```
Job-Monitoring-MIB DEFINITIONS ::= BEGIN
--  Version 1.1 - adds Service and Job traps for IPP Notifications
--  (see revision history in MODULE-IDENTITY macro below)

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    OBJECT-IDENTITY, NOTIFICATION-TYPE,
    enterprises, Integer32, TimeTicks   FROM SNMPv2-SMI
    TEXTUAL-CONVENTION                  FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP,
    NOTIFICATION-GROUP                  FROM SNMPv2-CONF
    SnmpAdminString                     FROM SNMP-FRAMEWORK-MIB;

    -- The following textual-conventions are needed to implement
    -- certain attributes, but are not needed to compile this MIB.
    -- They are provided here for convenience:
    -- hrDeviceIndex                    FROM HOST-RESOURCES-MIB
    -- DateAndTime                      FROM SNMPv2-TC
    -- PrtInterpreterLangFamilyTC,
    -- CodedCharSet                     FROM Printer-MIB

-- Use the enterprises arc assigned to the PWG which is pwg(2699).
-- Group all PWG mibs under mibs(1).

jobmonMIB MODULE-IDENTITY
    LAST-UPDATED "0007060000Z" -- 6 July 2000
    ORGANIZATION "Printer Working Group (PWG)"
    CONTACT-INFO
        "Tom Hastings
        Postal:  Xerox Corp.
                 Mail stop ESAE-231
                 701 S. Aviation Blvd.
                 El Segundo, CA 90245

        Tel:     (301)333-6413
        Fax:     (301)333-5514
        E-mail:  hastings@cp10.es.xerox.com

        Ira McDonald
        Postal:  High North Inc.
                 PO Box 221
                 Grand Marais, MI  49839

        Tel:     (906)494-2434
        Email:   imcdonal@sdsp.mc.xerox.com

        Send questions and comments to the Printer Working Group (PWG)
        using the Job Monitoring Project (JMP) Mailing List:
        jmp@pwg.org

        For further information, including how to subscribe to the
        jmp mailing list, access the PWG web page under 'JMP':

            http://www.pwg.org/

        Implementers of this specification are encouraged to join the
        jmp mailing list in order to participate in discussions on any
        clarifications needed and registration proposals being reviewed
        in order to achieve consensus."
    DESCRIPTION
        "The MIB module for monitoring job in servers, printers, and
        other devices.

        Version: 1.1"

    -- revision history
    REVISION    "0007060000Z" -- 6 July 2000
    DESCRIPTION
        "Version 1.1 - adds support for Service and Job events:

        1)  Added 'SnmpAdminString' to IMPORTS clause for new objects.
        2)  Corrected OID in MODULE-IDENTITY to use forward reference to
            definition of 'pwg' from 'enterprises' and 'mibs' from 'pwg'
        3)  Added 'JmServiceStateTC' textual convention
        4)  Added 'jmMirrorAttr' and 'jmSystem' object identifiers
            reserved for future extensions
        5)  Added Service, Job, Job Completed, and Job Progress traps
            for job notifications (aligned with IPP Notifications)
        6)  Added Service, Service Event, Job Event, and Job Progress
            object groups to support above traps"
    REVISION    "9902190000Z" -- 19 February 1999
    DESCRIPTION
        "Version 1.0 - published as RFC 2707"

    ::= { mibs 1 }                      -- forward reference

pwg         OBJECT IDENTIFIER ::= { enterprises 2699 }
mibs        OBJECT IDENTIFIER ::= { pwg 1 }

-- Textual conventions for this MIB module

JmUTF8StringTC ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "255a"
    STATUS      current
    DESCRIPTION
        "To facilitate internationalization, this TC represents
        information taken from the ISO/IEC IS 10646-1 character set,
        encoded as an octet string using the UTF-8 character encoding
        scheme.

        See section 3.6.1, entitled: 'Text generated by the server or
        device'."
    SYNTAX      OCTET STRING (SIZE (0..63))

JmJobStringTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "To facilitate internationalization, this TC represents
        information using any coded character set registered by IANA as
        specified in section 3.7.  While it is recommended that the
        coded character set be UTF-8 [UTF-8], the actual coded
        character set SHALL be indicated by the value of the
        jobCodedCharSet(8) attribute for the job.

        See section 3.6.2, entitled: 'Text supplied by the job
        submitter'."
    SYNTAX      OCTET STRING (SIZE (0..63))


JmNaturalLanguageTagTC  ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "An IETF RFC 1766-compliant 'language tag', with zero or more
        sub-tags that identify a natural language.  While RFC 1766
        specifies that the US-ASCII values are case-insensitive, this
        MIB specification requires that all characters SHALL be lower
        case in order to simplify comparing by management applications.

        See section 3.6.1, entitled: 'Text generated by the server or
        device' and section 3.6.2, entitled: 'Text supplied by the job
        submitter'."
    SYNTAX      OCTET STRING (SIZE (0..63))

JmTimeStampTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The simple time at which an event took place.  The units are
        in seconds since the system was booted.

        NOTE - JmTimeStampTC is defined in units of seconds, rather
        than 100ths of seconds, so as to be simpler for agents to
        implement (even if they have to implement the 100ths of a
        second to comply with implementing sysUpTime in MIB-II[mib-
        II].)

        NOTE - JmTimeStampTC is defined as an Integer32 so that it can
        be used as a value of an attribute, i.e., as a value of the
        jmAttributeValueAsInteger object.  The TimeStamp textual-
        convention defined in SNMPv2-TC [SMIv2-TC] is defined as an
        APPLICATION 3 IMPLICIT INTEGER tag, not an Integer32 which is
        defined in SNMPv2-SMI [SMIv2-TC] as UNIVERSAL 2 IMPLICIT
        INTEGER, so cannot be used in this MIB as one of the values of
        jmAttributeValueAsInteger."
    SYNTAX      INTEGER (0..2147483647)

JmJobSourcePlatformTypeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The source platform type that can submit jobs to servers or
        devices in any of the 3 configurations.

        This is a type 2 enumeration.  See Section 3.7.1.2.  See also
        IANA operating-system-names registry."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
        sptUNIX(3),           -- UNIX
        sptOS2(4),            -- OS/2
        sptPCDOS(5),          -- DOS
        sptNT(6),             -- NT
        sptMVS(7),            -- MVS
        sptVM(8),             -- VM
        sptOS400(9),          -- OS/400
        sptVMS(10),           -- VMS
        sptWindows(11),       -- Windows
        sptNetWare(12)        -- NetWare
    }

JmFinishingTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The type of finishing operation.

        These values are the same as the enum values of the IPP
        'finishings' attribute.  See Section 3.7.1.2.

        other(1),
            Some other finishing operation besides one of the specified
            or registered values.

        unknown(2),
            The finishing is unknown.

        none(3),
            Perform no finishing.

        staple(4),
            Bind the document(s) with one or more staples. The exact
            number and placement of the staples is site-defined.

        punch(5),
            Holes are required in the finished document. The exact
            number and placement of the holes is site-defined.  The
            punch specification MAY be satisfied (in a site- and
            implementation-specific manner) either by
            drilling/punching, or by substituting pre-drilled media.

        cover(6),
            Select a non-printed (or pre-printed) cover for the
            document. This does not supplant the specification of a
            printed cover (on cover stock medium) by the document
            itself.

        bind(7)
            Binding is to be applied to the document; the type and
            placement of the binding is product-specific.

        This is a type 2 enumeration.  See Section 3.7.1.2."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
        none(3),
        staple(4),
        punch(5),
        cover(6),
        bind(7)
    }
```

```
JmPrintQualityTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Print quality settings.

        These values are the same as the enum values of the IPP 'print-
        quality' attribute.  See Section 3.7.1.2.

        This is a type 2 enumeration.  See Section 3.7.1.2."
    SYNTAX      INTEGER {
        other(1),    -- Not one of the specified or registered
                     -- values.
        unknown(2),  -- The actual value is unknown.
        draft(3),    -- Lowest quality available on the printer.
        normal(4),   -- Normal or intermediate quality on the
                     -- printer.
        high(5)      -- Highest quality available on the printer.
    }

JmPrinterResolutionTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Printer resolutions.

        Nine octets consisting of two 4-octet SIGNED-INTEGERs followed
        by a SIGNED-BYTE.  The values are the same as those specified
        in the Printer MIB [printmib]. The first SIGNED-INTEGER
        contains the value of prtMarkerAddressabilityXFeedDir.  The
        second SIGNED-INTEGER contains the value of
        prtMarkerAddressabilityFeedDir.  The SIGNED-BYTE contains the
        value of prtMarkerAddressabilityUnit.

        Note: the latter value is either 3 (tenThousandsOfInches) or 4
        (micrometers) and the addressability is in 10,000 units of
        measure. Thus the SIGNED-INTEGERs represent integral values in
        either dots-per-inch or dots-per-centimeter.

        The syntax is the same as the IPP 'printer-resolution'
        attribute.  See Section 3.7.1.2."
    SYNTAX      OCTET STRING (SIZE(9))

JmTonerEconomyTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Toner economy settings.

        This is a type 2 enumeration.  See Section 3.7.1.2."
    SYNTAX      INTEGER {
        unknown(2),    -- unknown.
        off(3),        -- Off.  Normal.  Use full toner.
        on(4)          -- On.  Use less toner than normal.
    }

JmBooleanTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Boolean true or false value.
        This is a type 2 enumeration.  See Section 3.7.1.2."
    SYNTAX      INTEGER {
        unknown(2),    -- unknown.
        false(3),      -- FALSE.
        true(4)        -- TRUE.
    }

JmMediumTypeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Identifies the type of medium.

        other(1),
            The type is neither one of the values listed in this
            specification nor a registered value.

        unknown(2),
            The type is not known.

        stationery(3),
            Separately cut sheets of an opaque material.

        transparency(4),
            Separately cut sheets of a transparent material.

        envelope(5),
            Envelopes that can be used for conventional mailing
            purposes.

        envelopePlain(6),
            Envelopes that are not preprinted and have no windows.

        envelopeWindow(7),
            Envelopes that have windows for addressing purposes.

        continuousLong(8),
            Continuously connected sheets of an opaque material
            connected along the long edge.

        continuousShort(9),
            Continuously connected sheets of an opaque material
            connected along the short edge.

        tabStock(10),
            Media with tabs.

        multiPartForm(11),
            Form medium composed of multiple layers not pre-attached to
            one another;  each sheet MAY be drawn separately from an
            input source.

        labels(12),
            Label-stock.

        multiLayer(13)
            Form medium composed of multiple layers which are pre-
            attached to one another, e.g. for use with impact printers.


        This is a type 2 enumeration.  See Section 3.7.1.2.  These enum
        values correspond to the keyword name strings of the
        prtInputMediaType object in the Printer MIB [print-mib].  There
        is no printer description attribute in IPP/1.0 that represents
        these values."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
        stationery(3),
        transparency(4),
        envelope(5),
        envelopePlain(6),
        envelopeWindow(7),
        continuousLong(8),
        continuousShort(9),
        tabStock(10),
        multiPartForm(11),
        labels(12),
        multiLayer(13)
    }

JmJobCollationTypeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This value is the type of job collation.  Implementations that
        don't support multiple documents or don't support multiple
        copies SHALL NOT support the uncollatedDocuments(5) value.

        This is a type 2 enumeration.  See Section 3.7.1.2. See also
        Section 3.4, entitled 'Monitoring Job Progress'."
    SYNTAX      INTEGER {
        other(1),
        unknown(2),
        uncollatedSheets(3),    -- sheets within each document copy
                                -- are not collated: 1 1 ..., 2 2 ...,
                                -- No corresponding value of IPP
                                -- "multiple-document-handling"
        collatedDocuments(4),   -- internal collated sheets,
                                -- documents: A, B, A, B, ...
                                -- Corresponds to IPP "multiple-
                                -- document-handling"='separate-
                                -- documents-collated-copies'
        uncollatedDocuments(5)  -- internal collated sheets,
                                -- documents: A, A, ..., B, B, ...
                                -- Corresponds to IPP "multiple-
                                -- document-handling"='separate-
                                -- documents-uncollated-copies'


    }

JmJobSubmissionIDTypeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
```

```
        "Identifies the format type of a job submission ID.

        Each job submission ID is a fixed-length, 48-octet printable
        US-ASCII [US-ASCII] coded character string containing no
        control characters, consisting of the fields defined in section
        3.5.1.

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

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

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

                Figure 4 - Normal Job State Transitions

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

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


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

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

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

        pendingHeld(4),
            The job is not a candidate for processing for any number of
            reasons but will return to the pending state as soon as the
            reasons are no longer present.  The job's
            jmJobStateReasons1 object and/or jobStateReasonsN (N=2..4)
            attributes SHALL indicate why the job is no longer a
            candidate for processing.  The reasons are represented as
            bits in the jmJobStateReasons1 object and/or
            jobStateReasonsN (N=2..4) attributes.  See the
            JmJobStateReasonsNTC (N=1..4) textual convention for the
            specification of each reason.

        processing(5),
            One or more of:

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

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

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

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

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

        processingStopped(6),
            The job has stopped while processing for any number of
            reasons and will return to the processing state as soon
            as the reasons are no longer present.

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

            NOTE - When an output device is stopped, the device
            usually indicates its condition in human readable form
            at the device.  The management application can obtain
             more complete device status remotely by querying the
            appropriate device MIB using the job's deviceIndex
            attribute(s), if the agent implements such a device MIB
        canceled(7),
            A client has canceled the job and the server or device
            has completed canceling the job AND all MIB objects and
            attributes have reached their final values for the job.
            While the server or device is canceling the job, the
            job's jmJobStateReasons1 object SHOULD contain the
            processingToStopPoint value and one of the
            canceledByUser, canceledByOperator, or canceledAtDevice
            values.  The canceledByUser, canceledByOperator, or
            canceledAtDevice values remain while the job is in the
            canceled state.

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


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

        This is a type 2 enumeration.  See Section 3.7.1.2."
    SYNTAX      INTEGER {
        unknown(2),
        pending(3),
        pendingHeld(4),
        processing(5),
        processingStopped(6),
        canceled(7),
        aborted(8),
        completed(9)
    }

JmAttributeTypeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The type of the attribute which identifies the attribute.

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

        Values in the range 2**30 to 2**31-1 are reserved for private
        or experimental usage.  This range corresponds to the same
        range reserved in IPP.  Implementers are warned that use of
        such values may conflict with other implementations.
        Implementers are encouraged to request registration of enum
        values following the procedures in Section 3.7.1.

        See Section 3.2 entitled 'The Attribute Mechanism' for a
        description of this textual-convention and its use in the
        jmAttributeTable.  See Section 3.3.8 for the specification of
        each attribute.  The comment(s) after each enum assignment
        specifies the data type(s) of the attribute.

        This is a type 2 enumeration.  See Section 3.7.1.2."


    SYNTAX      INTEGER {
        other(1),                       -- Integer32 (-2..2147483647)
                                        -- AND/OR
                                        -- OCTET STRING(SIZE(0..63))

        -- Job State attributes:
        jobStateReasons2(3),            -- JmJobStateReasons2TC
        jobStateReasons3(4),            -- JmJobStateReasons3TC
        jobStateReasons4(5),            -- JmJobStateReasons4TC
        processingMessage(6),           -- JmUTF8StringTC (SIZE(0..63))
        processingMessageNaturalLangTag(7),
                                        -- OCTET STRING(SIZE(0..63))
        jobCodedCharSet(8),             -- CodedCharSet
        jobNaturalLanguageTag(9),       -- OCTET STRING(SIZE(0..63))

        -- Job Identification attributes:
        jobURI(20),                     -- OCTET STRING(SIZE(0..63))
        jobAccountName(21),             -- OCTET STRING(SIZE(0..63))
        serverAssignedJobName(22),      -- JmJobStringTC (SIZE(0..63))
        jobName(23),                    -- JmJobStringTC (SIZE(0..63))
        jobServiceTypes(24),            -- JmJobServiceTypesTC
        jobSourceChannelIndex(25),      -- Integer32 (0..2147483647)
        jobSourcePlatformType(26),      -- JmJobSourcePlatformTypeTC
        submittingServerName(27),       -- JmJobStringTC (SIZE(0..63))
        submittingApplicationName(28),  -- JmJobStringTC (SIZE(0..63))
        jobOriginatingHost(29),         -- JmJobStringTC (SIZE(0..63))
        deviceNameRequested(30),        -- JmJobStringTC (SIZE(0..63))
        queueNameRequested(31),         -- JmJobStringTC (SIZE(0..63))
        physicalDevice(32),             -- hrDeviceIndex
                                        -- AND/OR
                                        -- JmUTF8StringTC (SIZE(0..63))
        numberOfDocuments(33),          -- Integer32 (-2..2147483647)
        fileName(34),                   -- JmJobStringTC (SIZE(0..63))
        documentName(35),               -- JmJobStringTC (SIZE(0..63))
        jobComment(36),                 -- JmJobStringTC (SIZE(0..63))
        documentFormatIndex(37),        -- Integer32 (0..2147483647)
        documentFormat(38),             -- PrtInterpreterLangFamilyTC
                                        -- AND/OR
                                        -- OCTET STRING(SIZE(0..63))

        -- Job Parameter attributes:
        jobPriority(50),                -- Integer32 (-2..100)
        jobProcessAfterDateAndTime(51), -- DateAndTime (SNMPv2-TC)
        jobHold(52),                    -- JmBooleanTC
        jobHoldUntil(53),               -- JmJobStringTC (SIZE(0..63))
        outputBin(54),                  -- Integer32 (0..2147483647)
                                        -- AND/OR
                                        -- JmJobStringTC (SIZE(0..63))
        sides(55),                      -- Integer32 (-2..2)
        finishing(56),                  -- JmFinishingTC

        -- Image Quality attributes:
        printQualityRequested(70),      -- JmPrintQualityTC
        printQualityUsed(71),           -- JmPrintQualityTC
        printerResolutionRequested(72), -- JmPrinterResolutionTC
        printerResolutionUsed(73),      -- JmPrinterResolutionTC
        tonerEcomonyRequested(74),      -- JmTonerEconomyTC
        tonerEcomonyUsed(75),           -- JmTonerEconomyTC
        tonerDensityRequested(76),      -- Integer32 (-2..100)
        tonerDensityUsed(77),           -- Integer32 (-2..100)

        -- Job Progress attributes:
        jobCopiesRequested(90),         -- Integer32 (-2..2147483647)
        jobCopiesCompleted(91),         -- Integer32 (-2..2147483647)
        documentCopiesRequested(92),    -- Integer32 (-2..2147483647)
        documentCopiesCompleted(93),    -- Integer32 (-2..2147483647)
        jobKOctetsTransferred(94),      -- Integer32 (-2..2147483647)
        sheetCompletedCopyNumber(95),   -- Integer32 (-2..2147483647)
        sheetCompletedDocumentNumber(96),
                                        -- Integer32 (-2..2147483647)
        jobCollationType(97),           -- JmJobCollationTypeTC

        -- Impression attributes:
        impressionsSpooled(110),        -- Integer32 (-2..2147483647)
        impressionsSentToDevice(111),   -- Integer32 (-2..2147483647)
        impressionsInterpreted(112),    -- Integer32 (-2..2147483647)
        impressionsCompletedCurrentCopy(113),
                                        -- Integer32 (-2..2147483647)
        fullColorImpressionsCompleted(114),
                                        -- Integer32 (-2..2147483647)
        highlightColorImpressionsCompleted(115),
                                        -- Integer32 (-2..2147483647)

        -- Page attributes:
        pagesRequested(130),            -- Integer32 (-2..2147483647)
        pagesCompleted(131),            -- Integer32 (-2..2147483647)
        pagesCompletedCurrentCopy(132), -- Integer32 (-2..2147483647)

        -- Sheet attributes:
        sheetsRequested(150),           -- Integer32 (-2..2147483647)
        sheetsCompleted(151),           -- Integer32 (-2..2147483647)
        sheetsCompletedCurrentCopy(152),-- Integer32 (-2..2147483647)

        -- Resource attributes:
        mediumRequested(170),           -- JmMediumTypeTC
                                        -- AND/OR
                                        -- JmJobStringTC (SIZE(0..63))
        mediumConsumed(171),            -- Integer32 (-2..2147483647)
                                        -- AND
                                        -- JmJobStringTC (SIZE(0..63))
```

```
        colorantRequested(172),         -- Integer32 (-2..2147483647)
                                        -- AND/OR
                                        -- JmJobStringTC (SIZE(0..63))
        colorantConsumed(173),          -- Integer32 (-2..2147483647)
                                        -- AND/OR
                                        -- JmJobStringTC (SIZE(0..63))
        mediumTypeConsumed(174),        -- Integer32 (-2..2147483647)
                                        -- AND
                                        -- JmJobStringTC (SIZE(0..63))
        mediumSizeConsumed(175),        -- Integer32 (-2..2147483647)
                                        -- AND
                                        -- JmJobStringTC (SIZE(0..63))

        -- Time attributes:
        jobSubmissionToServerTime(190), -- JmTimeStampTC
                                        -- AND/OR
                                        -- DateAndTime
        jobSubmissionTime(191),         -- JmTimeStampTC
                                        -- AND/OR
                                        -- DateAndTime
        jobStartedBeingHeldTime(192),   -- JmTimeStampTC
                                        -- AND/OR
                                        -- DateAndTime
        jobStartedProcessingTime(193),  -- JmTimeStampTC
                                        -- AND/OR
                                        -- DateAndTime
        jobCompletionTime(194),         -- JmTimeStampTC
                                        -- AND/OR
                                        -- DateAndTime
        jobProcessingCPUTime(195)       -- Integer32 (-2..2147483647)
    }

JmJobServiceTypesTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Specifies the type(s) of service to which the job has been
        submitted (print, fax, scan, etc.).  The service type is
        represented as an enum that is bit encoded with each job
        service type so that more general and arbitrary services can be
        created, such as services with more than one destination type,
        or ones with only a source or only a destination.  For example,
        a job service might scan, faxOut, and print a single job.  In
        this case, three bits would be set in the jobServiceTypes
        attribute, corresponding to the hexadecimal values: 0x8 + 0x20
        + 0x4, respectively, yielding: 0x2C.

        Whether this attribute is set from a job attribute supplied by
        the job submission client or is set by the recipient job
        submission server or device depends on the job submission
        protocol.  With either implementation, the agent SHALL return a
        non-zero value for this attribute indicating the type of the
        job.

        One of the purposes of this attribute is to permit a requester
        to filter out jobs that are not of interest.  For example, a
        printer operator MAY only be interested in jobs that include
        printing.  That is why the attribute is in the job
        identification category.

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

        other                             0x1
            The job contains some instructions that are not one of the
            identified types.

        unknown                           0x2
            The job contains some instructions whose type is unknown to
            the agent.

        print                             0x4
            The job contains some instructions that specify printing

        scan                              0x8
            The job contains some instructions that specify scanning

        faxIn                             0x10
            The job contains some instructions that specify receive fax

        faxOut                            0x20
            The job contains some instructions that specify sending fax

        getFile                           0x40
            The job contains some instructions that specify accessing
            files or documents

        putFile                           0x80
            The job contains some instructions that specify storing
            files or documents

        mailList                          0x100
            The job contains some instructions that specify
            distribution of documents using an electronic mail system.

        These bit definitions are the equivalent of a type 2 enum
        except that combinations of them MAY be used together.  See
        section 3.7.1.2."
    SYNTAX      INTEGER (0..2147483647)   -- 31 bits, all but sign bit

JmJobStateReasons1TC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The JmJobStateReasonsNTC (N=1..4) textual-conventions are used
        with the jmJobStateReasons1 object and jobStateReasonsN
        (N=2..4), respectively, to provide additional information
        regarding the current jmJobState object value.  These values
        MAY be used with any job state or states for which the reason
        makes sense.  See section 3.3.9.1 for the specification of each
        bit value defined for use with the JmJobStateReasons1TC.

        These bit definitions are the equivalent of a type 2 enum
        except that combinations of bits may be used together.  See
        section 3.7.1.2."
    SYNTAX      INTEGER (0..2147483647)   -- 31 bits, all but sign bit

JmJobStateReasons2TC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This textual-convention is used with the jobStateReasons2
        attribute to provides additional information regarding the
        jmJobState object.  See section 3.3.9.2 for the specification
        of JmJobStateReasons2TC.  See section 3.3.9.1 for the
        description under JmJobStateReasons1TC for additional
        information that applies to all reasons.

        These bit definitions are the equivalent of a type 2 enum
        except that combinations of them may be used together.  See
        section 3.7.1.2."
    SYNTAX      INTEGER (0..2147483647)   -- 31 bits, all but sign bit

JmJobStateReasons3TC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This textual-convention is used with the jobStateReasons3
        attribute to provides additional information regarding the
        jmJobState object.  See section 3.3.9.3 for the specification
        of JmJobStateReasons3TC.  See section 3.3.9.1 for the
        description under JmJobStateReasons1TC for additional
        information that applies to all reasons.

        These bit definitions are the equivalent of a type 2 enum
        except that combinations of them may be used together.  See
        section 3.7.1.2.  "
    SYNTAX      INTEGER (0..2147483647)   -- 31 bits, all but sign bit

JmJobStateReasons4TC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This textual-convention is used in the jobStateReasons4
        attribute to provides additional information regarding the
        jmJobState object.  See section 3.3.9.4 for the specification
        of JmJobStateReasons4TC.  See section 3.3.9.1 for the
        description under JmJobStateReasons1TC for additional
        information that applies to all reasons.

        These bit definitions are the equivalent of a type 2 enum
        except that combinations of them may be used together.  See
        section 3.7.1.2."
    SYNTAX      INTEGER (0..2147483647)   -- 31 bits, all but sign bit

JmServiceStateTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The current state of the service.

        See:    Section 4.4.11 'printer-state' in [IPP-MOD].

        This is a type 2 enumeration.  See Section 3.7.1.2."
    SYNTAX      INTEGER {
        unknown(2),
        idle(3),                        -- new jobs start immediately
        processing(4),                  -- jobs are processing
        stopped(5)                      -- intervention required
    }

jobmonMIBObjects  OBJECT IDENTIFIER  ::= { jobmonMIB 1 }

-- The General Group (MANDATORY)

-- The jmGeneralGroup consists entirely of the jmGeneralTable.

jmGeneral  OBJECT IDENTIFIER ::= { jobmonMIBObjects 1 }

jmGeneralTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmGeneralEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmGeneralTable consists of information of a general nature
        that are per-job-set, but are not per-job.  See Section 2
        entitled 'Terminology and Job Model' for the definition of a
        job set.

        The MANDATORY-GROUP macro specifies that this group is
        MANDATORY."
    ::= { jmGeneral 1 }

jmGeneralEntry  OBJECT-TYPE
    SYNTAX      JmGeneralEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information about a job set (queue).

        An entry SHALL exist in this table for each job set."
    INDEX  { jmGeneralJobSetIndex }
    ::= { jmGeneralTable 1 }

JmGeneralEntry ::= SEQUENCE {
    jmGeneralJobSetIndex                  Integer32 (1..32767),
    jmGeneralNumberOfActiveJobs           Integer32 (0..2147483647),
    jmGeneralOldestActiveJobIndex         Integer32 (0..2147483647),
    jmGeneralNewestActiveJobIndex         Integer32 (0..2147483647),
    jmGeneralJobPersistence               Integer32 (15..2147483647),
    jmGeneralAttributePersistence         Integer32 (15..2147483647),
    jmGeneralJobSetName                   JmUTF8StringTC (SIZE(0..63))
}

jmGeneralJobSetIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..32767)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value for each job set in this MIB.  The jmJobTable
        and jmAttributeTable tables have this same index as their
        primary index.

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

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

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

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

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

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

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

jmGeneralNewestActiveJobIndex  OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-only


    STATUS      current
    DESCRIPTION
        "The jmJobIndex of the newest job that is in one of the
        'active' states (pending, processing, or processingStopped).
        In other words, the index of the 'active' job that has been
        most recently added to the job tables.

        When all jobs become 'inactive', i.e., enter the pendingHeld,
        completed, canceled, or aborted states, the agent SHALL set the
        value of this object to 0.

        See Section 3.2 entitled 'The Job Tables and the Oldest Active
        and Newest Active Indexes' for a description of the usage of
        this object."
    DEFVAL      { 0 }      -- no active jobs
    ::= { jmGeneralEntry 4 }

jmGeneralJobPersistence OBJECT-TYPE
    SYNTAX      Integer32 (15..2147483647)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The minimum time in seconds for this instance of the Job Set
        that an entry SHALL remain in the jmJobIDTable and jmJobTable
        after processing has completed, i.e., the minimum time in
        seconds starting when the job enters the completed, canceled,
        or aborted state.

        Configuring this object is implementation-dependent.

        This value SHALL be equal to or greater than the value of
        jmGeneralAttributePersistence.  This value SHOULD be at least
        60 which gives a monitoring or accounting application one
        minute in which to poll for job data."
    DEFVAL      { 60 }          -- one minute
    ::= { jmGeneralEntry 5 }

jmGeneralAttributePersistence OBJECT-TYPE
    SYNTAX      Integer32 (15..2147483647)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The minimum time in seconds for this instance of the Job Set
        that an entry SHALL remain in the jmAttributeTable after
        processing has completed , i.e., the time in seconds starting
        when the job enters the completed, canceled, or aborted state.

        Configuring this object is implementation-dependent.

        This value SHOULD be at least 60 which gives a monitoring or
        accounting application one minute in which to poll for job
        data."
    DEFVAL      { 60 }          -- one minute
    ::= { jmGeneralEntry 6 }

jmGeneralJobSetName OBJECT-TYPE
    SYNTAX      JmUTF8StringTC (SIZE(0..63))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "The human readable name of this job set assigned by the system
        administrator (by means outside of this MIB).  Typically, this
        name SHOULD be the name of the job queue.  If a server or
        device has only a single job set, this object can be the
        administratively assigned name of the server or device itself.
        This name does not need to be unique, though each job set in a
        single Job Monitoring MIB SHOULD have distinct names.

        NOTE - If the job set corresponds to a single printer and the
        Printer MIB is implemented, this value SHOULD be the same as
        the prtGeneralPrinterName object in the draft Printer MIB
        [print-mib-draft].  If the job set corresponds to an IPP
        Printer, this value SHOULD be the same as the IPP 'printer-
        name' Printer attribute.

        NOTE - The purpose of this object is to help the user of the
        job monitoring application distinguish between several job sets
        in implementations that support more than one job set.

        See the OBJECT compliance macro for the minimum maximum length
        required for conformance."
    DEFVAL      { ''H }      -- empty string
    ::= { jmGeneralEntry 7 }

-- The Job ID Group (MANDATORY)

-- The jmJobIDGroup consists entirely of the jmJobIDTable.


jmJobID  OBJECT IDENTIFIER ::= { jobmonMIBObjects 2 }

jmJobIDTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmJobIDEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmJobIDTable provides a correspondence map (1) between the
        job submission ID that a client uses to refer to a job and (2)
        the jmGeneralJobSetIndex and jmJobIndex that the Job Monitoring
        MIB agent assigned to the job and that are used to access the
        job in all of the other tables in the MIB.  If a monitoring
        application already knows the jmGeneralJobSetIndex and the
        jmJobIndex of the job it is querying, that application NEED NOT
        use the jmJobIDTable.

        The MANDATORY-GROUP macro specifies that this group is
        MANDATORY."
    ::= { jmJobID 1 }

jmJobIDEntry  OBJECT-TYPE
    SYNTAX      JmJobIDEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The map from (1) the jmJobSubmissionID to (2) the
        jmGeneralJobSetIndex and jmJobIndex.

        An entry SHALL exist in this table for each job currently known
        to the agent for all job sets and job states.  There MAY be
        more than one jmJobIDEntry that maps to a single job.  This
        many to one mapping can occur when more than one network entity
        along the job submission path supplies a job submission ID.
        See Section 3.5.  However, each job SHALL appear once and in
        one and only one job set."
    INDEX  { jmJobSubmissionID }
    ::= { jmJobIDTable 1 }

JmJobIDEntry ::= SEQUENCE {
    jmJobSubmissionID                     OCTET STRING(SIZE(48)),
    jmJobIDJobSetIndex                    Integer32 (0..32767),
    jmJobIDJobIndex                       Integer32 (0..2147483647)
}

jmJobSubmissionID OBJECT-TYPE
    SYNTAX      OCTET STRING(SIZE(48))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A quasi-unique 48-octet fixed-length string ID which
        identifies the job within a particular client-server
        environment.  There are multiple formats for the
        jmJobSubmissionID.  Each format SHALL be uniquely identified.
        See the JmJobSubmissionIDTypeTC textual convention.  Each
        format SHALL be registered using the procedures of a type 2
        enum.  See section 3.7.3 entitled: 'PWG Registration of Job
        Submission Id Formats'.

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

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

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

        See the JmJobSubmissionIDTypeTC textual convention.
        See APPENDIX B - Support of Job Submission Protocols."
    ::= { jmJobIDEntry 1 }

jmJobIDJobSetIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..32767)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the value of the jmGeneralJobSetIndex for
        the job with the jmJobSubmissionID value, i.e., the job set
        index of the job set in which the job was placed when that
        server or device accepted the job.  This 16-bit value in
        combination with the jmJobIDJobIndex value permits the
        management application to access the other tables to obtain the
        job-specific objects for this job.

        See jmGeneralJobSetIndex in the jmGeneralTable."
    DEFVAL      { 0 }      -- 0 indicates no job set index
    ::= { jmJobIDEntry 2 }

jmJobIDJobIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the value of the jmJobIndex for the job
        with the jmJobSubmissionID value, i.e., the job index for the
        job when the server or device accepted the job.  This value, in
        combination with the jmJobIDJobSetIndex value, permits the
        management application to access the other tables to obtain the
        job-specific objects for this job.

        See jmJobIndex in the jmJobTable."
    DEFVAL      { 0 }      -- 0 indicates no jmJobIndex value.
    ::= { jmJobIDEntry 3 }

-- The Job Group (MANDATORY)

-- The jmJobGroup consists entirely of the jmJobTable.

jmJob  OBJECT IDENTIFIER ::= { jobmonMIBObjects 3 }

jmJobTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmJobEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmJobTable consists of basic job state and status
        information for each job in a job set that (1) monitoring
        applications need to be able to access in a single SNMP Get
        operation, (2) that have a single value per job, and (3) that
        SHALL always be implemented.

        The MANDATORY-GROUP macro specifies that this group is
        MANDATORY."
    ::= { jmJob 1 }


jmJobEntry  OBJECT-TYPE
    SYNTAX      JmJobEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
        "Basic per-job state and status information.

        An entry SHALL exist in this table for each job, no matter what
        the state of the job is.  Each job SHALL appear in one and only
        one job set.

        See Section 3.2 entitled 'The Job Tables'."
    INDEX  { jmGeneralJobSetIndex, jmJobIndex }
    ::= { jmJobTable 1 }

JmJobEntry ::= SEQUENCE {
    jmJobIndex                            Integer32 (1..2147483647),
    jmJobState                            JmJobStateTC,
    jmJobStateReasons1                    JmJobStateReasons1TC,
    jmNumberOfInterveningJobs             Integer32 (-2..2147483647),
    jmJobKOctetsPerCopyRequested          Integer32 (-2..2147483647),
    jmJobKOctetsProcessed                 Integer32 (-2..2147483647),
    jmJobImpressionsPerCopyRequested      Integer32 (-2..2147483647),
    jmJobImpressionsCompleted             Integer32 (-2..2147483647),
    jmJobOwner                            JmJobStringTC (SIZE(0..63))
}

jmJobIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The sequential, monatonically increasing identifier index for
        the job generated by the server or device when that server or
        device accepted the job.  This index value permits the
        management application to access the other tables to obtain the
        job-specific row entries.

        See Section 3.2 entitled 'The Job Tables and the Oldest Active
        and Newest Active Indexes'.
        See Section 3.5 entitled 'Job Identification'.
        See also jmGeneralNewestActiveJobIndex for the largest value of
        jmJobIndex.
        See JmJobSubmissionIDTypeTC for a limit on the size of this
        index if the agent represents it as an 8-digit decimal number."
    ::= { jmJobEntry 1 }


jmJobState OBJECT-TYPE
    SYNTAX      JmJobStateTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current state of the job (pending, processing, completed,
        etc.).  Agents SHALL implement only those states which are
        appropriate for the particular implementation.  However,
        management applications SHALL be prepared to receive all the
        standard job states.

        The final value for this object SHALL be one of: completed,
        canceled, or aborted.  The minimum length of time that the
        agent SHALL maintain MIB data for a job in the completed,
        canceled, or aborted state before removing the job data from
        the jmJobIDTable and jmJobTable is specified by the value of
        the jmGeneralJobPersistence object."
    DEFVAL      { unknown }      -- default is unknown
    ::= { jmJobEntry 2 }

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

        Implementation of any reason values is OPTIONAL, but an agent
        SHOULD return any reason information available.  These values
        MAY be used with any job state or states for which the reason
        makes sense.  Since the Job State Reasons will be more dynamic
        than the Job State, it is recommended that a job monitoring
        application read this object every time jmJobState is read.
        When the agent cannot provide a reason for the current state of
        the job, the value of the jmJobStateReasons1 object and
        jobStateReasonsN attributes SHALL be 0.

        The jobStateReasonsN (N=2..4) attributes provide further
        additional information about the job's current state."
    DEFVAL      { 0 }      -- no reasons
    ::= { jmJobEntry 3 }


jmNumberOfInterveningJobs OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of jobs that are expected to complete processing
        before this job has completed processing according to the
        implementation's queuing algorithm, if no other jobs were to be
        submitted.  In other words, this value is the job's queue
        position.  The agent SHALL return a value of 0 for this
        attribute when the job is the next job to complete processing
        (or has completed processing)."
    DEFVAL      { 0 }      -- default is no intervening jobs.
    ::= { jmJobEntry 4 }

jmJobKOctetsPerCopyRequested OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total size in K (1024) octets of the document(s) being
        requested to be processed in the job.  The agent SHALL round
        the actual number of octets up to the next highest K.  Thus 0
        octets is represented as '0', 1-1024 octets is represented as
        '1', 1025-2048 is represented as '2', etc.
        In computing this value, the server/device SHALL NOT include
        the multiplicative factors contributed by (1) the number of
        document copies, and (2) the number of job copies, independent
        of whether the device can process multiple copies of the job or
        document without making multiple passes over the job or
        document data and independent of whether the output is collated
        or not.  Thus the server/device computation is independent of
        the implementation and indicates the size of the document(s)
        measured in K octets independent of the number of copies."
    DEFVAL      { -2 }      -- the default is unknown(-2)
    ::= { jmJobEntry 5 }

jmJobKOctetsProcessed OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of octets processed by the server or device
        measured in units of K (1024) octets so far.  The agent SHALL
        round the actual number of octets processed up to the next
        higher K.  Thus 0 octets is represented as '0', 1-1024 octets
        is represented as '1', 1025-2048 octets is '2', etc.  For
        printing devices, this value is the number interpreted by the
        page description language interpreter rather than what has been
        marked on media.

        For implementations where multiple copies are produced by the
        interpreter with only a single pass over the data, the final
        value SHALL be equal to the value of the
        jmJobKOctetsPerCopyRequested object.  For implementations where
        multiple copies are produced by the interpreter by processing
        the data for each copy, the final value SHALL be a multiple of
        the value of the jmJobKOctetsPerCopyRequested object.

        NOTE - See the impressionsCompletedCurrentCopy and
        pagesCompletedCurrentCopy attributes for attributes that are
        reset on each document copy.

        NOTE - The jmJobKOctetsProcessed object can be used with the
        jmJobKOctetsPerCopyRequested object to provide an indication of
        the relative progress of the job, provided that the
        multiplicative factor is taken into account for some
        implementations of multiple copies."
    DEFVAL      { 0 }      -- default is no octets processed.
    ::= { jmJobEntry 6 }

jmJobImpressionsPerCopyRequested OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total size in number of impressions of the document(s)
        submitted.

        In computing this value, the server/device SHALL NOT include
        the multiplicative factors contributed by (1) the number of
        document copies, and (2) the number of job copies, independent
        of whether the device can process multiple copies of the job or
        document without making multiple passes over the job or
        document data and independent of whether the output is collated
        or not.  Thus the server/device computation is independent of
        the implementation and reflects the size of the document(s)
        measured in impressions independent of the number of copies.

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

jmJobImpressionsCompleted OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of impressions completed for this job so far.
        For printing devices, the impressions completed includes
        interpreting, marking, and stacking the output.  For other
        types of job services, the number of impressions completed
        includes the number of impressions processed.

        NOTE - See the impressionsCompletedCurrentCopy and
        pagesCompletedCurrentCopy attributes for attributes that are
        reset on each document copy.

        NOTE - The jmJobImpressionsCompleted object can be used with
        the jmJobImpressionsPerCopyRequested object to provide an
        indication of the relative progress of the job, provided that
        the multiplicative factor is taken into account for some
        implementations of multiple copies.

        See the definition of the term 'impression' in Section 2 and
        the counting example in Section 3.4 entitled 'Monitoring Job
        Progress'."
    DEFVAL      { 0 }      -- default is no octets
    ::= { jmJobEntry 8 }

jmJobOwner OBJECT-TYPE
    SYNTAX      JmJobStringTC (SIZE(0..63))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The coded character set name of the user that submitted the
        job.  The method of assigning this user name will be system
        and/or site specific but the method MUST ensure that the name
        is unique to the network that is visible to the client and
        target device.

        This value SHOULD be the most authenticated name of the user
        submitting the job.

        See the OBJECT compliance macro for the minimum maximum length
        required for conformance."
    DEFVAL      { ''H }      -- default is empty string
    ::= { jmJobEntry 9 }
```

```
-- The Attribute Group (MANDATORY)

-- The jmAttributeGroup consists entirely of the jmAttributeTable.
--
-- Implementation of the objects in this group is MANDATORY.
-- See Section 3.1 entitled 'Conformance Considerations'.
-- An agent SHALL implement any attribute if (1) the server or device
-- supports the functionality represented by the attribute and (2) the
-- information is available to the agent.

jmAttribute  OBJECT IDENTIFIER ::= { jobmonMIBObjects 4 }

jmAttributeTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmAttributeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmAttributeTable SHALL contain attributes of the job and
        document(s) for each job in a job set.  Instead of allocating
        distinct objects for each attribute, each attribute is
        represented as a separate row in the jmAttributeTable.

        The MANDATORY-GROUP macro specifies that this group is
        MANDATORY.  An agent SHALL implement any attribute if (1) the
        server or device supports the functionality represented by the
        attribute and (2) the information is available to the agent. "
    ::= { jmAttribute 1 }

jmAttributeEntry  OBJECT-TYPE
    SYNTAX      JmAttributeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Attributes representing information about the job and
        document(s) or resources required and/or consumed.

        Each entry in the jmAttributeTable is a per-job entry with an
        extra index for each type of attribute (jmAttributeTypeIndex)
        that a job can have and an additional index
        (jmAttributeInstanceIndex) for those attributes that can have
        multiple instances per job.  The jmAttributeTypeIndex object
        SHALL contain an enum type that indicates the type of attribute
        (see the JmAttributeTypeTC textual-convention).  The value of
        the attribute SHALL be represented in either the
        jmAttributeValueAsInteger or jmAttributeValueAsOctets objects,
        and/or both, as specified in the JmAttributeTypeTC textual-
        convention.

        The agent SHALL create rows in the jmAttributeTable as the
        server or device is able to discover the attributes either from
        the job submission protocol itself or from the document PDL.
        As the documents are interpreted, the interpreter MAY discover
        additional attributes and so the agent adds additional rows to
        this table.  As the attributes that represent resources are
        actually consumed, the usage counter contained in the
        jmAttributeValueAsInteger object is incremented according to
        the units indicated in the description of the JmAttributeTypeTC
        enum.

        The agent SHALL maintain each row in the jmAttributeTable for
        at least the minimum time after a job completes as specified by
        the jmGeneralAttributePersistence object.

        Zero or more entries SHALL exist in this table for each job in
        a job set.

        See Section 3.3 entitled 'The Attribute Mechanism' for a
        description of the jmAttributeTable."
    INDEX  { jmGeneralJobSetIndex, jmJobIndex, jmAttributeTypeIndex,
    jmAttributeInstanceIndex }
    ::= { jmAttributeTable 1 }

JmAttributeEntry ::= SEQUENCE {
    jmAttributeTypeIndex                  JmAttributeTypeTC,
    jmAttributeInstanceIndex              Integer32 (1..32767),
    jmAttributeValueAsInteger             Integer32 (-2..2147483647),
    jmAttributeValueAsOctets              OCTET STRING(SIZE(0..63))
}

jmAttributeTypeIndex OBJECT-TYPE
    SYNTAX      JmAttributeTypeTC
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The type of attribute that this row entry represents.

        The type MAY identify information about the job or document(s)
        or MAY identify a resource required to process the job before
        the job start processing and/or consumed by the job as the job
        is processed.

        Examples of job attributes (i.e., apply to the job as a whole)
        that have only one instance per job include:
        jobCopiesRequested(90), documentCopiesRequested(92),
        jobCopiesCompleted(91), documentCopiesCompleted(93), while
        examples of job attributes that may have more than one instance
        per job include:  documentFormatIndex(37), and
        documentFormat(38).

        Examples of document attributes (one instance per document)
        include: fileName(34), and documentName(35).

        Examples of required and consumed resource attributes include:
        pagesRequested(130), mediumRequested(170), pagesCompleted(131),
        and mediumConsumed(171), respectively."
    ::= { jmAttributeEntry 1 }

jmAttributeInstanceIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..32767)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A running 16-bit index of the attributes of the same type for
        each job.  For those attributes with only a single instance per
        job, this index value SHALL be 1.  For those attributes that
        are a single value per document, the index value SHALL be the
        document number, starting with 1 for the first document in the
        job.  Jobs with only a single document SHALL use the index
        value of 1.  For those attributes that can have multiple values
        per job or per document, such as documentFormatIndex(37) or
        documentFormat(38), the index SHALL be a running index for the
        job as a whole, starting at 1."
    ::= { jmAttributeEntry 2 }

jmAttributeValueAsInteger OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The integer value of the attribute.  The value of the
        attribute SHALL be represented as an integer if the enum
        description in the JmAttributeTypeTC textual-convention
        definition has the tag: 'INTEGER:'.

        Depending on the enum definition, this object value MAY be an
        integer, a counter, an index, or an enum, depending on the
        jmAttributeTypeIndex value.  The units of this value are
        specified in the enum description.

        For those attributes that are accumulating job consumption as
        the job is processed as specified in the JmAttributeTypeTC
        textual-convention, SHALL contain the final value after the job
        completes processing, i.e., this value SHALL indicate the total
        usage of this resource made by the job.

        A monitoring application is able to copy this value to a
        suitable longer term storage for later processing as part of an
        accounting system.

        Since the agent MAY add attributes representing resources to
        this table while the job is waiting to be processed or being
        processed, which can be a long time before any of the resources
        are actually used, the agent SHALL set the value of the
        jmAttributeValueAsInteger object to 0 for resources that the
        job has not yet consumed.

        Attributes for which the concept of an integer value is
        meaningless, such as fileName(34), jobName, and
        processingMessage, do not have the 'INTEGER:' tag in the
        JmAttributeTypeTC definition and so an agent SHALL always
        return a value of '-1' to indicate 'other' for the value of the
        jmAttributeValueAsInteger object for these attributes.

        For attributes which do have the 'INTEGER:' tag in the
        JmAttributeTypeTC definition, if the integer value is not (yet)
        known, the agent either (1) SHALL not materialize the row in
        the jmAttributeTable until the value is known or (2) SHALL
        return a '-2' to represent an 'unknown' counting integer value,
        a '0' to represent an 'unknown' index value, and a '2' to
        represent an 'unknown(2)' enum value."
    DEFVAL      { -2 }      -- default value is unknown(-2)
    ::= { jmAttributeEntry 3 }

jmAttributeValueAsOctets OBJECT-TYPE
    SYNTAX      OCTET STRING(SIZE(0..63))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The octet string value of the attribute.  The value of the
        attribute SHALL be represented as an OCTET STRING if the enum
        description in the JmAttributeTypeTC textual-convention
        definition has the tag: 'OCTETS:'.

        Depending on the enum definition, this object value MAY be a
        coded character set string (text), such as 'JmUTF8StringTC', or
        a binary octet string, such as 'DateAndTime'.

        Attributes for which the concept of an octet string value is
        meaningless, such as pagesCompleted, do not have the tag
        'OCTETS:' in the JmAttributeTypeTC definition and so the agent
        SHALL always return a zero length string for the value of the
        jmAttributeValueAsOctets object.

        For attributes which do have the 'OCTETS:' tag in the
        JmAttributeTypeTC definition, if the OCTET STRING value is not
        (yet) known, the agent either SHALL NOT materialize the row in
        the jmAttributeTable until the value is known or SHALL return a
        zero-length string."
    DEFVAL      { ''H }      -- empty string
    ::= { jmAttributeEntry 4 }

--  Reserved identifiers for proposed future extensions

jmMirrorAttr  OBJECT IDENTIFIER ::= { jobmonMIBObjects 5 }
jmSystem      OBJECT IDENTIFIER ::= { jobmonMIBObjects 6 }


--  The Service Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which show Service state via SNMP.

--  The jmServiceGroup consists entirely of the jmServiceTable

jmService  OBJECT IDENTIFIER ::= { jobmonMIBObjects 7 }

jmServiceTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmServiceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmServiceTable consists of basic service state and status
        information for each service which offers one or more job
        services on this managed system.

        An entry SHALL exist in this table for each service, no matter
        what the state of that service.  A service MAY support multiple
        configured job sets and configured devices.

        See:    'jmServiceJobSetsConfigured' and
                'jmServiceDevicesConfigured' bit-arrays in this MIB."
    ::= { jmService 1 }


jmServiceEntry  OBJECT-TYPE
    SYNTAX      JmServiceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Basic service state and status information."
    INDEX  { jmServiceIndex }
    ::= { jmServiceTable 1 }


JmServiceEntry ::= SEQUENCE {
    jmServiceIndex                      Integer32 (1..2147483647),
    jmServiceName                       JmUTF8StringTC (SIZE (0..63)),
    jmServiceURI                        JmUTF8StringTC (SIZE (0..63)),
    jmServiceJobServiceTypes            JmJobServiceTypesTC,
    jmServiceJobSetsConfigured          OCTET STRING (SIZE (0..255)),
    jmServiceDevicesConfigured          OCTET STRING (SIZE (0..255)),
    jmServiceState                      JmServiceStateTC,
    jmServiceStateReasons               SnmpAdminString (SIZE (0..255))
}

jmServiceIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The unique identifier for this service on this managed system.

        See:    'jmServiceEventServiceIndex' object in this MIB."
    ::= { jmServiceEntry 1 }

jmServiceName OBJECT-TYPE
    SYNTAX      JmUTF8StringTC (SIZE(0..63))    -- 127 in IPP
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The human readable name of this managed service.

        See:    'deviceNameRequested' job attribute in this MIB;
                'physicalDevice' job attribute in this MIB;
                Section 4.4.4 'printer-name' in [IPP-MOD]."
    DEFVAL      { ''H }                 -- no service name
    ::= { jmServiceEntry 2 }

jmServiceURI OBJECT-TYPE
    SYNTAX      JmUTF8StringTC (SIZE(0..63))    -- 1023 in IPP
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A URI for this managed service (valid for job services).

        See:    Section 4.3.3 'job-printer-uri' in [IPP-MOD];
                Section 4.4.1 'printer-uri-supported' in [IPP-MOD]."
    DEFVAL      { ''H }                 -- no service URI
    ::= { jmServiceEntry 3 }

jmServiceJobServiceTypes OBJECT-TYPE
    SYNTAX      JmJobServiceTypesTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The types of job services supported by this managed service.

        See:    'JmJobServiceTypesTC' textual convention in this MIB;
                'jobServiceTypes' job attribute in this MIB."
    DEFVAL      { 0 }                   -- no job services
    ::= { jmServiceEntry 4 }

jmServiceJobSetsConfigured OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A bit-array that specifies the job sets configured for this
        service, where each bit '2**n' is set if 'jmGeneralJobSetIndex'
        is a configured job set.   Uses network byte order (big-endian)
        rules - the high-order bit of the first octet corresponds to
        'jmGeneralJobSetIndex' of '0' (reserved) - the low-order bit of
        the first octet corresponds to 'jmGeneralJobSetIndex' of '7'.
        Supports values of 'jmGeneralJobSetIndex' from '1' to '2039'.

        Compare to the BITS pseudotype defined in IETF SMIv2 (RFC 2578)
        which has the same bit ordering rules (big-endian).

        See:    'queueNameRequested' job attribute in this MIB;
                'jmGeneralJobSetIndex' table index in this MIB."
    DEFVAL      { ''H }                 -- no job sets configured
    ::= { jmServiceEntry 5 }

jmServiceDevicesConfigured OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A bit-array that specifies the devices configured for this
        service, where each bit '2**n' is set if 'hrDeviceIndex'
        is a configured device.   Uses network byte order (big-endian)
        rules - the high-order bit of the first octet corresponds to
        'hrDeviceIndex' of '0' (reserved) - the low-order bit of
        the first octet corresponds to 'hrDeviceIndex' of '7'.
        Supports values of 'hrDeviceIndex' from '1' to '2039'.

        Compare to the BITS pseudotype defined in IETF SMIv2 (RFC 2578)
        which has the same bit ordering rules (big-endian).

        See:    'physicalDevice' job attribute in this MIB;
                'hrDeviceIndex' in IETF Host MIB (RFC 2790)."
    DEFVAL      { ''H }                 -- no devices configured
    ::= { jmServiceEntry 6 }

jmServiceState OBJECT-TYPE
    SYNTAX      JmServiceStateTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current state of this managed service.

        See:    'jmServiceEventServiceState' object in this MIB;
                'jmJobState' object in this MIB;
                Section 4.4.11 'printer-state' in [IPP-MOD]."
    DEFVAL      { unknown }             -- unknown service state
    ::= { jmServiceEntry 7 }

jmServiceStateReasons OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255)) -- multi-valued in IPP
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The service state reasons associated with this service state
        (as a comma-separated list) or the empty string (if none).

        See:    'jmServiceEventServiceStateReasons' object in this MIB;
                'jmJobStateReasons1' object in this MIB;
                Section 4.4.12 'printer-state-reasons' in [IPP-MOD]."
    DEFVAL      { ''H }                 -- no service state reasons
    ::= { jmServiceEntry 8 }


--  The Service Event Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which show Service events via SNMP.

--  The jmServiceEventGroup consists entirely of the jmServiceEventTable

jmServiceEvent  OBJECT IDENTIFIER ::= { jobmonMIBObjects 8 }

jmServiceEventTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmServiceEventEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmServiceEventTable contains service event entries for the
        services which offer job services on this managed system."
    ::= { jmServiceEvent 1 }


jmServiceEventEntry  OBJECT-TYPE
    SYNTAX      JmServiceEventEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Basic service event information."
    INDEX  { jmServiceEventIndex }
    ::= { jmServiceEventTable 1 }
```

```


JmServiceEventEntry ::= SEQUENCE {
    jmServiceEventIndex                 Integer32 (1..2147483647),
    jmServiceEventNotifyEvent           SnmpAdminString (SIZE (0..63)),
    jmServiceEventNotifyTime            TimeTicks,
    jmServiceEventServiceIndex          Integer32 (1..2147483647),
    jmServiceEventServiceState          JmServiceStateTC,
    jmServiceEventServiceStateReasons   SnmpAdminString (SIZE (0..255))
}

jmServiceEventIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The unique identifier for this event on this managed system."
    ::= { jmServiceEventEntry 1 }

jmServiceEventNotifyEvent OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..63))   -- 255 in [IPP-NOT]
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The event type that created this row in 'jmServiceEventTable'.

        Standard Printer event types defined in [IPP-NOT] are:
        - 'printer-state-changed'
          - 'printer-restarted'
          - 'printer-shutdown'
        - 'printer-config-changed'
          - 'printer-media-changed'
          - 'printer-finishings-changed'
        - 'printer-queue-order-changed'
        - 'printer-full'
        - 'printer-no-longer-full'
        - 'printer-almost-idle'
        - 'printer-not-almost-idle'

        Standard Service event types generalized from [IPP-NOT] are:
        - 'service-state-changed'
          - 'service-restarted'
          - 'service-shutdown'
        - 'service-config-changed'
          - 'service-media-changed'
          - 'service-finishings-changed'
        - 'service-queue-order-changed'
        - 'service-full'
        - 'service-no-longer-full'
        - 'service-almost-idle'
        - 'service-not-almost-idle'

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

        Conformance:    This subscribed event type SHALL be valid
        and reported in ALL Job Monitoring MIB notifications.
        See:    Section 8.1 'notify-subscribed-event' in [IPP-NOT]."
    DEFVAL      { ''H }                 -- no notify subscribed event
    ::= { jmServiceEventEntry 2 }

jmServiceEventNotifyTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The time of this service event.

        Usage:  Conforming management agents, which MUST implement the
        IETF MIB-II (RFC 1213), SHALL set 'jmServiceEventNotifyTime' to
        'sysUpTime' when a service event row is created.

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

jmServiceEventServiceIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The unique identifier for this service on this managed system.

        See:    'jmServiceIndex' object in this MIB."
    ::= { jmServiceEventEntry 4 }

jmServiceEventServiceState OBJECT-TYPE
    SYNTAX      JmServiceStateTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The state of this managed service at the time of this event.

        See:    'jmServiceState' object in this MIB;
                'jmJobState' object in this MIB;
                Section 4.4.11 'printer-state' in [IPP-MOD]."
    DEFVAL      { unknown }             -- unknown service state
    ::= { jmServiceEventEntry 5 }

jmServiceEventServiceStateReasons OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..255)) -- multi-valued in IPP
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The service state reasons associated with this service state
        (as a comma-separated list) or the empty string (if none).

        See:    'jmServiceStateReasons' object in this MIB;
                'jmJobStateReasons1' object in this MIB;
                Section 4.4.12 'printer-state-reasons' in [IPP-MOD]."
    DEFVAL      { ''H }                 -- no service state reasons
    ::= { jmServiceEventEntry 6 }
```

```
--  The Job Event Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which show Job events via SNMP.

--  The jmJobEventGroup consists entirely of the jmJobEventTable

jmJobEvent  OBJECT IDENTIFIER ::= { jobmonMIBObjects 9 }

jmJobEventTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF JmJobEventEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The jmJobEventTable contains job event entries for the
        jobs present on this managed system."
    ::= { jmJobEvent 1 }


jmJobEventEntry  OBJECT-TYPE
    SYNTAX      JmJobEventEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Basic job event information."
    INDEX  { jmJobEventIndex }
    ::= { jmJobEventTable 1 }


JmJobEventEntry ::= SEQUENCE {
    jmJobEventIndex                     Integer32 (1..2147483647),
    jmJobEventNotifyEvent               SnmpAdminString (SIZE (0..63)),
    jmJobEventNotifyTime                TimeTicks,
    jmJobEventJobSetIndex               Integer32 (1..32767),
    jmJobEventJobIndex                  Integer32 (1..2147483647),
    jmJobEventJobState                  JmJobStateTC,
    jmJobEventJobStateReasons           OCTET STRING (SIZE (4..16))
}

jmJobEventIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The unique identifier for this event on this managed system."
    ::= { jmJobEventEntry 1 }

jmJobEventNotifyEvent OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..63))   -- 255 in [IPP-NOT]
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The event type that created this row in 'jmJobEventTable'.

        Standard Job event types defined in [IPP-NOT] are:
```

```
        - 'job-state-changed'
          - 'job-created'
          - 'job-completed'
          - 'job-purged'
        - 'job-config-changed'
        - 'job-progress'

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

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

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

jmJobEventNotifyTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The time of this job event.

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

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

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

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

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

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

jmJobEventJobState OBJECT-TYPE
    SYNTAX      JmJobStateTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The state of this managed job at the time of this event.

        See:    'jmJobState' in this MIB;
                Section 4.3.7 'job-state' in [IPP-MOD]."
    DEFVAL      { unknown }             -- unknown job state
    ::= { jmJobEventEntry 6 }

jmJobEventJobStateReasons OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (4..16)) -- multi-valued in IPP
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The job state reasons associated with this job state
        represented as one to four concatenated 32-bit integers
        in network byte order (big-endian).

        Usage:  Conforming management stations SHALL always report the
        value of the object 'jmJobStateReasons1' in the first four
        octets of 'jmJobEventJobStateReasons' and SHOULD report values
        of the attributes 'jobStateReasons2', 'jobStateReasons3', and
        'jobStateReasons4' in subsequent octets.

        See:    'jmJobStateReasons1' in this MIB;
                Section 4.3.8 'job-state-reasons' in [IPP-MOD]."
    DEFVAL      { '00000000'H }         -- no job state reasons
    ::= { jmJobEventEntry 7 }


--  The Job Progress Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which send Job progress traps via SNMP.

--  The jmProgressGroup consists entirely of leaf objects for traps

jmProgress  OBJECT IDENTIFIER ::= { jobmonMIBObjects 10 }

jmProgressJobCopiesRequested OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of copies of this job requested by the client.

        See:    'jobCopiesRequested' attribute in this MIB."
    DEFVAL      { -2 }                  -- unknown job copies
    ::= { jmProgress 1 }

jmProgressJobCollationType OBJECT-TYPE
    SYNTAX      JmJobCollationTypeTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of copies of this job requested by the client.

        See:    'jobCollationType' attribute in this MIB;
                'job-collation-type' in [IPP-PROG];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    DEFVAL      { unknown }             -- unknown job collation type
    ::= { jmProgress 2 }

jmProgressMediaSheetsCompleted OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of media sheets completed for this job.

        See:    'sheetsCompleted' attribute in this MIB;
                Section 4.3.18.3 'job-media-sheets-completed'
                in [IPP-MOD];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    DEFVAL      { -2 }                  -- unknown job progress
    ::= { jmProgress 3 }

jmProgressSheetCompletedCopyNum OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of the job copy currently being stacked for the
        current document or zero (if none) or '-2' (unknown).

        See:    'sheetCompletedCopyNumber' attribute in this MIB;
                'sheet-completed-copy-number' in [IPP-PROG];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    DEFVAL      { -2 }                  -- unknown sheet complete copy
    ::= { jmProgress 4 }

jmProgressSheetCompletedDocNum OBJECT-TYPE
    SYNTAX      Integer32 (-2..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of the job document currently being stacked for the
        current job or zero (if none) or '-2' (unknown).

        See:    'sheetCompletedDocumentNumber' attribute in this MIB;
                'sheet-completed-document-number' in [IPP-PROG];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    DEFVAL      { -2 }                  -- unknown sheet complete doc
    ::= { jmProgress 5 }

-- Notifications (Traps and Informs)

jobmonMIBNotifications  OBJECT IDENTIFIER  ::= { jobmonMIB 2 }

--  Service Basic Event Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which send this Service traps via SNMP.
--  The jmServiceBasicEventGroup consists entirely of the
--  jmServiceBasicV2Event notification.

jmServiceBasicV1Enterprise OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "The value of the enterprise-specific OID in an SNMPv1 trap
        for a Service basic event sent by this managed system."
    ::= { jobmonMIBNotifications 1 }

jmServiceBasicV2EventPrefix
    OBJECT IDENTIFIER ::= { jmServiceBasicV1Enterprise 0 }

jmServiceBasicV2Event NOTIFICATION-TYPE
    OBJECTS {
        jmServiceEventNotifyEvent,
        jmServiceState,
        jmServiceStateReasons
    }
    STATUS      current
    DESCRIPTION
        "This SMIv2 trap corresponds to an IPP Printer basic event.

        The value of 'jmServiceIndex' for
        use with 'jmServiceTable' for this Service is conveyed in the
        instance qualifier (OID suffix) of 'jmServiceState'.

        This trap is sent when requested by a prior subscription.
        The subscribed event type is in 'jmServiceEventNotifyEvent'.

        Standard Printer event types defined in [IPP-NOT] are:
        - 'printer-state-changed'
          - 'printer-restarted'
          - 'printer-shutdown'
        - 'printer-config-changed'
          - 'printer-media-changed'
          - 'printer-finishings-changed'
        - 'printer-queue-order-changed'
        - 'printer-full'
        - 'printer-no-longer-full'
        - 'printer-almost-idle'
        - 'printer-not-almost-idle'

        Standard Service event types generalized from [IPP-NOT] are:
        - 'service-state-changed'
          - 'service-restarted'
          - 'service-shutdown'
        - 'service-config-changed'
          - 'service-media-changed'
          - 'service-finishings-changed'
        - 'service-queue-order-changed'
        - 'service-full'
        - 'service-no-longer-full'
        - 'service-almost-idle'
        - 'service-not-almost-idle'
        - and (optionally) vendor extension event types
        Additional variable-bindings SHOULD be appended to this trap:
        - Systems with the Host Resources MIB (RFC 2790) SHOULD add
          'hrSystemDate'
          (compare to IPP 'printer-current-time')

        Additional variable-bindings MAY be appended to this trap
        for all printer-specific events:
        - Systems with the Host Resources MIB (RFC 2790) MAY add
          'hrDeviceStatus'
          (compare to IPP 'printer-state')
          'hrPrinterStatus'
          (compare to IPP 'printer-state') and
          'hrPrinterDetectedErrorState'
          (compare to IPP 'printer-state-reasons')

        Systems MAY add other variable-bindings from any MIB.

        See:    Section 5.3.2 'notify-events' in [IPP-NOT];
                Section 8.1 'notify-subscribed-event' in [IPP-NOT];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    ::= { jmServiceBasicV2EventPrefix 1 }

--  Job Basic Event Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which send this Job traps via SNMP.

--  The jmJobBasicEventGroup consists entirely of the
--  jmJobBasicV2Event notification.

jmJobBasicV1Enterprise OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "The value of the enterprise-specific OID in an SNMPv1 trap
        for a Job basic event sent by this managed system."
    ::= { jobmonMIBNotifications 2 }

jmJobBasicV2EventPrefix
    OBJECT IDENTIFIER ::= { jmJobBasicV1Enterprise 0 }

jmJobBasicV2Event NOTIFICATION-TYPE
    OBJECTS {
        jmJobEventNotifyEvent,
        jmJobState,
        jmJobEventJobStateReasons
    }
    STATUS      current
    DESCRIPTION
        "This SMIv2 trap corresponds to an IPP Job basic event.

        The values of 'jmGeneralJobSetIndex' and 'jmJobIndex' for
        use with 'jmJobTable' for this Job are conveyed in the
        instance qualifier (OID suffix) of 'jmJobState'.

        This trap is sent when requested by a prior subscription.
        The subscribed event type is in 'jmJobEventNotifyEvent'.

        Event types reported via 'jmJobBasicV2Event' include:
        - 'job-state-changed'
          - 'job-created'
          - 'job-completed'
          - 'job-purged'
        - 'job-config-changed'
        - and (optionally) vendor extension event types

        Additional variable-bindings SHOULD be appended to this trap:
        - Systems with the Host Resources MIB (RFC 2790) SHOULD add
          'hrSystemDate'
          (compare to IPP 'printer-current-time')

        Systems MAY add other variable-bindings from any MIB.

        See:    Section 5.3.2 'notify-events' in [IPP-NOT];
                Section 8.1 'notify-subscribed-event' in [IPP-NOT];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    ::= { jmJobBasicV2EventPrefix 1 }

--  Job Completed Event Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which send this Job traps via SNMP.

jmJobCompletedV1Enterprise OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "The value of the enterprise-specific OID in an SNMPv1 trap
        for a Job completed event sent by this managed system."
    ::= { jobmonMIBNotifications 3 }

jmJobCompletedV2EventPrefix
    OBJECT IDENTIFIER ::= { jmJobCompletedV1Enterprise 0 }

jmJobCompletedV2Event NOTIFICATION-TYPE
    OBJECTS {
        jmJobEventNotifyEvent,
        jmJobState,
        jmJobEventJobStateReasons,
        jmJobKOctetsProcessed,
        jmJobImpressionsCompleted
    }
    STATUS      current
    DESCRIPTION
        "This SMIv2 trap corresponds to an IPP 'job-completed' event.

        The values of 'jmGeneralJobSetIndex' and 'jmJobIndex' for
        use with 'jmJobTable' for this Job are conveyed in the
        instance qualifier (OID suffix) of 'jmJobState'.

        This trap is sent when requested by a prior subscription.
        The subscribed event type is in 'jmJobEventNotifyEvent'.

        Event types reported via 'jmJobCompletedV2Event' include:
        - 'job-completed'
        - 'job-purged'
        - and (optionally) vendor extension event types

        Additional variable-bindings SHOULD be appended to this trap:
        - Systems with the Host Resources MIB (RFC 2790) SHOULD add
          'hrSystemDate'
          (compare to IPP 'printer-current-time')

        Systems MAY add other variable-bindings from any MIB.

        See:    Section 5.3.2 'notify-events' in [IPP-NOT];
                Section 8.1 'notify-subscribed-event' in [IPP-NOT];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    ::= { jmJobCompletedV2EventPrefix 1 }

--  Job Progress Event Group (CONDITIONALLY MANDATORY)
--
--  Implementation of this group is conditionally mandatory;
--  mandatory for systems which send this Job traps via SNMP.

jmJobProgressV1Enterprise OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "The value of the enterprise-specific OID in an SNMPv1 trap
        for a Job progress event sent by this managed system."
    ::= { jobmonMIBNotifications 4 }

jmJobProgressV2EventPrefix
    OBJECT IDENTIFIER ::= { jmJobProgressV1Enterprise 0 }

jmJobProgressV2Event NOTIFICATION-TYPE
    OBJECTS {
        jmJobKOctetsPerCopyRequested,
        jmJobKOctetsProcessed,
        jmJobImpressionsPerCopyRequested,
        jmJobImpressionsCompleted,
        jmProgressJobCopiesRequested,
        jmProgressJobCollationType,
        jmProgressMediaSheetsCompleted,
        jmProgressSheetCompletedCopyNum,
        jmProgressSheetCompletedDocNum
    }
    STATUS      current
    DESCRIPTION
        "This SMIv2 trap corresponds to an IPP 'job-progress' event.

        The values of 'jmGeneralJobSetIndex' and 'jmJobIndex' for
        use with 'jmJobTable' for this Job are conveyed in the
        instance qualifier (OID suffix) of 'jmJobKOctetsProcessed'.

        This trap is sent when requested by a prior subscription.
        The event type is 'job-progress'.

        Additional variable-bindings SHOULD be appended to this trap:
        - Systems with the Host Resources MIB (RFC 2790) SHOULD add
          'hrSystemDate'
          (compare to IPP 'printer-current-time')

        Systems MAY add other variable-bindings from any MIB.

        See:    Section 5.3.2 'notify-events' in [IPP-NOT];
                Section 8.1 'notify-subscribed-event' in [IPP-NOT];
                Section 9 'Event Notification Content' in [IPP-NOT]."
    ::= { jmJobProgressV2EventPrefix 1 }

-- Conformance Information

jmMIBConformance  OBJECT IDENTIFIER ::= { jobmonMIB 3 }

-- compliance statements
jmMIBCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
        "The compliance statement for agents that implement the
        job monitoring MIB."
    MODULE -- this module
    MANDATORY-GROUPS {
        jmGeneralGroup, jmJobIDGroup, jmJobGroup, jmAttributeGroup }


    GROUP   jmServiceGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which show Service states via SNMP."

    GROUP   jmServiceEventGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which show Service events via SNMP."

    GROUP   jmJobEventGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which show Job events via SNMP."

    GROUP   jmProgressGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which send Job progress traps via SNMP."

    GROUP   jmServiceBasicEventGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which send this Service traps via SNMP."

    GROUP   jmJobBasicEventGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which send this Job traps via SNMP."

    GROUP   jmJobCompletedEventGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which send this Job traps via SNMP."

    GROUP   jmJobProgressEventGroup
    DESCRIPTION
        "Implementation of this group is conditionally mandatory;
        mandatory for systems which send this Job traps via SNMP."

    OBJECT  jmGeneralJobSetName
    SYNTAX  JmUTF8StringTC (SIZE(0..8))
    DESCRIPTION
        "Only 8 octets maximum string length NEED be supported by the
        agent."

    OBJECT  jmJobOwner
    SYNTAX  JmJobStringTC (SIZE(0..16))
    DESCRIPTION
        "Only 16 octets maximum string length NEED be supported by the
        agent."

    ::= { jmMIBConformance 1 }

jmMIBGroups  OBJECT IDENTIFIER ::= { jmMIBConformance 2 }

jmGeneralGroup OBJECT-GROUP
    OBJECTS {
        jmGeneralNumberOfActiveJobs,   jmGeneralOldestActiveJobIndex,
        jmGeneralNewestActiveJobIndex, jmGeneralJobPersistence,
        jmGeneralAttributePersistence, jmGeneralJobSetName}
    STATUS  current
    DESCRIPTION
        "The general object group."
    ::= { jmMIBGroups 1 }

jmJobIDGroup OBJECT-GROUP
    OBJECTS {
        jmJobIDJobSetIndex, jmJobIDJobIndex }
    STATUS  current
    DESCRIPTION
        "The job ID object group."
    ::= { jmMIBGroups 2 }

jmJobGroup OBJECT-GROUP
    OBJECTS {
        jmJobState, jmJobStateReasons1, jmNumberOfInterveningJobs,
        jmJobKOctetsPerCopyRequested, jmJobKOctetsProcessed,
        jmJobImpressionsPerCopyRequested, jmJobImpressionsCompleted,
        jmJobOwner }
    STATUS  current


    DESCRIPTION
        "The job object group."
    ::= { jmMIBGroups 3 }

jmAttributeGroup OBJECT-GROUP
    OBJECTS {
        jmAttributeValueAsInteger, jmAttributeValueAsOctets }
    STATUS  current
    DESCRIPTION
```

```
        "The job attribute object group."
    ::= { jmMIBGroups 4 }

jmServiceGroup OBJECT-GROUP
    OBJECTS {
        jmServiceName,
        jmServiceURI,
        jmServiceJobServiceTypes,
        jmServiceJobSetsConfigured,
        jmServiceDevicesConfigured,
        jmServiceState,
        jmServiceStateReasons
    }
    STATUS  current
    DESCRIPTION
        "The service object group."
    ::= { jmMIBGroups 7 }

jmServiceEventGroup OBJECT-GROUP
    OBJECTS {
        jmServiceEventNotifyEvent,
        jmServiceEventNotifyTime,
        jmServiceEventServiceIndex,
        jmServiceEventServiceState,
        jmServiceEventServiceStateReasons
    }
    STATUS  current
    DESCRIPTION
        "The service event object group."
    ::= { jmMIBGroups 8 }

jmJobEventGroup OBJECT-GROUP
    OBJECTS {
        jmJobEventNotifyEvent,
        jmJobEventNotifyTime,
        jmJobEventJobSetIndex,
        jmJobEventJobIndex,
        jmJobEventJobState,
        jmJobEventJobStateReasons
    }
    STATUS  current
    DESCRIPTION
        "The job event object group."
    ::= { jmMIBGroups 9 }

jmProgressGroup OBJECT-GROUP
    OBJECTS {
        jmProgressJobCopiesRequested,
        jmProgressJobCollationType,
        jmProgressMediaSheetsCompleted,
        jmProgressSheetCompletedCopyNum,
        jmProgressSheetCompletedDocNum
    }
    STATUS  current
    DESCRIPTION
        "The job progress object group."
    ::= { jmMIBGroups 10 }
```

```
jmMIBNotifyGroups  OBJECT IDENTIFIER ::= { jmMIBConformance 3 }

jmServiceBasicEventGroup NOTIFICATION-GROUP
    NOTIFICATIONS { jmServiceBasicV2Event }
    STATUS  current
    DESCRIPTION
        "The service basic event notification group."
    ::= { jmMIBNotifyGroups 1 }

jmJobBasicEventGroup NOTIFICATION-GROUP
    NOTIFICATIONS { jmJobBasicV2Event }
    STATUS  current
    DESCRIPTION
        "The job basic event notification group."
    ::= { jmMIBNotifyGroups 2 }

jmJobCompletedEventGroup NOTIFICATION-GROUP
    NOTIFICATIONS { jmJobCompletedV2Event }
    STATUS  current
    DESCRIPTION
        "The job completed event notification group."
    ::= { jmMIBNotifyGroups 3 }

jmJobProgressEventGroup NOTIFICATION-GROUP
    NOTIFICATIONS { jmJobProgressV2Event }
    STATUS  current
    DESCRIPTION
        "The job progress event notification group."
    ::= { jmMIBNotifyGroups 4 }

END
```