```
-- extracted from rfc2737.txt
-- at Wed Dec 15 07:07:49 1999

ENTITY-MIB DEFINITIONS ::= BEGIN

IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE,
        mib-2,
        NOTIFICATION-TYPE
                FROM SNMPv2-SMI
        TDomain,
        TAddress,
        TEXTUAL-CONVENTION,
        AutonomousType,
        RowPointer,
        TimeStamp,
        TruthValue
                FROM SNMPv2-TC
        MODULE-COMPLIANCE,
        OBJECT-GROUP,
        NOTIFICATION-GROUP
                FROM SNMPv2-CONF
        SnmpAdminString
                FROM SNMP-FRAMEWORK-MIB;

entityMIB MODULE-IDENTITY
        LAST-UPDATED "9912070000Z"        -- Dec 7, 1999 12:00:00 AM
        ORGANIZATION "IETF ENTMIB Working Group"
        CONTACT-INFO
                "WG E-mail: entmib@cisco.com
                Subscribe: majordomo@cisco.com
                       msg body: subscribe entmib

                   Keith McCloghrie
                   ENTMIB Working Group Chair
                   Cisco Systems Inc.
                   170 West Tasman Drive
                   San Jose, CA 95134
                   +1 408-526-5260
                   kzm@cisco.com

                   Andy Bierman
                   ENTMIB Working Group Editor
                   Cisco Systems Inc.
                   170 West Tasman Drive
                   San Jose, CA 95134
                   +1 408-527-3711
                   abierman@cisco.com"
        DESCRIPTION
                "The MIB module for representing multiple logical
                entities supported by a single SNMP agent."

        REVISION "9912070000Z"        -- Dec 7, 1999 12:00:00 AM
        DESCRIPTION
                "Initial Version of Entity MIB (Version 2).
                This revision obsoletes RFC 2037.
                This version published as RFC 2737."

        REVISION "9610310000Z"        -- Oct 31, 1996 12:00:00 AM
        DESCRIPTION
                "Initial version (version 1), published as
                RFC 2037."
 -- 1.3.6.1.2.1.47 --  ::= { mib-2 47 }

entityMIBObjects OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.1 --  ::= { entityMIB 1 }
-- MIB contains four groups
```

```
entityPhysical OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.1.1 --  ::= { entityMIBObjects 1 }

entityLogical OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.1.2 --  ::= { entityMIBObjects 2 }

entityMapping OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.1.3 --  ::= { entityMIBObjects 3 }

entityGeneral OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.1.4 --  ::= { entityMIBObjects 4 }
-- Textual Conventions

PhysicalIndex ::= TEXTUAL-CONVENTION
        STATUS      current
        DESCRIPTION
                "An arbitrary value which uniquely identifies the physical
                entity.  The value should be a small positive integer; index
                values for different physical entities are not necessarily
                contiguous."
        SYNTAX      INTEGER (1..2147483647)

PhysicalClass ::= TEXTUAL-CONVENTION
        STATUS      current
        DESCRIPTION
                "An enumerated value which provides an indication of the
                general hardware type of a particular physical entity.
                There are no restrictions as to the number of
                entPhysicalEntries of each entPhysicalClass, which must be
                instantiated by an agent.

                The enumeration 'other' is applicable if the physical entity
                class is known, but does not match any of the supported
                values.

                The enumeration 'unknown' is applicable if the physical
                entity class is unknown to the agent.

                The enumeration 'chassis' is applicable if the physical
                entity class is an overall container for networking
                equipment.  Any class of physical entity except a stack may
                be contained within a chassis, and a chassis may only be
                contained within a stack.

                The enumeration 'backplane' is applicable if the physical
                entity class is some sort of device for aggregating and
                forwarding networking traffic, such as a shared backplane in
                a modular ethernet switch.  Note that an agent may model a
                backplane as a single physical entity, which is actually
                implemented as multiple discrete physical components (within
                a chassis or stack).

                The enumeration 'container' is applicable if the physical
                entity class is capable of containing one or more removable
                physical entities, possibly of different types. For example,
                each (empty or full) slot in a chassis will be modeled as a
                container. Note that all removable physical entities should
                be modeled within a container entity, such as field-
                replaceable modules, fans, or power supplies.  Note that all
                known containers should be modeled by the agent, including
                empty containers.

                The enumeration 'powerSupply' is applicable if the physical
                entity class is a power-supplying component.

                The enumeration 'fan' is applicable if the physical entity
                class is a fan or other heat-reduction component.
                The enumeration 'sensor' is applicable if the physical
                entity class is some sort of sensor, such as a temperature
                sensor within a router chassis.

                The enumeration 'module' is applicable if the physical
                entity class is some sort of self-contained sub-system.  If
                it is removable, then it should be modeled within a
                container entity, otherwise it should be modeled directly
                within another physical entity (e.g., a chassis or another
                module).

                The enumeration 'port' is applicable if the physical entity
                class is some sort of networking port, capable of receiving
                and/or transmitting networking traffic.

                The enumeration 'stack' is applicable if the physical entity
                class is some sort of super-container (possibly virtual),
                intended to group together multiple chassis entities.  A
                stack may be realized by a 'virtual' cable, a real
                interconnect cable, attached to multiple chassis, or may in
                fact be comprised of multiple interconnect cables. A stack
                should not be modeled within any other physical entities,
                but a stack may be contained within another stack.  Only
                chassis entities should be contained within a stack."
        SYNTAX      INTEGER {
                        other(1),
                        unknown(2),
                        chassis(3),
                        backplane(4),
                        container(5),          -- e.g., chassis slot or daughter-card
holder
                        powerSupply(6),
                        fan(7),
                        sensor(8),
                        module(9),             -- e.g., plug-in card or daughter-card
                        port(10),
                        stack(11)              -- e.g., stack of multiple chassis
entities
                        }

SnmpEngineIdOrNone ::= TEXTUAL-CONVENTION
        STATUS      current
        DESCRIPTION
                "A specially formatted SnmpEngineID string for use with the
                Entity MIB.

                If an instance of an object of SYNTAX SnmpEngineIdOrNone has
                a non-zero length, then the object encoding and semantics
                are defined by the SnmpEngineID textual convention (see RFC
                2571 [RFC2571]).

                If an instance of an object of SYNTAX SnmpEngineIdOrNone
                contains a zero-length string, then no appropriate
                SnmpEngineID is associated with the logical entity (i.e.,
                SNMPv3 not supported)."
        SYNTAX      OCTET STRING (SIZE (0..32))
-- empty string or SnmpEngineID
--           The Physical Entity Table

entPhysicalTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF EntPhysicalEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "This table contains one row per physical entity.  There is
                always at least one row for an 'overall' physical entity."
 -- 1.3.6.1.2.1.47.1.1.1 --  ::= { entityPhysical 1 }

entPhysicalEntry OBJECT-TYPE
```

```
        SYNTAX      EntPhysicalEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "Information about a particular physical entity.

                Each entry provides objects (entPhysicalDescr,
                entPhysicalVendorType, and entPhysicalClass) to help an NMS
                identify and characterize the entry, and objects
                (entPhysicalContainedIn and entPhysicalParentRelPos) to help
                an NMS relate the particular entry to other entries in this
                table."
        INDEX {
                entPhysicalIndex
        }
 -- 1.3.6.1.2.1.47.1.1.1.1 --  ::= { entPhysicalTable 1 }

EntPhysicalEntry ::= SEQUENCE {
                entPhysicalIndex        PhysicalIndex,
                entPhysicalDescr        SnmpAdminString,
                entPhysicalVendorType   AutonomousType,
                entPhysicalContainedIn  INTEGER,
                entPhysicalClass        PhysicalClass,
                entPhysicalParentRelPos INTEGER,
                entPhysicalName         SnmpAdminString,
                entPhysicalHardwareRev  SnmpAdminString,
                entPhysicalFirmwareRev  SnmpAdminString,
                entPhysicalSoftwareRev  SnmpAdminString,
                entPhysicalSerialNum    SnmpAdminString,
                entPhysicalMfgName      SnmpAdminString,
                entPhysicalModelName    SnmpAdminString,
                entPhysicalAlias        SnmpAdminString,
                entPhysicalAssetID      SnmpAdminString,
                entPhysicalIsFRU        TruthValue
        }


entPhysicalIndex OBJECT-TYPE
        SYNTAX      PhysicalIndex
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "The index for this entry."
 -- 1.3.6.1.2.1.47.1.1.1.1.1 --  ::= { entPhysicalEntry 1 }

entPhysicalDescr OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "A textual description of physical entity.  This object
                should contain a string which identifies the manufacturer's
                name for the physical entity, and should be set to a
                distinct value for each version or model of the physical
                entity."
 -- 1.3.6.1.2.1.47.1.1.1.1.2 --  ::= { entPhysicalEntry 2 }

entPhysicalVendorType OBJECT-TYPE
        SYNTAX      AutonomousType
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "An indication of the vendor-specific hardware type of the
                physical entity.  Note that this is different from the
                definition of MIB-II's sysObjectID.

                An agent should set this object to a enterprise-specific
                registration identifier value indicating the specific
                equipment type in detail.  The associated instance of
```

```
             entPhysicalClass is used to indicate the general type of
             hardware device.

             If no vendor-specific registration identifier exists for
             this physical entity, or the value is unknown by this agent,
             then the value { 0 0 } is returned."
 -- 1.3.6.1.2.1.47.1.1.1.1.3 --  ::= { entPhysicalEntry 3 }

entPhysicalContainedIn OBJECT-TYPE
        SYNTAX      INTEGER (0..2147483647)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of entPhysicalIndex for the physical entity which
             'contains' this physical entity.  A value of zero indicates
             this physical entity is not contained in any other physical
             entity.  Note that the set of 'containment' relationships
             define a strict hierarchy; that is, recursion is not
             allowed.

             In the event a physical entity is contained by more than one
             physical entity (e.g., double-wide modules), this object
             should identify the containing entity with the lowest value
             of entPhysicalIndex."
 -- 1.3.6.1.2.1.47.1.1.1.1.4 --  ::= { entPhysicalEntry 4 }

entPhysicalClass OBJECT-TYPE
        SYNTAX      PhysicalClass
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "An indication of the general hardware type of the physical
             entity.

             An agent should set this object to the standard enumeration
             value which most accurately indicates the general class of
             the physical entity, or the primary class if there is more
             than one.

             If no appropriate standard registration identifier exists
             for this physical entity, then the value 'other(1)' is
             returned. If the value is unknown by this agent, then the
             value 'unknown(2)' is returned."
 -- 1.3.6.1.2.1.47.1.1.1.1.5 --  ::= { entPhysicalEntry 5 }

entPhysicalParentRelPos OBJECT-TYPE
        SYNTAX      INTEGER (-1..2147483647)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "An indication of the relative position of this 'child'
             component among all its 'sibling' components. Sibling
             components are defined as entPhysicalEntries which share the
             same instance values of each of the entPhysicalContainedIn
             and entPhysicalClass objects.

             An NMS can use this object to identify the relative ordering
             for all sibling components of a particular parent
             (identified by the entPhysicalContainedIn instance in each
             sibling entry).

             This value should match any external labeling of the
             physical component if possible. For example, for a container
             (e.g., card slot) labeled as 'slot #3',
             entPhysicalParentRelPos should have the value '3'.  Note
             that the entPhysicalEntry for the module plugged in slot 3
             should have an entPhysicalParentRelPos value of '1'.

             If the physical position of this component does not match
```

```
                any external numbering or clearly visible ordering, then
                user documentation or other external reference material
                should be used to determine the parent-relative position. If
                this is not possible, then the the agent should assign a
                consistent (but possibly arbitrary) ordering to a given set
                of 'sibling' components, perhaps based on internal
                representation of the components.

                If the agent cannot determine the parent-relative position
                for some reason, or if the associated value of
                entPhysicalContainedIn is '0', then the value '-1' is
                returned. Otherwise a non-negative integer is returned,
                indicating the parent-relative position of this physical
                entity.

                Parent-relative ordering normally starts from '1' and
                continues to 'N', where 'N' represents the highest
                positioned child entity.  However, if the physical entities
                (e.g., slots) are labeled from a starting position of zero,
                then the first sibling should be associated with a
                entPhysicalParentRelPos value of '0'.  Note that this
                ordering may be sparse or dense, depending on agent
                implementation.

                The actual values returned are not globally meaningful, as
                each 'parent' component may use different numbering
                algorithms. The ordering is only meaningful among siblings
                of the same parent component.

                The agent should retain parent-relative position values
                across reboots, either through algorithmic assignment or use
                of non-volatile storage."
 -- 1.3.6.1.2.1.47.1.1.1.1.6 --  ::= { entPhysicalEntry 6 }

entPhysicalName OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The textual name of the physical entity.  The value of this
                object should be the name of the component as assigned by
                the local device and should be suitable for use in commands
                entered at the device's `console'.  This might be a text
                name, such as `console' or a simple component number (e.g.,
                port or module number), such as `1', depending on the
                physical component naming syntax of the device.

                If there is no local name, or this object is otherwise not
                applicable, then this object contains a zero-length string.

                Note that the value of entPhysicalName for two physical
                entities will be the same in the event that the console
                interface does not distinguish between them, e.g., slot-1
                and the card in slot-1."
 -- 1.3.6.1.2.1.47.1.1.1.1.7 --  ::= { entPhysicalEntry 7 }

entPhysicalHardwareRev OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The vendor-specific hardware revision string for the
                physical entity.  The preferred value is the hardware
                revision identifier actually printed on the component itself
                (if present).

                Note that if revision information is stored internally in a
                non-printable (e.g., binary) format, then the agent must
                convert such information to a printable format, in an
                implementation-specific manner.

                If no specific hardware revision string is associated with
                the physical component, or this information is unknown to
                the agent, then this object will contain a zero-length
                string."
 -- 1.3.6.1.2.1.47.1.1.1.1.8 --  ::= { entPhysicalEntry 8 }

entPhysicalFirmwareRev OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The vendor-specific firmware revision string for the
                physical entity.

                Note that if revision information is stored internally in a
                non-printable (e.g., binary) format, then the agent must
                convert such information to a printable format, in an
                implementation-specific manner.

                If no specific firmware programs are associated with the
                physical component, or this information is unknown to the
                agent, then this object will contain a zero-length string."
 -- 1.3.6.1.2.1.47.1.1.1.1.9 --  ::= { entPhysicalEntry 9 }

entPhysicalSoftwareRev OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The vendor-specific software revision string for the
                physical entity.

                Note that if revision information is stored internally in a
                non-printable (e.g., binary) format, then the agent must
                convert such information to a printable format, in an
                implementation-specific manner.

                If no specific software programs are associated with the
                physical component, or this information is unknown to the
                agent, then this object will contain a zero-length string."
 -- 1.3.6.1.2.1.47.1.1.1.1.10 --  ::= { entPhysicalEntry 10 }

entPhysicalSerialNum OBJECT-TYPE
        SYNTAX      SnmpAdminString (SIZE (0..32))
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                "The vendor-specific serial number string for the physical
                entity.  The preferred value is the serial number string
                actually printed on the component itself (if present).

                On the first instantiation of an physical entity, the value
                of entPhysicalSerialNum associated with that entity is set
                to the correct vendor-assigned serial number, if this
                information is available to the agent.  If a serial number
                is unknown or non-existent, the entPhysicalSerialNum will be
                set to a zero-length string instead.

                Note that implementations which can correctly identify the
                serial numbers of all installed physical entities do not
                need to provide write access to the entPhysicalSerialNum
                object. Agents which cannot provide non-volatile storage for
                the entPhysicalSerialNum strings are not required to
                implement write access for this object.

                Not every physical component will have a serial number, or
                even need one.  Physical entities for which the associated
                value of the entPhysicalIsFRU object is equal to 'false(2)'
                (e.g., the repeater ports within a repeater module), do not
                need their own unique serial number. An agent does not have
                to provide write access for such entities, and may return a
                zero-length string.

                If write access is implemented for an instance of
                entPhysicalSerialNum, and a value is written into the
                instance, the agent must retain the supplied value in the
                entPhysicalSerialNum instance associated with the same
                physical entity for as long as that entity remains
                instantiated. This includes instantiations across all re-
                initializations/reboots of the network management system,
                including those which result in a change of the physical
                entity's entPhysicalIndex value."
 -- 1.3.6.1.2.1.47.1.1.1.1.11 --  ::= { entPhysicalEntry 11 }

entPhysicalMfgName OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The name of the manufacturer of this physical component.
                The preferred value is the manufacturer name string actually
                printed on the component itself (if present).
                Note that comparisons between instances of the
                entPhysicalModelName, entPhysicalFirmwareRev,
                entPhysicalSoftwareRev, and the entPhysicalSerialNum
                objects, are only meaningful amongst entPhysicalEntries with
                the same value of entPhysicalMfgName.

                If the manufacturer name string associated with the physical
                component is unknown to the agent, then this object will
                contain a zero-length string."
 -- 1.3.6.1.2.1.47.1.1.1.1.12 --  ::= { entPhysicalEntry 12 }

entPhysicalModelName OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The vendor-specific model name identifier string associated
                with this physical component.  The preferred value is the
                customer-visible part number, which may be printed on the
                component itself.

                If the model name string associated with the physical
                component is unknown to the agent, then this object will
                contain a zero-length string."
 -- 1.3.6.1.2.1.47.1.1.1.1.13 --  ::= { entPhysicalEntry 13 }

entPhysicalAlias OBJECT-TYPE
        SYNTAX      SnmpAdminString (SIZE (0..32))
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                "This object is an 'alias' name for the physical entity as
                specified by a network manager, and provides a non-volatile
                'handle' for the physical entity.

                On the first instantiation of an physical entity, the value
                of entPhysicalAlias associated with that entity is set to
                the zero-length string.  However, agent may set the value to
                a locally unique default value, instead of a zero-length
                string.

                If write access is implemented for an instance of
                entPhysicalAlias, and a value is written into the instance,
                the agent must retain the supplied value in the
```

```
            entPhysicalAlias instance associated with the same physical
            entity for as long as that entity remains instantiated.
            This includes instantiations across all re-
            initializations/reboots of the network management system,
            including those which result in a change of the physical
            entity's entPhysicalIndex value."
 -- 1.3.6.1.2.1.47.1.1.1.1.14 --  ::= { entPhysicalEntry 14 }

entPhysicalAssetID OBJECT-TYPE
        SYNTAX      SnmpAdminString (SIZE (0..32))
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                "This object is a user-assigned asset tracking identifier
                for the physical entity as specified by a network manager,
                and provides non-volatile storage of this information.

                On the first instantiation of an physical entity, the value
                of entPhysicalAssetID associated with that entity is set to
                the zero-length string.

                Not every physical component will have a asset tracking
                identifier, or even need one.  Physical entities for which
                the associated value of the entPhysicalIsFRU object is equal
                to 'false(2)' (e.g., the repeater ports within a repeater
                module), do not need their own unique asset tracking
                identifier. An agent does not have to provide write access
                for such entities, and may instead return a zero-length
                string.

                If write access is implemented for an instance of
                entPhysicalAssetID, and a value is written into the
                instance, the agent must retain the supplied value in the
                entPhysicalAssetID instance associated with the same
                physical entity for as long as that entity remains
                instantiated.  This includes instantiations across all re-
                initializations/reboots of the network management system,
                including those which result in a change of the physical
                entity's entPhysicalIndex value.

                If no asset tracking information is associated with the
                physical component, then this object will contain a zero-
                length string."
 -- 1.3.6.1.2.1.47.1.1.1.1.15 --  ::= { entPhysicalEntry 15 }

entPhysicalIsFRU OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "This object indicates whether or not this physical entity
                is considered a 'field replaceable unit' by the vendor.  If
                this object contains the value 'true(1)' then this
                entPhysicalEntry identifies a field replaceable unit.  For
                all entPhysicalEntries which represent components that are
                permanently contained within a field replaceable unit, the
                value 'false(2)' should be returned for this object."
 -- 1.3.6.1.2.1.47.1.1.1.1.16 --  ::= { entPhysicalEntry 16 }
--           The Logical Entity Table

entLogicalTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF EntLogicalEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "This table contains one row per logical entity.  For agents
                which implement more than one naming scope, at least one
                entry must exist. Agents which instantiate all MIB objects
                within a single naming scope are not required to implement
                this table."
 -- 1.3.6.1.2.1.47.1.2.1 --  ::= { entityLogical 1 }

entLogicalEntry OBJECT-TYPE
        SYNTAX      EntLogicalEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "Information about a particular logical entity.  Entities
                may be managed by this agent or other SNMP agents (possibly)
                in the same chassis."
        INDEX {
                entLogicalIndex
        }
 -- 1.3.6.1.2.1.47.1.2.1.1 --  ::= { entLogicalTable 1 }

EntLogicalEntry ::= SEQUENCE {
                entLogicalIndex            INTEGER,
                entLogicalDescr            SnmpAdminString,
                entLogicalType             AutonomousType,
                entLogicalCommunity        OCTET STRING,
                entLogicalTAddress         TAddress,
                entLogicalTDomain          TDomain,
                entLogicalContextEngineID SnmpEngineIdOrNone,
                entLogicalContextName      SnmpAdminString
        }


entLogicalIndex OBJECT-TYPE
        SYNTAX      INTEGER (1..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "The value of this object uniquely identifies the logical
                entity. The value should be a small positive integer; index
                values for different logical entities are are not
                necessarily contiguous."
 -- 1.3.6.1.2.1.47.1.2.1.1.1 --  ::= { entLogicalEntry 1 }

entLogicalDescr OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "A textual description of the logical entity.  This object
                should contain a string which identifies the manufacturer's
                name for the logical entity, and should be set to a distinct
                value for each version of the logical entity."
 -- 1.3.6.1.2.1.47.1.2.1.1.2 --  ::= { entLogicalEntry 2 }

entLogicalType OBJECT-TYPE
        SYNTAX      AutonomousType
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "An indication of the type of logical entity.  This will
                typically be the OBJECT IDENTIFIER name of the node in the
                SMI's naming hierarchy which represents the major MIB
                module, or the majority of the MIB modules, supported by the
                logical entity.  For example:
                   a logical entity of a regular host/router -> mib-2
                   a logical entity of a 802.1d bridge -> dot1dBridge
                   a logical entity of a 802.3 repeater -> snmpDot3RptrMgmt
                If an appropriate node in the SMI's naming hierarchy cannot
                be identified, the value 'mib-2' should be used."
 -- 1.3.6.1.2.1.47.1.2.1.1.3 --  ::= { entLogicalEntry 3 }

entLogicalCommunity OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE (0..255))
        MAX-ACCESS  read-only
        STATUS      deprecated
        DESCRIPTION
                "An SNMPv1 or SNMPv2C community-string which can be used to
                access detailed management information for this logical
                entity.  The agent should allow read access with this
                community string (to an appropriate subset of all managed
                objects) and may also return a community string based on the
                privileges of the request used to read this object.  Note
                that an agent may return a community string with read-only
                privileges, even if this object is accessed with a read-
                write community string. However, the agent must take care
                not to return a community string which allows more
                privileges than the community string used to access this
                object.

                A compliant SNMP agent may wish to conserve naming scopes by
                representing multiple logical entities in a single 'default'
                naming scope.  This is possible when the logical entities
                represented by the same value of entLogicalCommunity have no
                object instances in common.  For example, 'bridge1' and
                'repeater1' may be part of the main naming scope, but at
                least one additional community string is needed to represent
                'bridge2' and 'repeater2'.

                Logical entities 'bridge1' and 'repeater1' would be
                represented by sysOREntries associated with the 'default'
                naming scope.

                For agents not accessible via SNMPv1 or SNMPv2C, the value
                of this object is the empty string.  This object may also
                contain an empty string if a community string has not yet
                been assigned by the agent, or no community string with
                suitable access rights can be returned for a particular SNMP
                request.

                Note that this object is deprecated. Agents which implement
                SNMPv3 access should use the entLogicalContextEngineID and
                entLogicalContextName objects to identify the context
                associated with each logical entity.  SNMPv3 agents may
                return a zero-length string for this object, or may continue
                to return a community string (e.g., tri-lingual agent
                support)."
 -- 1.3.6.1.2.1.47.1.2.1.1.4 --  ::= { entLogicalEntry 4 }

entLogicalTAddress OBJECT-TYPE
        SYNTAX      TAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The transport service address by which the logical entity
                receives network management traffic, formatted according to
                the corresponding value of entLogicalTDomain.

                For snmpUDPDomain, a TAddress is 6 octets long, the initial
                4 octets containing the IP-address in network-byte order and
                the last 2 containing the UDP port in network-byte order.
                Consult 'Transport Mappings for Version 2 of the Simple
                Network Management Protocol' (RFC 1906 [RFC1906]) for
                further information on snmpUDPDomain."
 -- 1.3.6.1.2.1.47.1.2.1.1.5 --  ::= { entLogicalEntry 5 }

entLogicalTDomain OBJECT-TYPE
        SYNTAX      TDomain
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "Indicates the kind of transport service by which the
                logical entity receives network management traffic.
                Possible values for this object are presently found in the
                Transport Mappings for SNMPv2 document (RFC 1906
                [RFC1906])."
 -- 1.3.6.1.2.1.47.1.2.1.1.6 --  ::= { entLogicalEntry 6 }

entLogicalContextEngineID OBJECT-TYPE
        SYNTAX      SnmpEngineIdOrNone
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The authoritative contextEngineID that can be used to send
                an SNMP message concerning information held by this logical
                entity, to the address specified by the associated
                'entLogicalTAddress/entLogicalTDomain' pair.

                This object, together with the associated
                entLogicalContextName object, defines the context associated
                with a particular logical entity, and allows access to SNMP
                engines identified by a contextEngineId and contextName
                pair.

                If no value has been configured by the agent, a zero-length
                string is returned, or the agent may choose not to
                instantiate this object at all."
 -- 1.3.6.1.2.1.47.1.2.1.1.7 --  ::= { entLogicalEntry 7 }

entLogicalContextName OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The contextName that can be used to send an SNMP message
                concerning information held by this logical entity, to the
                address specified by the associated
                'entLogicalTAddress/entLogicalTDomain' pair.

                This object, together with the associated
                entLogicalContextEngineID object, defines the context
                associated with a particular logical entity, and allows
                access to SNMP engines identified by a contextEngineId and
                contextName pair.

                If no value has been configured by the agent, a zero-length
                string is returned, or the agent may choose not to
                instantiate this object at all."
 -- 1.3.6.1.2.1.47.1.2.1.1.8 --  ::= { entLogicalEntry 8 }

entLPMappingTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF EntLPMappingEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "This table contains zero or more rows of logical entity to
                physical equipment associations. For each logical entity
                known by this agent, there are zero or more mappings to the
                physical resources which are used to realize that logical
                entity.

                An agent should limit the number and nature of entries in
                this table such that only meaningful and non-redundant
                information is returned. For example, in a system which
                contains a single power supply, mappings between logical
                entities and the power supply are not useful and should not
                be included.

                Also, only the most appropriate physical component which is
                closest to the root of a particular containment tree should
                be identified in an entLPMapping entry.
                For example, suppose a bridge is realized on a particular
                module, and all ports on that module are ports on this
                bridge. A mapping between the bridge and the module would be
                useful, but additional mappings between the bridge and each
                of the ports on that module would be redundant (since the
                entPhysicalContainedIn hierarchy can provide the same
                information). If, on the other hand, more than one bridge
                was utilizing ports on this module, then mappings between
                each bridge and the ports it used would be appropriate.

                Also, in the case of a single backplane repeater, a mapping
                for the backplane to the single repeater entity is not
                necessary."
 -- 1.3.6.1.2.1.47.1.3.1 --  ::= { entityMapping 1 }

entLPMappingEntry OBJECT-TYPE
        SYNTAX      EntLPMappingEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "Information about a particular logical entity to physical
                equipment association. Note that the nature of the
                association is not specifically identified in this entry.
                It is expected that sufficient information exists in the
                MIBs used to manage a particular logical entity to infer how
                physical component information is utilized."
        INDEX {
                entLogicalIndex,
                entLPPhysicalIndex
        }
 -- 1.3.6.1.2.1.47.1.3.1.1 --  ::= { entLPMappingTable 1 }

EntLPMappingEntry ::= SEQUENCE {
                entLPPhysicalIndex PhysicalIndex
        }


entLPPhysicalIndex OBJECT-TYPE
        SYNTAX      PhysicalIndex
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The value of this object identifies the index value of a
                particular entPhysicalEntry associated with the indicated
                entLogicalEntity."
 -- 1.3.6.1.2.1.47.1.3.1.1.1 --  ::= { entLPMappingEntry 1 }
-- logical entity/component to alias table

entAliasMappingTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF EntAliasMappingEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "This table contains zero or more rows, representing
                mappings of logical entity and physical component to
                external MIB identifiers.  Each physical port in the system
                may be associated with a mapping to an external identifier,
                which itself is associated with a particular logical
                entity's naming scope.  A 'wildcard' mechanism is provided
                to indicate that an identifier is associated with more than
                one logical entity."
 -- 1.3.6.1.2.1.47.1.3.2 --  ::= { entityMapping 2 }

entAliasMappingEntry OBJECT-TYPE
        SYNTAX      EntAliasMappingEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "Information about a particular physical equipment, logical
                entity to external identifier binding. Each logical
                entity/physical component pair may be associated with one
                alias mapping.  The logical entity index may also be used as
                a 'wildcard' (refer to the entAliasLogicalIndexOrZero object
                DESCRIPTION clause for details.)

                Note that only entPhysicalIndex values which represent
                physical ports (i.e. associated entPhysicalClass value is
                'port(10)') are permitted to exist in this table."
        INDEX {
                entPhysicalIndex,
                entAliasLogicalIndexOrZero
        }
 -- 1.3.6.1.2.1.47.1.3.2.1 --  ::= { entAliasMappingTable 1 }

EntAliasMappingEntry ::= SEQUENCE {
                entAliasLogicalIndexOrZero INTEGER,
                entAliasMappingIdentifier  RowPointer
        }


entAliasLogicalIndexOrZero OBJECT-TYPE
        SYNTAX      INTEGER (0..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "The value of this object identifies the logical entity
                which defines the naming scope for the associated instance
                of the 'entAliasMappingIdentifier' object.

                If this object has a non-zero value, then it identifies the
                logical entity named by the same value of entLogicalIndex.

                If this object has a value of zero, then the mapping between
                the physical component and the alias identifier for this
                entAliasMapping entry is associated with all unspecified
                logical entities. That is, a value of zero (the default
                mapping) identifies any logical entity which does not have
                an explicit entry in this table for a particular
                entPhysicalIndex/entAliasMappingIdentifier pair.

                For example, to indicate that a particular interface (e.g.,
                physical component 33) is identified by the same value of
                ifIndex for all logical entities, the following instance
                might exist:

                        entAliasMappingIdentifier.33.0 = ifIndex.5

                In the event an entPhysicalEntry is associated differently
                for some logical entities, additional entAliasMapping
                entries may exist, e.g.:

                        entAliasMappingIdentifier.33.0 = ifIndex.6
                        entAliasMappingIdentifier.33.4 =  ifIndex.1
                        entAliasMappingIdentifier.33.5 =  ifIndex.1
                        entAliasMappingIdentifier.33.10 = ifIndex.12

                Note that entries with non-zero entAliasLogicalIndexOrZero
                index values have precedence over any zero-indexed entry. In
                this example, all logical entities except 4, 5, and 10,
                associate physical entity 33 with ifIndex.6."
 -- 1.3.6.1.2.1.47.1.3.2.1.1 --  ::= { entAliasMappingEntry 1 }

entAliasMappingIdentifier OBJECT-TYPE
        SYNTAX      RowPointer
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The value of this object identifies a particular conceptual
                row associated with the indicated entPhysicalIndex and
                entLogicalIndex pair.

                Since only physical ports are modeled in this table, only
                entries which represent interfaces or ports are allowed.  If
                an ifEntry exists on behalf of a particular physical port,
                then this object should identify the associated 'ifEntry'.
                For repeater ports, the appropriate row in the
                'rptrPortGroupTable' should be identified instead.

                For example, suppose a physical port was represented by
                entPhysicalEntry.3, entLogicalEntry.15 existed for a
                repeater, and entLogicalEntry.22 existed for a bridge.  Then
                there might be two related instances of
                entAliasMappingIdentifier:
                   entAliasMappingIdentifier.3.15 == rptrPortGroupIndex.5.2
                   entAliasMappingIdentifier.3.22 == ifIndex.17
                It is possible that other mappings (besides interfaces and
                repeater ports) may be defined in the future, as required.

                Bridge ports are identified by examining the Bridge MIB and
                appropriate ifEntries associated with each 'dot1dBasePort',
                and are thus not represented in this table."
 -- 1.3.6.1.2.1.47.1.3.2.1.2 --  ::= { entAliasMappingEntry 2 }
-- physical mapping table

entPhysicalContainsTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF EntPhysicalContainsEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "A table which exposes the container/'containee'
                relationships between physical entities. This table provides
                all the information found by constructing the virtual
                containment tree for a given entPhysicalTable, but in a more
                direct format.

                In the event a physical entity is contained by more than one
                other physical entity (e.g., double-wide modules), this
                table should include these additional mappings, which cannot
                be represented in the entPhysicalTable virtual containment
                tree."
 -- 1.3.6.1.2.1.47.1.3.3 --  ::= { entityMapping 3 }

entPhysicalContainsEntry OBJECT-TYPE
        SYNTAX      EntPhysicalContainsEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "A single container/'containee' relationship."
        INDEX {
                entPhysicalIndex,
                entPhysicalChildIndex
        }
 -- 1.3.6.1.2.1.47.1.3.3.1 --  ::= { entPhysicalContainsTable 1 }

EntPhysicalContainsEntry ::= SEQUENCE {
                entPhysicalChildIndex PhysicalIndex
        }


entPhysicalChildIndex OBJECT-TYPE
        SYNTAX      PhysicalIndex
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The value of entPhysicalIndex for the contained physical
                entity."
 -- 1.3.6.1.2.1.47.1.3.3.1.1 --  ::= { entPhysicalContainsEntry 1 }
-- last change time stamp for the whole MIB

entLastChangeTime OBJECT-TYPE
        SYNTAX      TimeStamp
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The value of sysUpTime at the time a conceptual row is
                created, modified, or deleted in any of these tables:
                        - entPhysicalTable
                        - entLogicalTable
                        - entLPMappingTable
                        - entAliasMappingTable
                        - entPhysicalContainsTable

                "
 -- 1.3.6.1.2.1.47.1.4.1 --  ::= { entityGeneral 1 }
-- Entity MIB Trap Definitions

entityMIBTraps OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.2 --  ::= { entityMIB 2 }

entityMIBTrapPrefix OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.2.0 --  ::= { entityMIBTraps 0 }

entConfigChange NOTIFICATION-TYPE
        STATUS      current
        DESCRIPTION
                "An entConfigChange notification is generated when the value
                of entLastChangeTime changes. It can be utilized by an NMS
                to trigger logical/physical entity table maintenance polls.

                An agent should not generate more than one entConfigChange
                'notification-event' in a given time interval (five seconds
                is the suggested default).  A 'notification-event' is the
                transmission of a single trap or inform PDU to a list of
                notification destinations.

                If additional configuration changes occur within the
                throttling period, then notification-events for these
                changes should be suppressed by the agent until the current
                throttling period expires.  At the end of a throttling
                period, one notification-event should be generated if any
                configuration changes occurred since the start of the
                throttling period. In such a case, another throttling period
                is started right away.

                An NMS should periodically check the value of
                entLastChangeTime to detect any missed entConfigChange
                notification-events, e.g., due to throttling or transmission
                loss."
 -- 1.3.6.1.2.1.47.2.0.1 --  ::= { entityMIBTrapPrefix 1 }
-- conformance information

entityConformance OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.3 --  ::= { entityMIB 3 }

entityCompliances OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.3.1 --  ::= { entityConformance 1 }

entityGroups OBJECT IDENTIFIER
 -- 1.3.6.1.2.1.47.3.2 --  ::= { entityConformance 2 }
-- compliance statements

entityCompliance MODULE-COMPLIANCE
        STATUS      deprecated
        DESCRIPTION
                "The compliance statement for SNMP entities which implement
                version 1 of the Entity MIB."
        MODULE
        MANDATORY-GROUPS {
                        entityPhysicalGroup,
                        entityLogicalGroup,
                        entityMappingGroup,
                        entityGeneralGroup,
                        entityNotificationsGroup
        }
 -- 1.3.6.1.2.1.47.3.1.1 --  ::= { entityCompliances 1 }

entity2Compliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION
                "The compliance statement for SNMP entities which implement
                version 2 of the Entity MIB."
        MODULE
        MANDATORY-GROUPS {
                        entityPhysicalGroup,
                        entityPhysical2Group,
                        entityGeneralGroup,
                        entityNotificationsGroup
        }
        VARIATION      entityLogical2Group
          DESCRIPTION
                "Implementation of this group is not mandatory for agents
                which model all MIB object instances within a single naming
                scope."
        VARIATION      entityMappingGroup
          DESCRIPTION
                "Implementation of the entPhysicalContainsTable is mandatory
                for all agents.  Implementation of the entLPMappingTable and
                entAliasMappingTables are not mandatory for agents which
                model all MIB object instances within a single naming scope.

                Note that the entAliasMappingTable may be useful for all
                agents, however implementation of the entityLogicalGroup or
                entityLogical2Group is required to support this table."
        OBJECT         entPhysicalSerialNum
          MIN-ACCESS   not-accessible
          DESCRIPTION
                "Read and write access is not required for agents which
                cannot identify serial number information for physical
                entities, and/or cannot provide non-volatile storage for
                NMS-assigned serial numbers.

                Write access is not required for agents which can identify
                serial number information for physical entities, but cannot
                provide non-volatile storage for NMS-assigned serial
                numbers.

                Write access is not required for physical entities for
                physical entities for which the associated value of the
                entPhysicalIsFRU object is equal to 'false(2)'."
        OBJECT         entPhysicalAlias
          MIN-ACCESS   read-only
          DESCRIPTION
                "Write access is required only if the associated
                entPhysicalClass value is equal to 'chassis(3)'."
        OBJECT         entPhysicalAssetID
          MIN-ACCESS   not-accessible
          DESCRIPTION
                "Read and write access is not required for agents which
                cannot provide non-volatile storage for NMS-assigned asset
                identifiers.

                Write access is not required for physical entities for which
                the associated value of entPhysicalIsFRU is equal to
                'false(2)'."
 -- 1.3.6.1.2.1.47.3.1.2 --  ::= { entityCompliances 2 }
-- MIB groupings

entityPhysicalGroup OBJECT-GROUP
        OBJECTS {
                entPhysicalDescr,
                entPhysicalVendorType,
                entPhysicalContainedIn,
                entPhysicalClass,
                entPhysicalParentRelPos,
                entPhysicalName
        }
        STATUS      current
        DESCRIPTION
                "The collection of objects which are used to represent
                physical system components, for which a single agent
                provides management information."
 -- 1.3.6.1.2.1.47.3.2.1 --  ::= { entityGroups 1 }

entityLogicalGroup OBJECT-GROUP
        OBJECTS {
                entLogicalDescr,
                entLogicalType,
                entLogicalCommunity,
                entLogicalTAddress,
                entLogicalTDomain
        }
        STATUS      deprecated
        DESCRIPTION
                "The collection of objects which are used to represent the
                list of logical entities for which a single agent provides
                management information."
 -- 1.3.6.1.2.1.47.3.2.2 --  ::= { entityGroups 2 }

entityMappingGroup OBJECT-GROUP
        OBJECTS {
                entLPPhysicalIndex,
                entAliasMappingIdentifier,
                entPhysicalChildIndex
        }
        STATUS      current
        DESCRIPTION
                "The collection of objects which are used to represent the
                associations between multiple logical entities, physical
                components, interfaces, and port identifiers for which a
                single agent provides management information."
 -- 1.3.6.1.2.1.47.3.2.3 --  ::= { entityGroups 3 }

entityGeneralGroup OBJECT-GROUP
        OBJECTS {
                entLastChangeTime
        }
        STATUS      current
        DESCRIPTION
                "The collection of objects which are used to represent
                general entity information for which a single agent provides
                management information."
 -- 1.3.6.1.2.1.47.3.2.4 --  ::= { entityGroups 4 }

entityNotificationsGroup NOTIFICATION-GROUP
        NOTIFICATIONS {
                entConfigChange
        }
        STATUS      current
        DESCRIPTION
                "The collection of notifications used to indicate Entity MIB
                data consistency and general status information."
 -- 1.3.6.1.2.1.47.3.2.5 --  ::= { entityGroups 5 }

entityPhysical2Group OBJECT-GROUP
```

```
        OBJECTS {
                entPhysicalHardwareRev,
                entPhysicalFirmwareRev,
                entPhysicalSoftwareRev,
                entPhysicalSerialNum,
                entPhysicalMfgName,
                entPhysicalModelName,
                entPhysicalAlias,
                entPhysicalAssetID,
                entPhysicalIsFRU
        }
        STATUS      current
        DESCRIPTION
                "The collection of objects which are used to represent
                physical system components, for which a single agent
                provides management information.  This group augments the
                objects contained in the entityPhysicalGroup."
 -- 1.3.6.1.2.1.47.3.2.6 --  ::= { entityGroups 6 }

entityLogical2Group OBJECT-GROUP
        OBJECTS {
                entLogicalDescr,
                entLogicalType,
                entLogicalTAddress,
                entLogicalTDomain,
                entLogicalContextEngineID,
                entLogicalContextName
        }
        STATUS      current
        DESCRIPTION
                "The collection of objects which are used to represent the
                list of logical entities for which a single SNMP entity
                provides management information."
 -- 1.3.6.1.2.1.47.3.2.7 --  ::= { entityGroups 7 }

END
```