECT-TYPE  
1408 SYNTAX JmJobStringTC (SIZE(0..63))  
1409 MAX-ACCESS read-only  
1410 STATUS current  
1411 DESCRIPTION  
1412 "The coded character set name of the user that submitted the  
1413 job. The method of assigning this user name will be system  
1414 and/or site specific but the method MUST ensure that the name  
1415 is unique to the network that is visible to the client and  
1416 target device.  
1417  
1418 This value SHOULD be the most authenticated name of the user  
1419 submitting the job.  
1420  
1421 See the OBJECT compliance macro for the minimum maximum length  
1422 required for conformance."  
1423 DEFVAL { 'H' } -- default is empty string  
1424 ::= { jmJobEntry 9 }  
1425

```

1426 -- The Attribute Group (MANDATORY)
1427
1428 -- The jmAttributeGroup consists entirely of the jmAttributeTable.
1429 --
1430 -- Implementation of the objects in this group is MANDATORY.
1431 -- See Section 3.1 entitled 'Conformance Considerations'.
1432 -- An agent SHALL implement any attribute if (1) the server or device
1433 -- supports the functionality represented by the attribute and (2) the
1434 -- information is available to the agent.
1435
1436 jmAttribute OBJECT IDENTIFIER ::= { jobmonMIBObjects 4 }
1437
1438 jmAttributeTable OBJECT-TYPE
1439     SYNTAX      SEQUENCE OF JmAttributeEntry
1440     MAX-ACCESS  not-accessible
1441     STATUS      current
1442     DESCRIPTION
1443         "The jmAttributeTable SHALL contain attributes of the job and
1444         document(s) for each job in a job set. Instead of allocating
1445         distinct objects for each attribute, each attribute is
1446         represented as a separate row in the jmAttributeTable.
1447
1448         The MANDATORY-GROUP macro specifies that this group is
1449         MANDATORY. An agent SHALL implement any attribute if (1) the
1450         server or device supports the functionality represented by the
1451         attribute and (2) the information is available to the agent. "
1452     ::= { jmAttribute 1 }
1453
1454 jmAttributeEntry OBJECT-TYPE
1455     SYNTAX      JmAttributeEntry
1456     MAX-ACCESS  not-accessible
1457     STATUS      current
1458     DESCRIPTION
1459         "Attributes representing information about the job and
1460         document(s) or resources required and/or consumed.
1461
1462         Each entry in the jmAttributeTable is a per-job entry with an
1463         extra index for each type of attribute (jmAttributeTypeIndex)
1464         that a job can have and an additional index
1465         (jmAttributeInstanceIndex) for those attributes that can have
1466         multiple instances per job. The jmAttributeTypeIndex object
1467         SHALL contain an enum type that indicates the type of attribute
1468         (see the JmAttributeTypeTC textual-convention). The value of
1469         the attribute SHALL be represented in either the
1470         jmAttributeValueAsInteger or jmAttributeValueAsOctets objects,
1471         and/or both, as specified in the JmAttributeTypeTC textual-
1472         convention.
1473
1474         The agent SHALL create rows in the jmAttributeTable as the
1475         server or device is able to discover the attributes either from
1476         the job submission protocol itself or from the document PDL.
1477         As the documents are interpreted, the interpreter MAY discover
1478         additional attributes and so the agent adds additional rows to
1479         this table. As the attributes that represent resources are
1480         actually consumed, the usage counter contained in the
1481         jmAttributeValueAsInteger object is incremented according to
1482         the units indicated in the description of the JmAttributeTypeTC

```

```

1483         enum.
1484
1485         The agent SHALL maintain each row in the jmAttributeTable for
1486         at least the minimum time after a job completes as specified by
1487         the jmGeneralAttributePersistence object.
1488
1489         Zero or more entries SHALL exist in this table for each job in
1490         a job set.
1491
1492         See Section 3.3 entitled 'The Attribute Mechanism' for a
1493         description of the jmAttributeTable."
1494     INDEX { jmGeneralJobSetIndex, jmJobIndex, jmAttributeTypeIndex,
1495     jmAttributeInstanceIndex }
1496     ::= { jmAttributeTable 1 }
1497
1498     JmAttributeEntry ::= SEQUENCE {
1499         jmAttributeTypeIndex          JmAttributeTypeTC,
1500         jmAttributeInstanceIndex      Integer32 (1..32767),
1501         jmAttributeValueAsInteger     Integer32 (-2..2147483647),
1502         jmAttributeValueAsOctets      OCTET STRING(SIZE(0..63))
1503     }
1504
1505     jmAttributeTypeIndex OBJECT-TYPE
1506         SYNTAX          JmAttributeTypeTC
1507         MAX-ACCESS      not-accessible
1508         STATUS          current
1509         DESCRIPTION
1510             "The type of attribute that this row entry represents.
1511
1512             The type MAY identify information about the job or document(s)
1513             or MAY identify a resource required to process the job before
1514             the job start processing and/or consumed by the job as the job
1515             is processed.
1516
1517             Examples of job attributes (i.e., apply to the job as a whole)
1518             that have only one instance per job include:
1519             jobCopiesRequested(90), documentCopiesRequested(92),
1520             jobCopiesCompleted(91), documentCopiesCompleted(93), while
1521             examples of job attributes that may have more than one instance
1522             per job include: documentFormatIndex(37), and
1523             documentFormat(38).
1524
1525             Examples of document attributes (one instance per document)
1526             include: fileName(34), and documentName(35).
1527
1528             Examples of required and consumed resource attributes include:
1529             pagesRequested(130), mediumRequested(170), pagesCompleted(131),
1530             and mediumConsumed(171), respectively."
1531     ::= { jmAttributeEntry 1 }
1532
1533     jmAttributeInstanceIndex OBJECT-TYPE
1534         SYNTAX          Integer32 (1..32767)
1535         MAX-ACCESS      not-accessible
1536         STATUS          current
1537         DESCRIPTION
1538             "A running 16-bit index of the attributes of the same type for
1539             each job. For those attributes with only a single instance per

```

```

1540     job, this index value SHALL be 1. For those attributes that
1541     are a single value per document, the index value SHALL be the
1542     document number, starting with 1 for the first document in the
1543     job. Jobs with only a single document SHALL use the index
1544     value of 1. For those attributes that can have multiple values
1545     per job or per document, such as documentFormatIndex(37) or
1546     documentFormat(38), the index SHALL be a running index for the
1547     job as a whole, starting at 1."
1548 ::= { jmAttributeEntry 2 }
1549
1550 jmAttributeValueAsInteger OBJECT-TYPE
1551     SYNTAX      Integer32 (-2..2147483647)
1552     MAX-ACCESS  read-only
1553     STATUS      current
1554     DESCRIPTION
1555         "The integer value of the attribute. The value of the
1556         attribute SHALL be represented as an integer if the enum
1557         description in the JmAttributeTypeTC textual-convention
1558         definition has the tag: 'INTEGER:'."
1559
1560         Depending on the enum definition, this object value MAY be an
1561         integer, a counter, an index, or an enum, depending on the
1562         jmAttributeTypeIndex value. The units of this value are
1563         specified in the enum description.
1564
1565         For those attributes that are accumulating job consumption as
1566         the job is processed as specified in the JmAttributeTypeTC
1567         textual-convention, SHALL contain the final value after the job
1568         completes processing, i.e., this value SHALL indicate the total
1569         usage of this resource made by the job.
1570
1571         A monitoring application is able to copy this value to a
1572         suitable longer term storage for later processing as part of an
1573         accounting system.
1574
1575         Since the agent MAY add attributes representing resources to
1576         this table while the job is waiting to be processed or being
1577         processed, which can be a long time before any of the resources
1578         are actually used, the agent SHALL set the value of the
1579         jmAttributeValueAsInteger object to 0 for resources that the
1580         job has not yet consumed.
1581
1582         Attributes for which the concept of an integer value is
1583         meaningless, such as fileName(34), jobName, and
1584         processingMessage, do not have the 'INTEGER:' tag in the
1585         JmAttributeTypeTC definition and so an agent SHALL always
1586         return a value of '-1' to indicate 'other' for the value of the
1587         jmAttributeValueAsInteger object for these attributes.
1588
1589         For attributes which do have the 'INTEGER:' tag in the
1590         JmAttributeTypeTC definition, if the integer value is not (yet)
1591         known, the agent either (1) SHALL not materialize the row in
1592         the jmAttributeTable until the value is known or (2) SHALL
1593         return a '-2' to represent an 'unknown' counting integer value,
1594         a '0' to represent an 'unknown' index value, and a '2' to
1595         represent an 'unknown(2)' enum value."
1596     DEFVAL      { -2 }      -- default value is unknown(-2)

```

```

1597     ::= { jmAttributeEntry 3 }
1598
1599 jmAttributeValueAsOctets OBJECT-TYPE
1600     SYNTAX      OCTET STRING(SIZE(0..63))
1601     MAX-ACCESS  read-only
1602     STATUS      current
1603     DESCRIPTION
1604         "The octet string value of the attribute.  The value of the
1605         attribute SHALL be represented as an OCTET STRING if the enum
1606         description in the JmAttributeTypeTC textual-convention
1607         definition has the tag: 'OCTETS:'.
1608
1609         Depending on the enum definition, this object value MAY be a
1610         coded character set string (text), such as 'JmUTF8StringTC', or
1611         a binary octet string, such as 'DateAndTime'.
1612
1613         Attributes for which the concept of an octet string value is
1614         meaningless, such as pagesCompleted, do not have the tag
1615         'OCTETS:' in the JmAttributeTypeTC definition and so the agent
1616         SHALL always return a zero length string for the value of the
1617         jmAttributeValueAsOctets object.
1618
1619         For attributes which do have the 'OCTETS:' tag in the
1620         JmAttributeTypeTC definition, if the OCTET STRING value is not
1621         (yet) known, the agent either SHALL NOT materialize the row in
1622         the jmAttributeTable until the value is known or SHALL return a
1623         zero-length string."
1624     DEFVAL      { ''H } -- empty string
1625     ::= { jmAttributeEntry 4 }
1626
1627 -- Reserved identifiers for proposed future extensions
1628
1629 jmMirrorAttr OBJECT IDENTIFIER ::= { jobmonMIBObjects 5 }
1630 jmSystem     OBJECT IDENTIFIER ::= { jobmonMIBObjects 6 }
1631
1632
1633 -- The Service Group (CONDITIONALLY MANDATORY)
1634 --
1635 -- Implementation of this group is conditionally mandatory;
1636 -- mandatory for systems which show Service state via SNMP.
1637
1638 -- The jmServiceGroup consists entirely of the jmServiceTable
1639
1640 jmService OBJECT IDENTIFIER ::= { jobmonMIBObjects 7 }
1641
1642 jmServiceTable OBJECT-TYPE
1643     SYNTAX      SEQUENCE OF JmServiceEntry
1644     MAX-ACCESS  not-accessible
1645     STATUS      current
1646     DESCRIPTION
1647         "The jmServiceTable consists of basic service state and status
1648         information for each service which offers one or more job
1649         services on this managed system.
1650
1651         An entry SHALL exist in this table for each service, no matter
1652         what the state of that service.  A service MAY support multiple
1653         configured job sets and configured devices.

```

```

1654
1655         See:      'jmServiceJobSetsConfigured' and
1656                 'jmServiceDevicesConfigured' bit-arrays in this MIB."
1657 ::= { jmService 1 }
1658
1659
1660 jmServiceEntry OBJECT-TYPE
1661     SYNTAX      JmServiceEntry
1662     MAX-ACCESS  not-accessible
1663     STATUS      current
1664     DESCRIPTION
1665         "Basic service state and status information."
1666     INDEX       { jmServiceIndex }
1667 ::= { jmServiceTable 1 }
1668
1669
1670 JmServiceEntry ::= SEQUENCE {
1671     jmServiceIndex      Integer32 (1..2147483647),
1672     jmServiceName       JmUTF8StringTC (SIZE (0..63)),
1673     jmServiceURI        JmUTF8StringTC (SIZE (0..63)),
1674     jmServiceJobServiceTypes JmJobServiceTypesTC,
1675     jmServiceJobSetsConfigured OCTET STRING (SIZE (0..255)),
1676     jmServiceDevicesConfigured OCTET STRING (SIZE (0..255)),
1677     jmServiceState      JmServiceStateTC,
1678     jmServiceStateReasons SnmpAdminString (SIZE (0..255))
1679 }
1680
1681 jmServiceIndex OBJECT-TYPE
1682     SYNTAX      Integer32 (1..2147483647)
1683     MAX-ACCESS  not-accessible
1684     STATUS      current
1685     DESCRIPTION
1686         "The unique identifier for this service on this managed system.
1687
1688         See:      'jmServiceEventServiceIndex' object in this MIB."
1689 ::= { jmServiceEntry 1 }
1690
1691 jmServiceName OBJECT-TYPE
1692     SYNTAX      JmUTF8StringTC (SIZE(0..63))    -- 127 in IPP
1693     MAX-ACCESS  read-only
1694     STATUS      current
1695     DESCRIPTION
1696         "The human readable name of this managed service.
1697
1698         See:      'deviceNameRequested' job attribute in this MIB;
1699                 'physicalDevice' job attribute in this MIB;
1700                 Section 4.4.4 'printer-name' in [IPP-MOD]."
1701     DEFVAL     { 'H' }                          -- no service name
1702 ::= { jmServiceEntry 2 }
1703
1704 jmServiceURI OBJECT-TYPE
1705     SYNTAX      JmUTF8StringTC (SIZE(0..63))    -- 1023 in IPP
1706     MAX-ACCESS  read-only
1707     STATUS      current
1708     DESCRIPTION
1709         "A URI for this managed service (valid for job services).
1710

```

```

1711         See:      Section 4.3.3 'job-printer-uri' in [IPP-MOD];
1712                Section 4.4.1 'printer-uri-supported' in [IPP-MOD].
1713     DEFVAL      { 'H' }                -- no service URI
1714     ::= { jmServiceEntry 3 }
1715
1716     jmServiceJobServiceTypes OBJECT-TYPE
1717     SYNTAX      JmJobServiceTypesTC
1718     MAX-ACCESS  read-only
1719     STATUS      current
1720     DESCRIPTION
1721         "The types of job services supported by this managed service.
1722
1723         See:      'JmJobServiceTypesTC' textual convention in this MIB;
1724                'jobServiceTypes' job attribute in this MIB."
1725     DEFVAL      { 0 }                  -- no job services
1726     ::= { jmServiceEntry 4 }
1727
1728     jmServiceJobSetsConfigured OBJECT-TYPE
1729     SYNTAX      OCTET STRING (SIZE(0..255))
1730     MAX-ACCESS  read-only
1731     STATUS      current
1732     DESCRIPTION
1733         "A bit-array that specifies the job sets configured for this
1734         service, where each bit '2**n' is set if 'jmGeneralJobSetIndex'
1735         is a configured job set.  Uses network byte order (big-endian)
1736         rules - the high-order bit of the first octet corresponds to
1737         'jmGeneralJobSetIndex' of '0' (reserved) - the low-order bit of
1738         the first octet corresponds to 'jmGeneralJobSetIndex' of '7'.
1739         Supports values of 'jmGeneralJobSetIndex' from '1' to '2039'.
1740
1741         Compare to the BITS pseudotype defined in IETF SMIV2 (RFC 2578)
1742         which has the same bit ordering rules (big-endian).
1743
1744         See:      'queueNameRequested' job attribute in this MIB;
1745                'jmGeneralJobSetIndex' table index in this MIB."
1746     DEFVAL      { 'H' }                -- no job sets configured
1747     ::= { jmServiceEntry 5 }
1748
1749     jmServiceDevicesConfigured OBJECT-TYPE
1750     SYNTAX      OCTET STRING (SIZE(0..255))
1751     MAX-ACCESS  read-only
1752     STATUS      current
1753     DESCRIPTION
1754         "A bit-array that specifies the devices configured for this
1755         service, where each bit '2**n' is set if 'hrDeviceIndex'
1756         is a configured device.  Uses network byte order (big-endian)
1757         rules - the high-order bit of the first octet corresponds to
1758         'hrDeviceIndex' of '0' (reserved) - the low-order bit of
1759         the first octet corresponds to 'hrDeviceIndex' of '7'.
1760         Supports values of 'hrDeviceIndex' from '1' to '2039'.
1761
1762         Compare to the BITS pseudotype defined in IETF SMIV2 (RFC 2578)
1763         which has the same bit ordering rules (big-endian).
1764
1765         See:      'physicalDevice' job attribute in this MIB;
1766                'hrDeviceIndex' in IETF Host MIB (RFC 2790).
1767     DEFVAL      { 'H' }                -- no devices configured

```

```

1768     ::= { jmServiceEntry 6 }
1769
1770 jmServiceState OBJECT-TYPE
1771     SYNTAX      JmServiceStateTC
1772     MAX-ACCESS  read-only
1773     STATUS      current
1774     DESCRIPTION
1775         "The current state of this managed service.
1776
1777         See:      'jmServiceEventServiceState' object in this MIB;
1778                 'jmJobState' object in this MIB;
1779                 Section 4.4.11 'printer-state' in [IPP-MOD]."
```

```

1780     DEFVAL      { unknown }          -- unknown service state
1781     ::= { jmServiceEntry 7 }
1782
1783 jmServiceStateReasons OBJECT-TYPE
1784     SYNTAX      SnmpAdminString (SIZE (0..255)) -- multi-valued in IPP
1785     MAX-ACCESS  read-only
1786     STATUS      current
1787     DESCRIPTION
1788         "The service state reasons associated with this service state
1789         (as a comma-separated list) or the empty string (if none).
1790
1791         See:      'jmServiceEventServiceStateReasons' object in this MIB;
1792                 'jmJobStateReasons1' object in this MIB;
1793                 Section 4.4.12 'printer-state-reasons' in [IPP-MOD]."
```

```

1794     DEFVAL      { 'H' }              -- no service state reasons
1795     ::= { jmServiceEntry 8 }
1796
1797
1798 -- The Service Event Group (CONDITIONALLY MANDATORY)
1799 --
1800 -- Implementation of this group is conditionally mandatory;
1801 -- mandatory for systems which show Service events via SNMP.
1802
1803 -- The jmServiceEventGroup consists entirely of the jmServiceEventTable
1804
1805 jmServiceEvent OBJECT IDENTIFIER ::= { jobmonMIBObjects 8 }
1806
1807 jmServiceEventTable OBJECT-TYPE
1808     SYNTAX      SEQUENCE OF JmServiceEventEntry
1809     MAX-ACCESS  not-accessible
1810     STATUS      current
1811     DESCRIPTION
1812         "The jmServiceEventTable contains service event entries for the
1813         services which offer job services on this managed system."
1814     ::= { jmServiceEvent 1 }
1815
1816
1817 jmServiceEventEntry OBJECT-TYPE
1818     SYNTAX      JmServiceEventEntry
1819     MAX-ACCESS  not-accessible
1820     STATUS      current
1821     DESCRIPTION
1822         "Basic service event information."
1823     INDEX      { jmServiceEventIndex }
1824     ::= { jmServiceEventTable 1 }
```



```

1825
1826
1827 JmServiceEventEntry ::= SEQUENCE {
1828     jmServiceEventIndex          Integer32 (1..2147483647),
1829     jmServiceEventNotifyEvent    SnmpAdminString (SIZE (0..63)),
1830     jmServiceEventNotifyTime     TimeTicks,
1831     jmServiceEventServiceIndex   Integer32 (1..2147483647),
1832     jmServiceEventServiceState   JmServiceStateTC,
1833     jmServiceEventServiceStateReasons SnmpAdminString (SIZE (0..255))
1834 }
1835
1836 jmServiceEventIndex OBJECT-TYPE
1837     SYNTAX      Integer32 (1..2147483647)
1838     MAX-ACCESS  not-accessible
1839     STATUS      current
1840     DESCRIPTION
1841         "The unique identifier for this event on this managed system."
1842     ::= { jmServiceEventEntry 1 }
1843
1844 jmServiceEventNotifyEvent OBJECT-TYPE
1845     SYNTAX      SnmpAdminString (SIZE (0..63))  -- 255 in [IPP-NOT]
1846     MAX-ACCESS  read-only
1847     STATUS      current
1848     DESCRIPTION
1849         "The event type that created this row in 'jmServiceEventTable'."
1850
1851         Standard Printer event types defined in [IPP-NOT] are:
1852         - 'printer-state-changed'
1853         - 'printer-restarted'
1854         - 'printer-shutdown'
1855         - 'printer-config-changed'
1856         - 'printer-media-changed'
1857         - 'printer-finishings-changed'
1858         - 'printer-queue-order-changed'
1859         - 'printer-full'
1860         - 'printer-no-longer-full'
1861         - 'printer-almost-idle'
1862         - 'printer-not-almost-idle'
1863
1864         Standard Service event types generalized from [IPP-NOT] are:
1865         - 'service-state-changed'
1866         - 'service-restarted'
1867         - 'service-shutdown'
1868         - 'service-config-changed'
1869         - 'service-media-changed'
1870         - 'service-finishings-changed'
1871         - 'service-queue-order-changed'
1872         - 'service-full'
1873         - 'service-no-longer-full'
1874         - 'service-almost-idle'
1875         - 'service-not-almost-idle'
1876
1877         Conformance:    The natural language for keywords
1878         in subscribed event type SHALL always be US English.
1879
1880         Conformance:    This subscribed event type SHALL be valid
1881         and reported in ALL Job Monitoring MIB notifications.

```

```

1882
1883     See:      Section 8.1 'notify-subscribed-event' in [IPP-NOT].
1884     DEFVAL    { 'H' }          -- no notify subscribed event
1885     ::= { jmServiceEventEntry 2 }
1886
1887 jmServiceEventNotifyTime OBJECT-TYPE
1888     SYNTAX      TimeTicks
1889     MAX-ACCESS  read-only
1890     STATUS      current
1891     DESCRIPTION
1892         "The time of this service event."
1893
1894     Usage:      Conforming management agents, which MUST implement the
1895     IETF MIB-II (RFC 1213), SHALL set 'jmServiceEventNotifyTime' to
1896     'sysUpTime' when a service event row is created.
1897
1898     See:        'sysUpTime' in IETF MIB-II (RFC 1213);
1899     Section 5.4.4 'notify-printer-up-time' in [IPP-NOT];
1900     Section 4.4.29 'printer-up-time' in [IPP-MOD].
1901     ::= { jmServiceEventEntry 3 }
1902
1903 jmServiceEventServiceIndex OBJECT-TYPE
1904     SYNTAX      Integer32 (1..2147483647)
1905     MAX-ACCESS  read-only
1906     STATUS      current
1907     DESCRIPTION
1908         "The unique identifier for this service on this managed system."
1909
1910     See:        'jmServiceIndex' object in this MIB."
1911     ::= { jmServiceEventEntry 4 }
1912
1913 jmServiceEventServiceState OBJECT-TYPE
1914     SYNTAX      JmServiceStateTC
1915     MAX-ACCESS  read-only
1916     STATUS      current
1917     DESCRIPTION
1918         "The state of this managed service at the time of this event."
1919
1920     See:        'jmServiceState' object in this MIB;
1921     'jmJobState' object in this MIB;
1922     Section 4.4.11 'printer-state' in [IPP-MOD].
1923     DEFVAL     { unknown }      -- unknown service state
1924     ::= { jmServiceEventEntry 5 }
1925
1926 jmServiceEventServiceStateReasons OBJECT-TYPE
1927     SYNTAX      SnmpAdminString (SIZE (0..255)) -- multi-valued in IPP
1928     MAX-ACCESS  read-only
1929     STATUS      current
1930     DESCRIPTION
1931         "The service state reasons associated with this service state
1932         (as a comma-separated list) or the empty string (if none)."
1933
1934     See:        'jmServiceStateReasons' object in this MIB;
1935     'jmJobStateReasons1' object in this MIB;
1936     Section 4.4.12 'printer-state-reasons' in [IPP-MOD].
1937     DEFVAL     { 'H' }          -- no service state reasons
1938     ::= { jmServiceEventEntry 6 }

```

```

1939
1940
1941 -- The Job Event Group (CONDITIONALLY MANDATORY)
1942 --
1943 -- Implementation of this group is conditionally mandatory;
1944 -- mandatory for systems which show Job events via SNMP.
1945
1946 -- The jmJobEventGroup consists entirely of the jmJobEventTable
1947
1948 jmJobEvent OBJECT IDENTIFIER ::= { jobmonMIBObjects 9 }
1949
1950 jmJobEventTable OBJECT-TYPE
1951     SYNTAX      SEQUENCE OF JmJobEventEntry
1952     MAX-ACCESS  not-accessible
1953     STATUS      current
1954     DESCRIPTION
1955         "The jmJobEventTable contains job event entries for the
1956         jobs present on this managed system."
1957     ::= { jmJobEvent 1 }
1958
1959
1960 jmJobEventEntry OBJECT-TYPE
1961     SYNTAX      JmJobEventEntry
1962     MAX-ACCESS  not-accessible
1963     STATUS      current
1964     DESCRIPTION
1965         "Basic job event information."
1966     INDEX      { jmJobEventIndex }
1967     ::= { jmJobEventTable 1 }
1968
1969
1970 JmJobEventEntry ::= SEQUENCE {
1971     jmJobEventIndex          Integer32 (1..2147483647),
1972     jmJobEventNotifyEvent    SnmpAdminString (SIZE (0..63)),
1973     jmJobEventNotifyTime     TimeTicks,
1974     jmJobEventJobSetIndex    Integer32 (1..32767),
1975     jmJobEventJobIndex       Integer32 (1..2147483647),
1976     jmJobEventJobState       JmJobStateTC,
1977     jmJobEventJobStateReasons OCTET STRING (SIZE (4..16))
1978 }
1979
1980 jmJobEventIndex OBJECT-TYPE
1981     SYNTAX      Integer32 (1..2147483647)
1982     MAX-ACCESS  not-accessible
1983     STATUS      current
1984     DESCRIPTION
1985         "The unique identifier for this event on this managed system."
1986     ::= { jmJobEventEntry 1 }
1987
1988 jmJobEventNotifyEvent OBJECT-TYPE
1989     SYNTAX      SnmpAdminString (SIZE (0..63)) -- 255 in [IPP-NOT]
1990     MAX-ACCESS  read-only
1991     STATUS      current
1992     DESCRIPTION
1993         "The event type that created this row in 'jmJobEventTable'.
1994
1995         Standard Job event types defined in [IPP-NOT] are:

```

1996 - 'job-state-changed'  
1997 - 'job-created'  
1998 - 'job-completed'  
1999 - 'job-purged'  
2000 - 'job-config-changed'  
2001 - 'job-progress'

2002  
2003 Conformance: The natural language for keywords  
2004 in subscribed event type SHALL always be US English.  
2005

2006 Conformance: This subscribed event type SHALL be valid  
2007 and reported in ALL Job Monitoring MIB notifications.  
2008

2009 See: Section 8.1 'notify-subscribed-event' in [IPP-NOT]."  
2010 DEFVAL { 'H } -- no notify subscribed event  
2011 ::= { jmJobEventEntry 2 }  
2012

2013 jmJobEventNotifyTime OBJECT-TYPE  
2014 SYNTAX TimeTicks  
2015 MAX-ACCESS read-only  
2016 STATUS current  
2017 DESCRIPTION  
2018 "The time of this job event.  
2019

2020 Usage: Conforming management agents, which MUST implement the  
2021 IETF MIB-II (RFC 1213), SHALL set 'jmJobEventNotifyTime' to  
2022 'sysUpTime' when a job event row is created.  
2023

2024 See: 'sysUpTime' in IETF MIB-II (RFC 1213);  
2025 Section 5.4.4 'notify-printer-up-time' in [IPP-NOT];  
2026 Section 4.4.29 'printer-up-time' in [IPP-MOD]."  
2027 ::= { jmJobEventEntry 3 }  
2028

2029 jmJobEventJobSetIndex OBJECT-TYPE  
2030 SYNTAX Integer32 (1..32767)  
2031 MAX-ACCESS read-only  
2032 STATUS current  
2033 DESCRIPTION  
2034 "The unique identifier for this job set on this managed system.  
2035

2036 See: 'jmGeneralJobSetIndex' object in this MIB."  
2037 ::= { jmJobEventEntry 4 }  
2038

2039 jmJobEventJobIndex OBJECT-TYPE  
2040 SYNTAX Integer32 (1..2147483647)  
2041 MAX-ACCESS read-only  
2042 STATUS current  
2043 DESCRIPTION  
2044 "The unique identifier for this job on this managed system,  
2045 when prefixed with 'jmJobEventJobSetIndex'.  
2046

2047 See: 'jmJobIndex' object in this MIB."  
2048 ::= { jmJobEventEntry 5 }  
2049

2050 jmJobEventJobState OBJECT-TYPE  
2051 SYNTAX JmJobStateTC  
2052 MAX-ACCESS read-only

```

2053     STATUS      current
2054     DESCRIPTION
2055         "The state of this managed job at the time of this event.
2056
2057         See:      'jmJobState' in this MIB;
2058                 Section 4.3.7 'job-state' in [IPP-MOD]."
2059     DEFVAL      { unknown }          -- unknown job state
2060     ::= { jmJobEventEntry 6 }
2061
2062 jmJobEventJobStateReasons OBJECT-TYPE
2063     SYNTAX      OCTET STRING (SIZE (4..16)) -- multi-valued in IPP
2064     MAX-ACCESS  read-only
2065     STATUS      current
2066     DESCRIPTION
2067         "The job state reasons associated with this job state
2068         represented as one to four concatenated 32-bit integers
2069         in network byte order (big-endian).
2070
2071         Usage:   Conforming management stations SHALL always report the
2072                 value of the object 'jmJobStateReasons1' in the first four
2073                 octets of 'jmJobEventJobStateReasons' and SHOULD report values
2074                 of the attributes 'jobStateReasons2', 'jobStateReasons3', and
2075                 'jobStateReasons4' in subsequent octets.
2076
2077         See:      'jmJobStateReasons1' in this MIB;
2078                 Section 4.3.8 'job-state-reasons' in [IPP-MOD]."
2079     DEFVAL      { '00000000'H }      -- no job state reasons
2080     ::= { jmJobEventEntry 7 }
2081
2082
2083 -- The Job Progress Group (CONDITIONALLY MANDATORY)
2084 --
2085 -- Implementation of this group is conditionally mandatory;
2086 -- mandatory for systems which send Job progress traps via SNMP.
2087
2088 -- The jmProgressGroup consists entirely of leaf objects for traps
2089
2090 jmProgress OBJECT IDENTIFIER ::= { jobmonMIBObjects 10 }
2091
2092 jmProgressJobCopiesRequested OBJECT-TYPE
2093     SYNTAX      Integer32 (-2..2147483647)
2094     MAX-ACCESS  read-only
2095     STATUS      current
2096     DESCRIPTION
2097         "The number of copies of this job requested by the client.
2098
2099         See:      'jobCopiesRequested' attribute in this MIB."
2100     DEFVAL      { -2 }              -- unknown job copies
2101     ::= { jmProgress 1 }
2102
2103 jmProgressJobCollationType OBJECT-TYPE
2104     SYNTAX      JmJobCollationTypeTC
2105     MAX-ACCESS  read-only
2106     STATUS      current
2107     DESCRIPTION
2108         "The number of copies of this job requested by the client.
2109

```

```

2110         See:      'jobCollationType' attribute in this MIB;
2111         'job-collation-type' in [IPP-PROG];
2112         Section 9 'Event Notification Content' in [IPP-NOT].
2113     DEFVAL      { unknown }          -- unknown job collation type
2114     ::= { jmProgress 2 }
2115
2116 jmProgressMediaSheetsCompleted OBJECT-TYPE
2117     SYNTAX      Integer32 (-2..2147483647)
2118     MAX-ACCESS  read-only
2119     STATUS      current
2120     DESCRIPTION
2121         "The number of media sheets completed for this job.
2122
2123         See:      'sheetsCompleted' attribute in this MIB;
2124         Section 4.3.18.3 'job-media-sheets-completed'
2125         in [IPP-MOD];
2126         Section 9 'Event Notification Content' in [IPP-NOT].
2127     DEFVAL      { -2 }              -- unknown job progress
2128     ::= { jmProgress 3 }
2129
2130 jmProgressSheetCompletedCopyNum OBJECT-TYPE
2131     SYNTAX      Integer32 (-2..2147483647)
2132     MAX-ACCESS  read-only
2133     STATUS      current
2134     DESCRIPTION
2135         "The number of the job copy currently being stacked for the
2136         current document or zero (if none) or '-2' (unknown).
2137
2138         See:      'sheetCompletedCopyNumber' attribute in this MIB;
2139         'sheet-completed-copy-number' in [IPP-PROG];
2140         Section 9 'Event Notification Content' in [IPP-NOT].
2141     DEFVAL      { -2 }              -- unknown sheet complete copy
2142     ::= { jmProgress 4 }
2143
2144 jmProgressSheetCompletedDocNum OBJECT-TYPE
2145     SYNTAX      Integer32 (-2..2147483647)
2146     MAX-ACCESS  read-only
2147     STATUS      current
2148     DESCRIPTION
2149         "The number of the job document currently being stacked for the
2150         current job or zero (if none) or '-2' (unknown).
2151
2152         See:      'sheetCompletedDocumentNumber' attribute in this MIB;
2153         'sheet-completed-document-number' in [IPP-PROG];
2154         Section 9 'Event Notification Content' in [IPP-NOT].
2155     DEFVAL      { -2 }              -- unknown sheet complete doc
2156     ::= { jmProgress 5 }
2157
2158 -- Notifications (Traps and Informs)
2159
2160 jobmonMIBNotifications OBJECT IDENTIFIER ::= { jobmonMIB 2 }
2161
2162 -- Service Basic Event Group (CONDITIONALLY MANDATORY)
2163 --
2164 -- Implementation of this group is conditionally mandatory;
2165 -- mandatory for systems which send this Service traps via SNMP.
2166

```

```

2167 -- The jmServiceBasicEventGroup consists entirely of the
2168 -- jmServiceBasicV2Event notification.
2169
2170 jmServiceBasicV1Enterprise OBJECT-IDENTITY
2171     STATUS          current
2172     DESCRIPTION
2173         "The value of the enterprise-specific OID in an SNMPv1 trap
2174         for a Service basic event sent by this managed system."
2175     ::= { jobmonMIBNotifications 1 }
2176
2177 jmServiceBasicV2EventPrefix
2178     OBJECT IDENTIFIER ::= { jmServiceBasicV1Enterprise 0 }
2179
2180 jmServiceBasicV2Event NOTIFICATION-TYPE
2181     OBJECTS {
2182         jmServiceEventNotifyEvent,
2183         jmServiceState,
2184         jmServiceStateReasons
2185     }
2186     STATUS          current
2187     DESCRIPTION
2188         "This SMIV2 trap corresponds to an IPP Printer basic event.
2189
2190         The value of 'jmServiceIndex' for
2191         use with 'jmServiceTable' for this Service is conveyed in the
2192         instance qualifier (OID suffix) of 'jmServiceState'.
2193
2194         This trap is sent when requested by a prior subscription.
2195         The subscribed event type is in 'jmServiceEventNotifyEvent'.
2196
2197         Standard Printer event types defined in [IPP-NOT] are:
2198         - 'printer-state-changed'
2199         - 'printer-restarted'
2200         - 'printer-shutdown'
2201         - 'printer-config-changed'
2202         - 'printer-media-changed'
2203         - 'printer-finishings-changed'
2204         - 'printer-queue-order-changed'
2205         - 'printer-full'
2206         - 'printer-no-longer-full'
2207         - 'printer-almost-idle'
2208         - 'printer-not-almost-idle'
2209
2210         Standard Service event types generalized from [IPP-NOT] are:
2211         - 'service-state-changed'
2212         - 'service-restarted'
2213         - 'service-shutdown'
2214         - 'service-config-changed'
2215         - 'service-media-changed'
2216         - 'service-finishings-changed'
2217         - 'service-queue-order-changed'
2218         - 'service-full'
2219         - 'service-no-longer-full'
2220         - 'service-almost-idle'
2221         - 'service-not-almost-idle'
2222         - and (optionally) vendor extension event types
2223

```

```

2224     Additional variable-bindings SHOULD be appended to this trap:
2225     - Systems with the Host Resources MIB (RFC 2790) SHOULD add
2226       'hrSystemDate'
2227       (compare to IPP 'printer-current-time')
2228
2229     Additional variable-bindings MAY be appended to this trap
2230     for all printer-specific events:
2231     - Systems with the Host Resources MIB (RFC 2790) MAY add
2232       'hrDeviceStatus'
2233       (compare to IPP 'printer-state')
2234       'hrPrinterStatus'
2235       (compare to IPP 'printer-state') and
2236       'hrPrinterDetectedErrorState'
2237       (compare to IPP 'printer-state-reasons')
2238
2239     Systems MAY add other variable-bindings from any MIB.
2240
2241     See:     Section 5.3.2 'notify-events' in [IPP-NOT];
2242             Section 8.1 'notify-subscribed-event' in [IPP-NOT];
2243             Section 9 'Event Notification Content' in [IPP-NOT].
2244     ::= { jmServiceBasicV2EventPrefix 1 }
2245
2246     -- Job Basic Event Group (CONDITIONALLY MANDATORY)
2247     --
2248     -- Implementation of this group is conditionally mandatory;
2249     -- mandatory for systems which send this Job traps via SNMP.
2250
2251     -- The jmJobBasicEventGroup consists entirely of the
2252     -- jmJobBasicV2Event notification.
2253
2254     jmJobBasicV1Enterprise OBJECT-IDENTITY
2255     STATUS      current
2256     DESCRIPTION
2257         "The value of the enterprise-specific OID in an SNMPv1 trap
2258         for a Job basic event sent by this managed system."
2259     ::= { jobmonMIBNotifications 2 }
2260
2261     jmJobBasicV2EventPrefix
2262     OBJECT IDENTIFIER ::= { jmJobBasicV1Enterprise 0 }
2263
2264     jmJobBasicV2Event NOTIFICATION-TYPE
2265     OBJECTS {
2266         jmJobEventNotifyEvent,
2267         jmJobState,
2268         jmJobEventJobStateReasons
2269     }
2270     STATUS      current
2271     DESCRIPTION
2272         "This SMIV2 trap corresponds to an IPP Job basic event.
2273
2274         The values of 'jmGeneralJobSetIndex' and 'jmJobIndex' for
2275         use with 'jmJobTable' for this Job are conveyed in the
2276         instance qualifier (OID suffix) of 'jmJobState'.
2277
2278         This trap is sent when requested by a prior subscription.
2279         The subscribed event type is in 'jmJobEventNotifyEvent'.
2280

```



```

2281     Event types reported via 'jmJobBasicV2Event' include:
2282     - 'job-state-changed'
2283     - 'job-created'
2284     - 'job-completed'
2285     - 'job-purged'
2286     - 'job-config-changed'
2287     - and (optionally) vendor extension event types
2288
2289     Additional variable-bindings SHOULD be appended to this trap:
2290     - Systems with the Host Resources MIB (RFC 2790) SHOULD add
2291       'hrSystemDate'
2292       (compare to IPP 'printer-current-time')
2293
2294     Systems MAY add other variable-bindings from any MIB.
2295
2296     See:     Section 5.3.2 'notify-events' in [IPP-NOT];
2297             Section 8.1 'notify-subscribed-event' in [IPP-NOT];
2298             Section 9 'Event Notification Content' in [IPP-NOT].
2299     ::= { jmJobBasicV2EventPrefix 1 }
2300
2301 -- Job Completed Event Group (CONDITIONALLY MANDATORY)
2302 --
2303 -- Implementation of this group is conditionally mandatory;
2304 -- mandatory for systems which send this Job traps via SNMP.
2305
2306 jmJobCompletedV1Enterprise OBJECT-IDENTITY
2307     STATUS          current
2308     DESCRIPTION
2309         "The value of the enterprise-specific OID in an SNMPv1 trap
2310         for a Job completed event sent by this managed system."
2311     ::= { jobmonMIBNotifications 3 }
2312
2313 jmJobCompletedV2EventPrefix
2314     OBJECT IDENTIFIER ::= { jmJobCompletedV1Enterprise 0 }
2315
2316 jmJobCompletedV2Event NOTIFICATION-TYPE
2317     OBJECTS {
2318         jmJobEventNotifyEvent,
2319         jmJobState,
2320         jmJobEventJobStateReasons,
2321         jmJobKOctetsProcessed,
2322         jmJobImpressionsCompleted
2323     }
2324     STATUS          current
2325     DESCRIPTION
2326         "This SMIV2 trap corresponds to an IPP 'job-completed' event.
2327
2328         The values of 'jmGeneralJobSetIndex' and 'jmJobIndex' for
2329         use with 'jmJobTable' for this Job are conveyed in the
2330         instance qualifier (OID suffix) of 'jmJobState'.
2331
2332         This trap is sent when requested by a prior subscription.
2333         The subscribed event type is in 'jmJobEventNotifyEvent'.
2334
2335         Event types reported via 'jmJobCompletedV2Event' include:
2336         - 'job-completed'
2337         - 'job-purged'

```

```

2338     - and (optionally) vendor extension event types
2339
2340     Additional variable-bindings SHOULD be appended to this trap:
2341     - Systems with the Host Resources MIB (RFC 2790) SHOULD add
2342       'hrSystemDate'
2343       (compare to IPP 'printer-current-time')
2344
2345     Systems MAY add other variable-bindings from any MIB.
2346
2347     See:   Section 5.3.2 'notify-events' in [IPP-NOT];
2348           Section 8.1 'notify-subscribed-event' in [IPP-NOT];
2349           Section 9 'Event Notification Content' in [IPP-NOT].
2350 ::= { jmJobCompletedV2EventPrefix 1 }
2351
2352 -- Job Progress Event Group (CONDITIONALLY MANDATORY)
2353 --
2354 -- Implementation of this group is conditionally mandatory;
2355 -- mandatory for systems which send this Job traps via SNMP.
2356
2357 jmJobProgressV1Enterprise OBJECT-IDENTITY
2358     STATUS          current
2359     DESCRIPTION
2360         "The value of the enterprise-specific OID in an SNMPv1 trap
2361         for a Job progress event sent by this managed system."
2362     ::= { jobmonMIBNotifications 4 }
2363
2364 jmJobProgressV2EventPrefix
2365     OBJECT IDENTIFIER ::= { jmJobProgressV1Enterprise 0 }
2366
2367 jmJobProgressV2Event NOTIFICATION-TYPE
2368     OBJECTS {
2369         jmJobKOctetsPerCopyRequested,
2370         jmJobKOctetsProcessed,
2371         jmJobImpressionsPerCopyRequested,
2372         jmJobImpressionsCompleted,
2373         jmProgressJobCopiesRequested,
2374         jmProgressJobCollationType,
2375         jmProgressMediaSheetsCompleted,
2376         jmProgressSheetCompletedCopyNum,
2377         jmProgressSheetCompletedDocNum
2378     }
2379     STATUS          current
2380     DESCRIPTION
2381         "This SMIV2 trap corresponds to an IPP 'job-progress' event.
2382
2383         The values of 'jmGeneralJobSetIndex' and 'jmJobIndex' for
2384         use with 'jmJobTable' for this Job are conveyed in the
2385         instance qualifier (OID suffix) of 'jmJobKOctetsProcessed'.
2386
2387         This trap is sent when requested by a prior subscription.
2388         The event type is 'job-progress'.
2389
2390     Additional variable-bindings SHOULD be appended to this trap:
2391     - Systems with the Host Resources MIB (RFC 2790) SHOULD add
2392       'hrSystemDate'
2393       (compare to IPP 'printer-current-time')
2394

```

```

2395         Systems MAY add other variable-bindings from any MIB.
2396
2397         See:      Section 5.3.2 'notify-events' in [IPP-NOT];
2398                 Section 8.1 'notify-subscribed-event' in [IPP-NOT];
2399                 Section 9 'Event Notification Content' in [IPP-NOT].
2400         ::= { jmJobProgressV2EventPrefix 1 }
2401
2402 -- Conformance Information
2403
2404 jmMIBConformance OBJECT IDENTIFIER ::= { jobmonMIB 3 }
2405
2406 -- compliance statements
2407 jmMIBCompliance MODULE-COMPLIANCE
2408     STATUS current
2409     DESCRIPTION
2410         "The compliance statement for agents that implement the
2411         job monitoring MIB."
2412     MODULE -- this module
2413     MANDATORY-GROUPS {
2414         jmGeneralGroup, jmJobIDGroup, jmJobGroup, jmAttributeGroup }
2415
2416     GROUP jmServiceGroup
2417     DESCRIPTION
2418         "Implementation of this group is conditionally mandatory;
2419         mandatory for systems which show Service states via SNMP."
2420
2421     GROUP jmServiceEventGroup
2422     DESCRIPTION
2423         "Implementation of this group is conditionally mandatory;
2424         mandatory for systems which show Service events via SNMP."
2425
2426     GROUP jmJobEventGroup
2427     DESCRIPTION
2428         "Implementation of this group is conditionally mandatory;
2429         mandatory for systems which show Job events via SNMP."
2430
2431     GROUP jmProgressGroup
2432     DESCRIPTION
2433         "Implementation of this group is conditionally mandatory;
2434         mandatory for systems which send Job progress traps via SNMP."
2435
2436     GROUP jmServiceBasicEventGroup
2437     DESCRIPTION
2438         "Implementation of this group is conditionally mandatory;
2439         mandatory for systems which send this Service traps via SNMP."
2440
2441     GROUP jmJobBasicEventGroup
2442     DESCRIPTION
2443         "Implementation of this group is conditionally mandatory;
2444         mandatory for systems which send this Job traps via SNMP."
2445
2446     GROUP jmJobCompletedEventGroup
2447     DESCRIPTION
2448         "Implementation of this group is conditionally mandatory;
2449         mandatory for systems which send this Job traps via SNMP."
2450
2451

```

```

2452     GROUP    jmJobProgressEventGroup
2453     DESCRIPTION
2454         "Implementation of this group is conditionally mandatory;
2455         mandatory for systems which send this Job traps via SNMP."
2456
2457     OBJECT    jmGeneralJobSetName
2458     SYNTAX    JmUTF8StringTC (SIZE(0..8))
2459     DESCRIPTION
2460         "Only 8 octets maximum string length NEED be supported by the
2461         agent."
2462
2463     OBJECT    jmJobOwner
2464     SYNTAX    JmJobStringTC (SIZE(0..16))
2465     DESCRIPTION
2466         "Only 16 octets maximum string length NEED be supported by the
2467         agent."
2468
2469     ::= { jmMIBConformance 1 }
2470
2471 jmMIBGroups OBJECT IDENTIFIER ::= { jmMIBConformance 2 }
2472
2473 jmGeneralGroup OBJECT-GROUP
2474     OBJECTS {
2475         jmGeneralNumberOfActiveJobs,    jmGeneralOldestActiveJobIndex,
2476         jmGeneralNewestActiveJobIndex,  jmGeneralJobPersistence,
2477         jmGeneralAttributePersistence,  jmGeneralJobSetName}
2478     STATUS current
2479     DESCRIPTION
2480         "The general object group."
2481     ::= { jmMIBGroups 1 }
2482
2483 jmJobIDGroup OBJECT-GROUP
2484     OBJECTS {
2485         jmJobIDJobSetIndex, jmJobIDJobIndex }
2486     STATUS current
2487     DESCRIPTION
2488         "The job ID object group."
2489     ::= { jmMIBGroups 2 }
2490
2491 jmJobGroup OBJECT-GROUP
2492     OBJECTS {
2493         jmJobState, jmJobStateReasons1, jmNumberOfInterveningJobs,
2494         jmJobKOctetsPerCopyRequested, jmJobKOctetsProcessed,
2495         jmJobImpressionsPerCopyRequested, jmJobImpressionsCompleted,
2496         jmJobOwner }
2497     STATUS current
2498
2499     DESCRIPTION
2500         "The job object group."
2501     ::= { jmMIBGroups 3 }
2502
2503
2504 jmAttributeGroup OBJECT-GROUP
2505     OBJECTS {
2506         jmAttributeValueAsInteger, jmAttributeValueAsOctets }
2507     STATUS current
2508     DESCRIPTION

```

```

2509         "The job attribute object group."
2510         ::= { jmMIBGroups 4 }
2511
2512 jmServiceGroup OBJECT-GROUP
2513     OBJECTS {
2514         jmServiceName,
2515         jmServiceURI,
2516         jmServiceJobServiceTypes,
2517         jmServiceJobSetsConfigured,
2518         jmServiceDevicesConfigured,
2519         jmServiceState,
2520         jmServiceStateReasons
2521     }
2522     STATUS current
2523     DESCRIPTION
2524         "The service object group."
2525     ::= { jmMIBGroups 7 }
2526
2527 jmServiceEventGroup OBJECT-GROUP
2528     OBJECTS {
2529         jmServiceEventNotifyEvent,
2530         jmServiceEventNotifyTime,
2531         jmServiceEventServiceIndex,
2532         jmServiceEventServiceState,
2533         jmServiceEventServiceStateReasons
2534     }
2535     STATUS current
2536     DESCRIPTION
2537         "The service event object group."
2538     ::= { jmMIBGroups 8 }
2539
2540 jmJobEventGroup OBJECT-GROUP
2541     OBJECTS {
2542         jmJobEventNotifyEvent,
2543         jmJobEventNotifyTime,
2544         jmJobEventJobSetIndex,
2545         jmJobEventJobIndex,
2546         jmJobEventJobState,
2547         jmJobEventJobStateReasons
2548     }
2549     STATUS current
2550     DESCRIPTION
2551         "The job event object group."
2552     ::= { jmMIBGroups 9 }
2553
2554 jmProgressGroup OBJECT-GROUP
2555     OBJECTS {
2556         jmProgressJobCopiesRequested,
2557         jmProgressJobCollationType,
2558         jmProgressMediaSheetsCompleted,
2559         jmProgressSheetCompletedCopyNum,
2560         jmProgressSheetCompletedDocNum
2561     }
2562     STATUS current
2563     DESCRIPTION
2564         "The job progress object group."
2565     ::= { jmMIBGroups 10 }

```

```

2566
2567 jmMIBNotifyGroups OBJECT IDENTIFIER ::= { jmMIBConformance 3 }
2568
2569 jmServiceBasicEventGroup NOTIFICATION-GROUP
2570 NOTIFICATIONS { jmServiceBasicV2Event }
2571 STATUS current
2572 DESCRIPTION
2573     "The service basic event notification group."
2574 ::= { jmMIBNotifyGroups 1 }
2575
2576 jmJobBasicEventGroup NOTIFICATION-GROUP
2577 NOTIFICATIONS { jmJobBasicV2Event }
2578 STATUS current
2579 DESCRIPTION
2580     "The job basic event notification group."
2581 ::= { jmMIBNotifyGroups 2 }
2582
2583 jmJobCompletedEventGroup NOTIFICATION-GROUP
2584 NOTIFICATIONS { jmJobCompletedV2Event }
2585 STATUS current
2586 DESCRIPTION
2587     "The job completed event notification group."
2588 ::= { jmMIBNotifyGroups 3 }
2589
2590 jmJobProgressEventGroup NOTIFICATION-GROUP
2591 NOTIFICATIONS { jmJobProgressV2Event }
2592 STATUS current
2593 DESCRIPTION
2594     "The job progress event notification group."
2595 ::= { jmMIBNotifyGroups 4 }
2596
2597 END

```