Network Working Group
Request for Comments: 1628
Category: Standards Track

J. Case, Editor
SNMP Research, Incorporated

May 1994

UPS Management Information Base

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing uninterruptible power supply (UPS) systems.

2. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- RFC 1442 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- STD 17, RFC 1213 defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- RFC 1445 which defines the administrative and other architectural aspects of the framework.
- RFC 1448 which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

3. Overview

This document defines the managed objects for Uninterruptible Power Supplies which are to be manageable via the Simple Network Management Protocol (SNMP).

4. Definitions

```
   UPS-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       OBJECT-IDENTITY, Counter32, Gauge32, Integer32
           FROM SNMPv2-SMI
       DisplayString, TimeStamp, TimeInterval, TestAndIncr,
         AutonomousType
           FROM SNMPv2-TC
       MODULE-COMPLIANCE, OBJECT-GROUP
           FROM SNMPv2-CONF;


   upsMIB MODULE-IDENTITY
       LAST-UPDATED "9402230000Z"
       ORGANIZATION "IETF UPS MIB Working Group"
       CONTACT-INFO
              "        Jeffrey D. Case

                Postal: SNMP Research, Incorporated
                        3001 Kimberlin Heights Road
                        Knoxville, TN  37920
                        US

                   Tel: +1 615 573 1434
                   Fax: +1 615 573 9197

                E-mail: case@snmp.com"
       DESCRIPTION
              "The MIB module to describe Uninterruptible Power
              Supplies."
       ::= { mib-2 33 }

   PositiveInteger ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS       current
       DESCRIPTION
              "This data type is a non-zero and non-negative value."
       SYNTAX       INTEGER (1..2147483647)

   NonNegativeInteger ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS       current
       DESCRIPTION
              "This data type is a non-negative value."
       SYNTAX       INTEGER (0..2147483647)
```

```
   upsObjects            OBJECT IDENTIFIER ::= { upsMIB 1 }


   --
   -- The Device Identification group.
   --      All objects in this group except for upsIdentName and
   --      upsIdentAttachedDevices are set at device initialization
   --      and remain static.
   --

   upsIdent              OBJECT IDENTIFIER ::= { upsObjects 1 }

   upsIdentManufacturer OBJECT-TYPE
       SYNTAX     DisplayString (SIZE (0..31))
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The name of the UPS manufacturer."
       ::= { upsIdent 1 }

   upsIdentModel OBJECT-TYPE
       SYNTAX     DisplayString (SIZE (0..63))
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The UPS Model designation."
       ::= { upsIdent 2 }

   upsIdentUPSSoftwareVersion OBJECT-TYPE
       SYNTAX     DisplayString (SIZE (0..63))
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The UPS firmware/software version(s).  This variable
               may or may not have the same value as
               upsIdentAgentSoftwareVersion in some implementations."
       ::= { upsIdent 3 }

   upsIdentAgentSoftwareVersion OBJECT-TYPE
       SYNTAX     DisplayString (SIZE (0..63))
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The UPS agent software version.  This variable may or
               may not have the same value as
               upsIdentUPSSoftwareVersion in some implementations."
       ::= { upsIdent 4 }
```

```
   upsIdentName OBJECT-TYPE
       SYNTAX     DisplayString (SIZE(0..63))
       MAX-ACCESS read-write
       STATUS     current
       DESCRIPTION
               "A string identifying the UPS.  This object should be
               set by the administrator."
       ::= { upsIdent 5 }

   upsIdentAttachedDevices OBJECT-TYPE
       SYNTAX     DisplayString (SIZE(0..63))
       MAX-ACCESS read-write
       STATUS     current
       DESCRIPTION
               "A string identifying the devices attached to the
               output(s) of the UPS.  This object should be set by
               the administrator."
       ::= { upsIdent 6 }


   --
   -- Battery Group
   --

   upsBattery            OBJECT IDENTIFIER ::= { upsObjects 2 }

   upsBatteryStatus OBJECT-TYPE
       SYNTAX     INTEGER {
           unknown(1),
           batteryNormal(2),
           batteryLow(3),
           batteryDepleted(4)
       }
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The indication of the capacity remaining in the UPS
               system's batteries.   A value of batteryNormal
               indicates that the remaining run-time is greater than
               upsConfigLowBattTime.  A value of batteryLow indicates
               that the remaining battery run-time is less than or
               equal to upsConfigLowBattTime.  A value of
               batteryDepleted indicates that the UPS will be unable
               to sustain the present load when and if the utility
               power is lost (including the possibility that the
               utility power is currently absent and the UPS is
               unable to sustain the output)."
       ::= { upsBattery 1 }
```

```
   upsSecondsOnBattery OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "seconds"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "If the unit is on battery power, the elapsed time
               since the UPS last switched to battery power, or the
               time since the network management subsystem was last
               restarted, whichever is less.  Zero shall be returned
               if the unit is not on battery power."
       ::= { upsBattery 2 }

   upsEstimatedMinutesRemaining OBJECT-TYPE
       SYNTAX     PositiveInteger
       UNITS      "minutes"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "An estimate of the time to battery charge depletion
               under the present load conditions if the utility power
               is off and remains off, or if it were to be lost and
               remain off."
       ::= { upsBattery 3 }

   upsEstimatedChargeRemaining OBJECT-TYPE
       SYNTAX     INTEGER (0..100)
       UNITS      "percent"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "An estimate of the battery charge remaining expressed
               as a percent of full charge."
       ::= { upsBattery 4 }

   upsBatteryVoltage OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "0.1 Volt DC"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The magnitude of the present battery voltage."
       ::= { upsBattery 5 }

   upsBatteryCurrent OBJECT-TYPE
       SYNTAX     Integer32
       UNITS      "0.1 Amp DC"
       MAX-ACCESS read-only
```

```
        STATUS     current
        DESCRIPTION
                "The present battery current."
        ::= { upsBattery 6 }

    upsBatteryTemperature OBJECT-TYPE
        SYNTAX     Integer32
        UNITS      "degrees Centigrade"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The ambient temperature at or near the UPS Battery
                casing."
        ::= { upsBattery 7 }


    --
    -- Input Group
    --

    upsInput              OBJECT IDENTIFIER ::= { upsObjects 3 }

    upsInputLineBads OBJECT-TYPE
        SYNTAX     Counter32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "A count of the number of times the input entered an
                out-of-tolerance condition as defined by the
                manufacturer.  This count is incremented by one each
                time the input transitions from zero out-of-tolerance
                lines to one or more input lines out-of-tolerance."
        ::= { upsInput 1 }


    upsInputNumLines OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The number of input lines utilized in this device.
                This variable indicates the number of rows in the
                input table."
        ::= { upsInput 2 }

    upsInputTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF UpsInputEntry
        MAX-ACCESS not-accessible
```

```
        STATUS     current
        DESCRIPTION
                "A list of input table entries.  The number of entries
                is given by the value of upsInputNumLines."
        ::= { upsInput 3 }

    upsInputEntry OBJECT-TYPE
        SYNTAX     UpsInputEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "An entry containing information applicable to a
                particular input line."
        INDEX { upsInputLineIndex }
        ::= { upsInputTable 1 }

    UpsInputEntry ::= SEQUENCE {
        upsInputLineIndex   PositiveInteger,
        upsInputFrequency   NonNegativeInteger,
        upsInputVoltage     NonNegativeInteger,
        upsInputCurrent     NonNegativeInteger,
        upsInputTruePower   NonNegativeInteger
    }

    upsInputLineIndex OBJECT-TYPE
        SYNTAX     PositiveInteger
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "The input line identifier."
        ::= { upsInputEntry 1 }

    upsInputFrequency OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "0.1 Hertz"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present input frequency."
        ::= { upsInputEntry 2 }

    upsInputVoltage OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "RMS Volts"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The magnitude of the present input voltage."
```

```
        ::= { upsInputEntry 3 }

    upsInputCurrent OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "0.1 RMS Amp"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The magnitude of the present input current."
        ::= { upsInputEntry 4 }

    upsInputTruePower OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "Watts"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The magnitude of the present input true power."
        ::= { upsInputEntry 5 }


    --
    -- The Output group.
    --

    upsOutput             OBJECT IDENTIFIER ::= { upsObjects 4 }

    upsOutputSource OBJECT-TYPE
        SYNTAX     INTEGER {
            other(1),
            none(2),
            normal(3),
            bypass(4),
            battery(5),
            booster(6),
            reducer(7)
        }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present source of output power.  The enumeration
                none(2) indicates that there is no source of output
                power (and therefore no output power), for example,
                the system has opened the output breaker."
        ::= { upsOutput 1 }

    upsOutputFrequency OBJECT-TYPE
        SYNTAX     NonNegativeInteger
```

```
        UNITS       "0.1 Hertz"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present output frequency."
        ::= { upsOutput 2 }

    upsOutputNumLines OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The number of output lines utilized in this device.
                This variable indicates the number of rows in the
                output table."
        ::= { upsOutput 3 }

    upsOutputTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF UpsOutputEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "A list of output table entries.  The number of
                entries is given by the value of upsOutputNumLines."
        ::= { upsOutput 4 }

    upsOutputEntry OBJECT-TYPE
        SYNTAX     UpsOutputEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "An entry containing information applicable to a
                particular output line."
        INDEX { upsOutputLineIndex }
        ::= { upsOutputTable 1 }

    UpsOutputEntry ::= SEQUENCE {
        upsOutputLineIndex   PositiveInteger,
        upsOutputVoltage     NonNegativeInteger,
        upsOutputCurrent     NonNegativeInteger,
        upsOutputPower       NonNegativeInteger,
        upsOutputPercentLoad INTEGER
    }

    upsOutputLineIndex OBJECT-TYPE
        SYNTAX     PositiveInteger
        MAX-ACCESS not-accessible
        STATUS     current
```

```
        DESCRIPTION
                "The output line identifier."
        ::= { upsOutputEntry 1 }

    upsOutputVoltage OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "RMS Volts"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present output voltage."
        ::= { upsOutputEntry 2 }

    upsOutputCurrent OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "0.1 RMS Amp"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present output current."
        ::= { upsOutputEntry 3 }

    upsOutputPower OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "Watts"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present output true power."
        ::= { upsOutputEntry 4 }

    upsOutputPercentLoad OBJECT-TYPE
        SYNTAX     INTEGER (0..200)
        UNITS      "percent"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The percentage of the UPS power capacity presently
                being used on this output line, i.e., the greater of
                the percent load of true power capacity and the
                percent load of VA."
        ::= { upsOutputEntry 5 }
```

```
   --
   -- The Bypass group.
   --

   upsBypass             OBJECT IDENTIFIER ::= { upsObjects 5 }

   upsBypassFrequency OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "0.1 Hertz"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The present bypass frequency."
       ::= { upsBypass 1 }

   upsBypassNumLines OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The number of bypass lines utilized in this device.
               This entry indicates the number of rows in the bypass
               table."
       ::= { upsBypass 2 }

   upsBypassTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF UpsBypassEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
               "A list of bypass table entries.  The number of
               entries is given by the value of upsBypassNumLines."
       ::= { upsBypass 3 }

   upsBypassEntry OBJECT-TYPE
       SYNTAX     UpsBypassEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
               "An entry containing information applicable to a
               particular bypass input."
       INDEX { upsBypassLineIndex }
       ::= { upsBypassTable 1 }

   UpsBypassEntry ::= SEQUENCE {
       upsBypassLineIndex  PositiveInteger,
       upsBypassVoltage    NonNegativeInteger,
       upsBypassCurrent    NonNegativeInteger,
```

```
        upsBypassPower      NonNegativeInteger
    }

    upsBypassLineIndex OBJECT-TYPE
        SYNTAX     PositiveInteger
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "The bypass line identifier."
        ::= { upsBypassEntry 1 }

    upsBypassVoltage OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "RMS Volts"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present bypass voltage."
        ::= { upsBypassEntry 2 }

    upsBypassCurrent OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "0.1 RMS Amp"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present bypass current."
        ::= { upsBypassEntry 3 }

    upsBypassPower OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "Watts"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The present true power conveyed by the bypass."
        ::= { upsBypassEntry 4 }


    --
    -- The Alarm group.
    --

    upsAlarm              OBJECT IDENTIFIER ::= { upsObjects 6 }

    upsAlarmsPresent OBJECT-TYPE
        SYNTAX     Gauge32
        MAX-ACCESS read-only
```

```
        STATUS     current
        DESCRIPTION
                "The present number of active alarm conditions."
        ::= { upsAlarm 1 }

    upsAlarmTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF UpsAlarmEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "A list of alarm table entries.  The table contains
                zero, one, or many rows at any moment, depending upon
                the number of alarm conditions in effect.  The table
                is initially empty at agent startup.  The agent
                creates a row in the table each time a condition is
                detected and deletes that row when that condition no
                longer pertains.  The agent creates the first row with
                upsAlarmId equal to 1, and increments the value of
                upsAlarmId each time a new row is created, wrapping to
                the first free value greater than or equal to 1 when
                the maximum value of upsAlarmId would otherwise be
                exceeded.  Consequently, after multiple operations,
                the table may become sparse, e.g., containing entries
                for rows 95, 100, 101, and 203 and the entries should
                not be assumed to be in chronological order because
                upsAlarmId might have wrapped.

                Alarms are named by an AutonomousType (OBJECT
                IDENTIFIER), upsAlarmDescr, to allow a single table to
                reflect well known alarms plus alarms defined by a
                particular implementation, i.e., as documented in the
                private enterprise MIB definition for the device.  No
                two rows will have the same value of upsAlarmDescr,
                since alarms define conditions.  In order to meet this
                requirement, care should be taken in the definition of
                alarm conditions to insure that a system cannot enter
                the same condition multiple times simultaneously.

                The number of rows in the table at any given time is
                reflected by the value of upsAlarmsPresent."
        ::= { upsAlarm 2 }

    upsAlarmEntry OBJECT-TYPE
        SYNTAX     UpsAlarmEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "An entry containing information applicable to a
```

```
                particular alarm."
        INDEX { upsAlarmId }
        ::= { upsAlarmTable 1 }

    UpsAlarmEntry ::= SEQUENCE {
        upsAlarmId          PositiveInteger,
        upsAlarmDescr       AutonomousType,
        upsAlarmTime        TimeStamp
    }

    upsAlarmId OBJECT-TYPE
        SYNTAX     PositiveInteger
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "A unique identifier for an alarm condition.  This
                value must remain constant."
        ::= { upsAlarmEntry 1 }

    upsAlarmDescr OBJECT-TYPE
        SYNTAX     AutonomousType
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "A reference to an alarm description object.  The
                object referenced should not be accessible, but rather
                be used to provide a unique description of the alarm
                condition."
        ::= { upsAlarmEntry 2 }

    upsAlarmTime OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The value of sysUpTime when the alarm condition was
                detected.  If the alarm condition was detected at the
                time of agent startup and presumably existed before
                agent startup, the value of upsAlarmTime shall equal
                0."
        ::= { upsAlarmEntry 3 }


    --
    -- Well known alarm conditions.
    --

    upsWellKnownAlarms    OBJECT IDENTIFIER ::= { upsAlarm 3 }
```

```
   upsAlarmBatteryBad OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "One or more batteries have been determined to require
               replacement."
       ::= { upsWellKnownAlarms  1 }

   upsAlarmOnBattery OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The UPS is drawing power from the batteries."
       ::= { upsWellKnownAlarms  2 }

   upsAlarmLowBattery OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The remaining battery run-time is less than or equal
               to upsConfigLowBattTime."
       ::= { upsWellKnownAlarms  3 }


   upsAlarmDepletedBattery OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The UPS will be unable to sustain the present load
               when and if the utility power is lost."
       ::= { upsWellKnownAlarms  4 }

   upsAlarmTempBad OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "A temperature is out of tolerance."
       ::= { upsWellKnownAlarms  5 }

   upsAlarmInputBad OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "An input condition is out of tolerance."
       ::= { upsWellKnownAlarms  6 }

   upsAlarmOutputBad OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "An output condition (other than OutputOverload) is
               out of tolerance."
       ::= { upsWellKnownAlarms  7 }

   upsAlarmOutputOverload OBJECT-IDENTITY
```

```
        STATUS     current
        DESCRIPTION
                "The output load exceeds the UPS output capacity."
        ::= { upsWellKnownAlarms  8 }

    upsAlarmOnBypass OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The Bypass is presently engaged on the UPS."
        ::= { upsWellKnownAlarms  9 }

    upsAlarmBypassBad OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The Bypass is out of tolerance."
        ::= { upsWellKnownAlarms 10 }

    upsAlarmOutputOffAsRequested OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The UPS has shutdown as requested, i.e., the output
                is off."
        ::= { upsWellKnownAlarms 11 }

    upsAlarmUpsOffAsRequested OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The entire UPS has shutdown as commanded."
        ::= { upsWellKnownAlarms 12 }

    upsAlarmChargerFailed OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "An uncorrected problem has been detected within the
                UPS charger subsystem."
        ::= { upsWellKnownAlarms 13 }

    upsAlarmUpsOutputOff OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The output of the UPS is in the off state."
        ::= { upsWellKnownAlarms 14 }

    upsAlarmUpsSystemOff OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The UPS system is in the off state."
        ::= { upsWellKnownAlarms 15 }
```

```
   upsAlarmFanFailure OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The failure of one or more fans in the UPS has been
               detected."
       ::= { upsWellKnownAlarms 16 }

   upsAlarmFuseFailure OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The failure of one or more fuses has been detected."
       ::= { upsWellKnownAlarms 17 }

   upsAlarmGeneralFault OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "A general fault in the UPS has been detected."
       ::= { upsWellKnownAlarms 18 }

   upsAlarmDiagnosticTestFailed OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The result of the last diagnostic test indicates a
               failure."
       ::= { upsWellKnownAlarms 19 }

   upsAlarmCommunicationsLost OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "A problem has been encountered in the communications
               between the agent and the UPS."
       ::= { upsWellKnownAlarms 20 }

   upsAlarmAwaitingPower OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The UPS output is off and the UPS is awaiting the
               return of input power."
       ::= { upsWellKnownAlarms 21 }

   upsAlarmShutdownPending OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "A upsShutdownAfterDelay countdown is underway."
       ::= { upsWellKnownAlarms 22 }

   upsAlarmShutdownImminent OBJECT-IDENTITY
       STATUS     current
```

```
        DESCRIPTION
                "The UPS will turn off power to the load in less than
                5 seconds; this may be either a timed shutdown or a
                low battery shutdown."
        ::= { upsWellKnownAlarms 23 }

    upsAlarmTestInProgress OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "A test is in progress, as initiated and indicated by
                the Test Group.  Tests initiated via other
                implementation-specific mechanisms can indicate the
                presence of the testing in the alarm table, if
                desired, via a OBJECT-IDENTITY macro in the MIB
                document specific to that implementation and are
                outside the scope of this OBJECT-IDENTITY."
        ::= { upsWellKnownAlarms 24 }


    --
    -- The Test Group
    --

    upsTest               OBJECT IDENTIFIER ::= { upsObjects 7 }

    upsTestId OBJECT-TYPE
        SYNTAX     OBJECT IDENTIFIER
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The test is named by an OBJECT IDENTIFIER which
                allows a standard mechanism for the initiation of
                tests, including the well known tests identified in
                this document as well as those introduced by a
                particular implementation, i.e., as documented in the
                private enterprise MIB definition for the device.

                Setting this variable initiates the named test. Sets
                to this variable require the presence of
                upsTestSpinLock in the same SNMP message.

                The set request will be rejected with an appropriate
                error message if the requested test cannot be
                performed, including attempts to start a test when
                another test is already in progress.  The status of
                the current or last test is maintained in
                upsTestResultsSummary. Tests in progress may be
                aborted by setting the upsTestId variable to
```

```
                upsTestAbortTestInProgress.

                Read operations return the value of the name of the
                test in progress if a test is in progress or the name
                of the last test performed if no test is in progress,
                unless no test has been run, in which case the well
                known value upsTestNoTestsInitiated is returned."
        ::= { upsTest 1 }

    -- see [6] for more information on the semantics of objects with
    -- syntax of TestAndIncr

    upsTestSpinLock OBJECT-TYPE
        SYNTAX     TestAndIncr
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "A spin lock on the test subsystem.  The spinlock is
                used as follows.

                Before starting a test, a manager-station should make
                sure that a test is not in progress as follows:

                    try_again:
                      get (upsTestSpinLock)
                      while (upsTestResultsSummary == inProgress) {
                        /* loop while a test is running for another
                manager */
                        short delay
                        get (upsTestSpinLock)
                      }
                      lock_value = upsTestSpinLock
                      /* no test in progress, start the test */
                      set (upsTestSpinLock = lock_value, upsTestId =
                requested_test)
                      if (error_index == 1) { /* (upsTestSpinLock
                failed) */
                        /* if problem is not access control, then
                            some other manager slipped in ahead of us
                */
                        goto try_again
                      }
                      if (error_index == 2) { /* (upsTestId) */
                        /* cannot perform the test */
                        give up
                      }
                      /* test started ok */
                      /* wait for test completion by polling
```

```
                upsTestResultsSummary */
                      get (upsTestSpinLock, upsTestResultsSummary,
                upsTestResultsDetail)
                      while (upsTestResultsSummary == inProgress) {
                        short delay
                        get (upsTestSpinLock, upsTestResultsSummary,
                upsTestResultsDetail)
                      }
                      /* when test completes, retrieve any additional
                test results */
                      /* if upsTestSpinLock == lock_value + 1, then
                these are our test */
                      /* results (as opposed to another manager's */
                      The initial value of upsTestSpinLock at agent
                initialization shall
                      be 1."
        ::= { upsTest 2 }

    upsTestResultsSummary OBJECT-TYPE
        SYNTAX     INTEGER {
            donePass(1),
            doneWarning(2),
            doneError(3),
            aborted(4),
            inProgress(5),
            noTestsInitiated(6)
        }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The results of the current or last UPS diagnostics
                test performed.  The values for donePass(1),
                doneWarning(2), and doneError(3) indicate that the
                test completed either successfully, with a warning, or
                with an error, respectively.  The value aborted(4) is
                returned for tests which are aborted by setting the
                value of upsTestId to upsTestAbortTestInProgress.
                Tests which have not yet concluded are indicated by
                inProgress(5).  The value noTestsInitiated(6)
                indicates that no previous test results are available,
                such as is the case when no tests have been run since
                the last reinitialization of the network management
                subsystem and the system has no provision for non-
                volatile storage of test results."
        ::= { upsTest 3 }

    upsTestResultsDetail OBJECT-TYPE
        SYNTAX     DisplayString (SIZE (0..255))
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "Additional information about upsTestResultsSummary.
                If no additional information available, a zero length
                string is returned."
        ::= { upsTest 4 }

    upsTestStartTime OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The value of sysUpTime at the time the test in
                progress was initiated, or, if no test is in progress,
                the time the previous test was initiated.  If the
                value of upsTestResultsSummary is noTestsInitiated(6),
                upsTestStartTime has the value 0."
        ::= { upsTest 5 }

    upsTestElapsedTime OBJECT-TYPE
        SYNTAX     TimeInterval
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The amount of time, in TimeTicks, since the test in
                progress was initiated, or, if no test is in progress,
                the previous test took to complete.  If the value of
                upsTestResultsSummary is noTestsInitiated(6),
                upsTestElapsedTime has the value 0."
        ::= { upsTest 6 }

    --
    -- Well known tests.
    --

    upsWellKnownTests     OBJECT IDENTIFIER ::= { upsTest 7 }


    upsTestNoTestsInitiated OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "No tests have been initiated and no test is in
                progress."
        ::= { upsWellKnownTests  1 }

    upsTestAbortTestInProgress OBJECT-IDENTITY
        STATUS     current
```

```
        DESCRIPTION
                "The test in progress is to be aborted / the test in
                progress was aborted."
        ::= { upsWellKnownTests  2 }

    upsTestGeneralSystemsTest OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The manufacturer's standard test of UPS device
                systems."
        ::= { upsWellKnownTests  3 }

    upsTestQuickBatteryTest OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "A test that is sufficient to determine if the battery
                needs replacement."
        ::= { upsWellKnownTests  4 }

    upsTestDeepBatteryCalibration OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The system is placed on battery to a discharge level,
                set by the manufacturer, sufficient to determine
                battery replacement and battery run-time with a high
                degree of confidence.  WARNING:  this test will leave
                the battery in a low charge state and will require
                time for recharging to a level sufficient to provide
                normal battery duration for the protected load."
        ::= { upsWellKnownTests  5 }


    --
    -- The Control group.
    --

    upsControl            OBJECT IDENTIFIER ::= { upsObjects 8 }

    upsShutdownType OBJECT-TYPE
        SYNTAX     INTEGER {
            output(1),
            system(2)
        }
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "This object determines the nature of the action to be
                taken at the time when the countdown of the
```

```
                upsShutdownAfterDelay and upsRebootWithDuration
                objects reaches zero.

                Setting this object to output(1) indicates that
                shutdown requests should cause only the output of the
                UPS to turn off.  Setting this object to system(2)
                indicates that shutdown requests will cause the entire
                UPS system to turn off."
        ::= { upsControl 1 }

    upsShutdownAfterDelay OBJECT-TYPE
        SYNTAX     INTEGER (-1..2147483648)
        UNITS      "seconds"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "Setting this object will shutdown (i.e., turn off)
                either the UPS output or the UPS system (as determined
                by the value of upsShutdownType at the time of
                shutdown) after the indicated number of seconds, or
                less if the UPS batteries become depleted. Setting
                this object to 0 will cause the shutdown to occur
                immediately.  Setting this object to -1 will abort the
                countdown.  If the system is already in the desired
                state at the time the countdown reaches 0, then
                nothing will happen.  That is, there is no additional
                action at that time if upsShutdownType = system and
                the system is already off.  Similarly, there is no
                additional action at that time if upsShutdownType =
                output and the output is already off.  When read,
                upsShutdownAfterDelay will return the number of
                seconds remaining until shutdown, or -1 if no shutdown
                countdown is in effect.  On some systems, if the agent
                is restarted while a shutdown countdown is in effect,
                the countdown may be aborted.  Sets to this object
                override any upsShutdownAfterDelay already in effect."
        ::= { upsControl 2 }

    upsStartupAfterDelay OBJECT-TYPE
        SYNTAX     INTEGER (-1..2147483648)
        UNITS      "seconds"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "Setting this object will start the output after the
                indicated number of seconds, including starting the
                UPS, if necessary.  Setting this object to 0 will
                cause the startup to occur immediately.  Setting this
```

```
                object to -1 will abort the countdown.  If the output
                is already on at the time the countdown reaches 0,
                then nothing will happen.  Sets to this object
                override the effect of any upsStartupAfterDelay
                countdown or upsRebootWithDuration countdown in
                progress.  When read, upsStartupAfterDelay will return
                the number of seconds until startup, or -1 if no
                startup countdown is in effect.  If the countdown
                expires during a utility failure, the startup shall
                not occur until the utility power is restored.  On
                some systems, if the agent is restarted while a
                startup countdown is in effect, the countdown is
                aborted."
        ::= { upsControl 3 }

    upsRebootWithDuration OBJECT-TYPE
        SYNTAX     INTEGER (-1..300)
        UNITS      "seconds"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "Setting this object will immediately shutdown (i.e.,
                turn off) either the UPS output or the UPS system (as
                determined by the value of upsShutdownType at the time
                of shutdown) for a period equal to the indicated
                number of seconds, after which time the output will be
                started, including starting the UPS, if necessary.  If
                the number of seconds required to perform the request
                is greater than the requested duration, then the
                requested shutdown and startup cycle shall be
                performed in the minimum time possible, but in no case
                shall this require more than the requested duration
                plus 60 seconds.  When read, upsRebootWithDuration
                shall return the number of seconds remaining in the
                countdown, or -1 if no countdown is in progress.  If
                the startup should occur during a utility failure, the
                startup shall not occur until the utility power is
                restored."
        ::= { upsControl 4 }

    upsAutoRestart OBJECT-TYPE
        SYNTAX     INTEGER {
            on(1),
            off(2)
        }
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
```

```
                "Setting this object to 'on' will cause the UPS system
                to restart after a shutdown if the shutdown occurred
                during a power loss as a result of either a
                upsShutdownAfterDelay or an internal battery depleted
                condition.  Setting this object to 'off' will prevent
                the UPS system from restarting after a shutdown until
                an operator manually or remotely explicitly restarts
                it.  If the UPS is in a startup or reboot countdown,
                then the UPS will not restart until that delay has
                been satisfied."
        ::= { upsControl 5 }


    --
    -- The Configuration group.
    --

    upsConfig             OBJECT IDENTIFIER ::= { upsObjects 9 }

    upsConfigInputVoltage OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "RMS Volts"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The magnitude of the nominal input voltage.  On those
                systems which support read-write access to this
                object, if there is an attempt to set this variable to
                a value that is not supported, the request must be
                rejected and the agent shall respond with an
                appropriate error message, i.e., badValue for SNMPv1,
                or inconsistentValue for SNMPv2."
        ::= { upsConfig 1 }

    upsConfigInputFreq OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "0.1 Hertz"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The nominal input frequency.  On those systems which
                support read-write access to this object, if there is
                an attempt to set this variable to a value that is not
                supported, the request must be rejected and the agent
                shall respond with an appropriate error message, i.e.,
                badValue for SNMPv1, or inconsistentValue for SNMPv2."
        ::= { upsConfig 2 }
```

```
   upsConfigOutputVoltage OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "RMS Volts"
       MAX-ACCESS read-write
       STATUS     current
       DESCRIPTION
               "The magnitude of the nominal output voltage.  On
               those systems which support read-write access to this
               object, if there is an attempt to set this variable to
               a value that is not supported, the request must be
               rejected and the agent shall respond with an
               appropriate error message, i.e., badValue for SNMPv1,
               or inconsistentValue for SNMPv2."
       ::= { upsConfig 3 }

   upsConfigOutputFreq OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "0.1 Hertz"
       MAX-ACCESS read-write
       STATUS     current
       DESCRIPTION
               "The nominal output frequency.  On those systems which
               support read-write access to this object, if there is
               an attempt to set this variable to a value that is not
               supported, the request must be rejected and the agent
               shall respond with an appropriate error message, i.e.,
               badValue for SNMPv1, or inconsistentValue for SNMPv2."
       ::= { upsConfig 4 }

   upsConfigOutputVA OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "Volt-Amps"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The magnitude of the nominal Volt-Amp rating."
       ::= { upsConfig 5 }

   upsConfigOutputPower OBJECT-TYPE
       SYNTAX     NonNegativeInteger
       UNITS      "Watts"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The magnitude of the nominal true power rating."
       ::= { upsConfig 6 }

   upsConfigLowBattTime OBJECT-TYPE
```

```
        SYNTAX     NonNegativeInteger
        UNITS      "minutes"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The value of upsEstimatedMinutesRemaining at which a
                lowBattery condition is declared.  For agents which
                support only discrete (discontinuous) values, then the
                agent shall round up to the next supported value.  If
                the requested value is larger than the largest
                supported value, then the largest supported value
                shall be selected."
        ::= { upsConfig 7 }

    upsConfigAudibleStatus OBJECT-TYPE
        SYNTAX     INTEGER {
            disabled(1),
            enabled(2),
            muted(3)
        }
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The requested state of the audible alarm.  When in
                the disabled state, the audible alarm should never
                sound.  The enabled state is self-describing.  Setting
                this object to muted(3) when the audible alarm is
                sounding shall temporarily silence the alarm.  It will
                remain muted until it would normally stop sounding and
                the value returned for read operations during this
                period shall equal muted(3).  At the end of this
                period, the value shall revert to enabled(2).  Writes
                of the value muted(3) when the audible alarm is not
                sounding shall be accepted but otherwise shall have no
                effect."
        ::= { upsConfig 8 }

    upsConfigLowVoltageTransferPoint OBJECT-TYPE
        SYNTAX     NonNegativeInteger
        UNITS      "RMS Volts"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The minimum input line voltage allowed before the UPS
                system transfers to battery backup."
        ::= { upsConfig 9 }

    upsConfigHighVoltageTransferPoint OBJECT-TYPE
```

```
        SYNTAX     NonNegativeInteger
        UNITS      "RMS Volts"
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "The maximum line voltage allowed before the UPS
                system transfers to battery backup."
        ::= { upsConfig 10 }


    --
    -- notifications, i.e., traps
    --
    upsTraps              OBJECT IDENTIFIER ::= { upsMIB 2 }

    -- This section defines the well-known notifications sent by
    -- UPS agents.
    -- Care must be taken to insure that no particular notification
    -- is sent to a single receiving entity more often than once
    -- every five seconds.

    upsTrapOnBattery NOTIFICATION-TYPE
        OBJECTS { upsEstimatedMinutesRemaining, upsSecondsOnBattery,
                  upsConfigLowBattTime }
        STATUS  current
        DESCRIPTION
                "The UPS is operating on battery power.  This trap is
                persistent and is resent at one minute intervals until
                the UPS either turns off or is no longer running on
                battery."
      ::= { upsTraps 1 }

    upsTrapTestCompleted NOTIFICATION-TYPE
        OBJECTS { upsTestId, upsTestSpinLock,
                  upsTestResultsSummary, upsTestResultsDetail,
                  upsTestStartTime, upsTestElapsedTime }
        STATUS  current
        DESCRIPTION
                "This trap is sent upon completion of a UPS diagnostic
                test."
      ::= { upsTraps 2 }

    upsTrapAlarmEntryAdded NOTIFICATION-TYPE
        OBJECTS { upsAlarmId, upsAlarmDescr }
        STATUS  current
        DESCRIPTION
                "This trap is sent each time an alarm is inserted into
                to the alarm table.  It is sent on the insertion of
```

```
                all alarms except for upsAlarmOnBattery and
                upsAlarmTestInProgress."
      ::= { upsTraps 3 }

   upsTrapAlarmEntryRemoved NOTIFICATION-TYPE
       OBJECTS { upsAlarmId, upsAlarmDescr }
       STATUS  current
       DESCRIPTION
                "This trap is sent each time an alarm is removed from
                the alarm table.  It is sent on the removal of all
                alarms except for upsAlarmTestInProgress."
      ::= { upsTraps 4 }


   --
   -- conformance information
   --
   upsConformance        OBJECT IDENTIFIER ::= { upsMIB 3 }

   upsCompliances        OBJECT IDENTIFIER ::= { upsConformance 1 }


   --
   -- compliance statements
   --

   upsSubsetCompliance MODULE-COMPLIANCE
       STATUS     current
       DESCRIPTION

                "The compliance statement for UPSs that only support
                the two-contact communication protocol."
       MODULE -- this module
           MANDATORY-GROUPS  { upsSubsetIdentGroup,
                       upsSubsetBatteryGroup, upsSubsetInputGroup,
                       upsSubsetOutputGroup, upsSubsetAlarmGroup,
                       upsSubsetControlGroup, upsSubsetConfigGroup }

       OBJECT     upsBatteryStatus
       SYNTAX     INTEGER {
           batteryNormal(2),
           batteryLow(3)
       }
       DESCRIPTION
                "Support of the values unknown(1) and
                batteryDepleted(4) is not required."

       OBJECT     upsAlarmDescr
```

```
        DESCRIPTION
                "Support of all `well known' alarm types is not
                required.  The well known alarm types which must be
                supported are: upsAlarmOnBattery, upsAlarmLowBattery,
                upsAlarmInputBad, upsAlarmUpsOutputOff,
                upsAlarmUpsSystemOff, and upsAlarmTestInProgress."

        OBJECT     upsOutputSource
        SYNTAX     INTEGER {
            normal(2),
            battery(4)
        }
        DESCRIPTION
                "Support of the values other(1), none(2), bypass(4),
                booster(6) and reducer(7) is not required."

        OBJECT     upsShutdownType
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required, i.e., compliant
                systems need not support more than one shutdown type."

        OBJECT     upsAutoRestart
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required, i.e., compliant
                systems need not support more than one restart type."

        OBJECT     upsConfigInputVoltage
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigInputFreq
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigOutputVoltage
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigOutputFreq
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."
```

```
        ::= { upsCompliances 1 }

    upsBasicCompliance MODULE-COMPLIANCE
        STATUS     current
        DESCRIPTION

                "The compliance statement for UPSs that support
                full-featured functions, such as control."
        MODULE -- this module
            MANDATORY-GROUPS { upsBasicIdentGroup,
                       upsBasicBatteryGroup, upsBasicInputGroup,
                       upsBasicOutputGroup, upsBasicAlarmGroup,
                       upsBasicTestGroup, upsBasicControlGroup,
                       upsBasicConfigGroup }


        OBJECT     upsAlarmDescr
        DESCRIPTION
                "Support of all `well known' alarm types is not
                required.  The well known alarm types which must be
                supported are: upsAlarmOnBattery, upsAlarmLowBattery,
                upsAlarmDepletedBattery, upsAlarmTempBad,
                upsAlarmInputBad, upsAlarmOutputOverload,
                upsAlarmOnBypass, upsAlarmBypassBad,
                upsAlarmOutputOffAsRequested,
                upsAlarmUpsOffAsRequested, upsAlarmUpsOutputOff,
                upsAlarmUpsSystemOff, upsAlarmGeneralFault,
                upsAlarmDiagnosticTestFailed,
                upsAlarmCommunicationsLost, upsAlarmShutdownPending,
                and upsAlarmTestInProgress."

        OBJECT     upsTestId
        DESCRIPTION
                "Support of all `well known' test types is not
                required.  If no tests are supported, then the only
                well known test type which must be supported is
                upsTestNoTestsInitiated."

        OBJECT     upsOutputSource
        SYNTAX     INTEGER {
            normal(2),
            battery(4)
        }
        DESCRIPTION
                "Support of the values other(1), none(2), bypass(4),
                booster(6) and reducer(7) is not required."

            GROUP upsBasicBypassGroup
```

```
            DESCRIPTION
                "The upsBasicBypassGroup is only required for UPSs
                that have a Bypass present."
        OBJECT     upsShutdownType
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required, i.e., compliant
                systems need not support more than one shutdown type."

        OBJECT     upsAutoRestart
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required, i.e., compliant
                systems need not support more than one restart type."

        OBJECT     upsConfigInputVoltage
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigInputFreq
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigOutputVoltage
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigOutputFreq
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigLowBattTime
        DESCRIPTION
                "Implementation of all possible values may be onerous
                for some systems.  Consequently, not all possible
                values must be supported.  However, at least two
                different manufacturer-selected values must be
                supported."

        ::= { upsCompliances 2 }

    upsFullCompliance MODULE-COMPLIANCE
        STATUS     current
        DESCRIPTION
```

```
            "The compliance statement for UPSs that support
            advanced full-featured functions."
        MODULE -- this module
            MANDATORY-GROUPS { upsFullIdentGroup, upsFullBatteryGroup,
                       upsFullInputGroup, upsFullOutputGroup,
                       upsFullAlarmGroup, upsFullTestGroup,
                       upsFullControlGroup, upsFullConfigGroup }

        OBJECT     upsAlarmDescr
        DESCRIPTION
                "Support of all `well known' alarm types is not
                required.  The well known alarm types which must be
                supported are: upsAlarmBatteryBad, upsAlarmOnBattery,
                upsAlarmLowBattery, upsAlarmDepletedBattery,
                upsAlarmTempBad, upsAlarmInputBad, upsAlarmOnBypass,
                upsAlarmBypassBad, upsAlarmOutputOffAsRequested,
                upsAlarmUpsOffAsRequested, upsAlarmUpsOutputOff,
                upsAlarmUpsSystemOff, upsAlarmGeneralFault,
                upsAlarmDiagnosticTestFailed,
                upsAlarmCommunicationsLost, upsAlarmShutdownPending,
                and upsAlarmTestInProgress."

        OBJECT     upsTestId
        DESCRIPTION
                "Support of all `well known' test types is not
                required.  The well known test types which must be
                supported are: upsTestNoTestsInitiated,
                upsTestGeneralSystemsTest, and
                upsTestQuickBatteryTest."

        OBJECT     upsOutputSource
        SYNTAX     INTEGER {
            normal(2),
            battery(4)
        }
        DESCRIPTION
                "Support of the values other(1), none(2), bypass(4),
                booster(6) and reducer(7) is not required."

            GROUP upsFullBypassGroup
            DESCRIPTION
                "The upsFullBypassGroup is only required for UPSs that
                have a Bypass present."

        OBJECT     upsShutdownType
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required, i.e., compliant
```

```
                systems need not support more than one shutdown type."

        OBJECT     upsAutoRestart
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required, i.e., compliant
                systems need not support more than one restart type."

        OBJECT     upsConfigInputVoltage
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigInputFreq
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigOutputVoltage
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigOutputFreq
        MIN-ACCESS read-only
        DESCRIPTION
                "Read-write access is not required."

        OBJECT     upsConfigLowBattTime
        DESCRIPTION
                "Implementation of all possible values may be onerous
                for some systems.  Consequently, not all possible
                values must be supported.  However, at least two
                different manufacturer-selected values must be
                supported."

        ::= { upsCompliances 3 }


    --
    -- units of conformance
    --

    -- summary at a glance:

    --                                      subset  basic   adv
    --upsIdentManufacturer                 x       x       x
    --upsIdentModel                        x       x       x
```

--upsIdentUPSSoftwareVersion		x	x	
--upsIdentAgentSoftwareVersion	x	x	x	
--upsIdentName	x	x	x	
--upsIdentAttachedDevices	x		x	
--				
--upsBatteryStatus	x	x	x	notes
--upsSecondsOnBattery	x	x	x	
--upsEstimatedMinutesRemaining			x	
--upsEstimatedChargeRemaining			x	
--upsBatteryVoltage				
--upsBatteryCurrent				
--upsBatteryTemperature				
--				
--upsInputLineBads	x	x	x	
--upsInputNumLines		x	x	
--upsInputFrequency		x	x	
--upsInputVoltage		x	x	
--upsInputCurrent				
--upsInputTruePower				
--				
--upsOutputSource	x	x	x	notes
--upsOutputFrequency		x	x	
--upsOutputNumLines		x	x	
--upsOutputVoltage		x	x	
--upsOutputCurrent			x	
--upsOutputPower			x	
--upsOutputPercentLoad			x	
--				
--				
--upsBypassFrequency		x	x	notes
--upsBypassNumLines		x	x	
--upsBypassVoltage		x	x	
--upsBypassCurrent				
--upsBypassPower				
--				
--				
--upsAlarmsPresent	x	x	x	
--upsAlarmDescr	x	x	x	notes
--upsAlarmTime	x	x	x	
--				
--upsTestId		x	x	notes
--upsTestSpinLock		x	x	
--upsTestResultsSummary		x	x	
--upsTestResultsDetail		x	x	
--upsTestStartTime		x	x	
--upsTestElapsedTime		x	x	
--				
--upsShutdownType	x	x	x	notes

```
   --upsShutdownAfterDelay                  x       x       x
   --upsStartupAfterDelay                           x       x
   --upsRebootWithDuration                          x       x
   --upsAutoRestart                         x       x       x  notes
   --
   --upsConfigInputVoltage                  x       x       x  notes
   --upsConfigInputFreq                     x       x       x  notes
   --upsConfigOutputVoltage                 x       x       x  notes
   --upsConfigOutputFreq                    x       x       x  notes
   --upsConfigOutputVA                      x       x       x
   --upsConfigOutputPower                   x       x       x
   --upsConfigLowBattTime                           x       x  notes
   --upsConfigAudibleStatus                         x       x
   --upsConfigLowVoltageTransferPoint
   --upsConfigHighVoltageTransferPoint


   -- units of conformance
   upsGroups             OBJECT IDENTIFIER ::= { upsConformance 2 }

   upsSubsetGroups       OBJECT IDENTIFIER ::= { upsGroups 1 }

   upsSubsetIdentGroup OBJECT-GROUP
       OBJECTS { upsIdentManufacturer, upsIdentModel,
                 upsIdentAgentSoftwareVersion, upsIdentName,
                 upsIdentAttachedDevices }
       STATUS  current
       DESCRIPTION
               "The upsSubsetIdentGroup defines objects which are
               common across all UPSs which meet subset compliance.
               Most devices which conform to the upsSubsetIdentGroup
               will provide access to these objects via a proxy
               agent.  If the proxy agent is compatible with multiple
               UPS types, configuration of the proxy agent will
               require specifying some of these values, either
               individually, or as a group (perhaps through a table
               lookup mechanism based on the UPS model number)."
       ::= { upsSubsetGroups 1 }

   upsSubsetBatteryGroup OBJECT-GROUP
       OBJECTS { upsBatteryStatus, upsSecondsOnBattery }
       STATUS  current
       DESCRIPTION
               "The upsSubsetBatteryGroup defines the objects that
               are common to battery groups of two-contact UPSs."
       ::= { upsSubsetGroups 2 }

   upsSubsetInputGroup OBJECT-GROUP
```

```
        OBJECTS { upsInputLineBads }
        STATUS  current
        DESCRIPTION
                "The upsSubsetInputGroup defines the objects that are
                common to the Input groups of two-contact UPSs."
        ::= { upsSubsetGroups 3 }

    upsSubsetOutputGroup OBJECT-GROUP
        OBJECTS { upsOutputSource }
        STATUS  current
        DESCRIPTION
                "The upsSubsetOutputGroup defines the objects that are
                common to the Output groups of two-contact UPSs."
        ::= { upsSubsetGroups 4 }


    -- { upsSubsetGroups 5 } is reserved for
    -- future use (upsSubsetBypassGroup)

    upsSubsetAlarmGroup OBJECT-GROUP
        OBJECTS { upsAlarmsPresent, upsAlarmDescr, upsAlarmTime }
        STATUS  current
        DESCRIPTION
                "The upsSubsetAlarmGroup defines the objects that are
                common to the Alarm groups of two-contact UPSs."
        ::= { upsSubsetGroups 6 }

    -- { upsSubsetGroups 7 } is reserved for
    -- future use (upsSubsetTestGroup)

    upsSubsetControlGroup OBJECT-GROUP
        OBJECTS { upsShutdownType, upsShutdownAfterDelay,
                        upsAutoRestart }
        STATUS  current
        DESCRIPTION
                "The upsSubsetControlGroup defines the objects that
                are common to the Control groups of two-contact UPSs."
        ::= { upsSubsetGroups 8 }

    upsSubsetConfigGroup OBJECT-GROUP
        OBJECTS { upsConfigInputVoltage, upsConfigInputFreq,
                  upsConfigOutputVoltage, upsConfigOutputFreq,
                  upsConfigOutputVA, upsConfigOutputPower }
        STATUS  current
        DESCRIPTION
                "The upsSubsetConfigGroup defines the objects that are
                common to the Config groups of two-contact UPSs."
        ::= { upsSubsetGroups 9 }
```

```
   upsBasicGroups        OBJECT IDENTIFIER ::= { upsGroups 2 }

   upsBasicIdentGroup OBJECT-GROUP
       OBJECTS { upsIdentManufacturer, upsIdentModel,
                 upsIdentUPSSoftwareVersion,
                 upsIdentAgentSoftwareVersion, upsIdentName }
       STATUS  current
       DESCRIPTION
               "The upsBasicIdentGroup defines objects which are
               common to the Ident group of compliant UPSs which
               support basic functions."
       ::= { upsBasicGroups 1 }

   upsBasicBatteryGroup OBJECT-GROUP
       OBJECTS { upsBatteryStatus, upsSecondsOnBattery }
       STATUS  current
       DESCRIPTION
               "The upsBasicBatteryGroup defines the objects that are
               common to the battery groups of compliant UPSs which
               support basic functions."
       ::= { upsBasicGroups 2 }

   upsBasicInputGroup OBJECT-GROUP
       OBJECTS { upsInputLineBads, upsInputNumLines,
                 upsInputFrequency, upsInputVoltage }
       STATUS  current
       DESCRIPTION
               "The upsBasicInputGroup defines the objects that are
               common to the Input groups of compliant UPSs which
               support basic functions."
       ::= { upsBasicGroups 3 }

   upsBasicOutputGroup OBJECT-GROUP
       OBJECTS { upsOutputSource, upsOutputFrequency,
                 upsOutputNumLines, upsOutputVoltage }
       STATUS  current
       DESCRIPTION
               "The upsBasicOutputGroup defines the objects that are
               common to the Output groups of compliant UPSs which
               support basic functions."
       ::= { upsBasicGroups 4 }

   upsBasicBypassGroup OBJECT-GROUP
       OBJECTS { upsBypassFrequency, upsBypassNumLines,
                 upsBypassVoltage }
       STATUS  current
       DESCRIPTION
               "The upsBasicBypassGroup defines the objects that are
```

```
                common to the Bypass groups of compliant UPSs which
                support basic functions."
        ::= { upsBasicGroups 5 }

    upsBasicAlarmGroup OBJECT-GROUP
        OBJECTS { upsAlarmsPresent, upsAlarmDescr, upsAlarmTime }
        STATUS  current
        DESCRIPTION

                "The upsBasicAlarmGroup defines the objects that are
                common to the Alarm  groups of compliant UPSs which
                support basic functions."
        ::= { upsBasicGroups 6 }

    upsBasicTestGroup OBJECT-GROUP
        OBJECTS { upsTestId, upsTestSpinLock,
                  upsTestResultsSummary, upsTestResultsDetail,
                  upsTestStartTime, upsTestElapsedTime }
        STATUS  current
        DESCRIPTION
                "The upsBasicTestGroup defines the objects that are
                common to the Test groups of compliant UPSs which
                support basic functions."
        ::= { upsBasicGroups 7 }

    upsBasicControlGroup OBJECT-GROUP
        OBJECTS { upsShutdownType, upsShutdownAfterDelay,
                  upsStartupAfterDelay, upsRebootWithDuration,
                  upsAutoRestart }
        STATUS  current
        DESCRIPTION
                "The upsBasicControlGroup defines the objects that are
                common to the Control groups of compliant UPSs which
                support basic functions."
        ::= { upsBasicGroups 8 }

    upsBasicConfigGroup OBJECT-GROUP
        OBJECTS { upsConfigInputVoltage, upsConfigInputFreq,
                  upsConfigOutputVoltage, upsConfigOutputFreq,
                  upsConfigOutputVA, upsConfigOutputPower,
                  upsConfigLowBattTime, upsConfigAudibleStatus }
        STATUS  current
        DESCRIPTION
                "The upsBasicConfigGroup defines the objects that are
                common to the Config groups of UPSs which support
                basic functions."
        ::= { upsBasicGroups 9 }
```

```
   upsFullGroups         OBJECT IDENTIFIER ::= { upsGroups 3 }

   upsFullIdentGroup OBJECT-GROUP
       OBJECTS { upsIdentManufacturer, upsIdentModel,
                 upsIdentUPSSoftwareVersion,
                 upsIdentAgentSoftwareVersion, upsIdentName,
                 upsIdentAttachedDevices }
       STATUS  current
       DESCRIPTION
               "The upsFullIdentGroup defines objects which are
               common to the Ident group of fully compliant UPSs."
       ::= { upsFullGroups 1 }

   upsFullBatteryGroup OBJECT-GROUP
       OBJECTS { upsBatteryStatus, upsSecondsOnBattery,
                 upsEstimatedMinutesRemaining,
                 upsEstimatedChargeRemaining }
       STATUS  current
       DESCRIPTION
               "The upsFullBatteryGroup defines the objects that are
               common to the battery groups of fully compliant UPSs."
       ::= { upsFullGroups 2 }

   upsFullInputGroup OBJECT-GROUP
       OBJECTS { upsInputLineBads, upsInputNumLines,
                 upsInputFrequency, upsInputVoltage }
       STATUS  current
       DESCRIPTION
               "The upsFullInputGroup defines the objects that are
               common to the Input groups of fully compliant UPSs."
       ::= { upsFullGroups 3 }

   upsFullOutputGroup OBJECT-GROUP
       OBJECTS { upsOutputSource, upsOutputFrequency,
                 upsOutputNumLines, upsOutputVoltage,
                 upsOutputCurrent, upsOutputPower,
                 upsOutputPercentLoad }
       STATUS  current
       DESCRIPTION
               "The upsFullOutputGroup defines the objects that are
               common to the Output groups of fully compliant UPSs."
       ::= { upsFullGroups 4 }

   upsFullBypassGroup OBJECT-GROUP
       OBJECTS { upsBypassFrequency, upsBypassNumLines,
                 upsBypassVoltage }
       STATUS  current
       DESCRIPTION
```

```
                "The upsFullBypassGroup defines the objects that are
                common to the Bypass groups of fully compliant UPSs."
        ::= { upsFullGroups 5 }

    upsFullAlarmGroup OBJECT-GROUP
        OBJECTS { upsAlarmsPresent, upsAlarmDescr, upsAlarmTime }
        STATUS  current
        DESCRIPTION

                "The upsFullAlarmGroup defines the objects that are
                common to the Alarm  groups of fully compliant UPSs."
        ::= { upsFullGroups 6 }

    upsFullTestGroup OBJECT-GROUP
        OBJECTS { upsTestId, upsTestSpinLock,
                  upsTestResultsSummary, upsTestResultsDetail,
                  upsTestStartTime, upsTestElapsedTime }
        STATUS  current
        DESCRIPTION
                "The upsFullTestGroup defines the objects that are
                common to the Test groups of fully compliant UPSs."
        ::= { upsFullGroups 7 }

    upsFullControlGroup OBJECT-GROUP
        OBJECTS { upsShutdownType, upsShutdownAfterDelay,
                  upsStartupAfterDelay, upsRebootWithDuration,
                  upsAutoRestart }
        STATUS  current
        DESCRIPTION
    "The upsFullControlGroup defines the objects that are
    common to the Control groups of fully compliant UPSs."
        ::= { upsFullGroups 8 }

    upsFullConfigGroup OBJECT-GROUP
        OBJECTS { upsConfigInputVoltage, upsConfigInputFreq,
                  upsConfigOutputVoltage, upsConfigOutputFreq,
                  upsConfigOutputVA, upsConfigOutputPower,
                  upsConfigLowBattTime, upsConfigAudibleStatus }
        STATUS  current
        DESCRIPTION
                "The upsFullConfigGroup defines the objects that are
                common to the Config groups of fully compliant UPSs."
        ::= { upsFullGroups 9 }

    END
```

5. Acknowledgements

The UPS MIB represents the combined work of the IETF UPS MIB Working Group, with particular, substantial authorship contributions from:

Mike Davison
Fibercom, Inc.

Ray Wasson
Consultant

Roger Draper
Liebert Corporation

Ken Key
SNMP Research, Incorporated

Pete Yoest
American Power Conversion

Doug Rademacher
American Power Conversion

Ron Pitt
Network Security Systems, Inc

Terry Zumwalt
International Power Machines

Lawren Markle
Tripp Lite

Bill Elliot
ONEAC

Tom Brennan
Exide Electronics

Brian Young
Best Power Technology

6. References

[1] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Inc., Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[2] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[3] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.

[4] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[5] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1444, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April, 1993.

[6] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1443, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

7. Security Considerations

Security issues are not discussed in this memo.

8. Author's Address

Jeffrey D. Case, Ph.D.
SNMP Research, Incorporated
3001 Kimberlin Heights Road
Knoxville, Tennessee 37920

Phone: (615) 573-1434
EMail: case@SNMP.COM