        Definitions of System-Level Managed Objects for Applications

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.

Copyright Notice

Table of Contents

1. Abstract

   This memo defines a portion of the Management Information Base (MIB)
   for use with network management protocols in the Internet community.
   In particular, it describes a basic set of managed objects for fault,
   configuration and performance management of applications from a
   systems perspective. More specifically, the managed objects are
   restricted to information that can be determined from the system
   itself and which does not require special instrumentation within the
   applications to make the information available.

   This memo does not specify a standard for the Internet community.

2. The SNMPv2 Network Management Framework

   The SNMPv2 Network Management Framework consists of the following
   major components:

   o   RFC 1902 Structure of Management Information for Version
         2 of the Simple Network Management Protocol (SNMPv2) [2]

   o   RFC 1903 Textual Conventions for Version 2 of the Simple
         Network Management Protocol (SNMPv2) [3]

   o   RFC 1904 Conformance Statements for Version 2 of the
         Simple Network Management Protocol (SNMPv2) [4]

   o   RFC 1905 Protocol Operations for Version 2 of the Simple
         Network Management Protocol (SNMPv2) [5]

   o   RFC 1906 Transport Mappings for Version 2 of the Simple
         Network Management Protocol (SNMPv2) [6]

   o   RFC 1907 Management Information Base for Version 2 of the
         Simple Network Management Protocol (SNMPv2) [7]

   o   RFC 1908 Coexistence between Version 1 and Version 2 of
         the Internet-standard Network Management Framework [8]

   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) [1],
   defined in the Structure of Management Information (SMI) (See RFC

1902 [2]). 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 object descriptor, to refer to the
object type.

3. Overview

The primary purpose of computing technologies is the execution of
application software. These applications, typically specialized
collections of executables, files, and interprocess communications,
exist to solve business, scientific or other "problems". The
configuration, fault detection, performance monitoring and control of
application software across its life on a host computer is of great
economic importance. For the purposes of our work, we define
applications as one or more units of executable code and other
resources, installed on a single host system that a manager may think
of as a single object for management purposes.

The information described by the objects in the System Application
MIB support configuration, fault, and performance management; they
represent some of the basic attributes of application software from a
systems (non-application specific) perspective. The information
allows for the description of applications as collections of
executables and files installed and executing on a host computer.

This memo is concerned primarily with, and defines a model for,
application information resident on a host computer which can be
determined from the system itself, and not from the individual
applications. This system-level view of applications is designed to
provide information about software applications installed and running
on the host system without requiring modifications and code additions
to the applications themselves. This approach was taken to insure
ease and speed of implementation, while allowing room for future
growth.

4. Architecture for Application Management

In the area of application management it is fully acknowledged and
even expected that additional MIB modules will be defined over time
to provide an even greater level of detail regarding applications.
This MIB module presents the most general case: a set of management
objects for providing generic information about applications and
whose object values can be determined from the computer system itself
without requiring instrumentation within the application.

A finer-grained level of detail is planned for the future "appl MIB"
which will be a common set of management objects relating to generic
applications, but which require some type of instrumentation in the
application in order to be determined. Since the applmib MIB module
will provide a finer level of detail, any connection to the sysAppl
MIB should be made by having references from the more detailed appl
MIB back to the more generic sysAppl MIB. Likewise, as application-
specific MIB modules such as the WWW MIB, etc., are developed over
time, these more specific MIBs should reference back to the more
generic MIBs.

While this MIB module does not attempt to provide every detailed
piece of information for managing applications, it does provide a
basic systems-level view of the applications and their components on
a single host system.

5. The Structure of the MIB

The System Application MIB structure models application packages as a
whole, and also models the individual elements (files and
executables) which collectively form an application. The MIB is
structured to model information regarding installed application
packages and the elements which make up each application package. The
MIB also models activity information on applications (and in turn,
their components) that are running or have previously run on the host
system. In modeling applications and their elements, this MIB module
provides the necessary link for associating executing processes with
the applications of which they are a part.

The objects are arranged into the following groups:

        - System Application Installed Group
             - sysApplInstallPkgTable
             - sysApplInstallElmtTable

        - System Application Run Group
             - sysApplRunTable
             - sysApplPastRunTable
             - sysApplElmtRunTable
             - sysApplElmtPastRunTable
             - (scalars for restricting table sizes)

        - System Application Map Group
             - sysApplMapTable

As can be seen by the arrangement above, for each category, the MIB
first treats an application package as a whole, and then breaks down
the package to provide information about each of the elements
(executable and non-executable files) of the package.

5.1. System Application Installed Group

The System Application Installed group consists of two tables.
Through these two tables, administrators will be able to determine
which applications have been installed on a system and what their
constituent components are. The first table, the
sysApplInstallPkgTable, lists the application packages installed on a
particular host. The second, the sysApplInstallElmtTable, provides
information regarding the executables and non-executable files, or
elements, which collectively compose an application.

NOTE: This MIB is intended to work with applications that have been
installed on a particular host, where "installed" means that the
existence of the application and the association between an
application and its component files can be discovered without
requiring additional instrumentation of the application itself. This
may require that certain conventions be used, such as using a central
software installation mechanism or registry, when installing
application packages. For example, many UNIX systems utilize a
"pkgadd" utility to track installed application packages, while many
PC systems utilize a global registry.

5.2. System Application Run Group

This group models activity information for applications that have
been invoked and are either currently running, or have previously
run, on the host system. Likewise, the individual elements of an
invoked application are also modeled to show currently running
processes, and processes that have run in the past. This information
is modeled using two pairs of tables: a pair of tables for currently
running applications and past run applications, and a pair of tables
for the currently running elements and the past run elements. Seven
scalars are also defined to control the size of the past run tables.

5.2.1. sysApplRunTable and sysApplPastRunTable

The sysApplRunTable and the sysApplPastRunTable make up the first
pair of tables. The sysApplRunTable contains the application
instances which are currently running on the host. Each time an
application is invoked, a new entry is created in the sysApplRunTable
to provide information about that particular invocation of the
application. An entry will remain in this table until the

application instance terminates, at which time the entry will be
deleted from the sysApplRunTable and placed in the
sysApplPastRunTable.

The sysApplPastRunTable maintains a history of instances of
applications which have previously executed on the host. Entries to
this table are made when an invoked application from the
sysApplRunTable terminates; the table entry which represents the
application instance is removed from the SysApplRunTable and a
corresponding entry is added to the sysApplPastRunTable.

Because the sysApplPastRunTable will continuously grow as
applications are executed and terminate, two scalars are defined to
control the aging-out of table entries. The value of
sysApplPastRunMaxRows specifies the maximum number of entries the
table may contain, while the sysApplPastRunTblTimeLimit specifies the
maximum age of the table entries. Oldest entries are removed first.

It is important to note that the sysApplRunTable and
sysApplPastRunTable contain entries for each INVOCATION of an
application. A single application package might be invoked multiple
times; each invocation is properly recorded by a separate entry in
the sysApplRunTable.

In order to implement this group, the agent must be able to recognize
that an application has been invoked, and be able to determine when
that invocation terminates. This poses a complex problem since a
single application invocation may involve numerous processes, some of
which may be required to remain running throughout the duration of
the application, others which might come and go. The
sysApplInstallElmtRole columnar object in the sysApplInstallElmtTable
is meant to assist in this task by indicating which element is the
application's primary executable, which elements must be running in
order for the application to be running, which elements are dependent
on required elements, etc. See the description of
sysApplInstallElmtRole for more details.

5.2.2. sysApplElmtRunTable and sysApplElmtPastRunTable

While the sysApplRunTable and sysApplPastRunTable focus on
applications as a whole, the sysApplElmtRunTable and
sysApplElmtPastRunTable provide information regarding an
application's executable elements, (processes), which are either
currently executing or have executed in the past.

The sysApplElmtRunTable contains an entry for every process currently
running on the host. An entry is created in this table for each
process at the time it is started, and will remain in the table until

the process terminates. Note that in order to provide complete
information on the load on the system, this table lists EVERY running
process, not just those processes that are running as part of an
identified application. However, when processes terminate, only
information from entries corresponding to elements of an identified
application are moved to the sysApplElmtPastRunTable.

The sysApplElmtPastRunTable maintains a history of processes which
have previously executed on the host as part of an application. When
a process from the sysApplElmtRunTable terminates, the entry's
information is moved to this sysApplElmtPastRunTable provided that
the process was part of an identified application. If the process
cannot be associated with any 'parent' application, then it is simply
removed from the sysApplElmtRunTable. This allows for processes like
'ps' or 'grep' to show up in the sysApplElmtRunTable, (where they are
consuming resources and are thus "interesting"), but not in the
sysApplElmtPastRunTable.

Because the sysApplElmtPastRunTable will continuously grow as
processes are executed and terminate, two scalars are defined to
control the aging-out of table entries. The value of
sysApplElmtPastRunMaxRows specifies the maximum number of entries the
table may contain, while the sysApplElmtPastRunTblTimeLimit specifies
the maximum age of the table entries. Oldest entries are removed
first.

5.3. System Application Map Group

The System Application Map group contains a single table, the
sysApplMapTable, whose sole purpose is to provide a backwards mapping
for determining the invoked application, installed element, and
installed application package given a known process identification
number.

6. Definitions

```
SYSAPPL-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    Unsigned32, TimeTicks, Counter32, Gauge32
        FROM SNMPv2-SMI
    DateAndTime, TEXTUAL-CONVENTION
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    mib-2 FROM SNMPv2-SMI;
```

    -- System Application MIB

    sysApplMIB MODULE-IDENTITY
        LAST-UPDATED "9710200000Z"
        ORGANIZATION "IETF Applications MIB Working Group"
        CONTACT-INFO
               "Cheryl Krupczak (Editor, WG Advisor)
                Postal: Empire Technologies, Inc.
                541 Tenth Street NW
                Suite 169
                Atlanta, GA 30318
                USA
                Phone: (770) 384-0184
                Email: cheryl@empiretech.com

                Jon Saperia (WG Chair)
                Postal: BGS Systems, Inc.
                One First Avenue
                Waltham, MA 02254-9111
                USA
                Phone: (617) 891-0000
                Email: saperia@networks.bgs.com"
        DESCRIPTION
            "The MIB module defines management objects that model
            applications as collections of executables and files
            installed and executing on a host system. The MIB
            presents a system-level view of applications; i.e.,
            objects in this MIB are limited to those attributes
            that can typically be obtained from the system itself
            without adding special instrumentation to the applications."
        ::= { mib-2 54 }

    sysApplOBJ   OBJECT IDENTIFIER ::= { sysApplMIB 1 }
    sysApplInstalled   OBJECT IDENTIFIER ::= { sysApplOBJ 1 }
    sysApplRun   OBJECT IDENTIFIER ::= { sysApplOBJ 2 }
    sysApplMap   OBJECT IDENTIFIER ::= { sysApplOBJ 3 }
    sysApplNotifications   OBJECT IDENTIFIER ::= { sysApplMIB 2 }
    sysApplConformance   OBJECT IDENTIFIER ::= { sysApplMIB 3 }

    -- Textual Conventions

    RunState ::= TEXTUAL-CONVENTION
        STATUS   current
        DESCRIPTION
            "This TC describes the current execution state of
            a running application or process. The possible
            values are:

```
                running(1),
                runnable(2),  - waiting for a resource (CPU, etc.)
                waiting(3),   - waiting for an event
                exiting(4),
                other(5)      - other invalid state"
        SYNTAX      INTEGER {
                    running (1),
                    runnable (2), -- waiting for resource (CPU, etc.)
                    waiting (3),  -- waiting for event
                    exiting (4),
                    other (5)     -- other invalid state
                    }

  LongUtf8String ::= TEXTUAL-CONVENTION
          DISPLAY-HINT "1024a"
          STATUS  current
          DESCRIPTION
                  "To facilitate internationalization, this TC
                   represents information taken from the ISO/IEC IS
                   10646-1 character set, encoded as an octet string
                   using the UTF-8 character encoding scheme described
                   in RFC 2044 [10].  For strings in 7-bit US-ASCII,
                   there is no impact since the UTF-8 representation
                   is identical to the US-ASCII encoding."
          SYNTAX  OCTET STRING (SIZE (0..1024))

  Utf8String ::= TEXTUAL-CONVENTION
          DISPLAY-HINT "255a"
          STATUS  current
          DESCRIPTION
                  "To facilitate internationalization, this TC
                   represents information taken from the ISO/IEC IS
                   10646-1 character set, encoded as an octet string
                   using the UTF-8 character encoding scheme described
                   in RFC 2044 [10].  For strings in 7-bit US-ASCII,
                   there is no impact since the UTF-8 representation
                   is identical to the US-ASCII encoding."
          SYNTAX  OCTET STRING (SIZE (0..255))


     -- sysApplInstalled Group
     -- This group provides information about application packages
     -- that have been installed on the host computer.  The group
     -- contains two tables.  The first, the sysApplInstallPkgTable,
     -- describes the application packages, the second, the
     -- sysApplInstallElmtTable, describes the constituent elements
     -- (files and executables) which compose an application package.
```

```
    --
    -- In order to appear in this group, an application and its
    -- component files must be discoverable by the system itself,
    -- possibly through some type of software installation mechanism
    -- or registry.


    -- sysApplInstallPkgTable
    -- The system installed application packages table provides
    -- information on the software packages installed on a system.
    -- These packages may consist of many different files including
    -- executable and non-executable files.

    sysApplInstallPkgTable OBJECT-TYPE
        SYNTAX        SEQUENCE OF SysApplInstallPkgEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "The table listing the software application packages
            installed on a host computer. In order to appear in
            this table, it may be necessary for the application
            to be installed using some type of software
            installation mechanism or global registry so that its
            existence can be detected by the agent implementation."

        ::= { sysApplInstalled 1 }

    sysApplInstallPkgEntry OBJECT-TYPE
        SYNTAX        SysApplInstallPkgEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "The logical row describing an installed application
            package."
        INDEX     { sysApplInstallPkgIndex }
        ::= { sysApplInstallPkgTable 1 }

    SysApplInstallPkgEntry ::= SEQUENCE {
        sysApplInstallPkgIndex              Unsigned32,
        sysApplInstallPkgManufacturer       Utf8String,
        sysApplInstallPkgProductName        Utf8String,
        sysApplInstallPkgVersion            Utf8String,
        sysApplInstallPkgSerialNumber       Utf8String,
        sysApplInstallPkgDate               DateAndTime,
        sysApplInstallPkgLocation           LongUtf8String
    }

    sysApplInstallPkgIndex OBJECT-TYPE
        SYNTAX        Unsigned32 (1..'ffffffff'h)
```

```
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An integer used only for indexing purposes.
            Generally monotonically increasing from 1 as new
            applications are installed.

            The value for each installed application must
            remain constant at least from one re-initialization of
            the network management entity which implements this
            MIB module to the next re-initialization.

            The specific value is meaningful only within a given SNMP
            entity. A sysApplInstallPkgIndex value must not be re-used
            until the next agent entity restart in the event the
            installed application entry is deleted."
        ::= { sysApplInstallPkgEntry 1 }

    sysApplInstallPkgManufacturer OBJECT-TYPE
        SYNTAX      Utf8String
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The Manufacturer of the software application package."
        ::= { sysApplInstallPkgEntry 2 }

    sysApplInstallPkgProductName OBJECT-TYPE
        SYNTAX      Utf8String
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The name assigned to the software application package
            by the Manufacturer."
        ::= { sysApplInstallPkgEntry 3 }

    sysApplInstallPkgVersion OBJECT-TYPE
        SYNTAX      Utf8String
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The version number assigned to the application package
            by the manufacturer of the software."
        ::= { sysApplInstallPkgEntry 4 }

    sysApplInstallPkgSerialNumber OBJECT-TYPE
        SYNTAX      Utf8String
        MAX-ACCESS  read-only
        STATUS      current
```

           DESCRIPTION
               "The serial number of the software assigned by the
               manufacturer."
           ::= { sysApplInstallPkgEntry 5 }

    sysApplInstallPkgDate OBJECT-TYPE
        SYNTAX   DateAndTime
        MAX-ACCESS read-only
        STATUS   current
        DESCRIPTION
            "The date and time this software application was installed
            on the host."
        ::= { sysApplInstallPkgEntry 6 }

    sysApplInstallPkgLocation OBJECT-TYPE
        SYNTAX   LongUtf8String
        MAX-ACCESS read-only
        STATUS   current
        DESCRIPTION
            "The complete path name where the application package
            is installed. For example, the value would be
            '/opt/MyapplDir' if the application package was installed
            in the /opt/MyapplDir directory."
        ::= { sysApplInstallPkgEntry 7 }

    -- sysApplInstallElmtTable
    -- The table describing the individual application package
    -- elements (files and executables) installed on the host computer.

    sysApplInstallElmtTable OBJECT-TYPE
        SYNTAX   SEQUENCE OF SysApplInstallElmtEntry
        MAX-ACCESS not-accessible
        STATUS   current
        DESCRIPTION
            "This table details the individual application package
            elements (files and executables) which comprise the
            applications defined in the sysApplInstallPkg Table.
            Each entry in this table has an index to the
            sysApplInstallPkg table to identify the application
            package of which it is a part. As a result, there may
            be many entries in this table for each instance in the
            sysApplInstallPkg Table.

            Table entries are indexed by sysApplInstallPkgIndex,
            sysApplInstallElmtIndex to facilitate retrieval of
            all elements associated with a particular installed
            application package."

```
    ::= { sysApplInstalled 2 }

sysApplInstallElmtEntry OBJECT-TYPE
    SYNTAX       SysApplInstallElmtEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The logical row describing an element of an installed
        application.  The element may be an executable or
        non-executable file."
    INDEX    {sysApplInstallPkgIndex, sysApplInstallElmtIndex}
    ::= { sysApplInstallElmtTable 1 }

SysApplInstallElmtEntry ::= SEQUENCE {
    sysApplInstallElmtIndex              Unsigned32,
    sysApplInstallElmtName               Utf8String,
    sysApplInstallElmtType               INTEGER,
    sysApplInstallElmtDate               DateAndTime,
    sysApplInstallElmtPath               LongUtf8String,
    sysApplInstallElmtSizeHigh           Unsigned32,
    sysApplInstallElmtSizeLow            Unsigned32,
    sysApplInstallElmtRole               BITS,
    sysApplInstallElmtModifyDate         DateAndTime,
    sysApplInstallElmtCurSizeHigh        Unsigned32,
    sysApplInstallElmtCurSizeLow         Unsigned32

}

sysApplInstallElmtIndex OBJECT-TYPE
    SYNTAX       Unsigned32 (1..'ffffffff'h)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An arbitrary integer used for indexing.  The value
        of this index is unique among all rows in this table
        that exist or have existed since the last agent restart."
    ::= { sysApplInstallElmtEntry 1 }


sysApplInstallElmtName OBJECT-TYPE
    SYNTAX       Utf8String
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The name of this element which is contained in the
        application."
    ::= { sysApplInstallElmtEntry 2 }
```

```
    sysApplInstallElmtType OBJECT-TYPE
        SYNTAX        INTEGER {
                      unknown(1),
                      nonexecutable(2),
                      operatingSystem(3),  -- executable
                      deviceDriver(4),     -- executable
                      application(5)       -- executable
                      }
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The type of element that is part of the installed
            application."
        ::= { sysApplInstallElmtEntry 3 }

    sysApplInstallElmtDate OBJECT-TYPE
        SYNTAX        DateAndTime
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The date and time that this component was installed on
            the system."
        ::= { sysApplInstallElmtEntry 4 }

    sysApplInstallElmtPath OBJECT-TYPE
        SYNTAX        LongUtf8String
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The full directory path where this element is installed.
            For example, the value would be '/opt/EMPuma/bin' for an
            element installed in the directory '/opt/EMPuma/bin'.
            Most application packages include information about the
            elements contained in the package. In addition, elements
            are typically installed in sub-directories under the
            package installation directory.  In cases where the
            element path names are not included in the package
            information itself, the path can usually be determined
            by a simple search of the sub-directories.  If the
            element is not installed in that location and there is
            no other information available to the agent implementation,
            then the path is unknown and null is returned."
        ::= { sysApplInstallElmtEntry 5}

    sysApplInstallElmtSizeHigh OBJECT-TYPE
        SYNTAX        Unsigned32
        MAX-ACCESS    read-only
        STATUS        current
```

        DESCRIPTION
            "The installed file size in 2^32 byte blocks. This is
            the size of the file on disk immediately after installation.

            For example, for a file with a total size of 4,294,967,296
            bytes, this variable would have a value of 1; for a file
            with a total size of 4,294,967,295 bytes this variable
            would be 0."
        ::= { sysApplInstallElmtEntry 6 }

    sysApplInstallElmtSizeLow OBJECT-TYPE
        SYNTAX   Unsigned32
        MAX-ACCESS read-only
        STATUS   current
        DESCRIPTION
            "The installed file size modulo 2^32 bytes. This is
            the size of the file on disk immediately after installation.

            For example, for a file with a total size of 4,294,967,296
            bytes this variable would have a value of 0; for a file with
            a total size of 4,294,967,295 bytes this variable would be
            4,294,967,295."
        ::= { sysApplInstallElmtEntry 7 }

    sysApplInstallElmtRole OBJECT-TYPE
        SYNTAX   BITS {
                    executable(0),
                        -- An application may have one or
                        -- more executable elements. The rest of the
                        -- bits have no meaning if the element is not
                        -- executable.
                    exclusive(1),
                        -- Only one copy of an exclusive element may be
                        -- running per invocation of the running
                        -- application.
                    primary(2),
                        -- The primary executable. An application can
                        -- have one, and only one element that is designated
                        -- as the primary executable. The execution of
                        -- this element constitutes an invocation of
                        -- the application. This is used by the agent
                        -- implementation to determine the initiation of
                        -- an application. The primary executable must
                        -- remain running long enough for the agent
                        -- implementation to detect its presence.
                    required(3),
                        -- An application may have zero or more required
                        -- elements. All required elements must be running

```
                    -- in order for the application to be judged to be
                    -- running and healthy.
                dependent(4),
                    -- An application may have zero or more
                    -- dependent elements. Dependent elements may
                    -- not be running unless required elements are.
                unknown(5)
                    -- Default value for the case when an operator
                    -- has not yet assigned one of the other values.
                    -- When set, bits 1, 2, 3, and 4 have no meaning.
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "An operator assigned value used in the determination of
        application status. This value is used by the agent to
        determine both the mapping of started processes to the
        initiation of an application, as well as to allow for a
        determination of application health. The default value,
        unknown(5), is used when an operator has not yet assigned
        one of the other values.  If unknown(5) is set, bits
        1 - 4 have no meaning.  The possible values are:

                executable(0),
                    An application may have one or
                    more executable elements.  The rest of the
                    bits have no meaning if the element is not
                    executable.
                exclusive(1),
                    Only one copy of an exclusive element may be
                    running per invocation of the running
                    application.
                primary(2),
                    The primary executable.  An application can
                    have one, and only one element that is designated
                    as the primary executable.  The execution of
                    this element constitutes an invocation of
                    the application.  This is used by the agent
                    implementation to determine the initiation of
                    an application.  The primary executable must
                    remain running long enough for the agent
                    implementation to detect its presence.
                required(3),
                    An application may have zero or more required
                    elements. All required elements must be running
                    in order for the application to be judged to be
                    running and healthy.
                dependent(4),
```

                             An application may have zero or more
                             dependent elements. Dependent elements may
                             not be running unless required elements are.
                    unknown(5)
                         Default value for the case when an operator
                         has not yet assigned one of the other values.
                         When set, bits 1, 2, 3, and 4 have no meaning.

            sysApplInstallElmtRole is used by the agent implementation
            in determining the initiation of an application, the
            current state of a running application (see
            sysApplRunCurrentState), when an application invocation is
            no longer running, and the exit status of a terminated
            application invocation (see sysApplPastRunExitState)."

        DEFVAL { { unknown } }
        ::= { sysApplInstallElmtEntry 8 }

    sysApplInstallElmtModifyDate OBJECT-TYPE
        SYNTAX   DateAndTime
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
            "The date and time that this element was last modified.
            Modification of the sysApplInstallElmtRole columnar
            object does NOT constitute a modification of the element
            itself and should not affect the value of this object."
        ::= { sysApplInstallElmtEntry 9 }

    sysApplInstallElmtCurSizeHigh OBJECT-TYPE
        SYNTAX   Unsigned32
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
            "The current file size in 2^32 byte blocks.
            For example, for a file with a total size of 4,294,967,296
            bytes, this variable would have a value of 1; for a file
            with a total size of 4,294,967,295 bytes this variable
            would be 0."
        ::= { sysApplInstallElmtEntry 10 }

    sysApplInstallElmtCurSizeLow OBJECT-TYPE
        SYNTAX   Unsigned32
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
            "The current file size modulo 2^32 bytes.
            For example, for a file with a total size of 4,294,967,296

        bytes this variable would have a value of 0; for a file with
        a total size of 4,294,967,295 bytes this variable would be
        4,294,967,295."
      ::= { sysApplInstallElmtEntry 11 }

    -- sysApplRun Group
    -- This group models activity information for applications
    -- that have been invoked and are either currently running,
    -- or have previously run on the host system. Likewise,
    -- the individual elements of an invoked application are
    -- also modeled to show currently running processes, and
    -- processes that have run in the past.

    -- sysApplRunTable
    -- The sysApplRunTable contains the application instances
    -- which are currently running on the host. Since a single
    -- application might be invoked multiple times, an entry is
    -- added to this table for each INVOCATION of an application.
    -- The table is indexed by sysApplInstallPkgIndex, sysApplRunIndex
    -- to enable managers to easily locate all invocations of
    -- a particular application package.

    sysApplRunTable OBJECT-TYPE
        SYNTAX   SEQUENCE OF SysApplRunEntry
        MAX-ACCESS   not-accessible
        STATUS   current
        DESCRIPTION
            "The table describes the applications which are executing
            on the host. Each time an application is invoked,
            an entry is created in this table. When an application ends,
            the entry is removed from this table and a corresponding
                    entry is created in the SysApplPastRunTable.

            A new entry is created in this table whenever the agent
            implementation detects a new running process that is an
            installed application element whose sysApplInstallElmtRole
            designates it as being the application's primary executable
            (sysApplInstallElmtRole = primary(2)).

            The table is indexed by sysApplInstallPkgIndex,
            sysApplRunIndex to enable managers to easily locate all
            invocations of a particular application package."
        ::= { sysApplRun 1 }

    sysApplRunEntry OBJECT-TYPE
        SYNTAX   SysApplRunEntry

```
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The logical row describing an application which is
            currently running on this host."
        INDEX    { sysApplInstallPkgIndex, sysApplRunIndex }
        ::= { sysApplRunTable   1 }

    SysApplRunEntry ::= SEQUENCE {
        sysApplRunIndex                     Unsigned32,
        sysApplRunStarted                   DateAndTime,
        sysApplRunCurrentState              RunState
    }

    sysApplRunIndex OBJECT-TYPE
        SYNTAX      Unsigned32 (1..'ffffffff'h)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Part of the index for this table. An arbitrary
            integer used only for indexing purposes. Generally
            monotonically increasing from 1 as new applications are
            started on the host, it uniquely identifies application
            invocations.

            The numbering for this index increases by 1 for each
            INVOCATION of an application, regardless of which
            installed application package this entry represents a
            running instance of.

            An example of the indexing for a couple of entries is
            shown below.

                            :
                    sysApplRunStarted.17.14
                    sysApplRunStarted.17.63
                    sysApplRunStarted.18.13
                            :

            In this example, the agent has observed 12 application
            invocations when the application represented by entry 18
            in the sysApplInstallPkgTable is invoked.  The next
            invocation detected by the agent is an invocation of
            installed application package 17.  Some time later,
            installed application 17 is invoked a second time.

            NOTE: this index is not intended to reflect a real-time
            (wall clock time) ordering of application invocations;
```

           it is merely intended to uniquely identify running
           instances of applications. Although the
           sysApplInstallPkgIndex is included in the INDEX clause
           for this table, it serves only to ease searching of
           this table by installed application and does not
           contribute to uniquely identifying table entries."
        ::= { sysApplRunEntry 1 }

    sysApplRunStarted OBJECT-TYPE
        SYNTAX   DateAndTime
        MAX-ACCESS read-only
        STATUS   current
        DESCRIPTION
           "The date and time that the application was started."
        ::= { sysApplRunEntry 2 }

    sysApplRunCurrentState OBJECT-TYPE
        SYNTAX   RunState
        MAX-ACCESS read-only
        STATUS   current
        DESCRIPTION
           "The current state of the running application instance.
           The possible values are running(1), runnable(2) but waiting
           for a resource such as CPU, waiting(3) for an event,
           exiting(4), or other(5). This value is based on an evaluation
           of the running elements of this application instance (see
           sysApplElmRunState) and their Roles as defined by
           sysApplInstallElmtRole. An agent implementation may
           detect that an application instance is in the process of
           exiting if one or more of its REQUIRED elements are no
           longer running. Most agent implementations will wait until
           a second internal poll has been completed to give the
           system time to start REQUIRED elements before marking the
           application instance as exiting."
         ::= { sysApplRunEntry 3 }

    -- sysApplPastRunTable
    -- The sysApplPastRunTable provides a history of applications
    -- previously run on the host computer. Entries are removed from
    -- the sysApplRunTable and corresponding entries are added to this
    -- table when an application becomes inactive. Entries remain in
    -- this table until they are aged out when either the table size
    -- reaches a maximum as determined by the sysApplPastRunMaxRows,
    -- or when an entry has aged to exceed a time limit as set be
    -- sysApplPastRunTblTimeLimit.
    --
    -- When aging out entries, the oldest entry, as determined by

```
    -- the value of sysApplPastRunTimeEnded, will be removed first.

    sysApplPastRunTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF SysApplPastRunEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A history of the applications that have previously run
            on the host computer.  An entry's information is moved to
            this table from the sysApplRunTable when the invoked
            application represented by the entry ceases to be running.

            An agent implementation can determine that an application
            invocation is no longer running by evaluating the running
            elements of the application instance and their Roles as
            defined by sysApplInstallElmtRole.  Obviously, if there
            are no running elements for the application instance,
            then the application invocation is no longer running.
            If any one of the REQUIRED elements is not running,
            the application instance may be in the process of exiting.
            Most agent implementations will wait until a second internal
            poll has been completed to give the system time to either
            restart partial failures or to give all elements time to
            exit.  If, after the second poll, there are REQUIRED
            elements that are not running, then the application
            instance may be considered by the agent implementation
            to no longer be running.

            Entries remain in the sysApplPastRunTable until they
            are aged out when either the table size reaches a maximum
            as determined by the sysApplPastRunMaxRows, or when an entry
            has aged to exceed a time limit as set by
            sysApplPastRunTblTimeLimit.

            Entries in this table are indexed by sysApplInstallPkgIndex,
            sysApplPastRunIndex to facilitate retrieval of all past
            run invocations of a particular installed application."
        ::= { sysApplRun 2 }

    sysApplPastRunEntry OBJECT-TYPE
        SYNTAX      SysApplPastRunEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The logical row describing an invocation of an application
            which was previously run and has terminated.  The entry
            is basically copied from the sysApplRunTable when the
            application instance terminates.  Hence, the entry's
```

```
        value for sysApplPastRunIndex is the same as its value was
        for sysApplRunIndex."
    INDEX    { sysApplInstallPkgIndex, sysApplPastRunIndex }
    ::= { sysApplPastRunTable   1 }

SysApplPastRunEntry ::= SEQUENCE {
    sysApplPastRunIndex                 Unsigned32,
    sysApplPastRunStarted               DateAndTime,
    sysApplPastRunExitState             INTEGER,
    sysApplPastRunTimeEnded             DateAndTime
}

sysApplPastRunIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..'ffffffff'h)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Part of the index for this table. An integer
        matching the value of the removed sysApplRunIndex
        corresponding to this row."
    ::= { sysApplPastRunEntry 1 }

sysApplPastRunStarted OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The date and time that the application was started."
    ::= { sysApplPastRunEntry 2 }

sysApplPastRunExitState OBJECT-TYPE
    SYNTAX      INTEGER {
                complete (1), -- normal exit at sysApplRunTimeEnded
                failed (2),   -- abnormal exit
                other (3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The state of the application instance when it terminated.
       This value is based on an evaluation of the running elements
       of an application and their Roles as defined by
       sysApplInstallElmtRole.  An application instance is said to
       have exited in a COMPLETE state and its entry is removed
       from the sysApplRunTable and added to the sysApplPastRunTable
       when the agent detects that ALL elements of an application
       invocation are no longer running.  Most agent implementations
       will wait until a second internal poll has been completed to
```

```
            give the system time to either restart partial failures or
            to give all elements time to exit.  A failed state occurs if,
            after the second poll, any elements continue to run but
            one or more of the REQUIRED elements are no longer running.
            All other combinations MUST be defined as OTHER."
        ::= { sysApplPastRunEntry 3 }

    sysApplPastRunTimeEnded OBJECT-TYPE
        SYNTAX        DateAndTime
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The DateAndTime the application instance was determined
            to be no longer running."
        ::= { sysApplPastRunEntry 4 }

    -- sysApplElmtRunTable
    -- The sysApplElmtRunTable contains an entry for each process that
    -- is currently running on the host.  An entry is created in
    -- this table for each process at the time it is started, and will
    -- remain in the table until the process terminates.
    --
    -- The table is indexed by sysApplElmtRunInstallPkg,
    -- sysApplElmtRunInvocID, and sysApplElmtRunIndex to make it easy
    -- to locate all running elements of a particular invoked application
    -- which has been installed on the system.


    sysApplElmtRunTable OBJECT-TYPE
        SYNTAX        SEQUENCE OF SysApplElmtRunEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "The table describes the processes which are
            currently executing on the host system. Each entry
            represents a running process and is associated with
            the invoked application of which that process is a part, if
            possible.  This table contains an entry for every process
            currently running on the system, regardless of whether its
            'parent' application can be determined.  So, for example,
            processes like 'ps' and 'grep' will have entries though they
            are not associated with an installed application package.

            Because a running application may involve
            more than one executable, it is possible to have
            multiple entries in this table for each application.
            Entries are removed from this table when the process
            terminates.
```

          The table is indexed by sysApplElmtRunInstallPkg,
          sysApplElmtRunInvocID, and sysApplElmtRunIndex to
          facilitate the retrieval of all running elements of a
          particular invoked application which has been installed on
          the system."
      ::= { sysApplRun 3 }

   sysApplElmtRunEntry OBJECT-TYPE
       SYNTAX   SysApplElmtRunEntry
       MAX-ACCESS not-accessible
       STATUS   current
       DESCRIPTION
           "The logical row describing a process currently
           running on this host. When possible, the entry is
           associated with the invoked application of which it
           is a part."
       INDEX   { sysApplElmtRunInstallPkg, sysApplElmtRunInvocID,
                  sysApplElmtRunIndex }
       ::= { sysApplElmtRunTable   1 }

   SysApplElmtRunEntry ::= SEQUENCE {
       sysApplElmtRunInstallPkg   Unsigned32,
       sysApplElmtRunInvocID   Unsigned32,
       sysApplElmtRunIndex   Unsigned32,
       sysApplElmtRunInstallID   Unsigned32,
       sysApplElmtRunTimeStarted   DateAndTime,
       sysApplElmtRunState   RunState,
       sysApplElmtRunName   LongUtf8String,
       sysApplElmtRunParameters   Utf8String,
       sysApplElmtRunCPU   TimeTicks,
       sysApplElmtRunMemory   Gauge32,
       sysApplElmtRunNumFiles   Gauge32,
       sysApplElmtRunUser   Utf8String
   }

   sysApplElmtRunInstallPkg OBJECT-TYPE
       SYNTAX   Unsigned32 (0..'ffffffff'h)
       MAX-ACCESS not-accessible
       STATUS   current
       DESCRIPTION
           "Part of the index for this table, this value
           identifies the installed software package for
           the application of which this process is a part.
           Provided that the process's 'parent' application can be
           determined, the value of this object is the same
           value as the sysApplInstallPkgIndex for the
           entry in the sysApplInstallPkgTable that corresponds
           to the installed application of which this process

                is a part.

                If, however, the 'parent' application cannot be
                determined, (for example the process is not part
                of a particular installed application), the value
                for this object is then '0', signifying that this
                process cannot be related back to an application,
                and in turn, an installed software package."
           ::= { sysApplElmtRunEntry 1 }

       sysApplElmtRunInvocID OBJECT-TYPE
           SYNTAX   Unsigned32 (0..'ffffffff'h)
           MAX-ACCESS   not-accessible
           STATUS   current
           DESCRIPTION
                "Part of the index for this table, this value
                identifies the invocation of an application of which
                this process is a part. Provided that the 'parent'
                application can be determined, the value of this object
                is the same value as the sysApplRunIndex for the
                corresponding application invocation in the
                sysApplRunTable.

                If, however, the 'parent' application cannot be
                determined, the value for this object is then '0',
                signifying that this process cannot be related back
                to an invocation of an application in the
                sysApplRunTable."
           ::= { sysApplElmtRunEntry 2 }

       sysApplElmtRunIndex OBJECT-TYPE
           SYNTAX   Unsigned32 (0..'ffffffff'h)
           MAX-ACCESS   not-accessible
           STATUS   current
           DESCRIPTION
                "Part of the index for this table. A unique value
                for each process running on the host. Wherever
                possible, this should be the system's native, unique
                identification number."
           ::= { sysApplElmtRunEntry 3 }

       sysApplElmtRunInstallID OBJECT-TYPE
           SYNTAX   Unsigned32 (0..'ffffffff'h)
           MAX-ACCESS   read-only
           STATUS   current
           DESCRIPTION
                "The index into the sysApplInstallElmtTable. The

          value of this object is the same value as the
          sysApplInstallElmtIndex for the application element
          of which this entry represents a running instance.
          If this process cannot be associated with an installed
          executable, the value should be '0'."
      ::= { sysApplElmtRunEntry 4 }

  sysApplElmtRunTimeStarted OBJECT-TYPE
      SYNTAX   DateAndTime
      MAX-ACCESS read-only
      STATUS   current
      DESCRIPTION
          "The time the process was started."
      ::= { sysApplElmtRunEntry 5 }

  sysApplElmtRunState OBJECT-TYPE
      SYNTAX   RunState
      MAX-ACCESS read-only
      STATUS   current
      DESCRIPTION
          "The current state of the running process. The
          possible values are running(1), runnable(2) but waiting
          for a resource such as CPU, waiting(3) for an event,
          exiting(4), or other(5)."
      ::= { sysApplElmtRunEntry 6 }

  sysApplElmtRunName OBJECT-TYPE
      SYNTAX   LongUtf8String
      MAX-ACCESS read-only
      STATUS   current
      DESCRIPTION
          "The full path and filename of the process.
          For example, '/opt/MYYpkg/bin/myyproc' would
          be returned for process 'myyproc' whose execution
          path is '/opt/MYYpkg/bin/myyproc'."
      ::= { sysApplElmtRunEntry 7 }

  sysApplElmtRunParameters OBJECT-TYPE
      SYNTAX   Utf8String
      MAX-ACCESS read-only
      STATUS   current
      DESCRIPTION
          "The starting parameters for the process."
   ::= { sysApplElmtRunEntry 8 }

  sysApplElmtRunCPU OBJECT-TYPE
      SYNTAX   TimeTicks
      MAX-ACCESS read-only

```
        STATUS        current
        DESCRIPTION
             "The number of centi-seconds of the total system's
             CPU resources consumed by this process.  Note that
             on a multi-processor system, this value may
             have been incremented by more than one centi-second
             in one centi-second of real (wall clock) time."
        ::= { sysApplElmtRunEntry 9 }

    sysApplElmtRunMemory OBJECT-TYPE
        SYNTAX        Gauge32
        UNITS         "Kbytes"
        MAX-ACCESS  read-only
        STATUS        current
        DESCRIPTION
             "The total amount of real system memory measured in
             Kbytes currently allocated to this process."

        ::= { sysApplElmtRunEntry 10 }

    sysApplElmtRunNumFiles OBJECT-TYPE
        SYNTAX        Gauge32
        MAX-ACCESS  read-only
        STATUS        current
        DESCRIPTION
             "The number of regular files currently open by the
             process.  Transport connections (sockets)
             should NOT be included in the calculation of
             this value, nor should operating system specific
             special file types."
        ::= { sysApplElmtRunEntry 11 }

    sysApplElmtRunUser OBJECT-TYPE
        SYNTAX        Utf8String
        MAX-ACCESS  read-only
        STATUS        current
        DESCRIPTION
             "The process owner's login name (e.g. root)."
        ::= { sysApplElmtRunEntry 12 }

    -- sysApplElmtPastRunTable
    -- The sysApplElmtPastRunTable maintains a history of
    -- processes which have previously executed on
    -- the host as part of an application. Upon termination
    -- of a process, the entry representing the process is removed from
    -- the sysApplElmtRunTable and a corresponding entry is created in
    -- this table provided that the process was part of an
    -- identifiable application.  If the process could not be associated
```

```
       -- with an invoked application, no corresponding entry is created.
       -- Hence, whereas the sysApplElmtRunTable contains an entry for
       -- every process currently executing on the system, the
       -- sysApplElmtPastRunTable only contains entries for processes
       -- that previously executed as part of an invoked application.
       --
       -- Entries remain in this table until they are aged out when
       -- either the number of entries in the table reaches a
       -- maximum as determined by sysApplElmtPastRunMaxRows, or
       -- when an entry has aged to exceed a time limit as set by
       -- sysApplElmtPastRunTblTimeLimit.  When aging out entries,
       -- the oldest entry, as determined by the value of
       -- sysApplElmtPastRunTimeEnded, will be removed first.
       --
       -- The table is indexed by sysApplInstallPkgIndex (from the
       -- sysApplInstallPkgTable), sysApplElmtPastRunInvocID, and
       -- sysApplElmtPastRunIndex to make it easy to locate all
       -- previously executed processes of a particular invoked application
       -- that has been installed on the system.


       sysApplElmtPastRunTable OBJECT-TYPE
           SYNTAX        SEQUENCE OF SysApplElmtPastRunEntry
           MAX-ACCESS  not-accessible
           STATUS        current
           DESCRIPTION
               "The table describes the processes which have previously
               executed on the host system as part of an application.
               Each entry represents a process which has previously
               executed and is associated with the invoked application
               of which it was a part.  Because an invoked application
               may involve more than one executable, it is possible
               to have multiple entries in this table for
               each application invocation. Entries are added
               to this table when the corresponding process in the
               sysApplElmtRun Table terminates.

               Entries remain in this table until they are aged out when
               either the number of entries in the table reaches a
               maximum as determined by sysApplElmtPastRunMaxRows, or
               when an entry has aged to exceed a time limit as set by
               sysApplElmtPastRunTblTimeLimit.  When aging out entries,
               the oldest entry, as determined by the value of
               sysApplElmtPastRunTimeEnded, will be removed first.

               The table is indexed by sysApplInstallPkgIndex (from the
               sysApplInstallPkgTable), sysApplElmtPastRunInvocID,
               and sysApplElmtPastRunIndex to make it easy to locate all
```

          previously executed processes of a particular invoked
          application that has been installed on the system."
        ::= { sysApplRun 4 }

    sysApplElmtPastRunEntry OBJECT-TYPE
        SYNTAX   SysApplElmtPastRunEntry
        MAX-ACCESS   not-accessible
        STATUS   current
        DESCRIPTION
            "The logical row describing a process which was
            previously executed on this host as part of an
            installed application. The entry is basically copied
            from the sysApplElmtRunTable when the process
            terminates. Hence, the entry's value for
            sysApplElmtPastRunIndex is the same as its value
            was for sysApplElmtRunIndex. Note carefully: only those
            processes which could be associated with an
            identified application are included in this table."
        INDEX   { sysApplInstallPkgIndex, sysApplElmtPastRunInvocID,
                   sysApplElmtPastRunIndex }
        ::= { sysApplElmtPastRunTable   1 }

    SysApplElmtPastRunEntry ::= SEQUENCE {
        sysApplElmtPastRunInvocID   Unsigned32,
        sysApplElmtPastRunIndex   Unsigned32,
        sysApplElmtPastRunInstallID   Unsigned32,
        sysApplElmtPastRunTimeStarted   DateAndTime,
        sysApplElmtPastRunTimeEnded   DateAndTime,
        sysApplElmtPastRunName   LongUtf8String,
        sysApplElmtPastRunParameters   Utf8String,
        sysApplElmtPastRunCPU   TimeTicks,
        sysApplElmtPastRunMemory   Unsigned32,
        sysApplElmtPastRunNumFiles   Unsigned32,
        sysApplElmtPastRunUser   Utf8String
    }

    sysApplElmtPastRunInvocID OBJECT-TYPE
        SYNTAX   Unsigned32 (1..'ffffffff'h)
        MAX-ACCESS   not-accessible
        STATUS   current
        DESCRIPTION
            "Part of the index for this table, this value
            identifies the invocation of an application of which
            the process represented by this entry was a part.
            The value of this object is the same value as the
            sysApplRunIndex for the corresponding application
            invocation in the sysApplRunTable. If the invoked
            application as a whole has terminated, it will be the

```
                    same as the sysApplPastRunIndex."
                ::= { sysApplElmtPastRunEntry 1 }


        sysApplElmtPastRunIndex OBJECT-TYPE
            SYNTAX        Unsigned32 (0..'ffffffff'h)
            MAX-ACCESS    not-accessible
            STATUS        current
            DESCRIPTION
                "Part of the index for this table. An integer
                assigned by the agent equal to the corresponding
                sysApplElmtRunIndex which was removed from the
                sysApplElmtRunTable and moved to this table
                when the element terminated.

                Note: entries in this table are indexed by
                sysApplElmtPastRunInvocID, sysApplElmtPastRunIndex.
                The possibility exists, though unlikely, of a
                collision occurring by a new entry which was run
                by the same invoked application (InvocID), and
                was assigned the same process identification number
                (ElmtRunIndex) as an element which was previously
                run by the same invoked application.

                Should this situation occur, the new entry replaces
                the old entry.

                See Section: 'Implementation Issues -
                sysApplElmtPastRunTable Entry Collisions' for the
                conditions that would have to occur in order for a
                collision to occur."
            ::= { sysApplElmtPastRunEntry 2 }


        sysApplElmtPastRunInstallID OBJECT-TYPE
            SYNTAX        Unsigned32 (1..'ffffffff'h)
            MAX-ACCESS    read-only
            STATUS        current
            DESCRIPTION
                "The index into the installed element table. The
                value of this object is the same value as the
                sysApplInstallElmtIndex for the application element
                of which this entry represents a previously executed
                process."
            ::= { sysApplElmtPastRunEntry 3 }

        sysApplElmtPastRunTimeStarted OBJECT-TYPE
            SYNTAX        DateAndTime
            MAX-ACCESS    read-only
```

```
         STATUS        current
         DESCRIPTION
             "The time the process was started."
         ::= { sysApplElmtPastRunEntry 4 }

     sysApplElmtPastRunTimeEnded OBJECT-TYPE
         SYNTAX        DateAndTime
         MAX-ACCESS    read-only
         STATUS        current
         DESCRIPTION
             "The time the process ended."
         ::= { sysApplElmtPastRunEntry 5 }

     sysApplElmtPastRunName OBJECT-TYPE
         SYNTAX        LongUtf8String
         MAX-ACCESS    read-only
         STATUS        current
         DESCRIPTION
             "The full path and filename of the process.
             For example, '/opt/MYYpkg/bin/myyproc' would
             be returned for process 'myyproc' whose execution
             path was '/opt/MYYpkg/bin/myyproc'."
         ::= { sysApplElmtPastRunEntry 6 }

     sysApplElmtPastRunParameters OBJECT-TYPE
         SYNTAX        Utf8String
         MAX-ACCESS    read-only
         STATUS        current
         DESCRIPTION
             "The starting parameters for the process."
         ::= { sysApplElmtPastRunEntry 7 }

     sysApplElmtPastRunCPU OBJECT-TYPE
         SYNTAX        TimeTicks
         MAX-ACCESS    read-only
         STATUS        current
         DESCRIPTION
              "The last known number of centi-seconds of the total
               system's CPU resources consumed by this process.
               Note that on a multi-processor system, this value may
               increment by more than one centi-second in one
               centi-second of real (wall clock) time."
         ::= { sysApplElmtPastRunEntry 8 }

     sysApplElmtPastRunMemory OBJECT-TYPE
         SYNTAX        Unsigned32 (0..'ffffffff'h)
         UNITS         "Kbytes"
         MAX-ACCESS    read-only
```

```
    STATUS        current
    DESCRIPTION
        "The last known total amount of real system memory
        measured in Kbytes allocated to this process before it
        terminated."
    ::= { sysApplElmtPastRunEntry 9 }

sysApplElmtPastRunNumFiles OBJECT-TYPE
    SYNTAX        Unsigned32 (0..'ffffffff'h)
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "The last known number of files open by the
        process before it terminated.  Transport
        connections (sockets) should NOT be included in
        the calculation of this value."
    ::= { sysApplElmtPastRunEntry 10 }

sysApplElmtPastRunUser OBJECT-TYPE
    SYNTAX        Utf8String
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "The process owner's login name (e.g. root)."
    ::= { sysApplElmtPastRunEntry 11 }


-- Additional Scalar objects to control table sizes

sysApplPastRunMaxRows OBJECT-TYPE
    SYNTAX        Unsigned32 (0..'ffffffff'h)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The maximum number of entries allowed in the
        sysApplPastRunTable.  Once the number of rows in
        the sysApplPastRunTable reaches this value, the
        management subsystem will remove the oldest entry
        in the table to make room for the new entry to be added.
        Entries will be removed on the basis of oldest
        sysApplPastRunTimeEnded value first.

        This object may be used to control the amount of
        system resources that can used for sysApplPastRunTable
        entries. A conforming implementation should attempt
        to support the default value, however, a lesser value
        may be necessary due to implementation-dependent issues
        and resource availability."
```

```
        DEFVAL        { 500 }
        ::= { sysApplRun 5 }


    sysApplPastRunTableRemItems OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS  read-only
        STATUS        current
        DESCRIPTION
            "A counter of the number of entries removed from
            the sysApplPastRunTable because of table size limitations
            as set in sysApplPastRunMaxRows.  This counter is the
            number of entries the management subsystem has had to
            remove in order to make room for new entries (so as not
            to exceed the limit set by sysApplPastRunMaxRows) since
            the last initialization of the management subsystem."
        ::= { sysApplRun 6 }


    sysApplPastRunTblTimeLimit OBJECT-TYPE
        SYNTAX        Unsigned32 (0..'ffffffff'h)
        UNITS         "seconds"
        MAX-ACCESS  read-write
        STATUS        current
        DESCRIPTION
            "The maximum time in seconds which an entry in the
             sysApplPastRunTable may exist before it is removed.
             Any entry that is older than this value will be
             removed (aged out) from the table.

             Note that an entry may be aged out prior to reaching
             this time limit if it is the oldest entry in the
             table and must be removed to make space for a new
             entry so as to not exceed sysApplPastRunMaxRows."
        DEFVAL        { 7200 }
        ::= { sysApplRun 7 }


    sysApplElemPastRunMaxRows OBJECT-TYPE
        SYNTAX        Unsigned32 (0..'ffffffff'h)
        MAX-ACCESS  read-write
        STATUS        current
        DESCRIPTION
            "The maximum number of entries allowed in the
            sysApplElmtPastRunTable.  Once the number of rows in
            the sysApplElmtPastRunTable reaches this value,
            the management subsystem will remove the oldest entry
            to make room for the new entry to be added.  Entries
            will be removed on the basis of oldest
            sysApplElmtPastRunTimeEnded value first.
```

           This object may be used to control the amount of
           system resources that can used for sysApplElemPastRunTable
           entries. A conforming implementation should attempt
           to support the default value, however, a lesser value
           may be necessary due to implementation-dependent issues
           and resource availability."
      DEFVAL   { 500 }
      ::= { sysApplRun 8 }

   sysApplElemPastRunTableRemItems OBJECT-TYPE
      SYNTAX   Counter32
      MAX-ACCESS   read-only
      STATUS   current
      DESCRIPTION
          "A counter of the number of entries removed from the
          sysApplElemPastRunTable because of table size limitations
          as set in sysApplElemPastRunMaxRows. This counter is the
          number of entries the management subsystem has had to
          remove in order to make room for new entries (so as not
          to exceed the limit set by sysApplElemPastRunMaxRows) since
          the last initialization of the management subsystem."
      ::= { sysApplRun 9 }

   sysApplElemPastRunTblTimeLimit OBJECT-TYPE
      SYNTAX   Unsigned32 (0..'ffffffff'h)
      UNITS   "seconds"
      MAX-ACCESS   read-write
      STATUS   current
      DESCRIPTION
          "The maximum time in seconds which an entry in the
           sysApplElemPastRunTable may exist before it is removed.
           Any entry that is older than this value will be
           removed (aged out) from the table.

           Note that an entry may be aged out prior to reaching
           this time limit if it is the oldest entry in the
           table and must be removed to make space for a new
           entry so as to not exceed sysApplElemPastRunMaxRows."
      DEFVAL   { 7200 }
      ::= { sysApplRun 10 }

   sysApplAgentPollInterval OBJECT-TYPE
      SYNTAX   Unsigned32 (0..'ffffffff'h)
      UNITS   "seconds"
      MAX-ACCESS   read-write
      STATUS   current
      DESCRIPTION
          "The minimum interval in seconds that the management

                 subsystem implementing this MIB will poll the status
                 of the managed resources. Because of the non-trivial
                 effort involved in polling the managed resources,
                 and because the method for obtaining the status of
                 the managed resources is implementation-dependent,
                 a conformant implementation may chose a lower bound
                 greater than 0.

                 A value of 0 indicates that there is no delay
                 in the passing of information from the managed
                 resources to the agent."
          DEFVAL   { 60 }
           ::= { sysApplRun 11 }

    -- sysApplMap Group
    -- This group contains a table, the sysApplMapTable,
    -- whose sole purpose is to provide a 'backwards'
    -- mapping so that, given a known sysApplElmtRunIndex
    -- (process identification number), the corresponding invoked
    -- application (sysApplRunIndex), installed element
    -- (sysApplInstallElmtIndex), and installed application
    -- package (sysApplInstallPkgIndex) can be quickly determined.
    --
    -- The table will contain one entry for each process
    -- currently running on the system.
    --
    -- A backwards mapping is extremely useful since the tables
    -- in this MIB module are typically indexed with the
    -- installed application package (sysApplInstallPkgIndex)
    -- as the primary key, and on down as required by the
    -- specific table, with the process ID number (sysApplElmtRunIndex)
    -- being the least significant key.
    --
    -- It is expected that management applications will use
    -- this mapping table by doing a 'GetNext' operation with
    -- the known process ID number (sysApplElmtRunIndex) as the partial
    -- instance identifier. Assuming that there is an entry for
    -- the process, the result should return a single columnar value,
    -- the sysApplMapInstallPkgIndex, with the sysApplElmtRunIndex,
    -- sysApplRunIndex, and sysApplInstallElmtIndex contained in the
    -- instance identifier for the returned MIB object value.
    --
    -- NOTE: if the process can not be associated back to an
    -- invoked application installed on the system, then the
    -- value returned for the columnar value sysApplMapInstallPkgIndex
    -- will be '0' and the instance portion of the object-identifier
    -- will be the process ID number (sysApplElmtRunIndex) followed

        -- by 0.0.

    sysApplMapTable OBJECT-TYPE
        SYNTAX   SEQUENCE OF SysApplMapEntry
        MAX-ACCESS   not-accessible
        STATUS   current
        DESCRIPTION
            "The sole purpose of this table is to provide a
            'backwards' mapping so that, given a known
            sysApplElmtRunIndex (process identification number),
            the corresponding invoked application (sysApplRunIndex),
            installed element (sysApplInstallElmtIndex), and
            installed application package (sysApplInstallPkgIndex)
            can be quickly determined.

            This table will contain one entry for each process
            that is currently executing on the system.

            It is expected that management applications will use
            this mapping table by doing a 'GetNext' operation with
            the known process ID number (sysApplElmtRunIndex) as the
            partial instance identifier. Assuming that there is an
            entry for the process, the result should return a single
            columnar value, the sysApplMapInstallPkgIndex, with the
            sysApplElmtRunIndex, sysApplRunIndex, and
            sysApplInstallElmtIndex contained in the instance identifier
            for the returned MIB object value.

            NOTE: if the process can not be associated back to an
            invoked application installed on the system, then the
            value returned for the columnar value
            sysApplMapInstallPkgIndex will be '0' and the instance
            portion of the object-identifier will be the process ID
            number (sysApplElmtRunIndex) followed by 0.0."
        ::= { sysApplMap 1 }

    sysApplMapEntry OBJECT-TYPE
        SYNTAX   SysApplMapEntry
        MAX-ACCESS   not-accessible
        STATUS   current
        DESCRIPTION
            "A logical row representing a process currently running
             on the system. This entry provides the index mapping from
             process identifier, back to the invoked application,
             installed element, and finally, the installed application
             package. The entry includes only one accessible columnar
             object, the sysApplMapInstallPkgIndex, but the
             invoked application and installed element can be

```
        determined from the instance identifier since they form
        part of the index clause."
    INDEX  { sysApplElmtRunIndex, sysApplElmtRunInvocID,
            sysApplMapInstallElmtIndex }
    ::= { sysApplMapTable 1 }

SysApplMapEntry ::= SEQUENCE {
    sysApplMapInstallElmtIndex    Unsigned32,
    sysApplMapInstallPkgIndex     Unsigned32
}

sysApplMapInstallElmtIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0..'ffffffff'h)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index into the sysApplInstallElmtTable. The
        value of this object is the same value as the
        sysApplInstallElmtIndex for the application element
        of which this entry represents a running instance.
        If this process cannot be associated to an installed
        executable, the value should be '0'."
    ::= { sysApplMapEntry 1 }

sysApplMapInstallPkgIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0..'ffffffff'h)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of this object identifies the installed
        software package for the application of which this
        process is a part.  Provided that the process's 'parent'
        application can be determined, the value of this object
        is the same value as the sysApplInstallPkgIndex for the
        entry in the sysApplInstallPkgTable that corresponds
        to the installed application of which this process
        is a part.

        If, however, the 'parent' application cannot be
        determined, (for example the process is not part
        of a particular installed application), the value
        for this object is then '0', signifying that this
        process cannot be related back to an application,
        and in turn, an installed software package."
    ::= { sysApplMapEntry 2 }


-- Conformance Macros
```

```
   sysApplMIBCompliances  OBJECT IDENTIFIER ::= { sysApplConformance 1 }
   sysApplMIBGroups       OBJECT IDENTIFIER ::= { sysApplConformance 2 }

   sysApplMIBCompliance MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
           "Describes the requirements for conformance to
           the System Application MIB"
       MODULE  -- this module
           MANDATORY-GROUPS { sysApplInstalledGroup,
                              sysApplRunGroup, sysApplMapGroup }
       ::= { sysApplMIBCompliances 1 }

   sysApplInstalledGroup OBJECT-GROUP
       OBJECTS { sysApplInstallPkgManufacturer,
                 sysApplInstallPkgProductName,
                 sysApplInstallPkgVersion,
                 sysApplInstallPkgSerialNumber,
                 sysApplInstallPkgDate,
                 sysApplInstallPkgLocation,
                 sysApplInstallElmtName,
                 sysApplInstallElmtType,
                 sysApplInstallElmtDate,
                 sysApplInstallElmtPath,
                 sysApplInstallElmtSizeHigh,
                 sysApplInstallElmtSizeLow,
                 sysApplInstallElmtRole,
                 sysApplInstallElmtModifyDate,
                 sysApplInstallElmtCurSizeHigh,
                 sysApplInstallElmtCurSizeLow }
       STATUS  current
       DESCRIPTION
           "The system application installed group contains
           information about applications and their constituent
           components which have been installed on the host system."
       ::= { sysApplMIBGroups 1 }

   sysApplRunGroup OBJECT-GROUP
       OBJECTS { sysApplRunStarted,
                 sysApplRunCurrentState,
                 sysApplPastRunStarted,
                 sysApplPastRunExitState,
                 sysApplPastRunTimeEnded,
                 sysApplElmtRunInstallID,
                 sysApplElmtRunTimeStarted,
                 sysApplElmtRunState,
                 sysApplElmtRunName,
                 sysApplElmtRunParameters,
```

```
                    sysApplElmtRunCPU,
                    sysApplElmtRunMemory,
                    sysApplElmtRunNumFiles,
                    sysApplElmtRunUser,
                    sysApplElmtPastRunInstallID,
                    sysApplElmtPastRunTimeStarted,
                    sysApplElmtPastRunTimeEnded,
                    sysApplElmtPastRunName,
                    sysApplElmtPastRunParameters,
                    sysApplElmtPastRunCPU,
                    sysApplElmtPastRunMemory,
                    sysApplElmtPastRunNumFiles,
                    sysApplElmtPastRunUser,
                    sysApplPastRunMaxRows,
                    sysApplPastRunTableRemItems,
                    sysApplPastRunTblTimeLimit,
                    sysApplElemPastRunMaxRows,
                    sysApplElemPastRunTableRemItems,
                    sysApplElemPastRunTblTimeLimit,
                    sysApplAgentPollInterval }
        STATUS  current
        DESCRIPTION
            "The system application run group contains information
            about applications and associated elements which have
            run or are currently running on the host system."
        ::= { sysApplMIBGroups 2 }

    sysApplMapGroup OBJECT-GROUP
        OBJECTS { sysApplMapInstallPkgIndex }
        STATUS  current
        DESCRIPTION
            "The Map Group contains a single table, sysApplMapTable,
            that provides a backwards mapping for determining the
            invoked application, installed element, and installed
            application package given a known process identification
            number."
    ::= { sysApplMIBGroups 3 }


    END
```

7. Implementation Issues

   This section discusses implementation issues that are important for
   both an agent developer, and a management application developer or
   user to understand with regards to this MIB module. Although this
   section does not attempt to prescribe a particular implementation
   strategy, it does attempt to recognize some of the real world
   limitations that could effect an implementation of this MIB module.

7.1. Implementation with Polling Agents

   Implementations of the System Application MIB on popular operating
   systems might require some considerable processing power to obtain
   status information from the managed resources. It might also be
   difficult to determine when an application or a process starts or
   finishes. Implementors of this MIB might therefore choose an
   implementation approach where the agent polls the managed resources
   at regular intervals. The information retrieved by every poll is used
   to update a cached version of this MIB maintained inside of the
   agent. SNMP request are processed based on the information found in
   this MIB cache.

   A scalar sysApplAgentPollInterval is defined to give the manager
   control over the polling frequency. There is a trade- off between the
   amount of resources consumed during every poll to update the MIB
   cache, and the accuracy of the information provided by the System
   Application MIB agent. A default value of 60 seconds is defined to
   keep the processing overhead low, while providing usable information
   for long-lived processes. A manager is expected to adjust this value
   if more accurate information about short-lived applications or
   processes is needed, or if the amount of resources consumed by the
   agent is too high.

7.2. sysApplElmtPastRunTable Entry Collisions

   The sysApplElmtPastRunTable maintains a history of processes which
   have previously executed on the host as part of an application.
   Information is moved from the sysApplElmtRunTable to this PastRun
   table when the process represented by the entry terminates.

   The sysApplElmtPastRunTable is indexed by the tuple,
   (sysApplElmtPastRunInvocID, sysApplElmtPastRunIndex), where the first
   part identifies the application invocation of which the process was a
   part, and the second part identifies the process itself.

   Recall that the sysApplElmtRunIndex represents the system's unique
   identification number assigned to a running process and that this
   value is mapped to sysApplElmtPastRunIndex when the process

terminates and the entry's information is moved from the
sysApplElmtRunTable to the sysApplElmtPastRunTable. Many systems
re-use process ID numbers which are no longer assigned to running
processes; typically, the process numbers wrap and the next available
process number is used.

It is therefore possible for two entries in the sysApplElmtPastRun
Table to have the same value for sysApplElmtPastRunIndex. For this
reason, entries in the ElmtPastRun table are indexed by the tuple
sysApplElmtPastRunInvocID, sysApplElmtPastRunIndex to reduce the
chance of a collision by two past run elements with the same
sysApplElmtPastRunIndex.

However, it is still possible, though unlikely, for a collision to
occur if the following happens:

1)   the invoked application (identified by InvocID), has an
      element which runs, terminates, and is moved into the
      sysApplElmtPastRun table (index: InvocID, RunIndex)

2)   the numbers used for the system's process identification
      numbering wrap

3)   that same invoked application (same InvocID), has another
      element process run, AND that process is assigned the same
      identification number as one of the processes previously run by
      that invoked application (same RunIndex), and finally,

4)   that element process terminates and is moved to the
      sysApplElmtPastRun table prior to the old, duplicate (InvocID,
      RunIndex) entry being aged out of the table by settings defined
      for sysApplElmtPastRunMaxRows and
      sysApplElmtPastRunTblTimeLimit.

In the event that a collision occurs, the new entry will replace the
old entry.

8. Security Considerations

In order to implement this MIB, an agent must make certain management
information available about various logical and physical entities
within a managed system which may be considered sensitive in some
network environments.

Therefore, a network administrator may wish to employ instance-level
access control, and configure the access mechanism (i.e., community
strings in SNMPv1 and SNMPv2C), such that certain instances within
this MIB are excluded from particular MIB views.

9. Acknowledgements

   This document was produced by the Application MIB working group.
   Special acknowledgement is made to:

      Rick Sturm
      Enterprise Management Professional Services, Inc.
      sturm@emi-summit.com
      For hosting the working group mailing list, and for his
      participation in the development of the initial draft.

      Jon Weinstock
      General Instrument Corporation
      jweinstock@gic.gi.com
      For his participation in the development of the initial drafts
      and for serving as editor for drafts 1 and 2.

      The editor would like to extend special thanks to the
      following working group members for their contributions
      to this effort.

      Harald Alvestrand, George Best, Ian Hanson, Harrie
      Hazewinkel, Carl Kalbfleisch, Bobby Krupczak, Randy
      Presuhn, Jon Saperia, Juergen Schoenwaelder

11. Author's Address

   Cheryl Krupczak
   Empire Technologies, Inc.
   541 Tenth Street, NW Suite 169
   Atlanta, GA 30318

   Phone: 770.384.0184
   EMail: cheryl@empiretech.com

   Jonathan Saperia
   BGS Systems Inc.
   saperia@networks.bgs.com

12. References

   [1] Information processing systems - Open Systems
        Interconnection - Specification of Abstract Syntax
        Notation One (ASN.1), International Organization for
        Standardization. International Standard 8824, (December,
        1987).

   [2] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Structure of Management Information
        for Version 2 of the Simple Network Management Protocol
        (SNMPv2)", RFC 1902, January 1996.

   [3] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Textual Conventions for Version 2 of
        the Simple Network Management Protocol (SNMPv2)", RFC
        1903, January 1996.

   [4] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Conformance Statements for Version 2
        of the Simple Network Management Protocol (SNMPv2)", RFC
        1904, January 1996.

   [5] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Protocol Operations for Version 2 of
        the Simple Network Management Protocol (SNMPv2)", RFC
        1905, January 1996.

   [6] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Transport Mappings for SNMPv2", RFC
        1906, January 1996.

   [7] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Management Information Base for
        Version 2 of the Simple Network Management Protocol
        (SNMPv2)", RFC 1907, January 1996.

   [8] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
        and S. Waldbusser, "Coexistence between Version 1 and
        Version 2 of the Internet-standard Network Management
        Framework", RFC 1908, January 1996.

   [9] Grillo, P., and S. Waldbusser, "Host Resources MIB", RFC 1514,
        September 1993.

   [10] Yergeau, F., "UTF-8, a transformation format of Unicode
        and ISO 10646", RFC 2044, October 1996.

   [11] Krupczak, C., and S. Waldbusser, "Applicability of Host
        Resources MIB to Application Management", Application MIB
        working group report, October 1995.

12. Full Copyright Statement